UCLA-ENG-7044
October 1970

NASAP-70
USER'S AND PROGRAMMER'S MANUAL

Howard Okrent
Lawrence P. McNamee

UCLA-ENG-7044
October 1970

NASAP-70 USER'S AND PROGRAMMER'S MANUAL

Prepared by

Howard Okrent
Lawrence P. McNamee

With Contributions and Assistance From

Alfred Tyrill
David Paletz
Steve Grandi
Donnamaie Meyerhoff
Dennis Sesar
Michael Wilson

School of Engineering and Applied Science
University of California
Los Angeles, California

PREFACE

The study for this report was carried out in the Computer Science Department, University of California, Los Angeles, under National Aeronautics and Space Administration Contract NAS 12-2138.

The development of the program and the preparation of this manual was a cooperative effort. We acknowledge the cooperation, patience, and understanding of Dr. W. W. Happ of the Electronics Research Center of NASA. We also express our appreciation to our sister universities and other NASAP-70 participants for their suggestions and help.

Within UCLA the cooperative spirit has also prevailed. Our thanks and appreciation go to: Alfred-Tyrill, who programmed the Fast Fourier Transform and plotter; David Paletz, who programmed the algorithm for building optimum circuit trees at a given frequency and the transfer function printout, Steve Grandi, who programmed the sensitivity analysis routines, Mrs. Donnamaie Meyerhoff for her excellent survey of transient models and the background study in radiation effects in solid state devices, Dennis Sesar for this critical review of the computational errors and run time limitations of NASAP-70, and Michael Wilson who investigated the application of NASAP-70 to large scale problems.

CONTENTS

PRECEDING PAGE BLANK NOT FILMED

CHAPTER 1

THE NASAP-70 PROJECT

1.0 Introduction

NASAP-70 is a digital circuit analysis program developed at UCLA with the cooperation and advice of a number of University, industrial, and governmental research groups located in virtually all sections of the United States. The program consists of a number of commonly required circuit analysis routines that are based upon the flowgraph techniques and algorithms of W. W. Happ and others.[1-3] As was pointed out by Happ, two distinct advantages can be realized (1) The requirement for writing a set of independent circuit equations is eliminated, and (2) the resulting flowgraph is unique. Because of these characteristics and the modular manner in which the program is constructed, algorithms can easily be mechanized, changed, and expanded for digital computation.

The program documented herein is the final installment of this cooperative project initiated by the Electronics Research Center of NASA four years ago.

1.1 Capabilities and Limitations

NASAP-70 can handle linear circuits consisting of constant-value passive elements, independent and dependent current sources, and independant and dependent voltage sources. The dependent sources must be linearly related to a voltage or current located in another part of the circuit. Nonlinear functional relationships (dependencies) and time-varying parameters cannot be handled in general. However, provisions have been included to accommodate a single nonlinear or external characteristic in resistors as advanced by Haag and Weber[4]

Central to the NASAP-70 program is its capability to formulate transfer functions from a topological description of the circuit and its numerical specification of components. The format of the input code, selected by a user, dictate the manner by which a transfer function is derived by the program. The following input codes, identified by the names NASAP, TREE, and CIRCUIT, can be employed

1. NASAP - directs the program to form its own tree and transfer function.
2. TREE - allows a user to define his own tree from which a transfer function can then be constructed by the program.
3. CIRCUIT - directs the program to select a tree by which a transfer function is constructed according to a user specified frequency to assure relatively accurate coefficients.

With one minor, but practical exception, a transfer function, relating a voltage across or a current through an element to either a driving voltage or current source, can in general be requested. Any number of transfer function requests can be made in any given computer run. The transfer function formulations are printed out as a ratio of two polynomials in S with numerical coefficients.

In conjunction with a particular transfer function request, a user can ask for

1. The sensitivity of a transfer function to changes in circuit parameters - A maximum of twenty valid sensitivity requests can be processed for each circuit. The sensitivity function is printed as a ratio of two polynomials in S with numerical coefficients.
2. A worst case analysis - For circuits of twenty elements or less (not counting independent voltage or current sources) a worst case analysis can be formulated as a ratio of two polynomials in S with numerical coefficients. Tolerances for each element can be user-specified or defaulted automatically with 10% values by the program.

3. Poles and zeros of all transfer functions
4. The sensitivity of poles - When pole sensitivity is requested, it is implied that it is formulated for all elements for which a sensitivity request had been issued.
5. Transient response - A user can request a transient response of any transfer function for an impulse, a step function, a sine wave, an exponential, a biased pulse, or pulse train driving function.
6. An A. C. frequency response (Bode Plot)

The output format for the transient, A. C. frequency response, sensitivity and worst case analysis functions can be presented by either tables, printer plots or both The input data is written on a free-field format. Diagnostics are provided for reading the input circuit description cards. If mistakes are relatively minor, corrections are made automatically by the program and are printed out to inform the user of the changes made. The program will not abort a run unless a serious error (one which cannot be easily interpreted) is encountered. The normal input is by cards. An on-line version of NASAP-70 has been developed but is not documented here.

NASAP-70 has been written completely in FORTRAN IV and has been developed on the IBM 360/91 and run on the SDS Sigma Seven Digital Computers at UCLA It has been successfully converted to other machines by other research groups at installations across the country By means of overlays, approximately 32K of core memory is required to handle circuits consisting of 30 elements.

The maximum number of elements that NASAP-70 can accommodate in practice is determined by the number of bits in the computer word. Even though computer words, in essence, are extended by software, circuits with more than 30 elements cannot be handled efficiently by the 32 bit IBM machines. Other machines with longer data words, such as the CDC 6600, can accommodate larger circuit problems.

1.2 Historical Perspective

To properly understand the place that NASAP-70 takes in the computer-aided design field, it is fitting to trace the development of the field, in general, and contrast it with NASAP-70. Typical circuit design programs, such as ECAP, SCEPTRE, NET-1, and CIRCUS derive their conceptual foundation from the treatise of Gabriel Kron, who was the first person to systematize circuit equations for solution by digital techniques.[5]

Kron noted that the topological-algebraic structure of networks could be treated separately in formulating mexh, nodal, and branch equations from a primitive set of ohmic relations. Even to this day, the influence of Kron's pioneering efforts are still being felt in all the major circuit program developments, including NASAP-70.

Kron's ideas, which were presented from an engineering point of view, were formally studied by J. Paul Roth who proved the interrelationship of the topological-algebraic structure of networks from an abstract mathematical position.[6] Even though Kron's work was known for a couple of decades, it never achieved popularity until Seshu and Reed published their papers and a book in which a rigorous proof of network theory was presented in a manner palatable to engineers.[7, 8]

Since that time, the field of computer-aided circuit design has exploded attracting outstanding investigators from many engineering and science fields. Noteworthy among the investigators are Branin, Bashkow, Calahan, Katzenelson, Malmberg, Kuh, Rohrer, Desoer, and many others.[9-16]

All of these researches advocated the matrix or state space method dealing with circuit problems. NASAP-70, on the other hand, with its flowgraph approach, traces its origin to Claude Shannon who discovered the topological gain formula while investigating the functional operation of an analog computer.[17*] Shannon's formula which is an analytic equation for calculating the gain for an open loop system (circuit) was generalized by W. W. Happ to include closed systems.[1] The Shannon-Happ formula fabricates the backbone of NASAP-70 upon which the transfer function, sensitivity, and frequency response of a large class of linear systems can be computed.

The NASAP project was started approximately four years ago at the Electronics Research Center of NASA with the Shannon-Happ formula as a programming kernal. The first year was devoted to program development which included transfer function, sensitivity, and A. C. frequency response capabilities. The second and third years brought further program development and application studies into areas of NASA research. This past year the NASAP project came to a close. The final version of the NASAP program, called NASAP-70, is documented herein. In addition, seven applications manuals are being written in the areas of Electronic Filter Design, Instrumentation Circuits, Biomedical Circuits, Communication Circuits, and others.[18-20]

1.3 Format of the Manual

The manual is divided into two parts (1) A User's Guide, which is made up of Chapters 2 and 3, and Appendices B and C, and (2) a Programmer's Manual which consists of Chapters 4 and 5, and the program listing and dictionary of program variables in Appendix A.

In Chapter 2, the fundamentals of setting up a problem for NASAP-70 are presented without any deep discussion of the theory involved It has been written, however, to show the program can be used for large problems, and how computing time and computational errors can be predicted and reconciled.

---

*During World War II, Claude Shannon developed his formula. But because of wartime restrictions, his work was not published and was virtually unknown to the academic community. In 1952, Samuel Mason rediscovered the same formula.

The theory behind NASAP-70 has been reserved for Chapter 3. Appendix B describes the most popular transistor models in use today, and includes a discourse of environmental effects (radiation) on transistor models.

Chapter 4 contains a detailed description of the NASAP program. This chapter should be read in conjunction with Appendix A, the program listing and dictionary of variables. Chapter 5 describes the system requirements for implementing NASAP-70.

The format of the User's Guide and Programmer's Manual has been designed to place those topics which can usually be self-contained in an individual chapter. It is for this reason that all references pertinent to a chapter have been placed at the end of that chapter. The format of the manual also provides a convenient method to add, delete, or change portions of the manual without disrupting larger segments of the entire manual.

REFERENCES

1. Happ, W. W., "Flowgraph Techniques for Closed Systems," IEEE Transactions Aerospace and Electronic Systems, AES-2, pp. 252-264, May 1966.

2. Happ, W. W., "Flowgraphs as a Teaching Aid," IEEE Transactions on Education, E-9, pp. 69-80, June 1966.

3. Gupta, S. C., Happ, W. W., "A Flowgraph Approach to the Laplace Transform and Transient Analysis of Discrete Systems," J. Franklin Institute, Vol. 280, pp. 150-163, August 1965.

4. Moe, M L., Schwartz, J. T., "Designers' Manual for Instrumentation Circuit," Final Report Contract NAS 12-2043, Denver Research Institute, University of Denver, Denver, Colorado.

5. Kron, G., Tensor Analysis of Circuits, Wiley, 1939.

6. Roth, J P., "An Application of Algebraic Topology to Numerical Analysis On the Existence of a Solution to the Network Problem," Proc. National Academy of Science, Vol. 41, pp. 518-521, 1955.

7. Reed, M. B., Seshu, S., "On Topology and Network Theory," Proceedings of the Symposium on Circuit Analysis, University of Illinois, Urbana, Illinois

8. Seshu, S , Reed, M. B., Linear Graphs and Electrical Networks, Addison-Wesley, 1961.

9. Branin, F. H. Jr., Computer Methods of Network Analysis with Applications in Science and Engineering, to be published by Prentice-Hall.

10. Bashkow, T. R , "The A Matrix, New Network Concept," IRE Transactions on Circuit Theory, Vol. CT-4, pp. 117-120, September 1957.

11. Calahan, D. A., Computer-Aided Network Design, McGraw-Hill, 1968.

12. Katzenelson, J., "An Algorithm for Solving Nonlinear Resistive Networks," Bell System Technical Journal, Vol. 44, pp. 1605-1620, November 1965.

13. Malmberg, A. F., et al., "Net-1 Network Analysis Program," Report LA-3119, Los Alamos Scientific Laboratory, Los Alamos, New Mexico.

14. Kuh, E. W., Rohrer, R. A., "The State Variable Approach to Network Analysis," Proceedings of IEEE, Vol. 53, pp. 672-686, July 1965.

15. Kuh, E.W., Desoer, C., Basic Circuit Theory, McGraw-Hill, 1966.

16. Special Issue on Computer-Aided Design, Proceedings of IEEE, Vol. 55, No. 11, November 1967.

17. Shannon, C.E., "The Theory and Design of Linear Differential Equation Moclines," National Defense Research Committee, OSRD Report 411, January 1942.

18. Moe, M.L., Schwartz, John T., "The Application of NASAP to the Design of Biomedical Instrumentation Circuits," Denver Research Institute Report DRI #2535, University of Denver, January 1970.

19. Haag, K.W., Weber, E.W., "Designers Manual for Computer-Aided Design of Communication Circuits," Department of Electrical Engineering, Illinois Institute of Technology, December 1969.

20. Zobrist, G., et al., "Designer's Manual for Electrical and Electronic Filters," Final Report Contract NAS 12-2041, Department of Electrical Engineering, University of Missouri, Columbia, Missouri.

PART I THE USER'S GUIDE

CHAPTER 2

A USER'S VIEWPOINT

2.0 Introduction

This chapter describes the step-by-step procedure of how to prepare a circuit for analysis by NASAP-70. The discussion has been restricted to an explanation of the rudiments of coding and does not include a detailed study of circuit theory or programming. That has been reserved for Chapter Chapters 3, 4 and 5. Examples have been dispersed throughout the discussion. Also included are some design tips from a user's viewpoint of NASAP-70, partitioning schemes, error analysis, and nonlinear techniques.

It must be emphasized, however, that if one is to really achieve a good design, a deeper understanding of circuit theory, and the advantages and limitations of computer methods contained herein are necessary. To that end the user is encouraged to read the articles and books referenced at the end of the chapter.[1-4]

2.1 Preliminary Considerations and Modeling

Before any problem is coded for a computer run, it is important to make a check on the validity of the circuit itself and the device models. Difficulty usually arises when some physical device, such as a transistor, diode, etc., is replaced by a lumped parameter model made up of some dependent active devices and passive elements, and inserted into the remaining network without really looking for such conflicts as two electrically difference current sources placed in series, or two electrically different voltage sources placed in parallel. In general, no loop in a circuit may consist entirely of voltage sources, including independent and dependent sources, nor should any node be formed entirely of branches in which only independent and dependent current sources are present. If such a case should occur, it would mean that NASAP-70 (or any other program for that matter) would not be able to formulate a set of independent equations necessary for solution since it violates the fundamental conservation of energy law.

It is also imperative to have a circuit designer select the appropriate model for a physical device. Quite often the model chosen has been derived for one type of operation and is assumed to be valid for another. All models are restricted to some extent. There are A.C., D.C., and transient models, some are considered small signal or large signal, while others are considered low frequency or high frequency. There are also the so-called environmental and radiation-effect models. Generally speaking, all models are appropriate only for the particular application for which they were derived, and if a circuit designer places a model in a universal catch-all role, only misleading results can occur.

In Appendix B descriptions of typical transistor models have been compiled and classified according to the intended use. Also include there is a tabulation of how the model parameters can be calculated from the manufacturer's device specifications. A rather in-depth study of solid-state modeling for a radiation environment is also given.

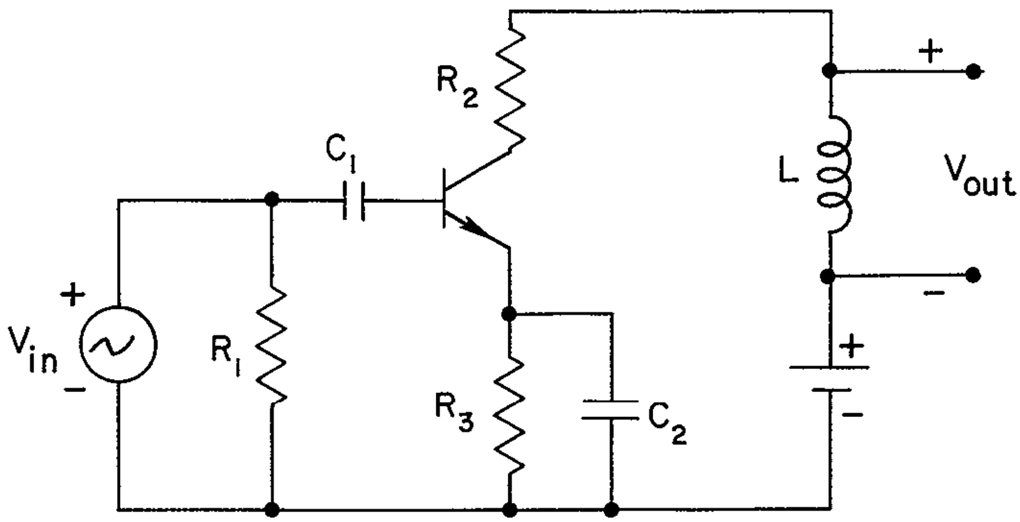
Besides modeling, other difficulties can appear when a complicated circuit is approximated by reducing the number of circuit. All too often, entire sections of a circuit are replaced by a simple series or parallel circuit, the cumulated practice of which can eliminate critical modes of circuit operations. As is usually the case, the circuit designer is so preoccupied with establishing models or reducing the size of the circuit in order to code it for the computer that he completely loses the physical realization of the problem itself. No matter how carefully a computer program is conceived, its operation can be completely frustrated by improper use by circuit designers who consistently violate sound engineering practice for the sake of accommodating a computer program.

2.2 Circuit Preparation

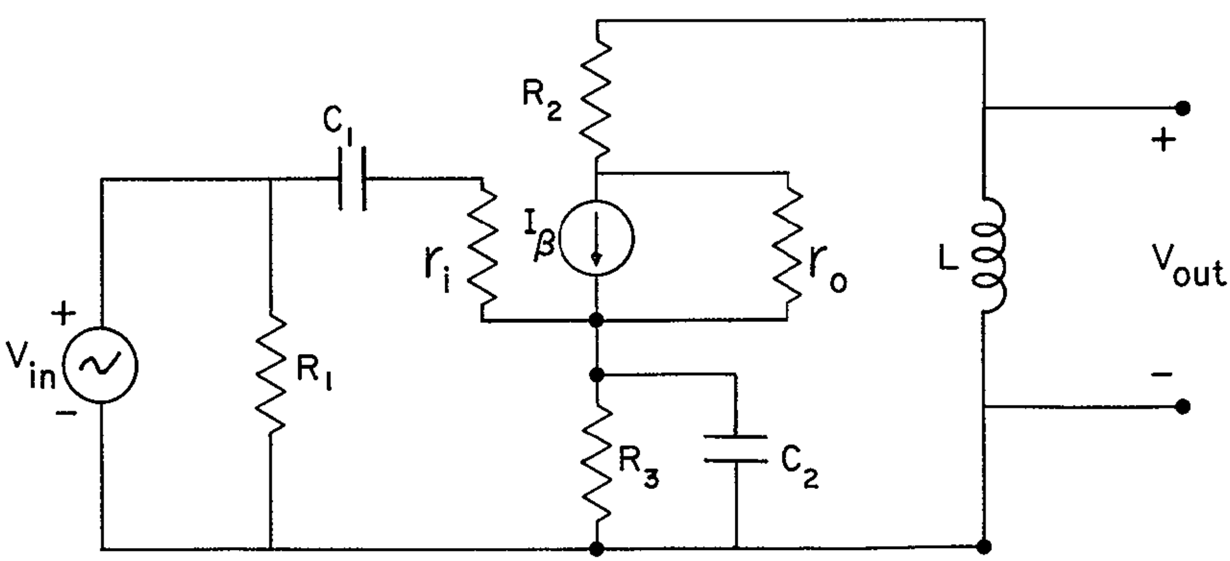
Once an appropriate circuit configuration has been established, making certain that no physical inconsistencies exist, the step-by-step preparation procedure for coding NASAP-70 can then be carried out. The procedure recommended here is described by means of an example circuit problem shown in Figure 2.1(a) which calls for an A.C. transfer function relating the voltage V_{out} across the inductor to the input voltage V_{in}.[4] The A.C. equivalent circuit is shown in Figure 2.1(b). Note that the battery has been replaced by its internal impedance of zero ohms and the transistor has been modeled by a simple input resistor and a dependent current generator placed in parallel with an output resistor. (High frequency effects are not considered here and actually are not required to illustrate the preparation procedure.)

Assuming that the circuit model in Figure 2.1(b) is adequate for the desired analysis, the initial preparation of coding for NASAP-70 requires that a unique identification be imposed on the topology of the circuit, the circuit elements, and the dependencies. Consequently, the nodes are numbered consecutively starting with the number 1 and the elements are redefined according to a letter-number format, where each component is unambiguously determined, according to type, by one of three possible coding schemes described in section 2.3. No two components can have the same code identifications. Dependencies are indicated on the circuit diagram by a dash line which designates the current or voltage variable upon which the sources are dependent This is done only as a crutch to remind the designer not to forget it when the circuit information is actually coded.

After the nodes are assigned numbers (topology), and the circuit elements and dependencies are distinguished, as shown in Figure 2.2 for the coding scheme NASAP, the unknown directed current variables must be assigned to each circuit branch element. Generally speaking, the current

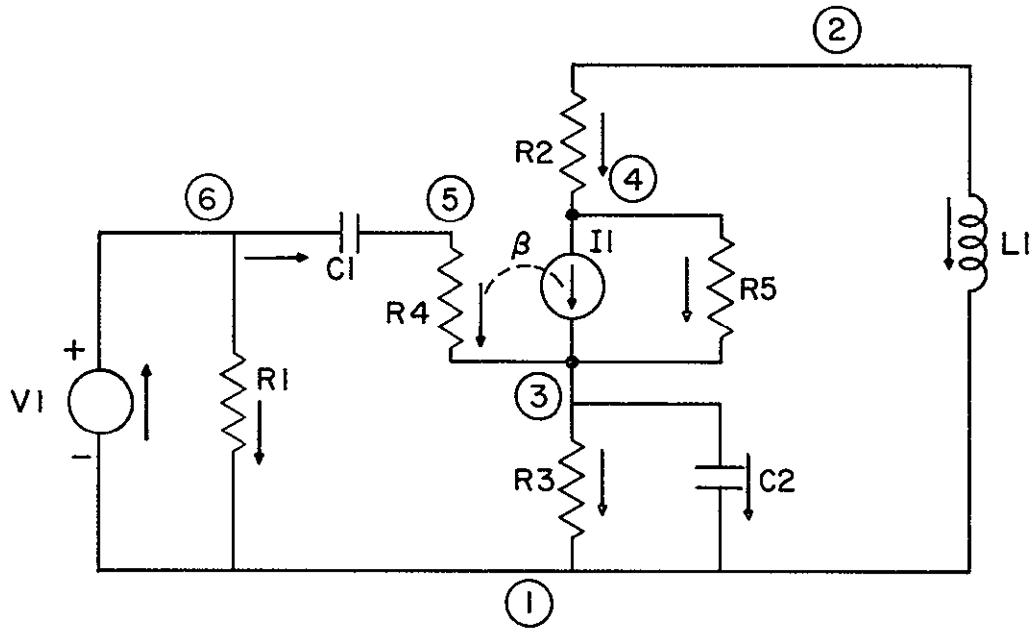


(a) CIRCUIT DIAGRAM



(b) A.C. MODEL

Figure 2 1



V1 = 1 volt

R1 = 50 ohms

C1 = 10.7 $\mu\mu f$

R4 = 25 ohms

R3 = 325 ohms

C2 = 10μf

β = 98

R5 = 2 megohms

R2 = 20,000 ohms

LI = 1.7 millihenrys

Figure 2.2 Prepared Circuit Diagram for Coding Scheme NASAP

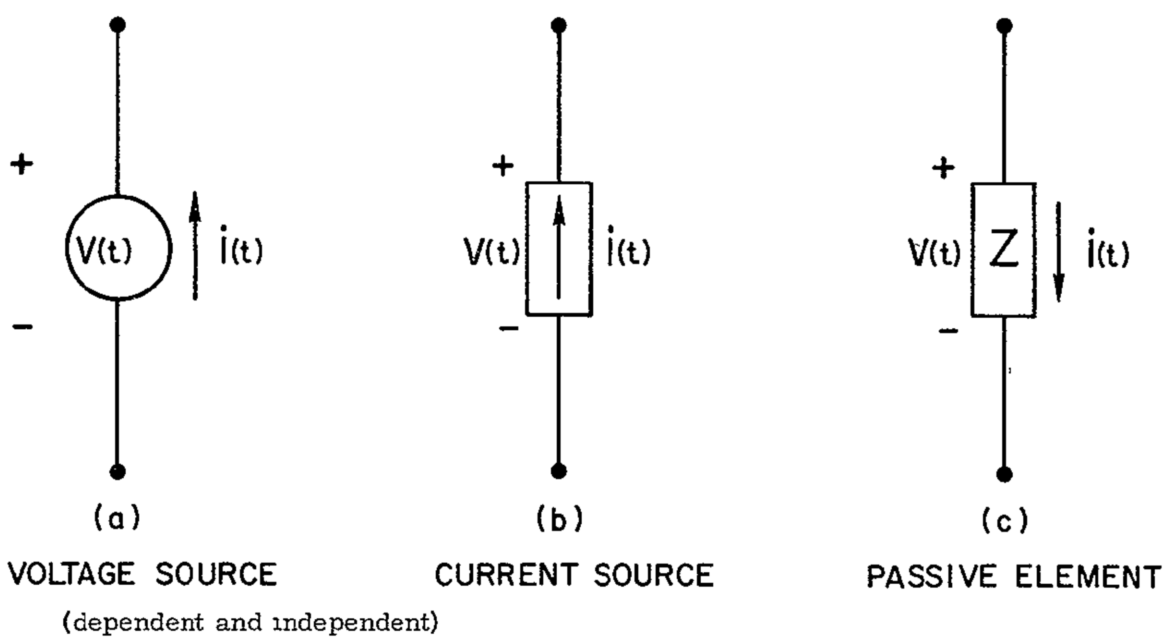


Figure 2.3 Voltage-Current Relationship of Elements

direction in each passive element can be assigned arbitrarily although some caution must be exercised in assigning currents to passive elements related to dependent sources. The current directions in active devices also must be handled with care so that the proper voltage-current relationship can be established.

Voltage Sources. The current direction associated with either an independent or dependent voltage source should be established in the positive sense of the voltage rise — from minus to plus — as shown in Figure 2.3(a). (An opposite rule could have been assumed but this would have required that a negative voltage value be read in with the input data. Since such a practice is somewhat unnatural to most circuit designers, it has been avoided here as a matter of good practice rather not that it cannot be done.) Following the current assignment rule as stated presents no difficulty when independent voltage sources are being considered, in general, the same rule can be applied to dependent voltage sources provided the dependency relationships are consistent with the physical phenomena involved. (A number of examples consisting of dependent sources have been included in this chapter and are described in sufficient detail to illustrate how the common current assignment rule for both dependent and independent voltage sources can be adopted.)

Current Source In dealing with either dependent or independent current sources, the positive voltage sense is taken in the direction of positive current flow as shown in Figure 2.3(b). Similar to the discussion concerning dependent voltage sources, the positive sense of the voltage across the dependent current source is the same as that defined for the independent current source provided the dependency relationship is consistent with the physical phenomenon. Examples are included to illustrate the procedure to be employed.

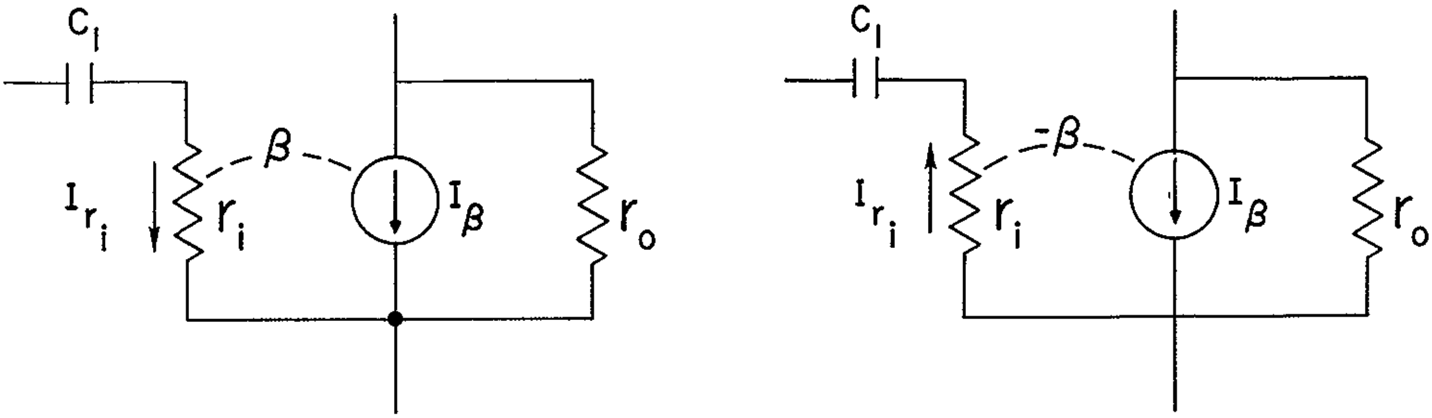
Figure 2.3 summarizes the current-voltage relationship for the dependent and independent active devices and passive elements. If the assignment of the current direction is in agreement with these configurations -

including the dependency considerations described below - the NASAP-70 program can then be properly coded.

Dependencies. A major concern of the NASAP-70 in programming dependency relationships is to assure that the proper sign is attached to the coded dependency parameter. Consider the transistor model shown in Figure 2.4(a) which has been extracted from the diagram of Figure 2.2. For this common emitter circuit, if a current flows into r_1, the dependent current source I_β will generate positive current flow as indicated in Figure 2.4(a). For the case depicted β is taken as a positive number, and thus the current can be expressed as $I_\beta = \beta I_{r_1}$. If the assigned current direction in r_1 is taken as in Figure 2.4(b) while the I_β direction of flow is retained, then the dependency parameter would be $-\beta$, a negative number. Therefore it is important that the physical dependency phenomenon be clearly understood for each device that is being modeled.

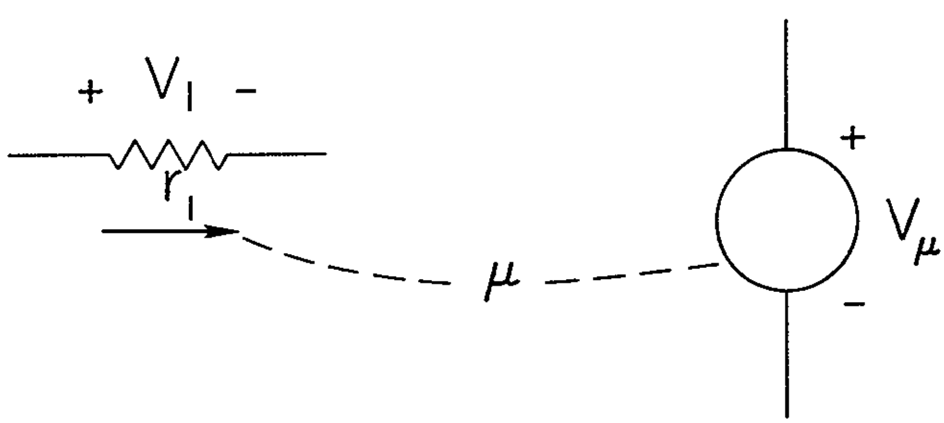
Another dependency example is shown in Figure 2.4(c). For this case a dependent voltage source V_μ is considered, where V_μ is linearly related to the voltage v_1 developed across the resistor r_1 as shown. In order to maintain the positive value of the parameter μ in the relation $V_\mu = \mu V_1$, it is necessary that the positive sense of V_1 be taken across r_1 as the physical phenomenon dictates. Thus, this requires that the current through r_1 be assigned so that the desired voltage polarity of V_1 is produced. Alternative measures can be taken, but practicality dictates that it is best to follow one rule because it leads to fewer errors and presents a better means for debugging. The user must be cognizant of the dependency relationship involved and code according to the plan best suited to his own tastes.

Once all the preparation steps are completed to uniquely identify the topology, the circuit elements, and the dependencies, and to assign the current direction in all the circuit elements - the circuit diagram is then ready for coding. Notice, for example, in Figure 2.2 that all the nodes



$$I_\beta = \beta\, I_{r_i}$$

(a) (b)



$$V_\mu = \mu V_1$$

(c)

Figure 2 4 Dependency Parameters

are numbered consecutively starting with the number 1. Zero and negative node numbers are not permitted. Also note that all circuit elements have been redefined to conform to the NASAP letter-number coding scheme (see Section 2.31). It must be emphasized that even though some of the elements could have the same value, they still have to be assigned different numbers in their respective letter-number symbol. The dependency relationship is illustrated by a broken line with the associated dependency parameter value placed in the middle of it.

2.3 Coding Mechanics

The general order of a NASAP-70 program deck is shown in Figure 2.5. The first group of cards allows a user to head the output listing with the title information punched on them. No more than ten cards are permitted.

The second group of cards contains the topology information, element designators, the values of the elements, and dependencies of the circuit to be analyzed. Three different circuit description coding rules can be utilized.

The last group of cards contain the desired output requests.

The title, circuit description, and output request cards are separated from each other by control cards. The necessary control card information is dictated by the circuit description format selected by a user.

Any number of individual problems can be processed during a computer run. However, the above mentioned order must be maintained for each problem. The last card of the entire group of cards must be a STOP card. Only one STOP card is allowed.

2.3.1 Circuit Description Codes

The circuit description coding rules are based upon a free-field format whereby each element identifier is written on a programming sheet, one to a line, with its corresponding topological, numerical, and dependency information. Thus a circuit composed of, say, 15 elements will require 15 cards, exclusive of the control and output request cards.

Title Cards

Control Card #1

Circuit Description Cards
Topology
Element Identification
Numerical Values
Dependencies

Control Card #2

Output Request Cards:
Transfer Function
Pole Sensitivity
Parameter Sensitivity
Worst Case
Transient Response
Poles and Zeros
Frequency Response
Nonlinear Parametric Response

Control Card #3

STOP Card

Figure 2 5

The three different code formats available in defining a circuit to NASAP-70 are identified to the program by punching one of the words: NASAP, TREE, or CIRCUIT on control card #1.

<u>NASAP</u>: For compatibility with other versions of NASAP, control card #1, with NASAP punched on it, will cause the program to accept the circuit description in the form presented in A USER'S GUIDE AND PROGRAMMER'S MANUAL FOR NASAP[5] and CODING INSTRUCTIONS FOR NASAP 69/I.[6] The NASAP coding of a circuit relieves the user of selecting a tree Instead, the program automatically builds a tree that results in a transfer function being formed in a minimum of computation time

The circuit coding requires that six fields of information be set down on each card, and since a free field format is employed, the ordering of the information must be taken into account This is done by inserting at least one blank to separate the fields from each other, permitting the information to be inputted without regard to column orientation

The general form of the circuit input data is.

(ELEMENT DESIGNATOR) (ORIGIN NODE) (TARGET NODE) (NUMERICAL OR DEPENDENCY VALUE) (UNITS (OPTIONAL)) (DEPENDENCY DESIGNATOR)

The parentheses are not allowed on the coding sheets, but are placed here for clarity sake.

The first letter of each ELEMENT DESIGNATOR must conform to the following relationship:

Table 2.1

V	Voltage Source	C	Capacitor
I	Current Source	L	Inductor
		R	Resistor

Any of those letters may be followed by up to 11 alphanumeric characters to bring the total number of characters in the ELEMENT DESIGNATOR to not more than 12.

The second and third fields contain the numbers of the origin node and the target (terminal) node, respectively, of the assigned current flow through the element. The fourth field accepts the numerical value of the circuit element, if the element is not a dependent source, and accepts the dependency parameter value if it is. The numerical values can be expressed in either decimal numbers with or without a decimal point, or in Fortran E-format.

The fifth field is an optional units field that allows a user to operate in other than the MKS system of units. In effect the units field is a scale factor multiplier to the numerical value field. The accepted symbols are given in Table 2 2 No mixing or interchanging of scale factors are permitted. If UNITS is not specified, MKS units are assumed

Table 2. 2

ELEMENT	SCALE FACTOR	MULTIPLIER
RESISTOR	K	10^{3}
	M	10^{6}
CAPACITOR	PF	10^{-12}
	UF	10^{-6}
INDUCTOR	UH	10^{-6}
	MH	10^{-3}

The sixth field position is employed only if a dependency exists, which is indicated by writing either an I, for current, or V, for voltage, followed by the element identifier of the element upon which the dependency exists.

With the "NASAP" coding rules, the prepared circuit diagram of Figure 2.2 can be coded as shown below:

ELEMENT DESIGNATOR	ORIGIN NODE	TARGET NODE	PARAMETER OR DEPENDENCY VALUE	UNITS	DEPENDENCY DESIGNATOR
V1	1	6	1		
R1	6	1	50		
C1	6	5	10.7	PF	
R4	5	3	25.		
R3	3	1	325		
C2	3	1	10	UF	
I1	4	3	98		IR4
R5	4	3	2	M	
R2	2	4	20	K	
L1	2	1	1.7	MH	

Note the coding peculiar to the dependent current source I1. Since I1 is related to current flow in the resistor R4, the dependency relation is indicated in the sixth field position as IR4 with the dependency parameter β (which is equal to a value of 98) coded in the fourth field. If other dependent sources were present, they too would follow the identical coding scheme.

Control card #2, for the NASAP coding rules, must contain either the word OUTPUT or the word END.

<u>TREE</u> The TREE input code can be employed by a user who wishes to form a transfer function based upon a user-defined tree. Control card #1, with the word TREE on it, alerts the program that the circuit description that follows contains a user specified tree in addition to the topological, elemental, and dependency information.

The general format of the circuit coding is

ELEMENT INDENTIFIER/DEPENDENCY (ORIGIN-TARGET) = VALUE UNITS

To obtain a worst case function, a card must be inserted immediately after control card #2 (the END or OUTPUT cards), with the word, WORST. Tolerance cards must also be provided, since the variation of each element must be known. Tolerance cards have the following format

TOL = ELEMENT DESIGNATOR = VALUE

where ELEMENT DESIGNATOR is a circuit element, and VALUE is the relative tolerance $\frac{\Delta Q}{Q}$ of the element Q. If no TOLERANCE card is provided for an element, a default value of .1 (10%) is assumed.

When a WORST CASE Function is requested, the sensitivity functions for all elements are also printed out automatically. Therefore no SENSITIVITY cards can be present when a WORST card is present. A typical WORST CASE analysis is given in the examples in Appendix C.

POLE SENSITIVITY The sensitivity of the poles of the transfer function is obtained by adding the word POLES to the ROOTS card:

ROOTS, POLES.

Pole sensitivity in NASAP-70 is defined as

$$PS = \frac{dS_i}{dQ} \; Q$$

where S_i is a pole and Q is a circuit parameter whose value changes causing a change in the root S_i. The sensitivity of the poles are produced for all elements for which a sensitivity request is made (or for all elements if a WORST card is included).

Pole sensitivity function is printed as the ratio of two polynomials in S, just like the parameter sensitivity function and is interpreted similarly.

Plot Requests: Printer plots can be requested of NASAP-70 to produce: (1) transient responses of transfer functions from a group of user-specified standard input functions, (2) a bounded frequency response of transfer functions, and (3) bounded (parameter, pole, and worst case) sensitivity responses.

The plot request is a single card of the form

PLOT (option 1/option 2 ......)

where the options are types, parameters, and limits of the printer plots. All options for any one plot request must appear on one card since continuation cards are not allowed. Not all the options are required since defaults are provided. Options may appear in any order on the plot card.

A transient response can be obtained for an impulse, step, sine, exponential, pulse, and pulse train by specifying an option according to

Table 2.6

OPTION FORMAT	DESIRED INPUT RESPONSE
TYPE = IMPULSE	Impulse
TYPE = STEP	Step function
TYPE = SINE	Sine Wave
TYPE = EXPONENTIAL	Exponential
TYPE = PULSE	Pulse or Pulse train

To provide control over the format of the printer plot itself the following options can be included

Table 2.7

Option Format	Interpretation
AMPLITUDE = a number	Represents the magnitude of any of the waveforms in Table 2.7
DENSITY = a number	Specifies the number of calculations per plotted output point
TIME = a number (in secs)	Specifies the duration of the applied input, it must always be present for any transient response request
STEP = a number (in secs)	Specifies the time between individual calculated points

For the PULSE type option, the following additional options can be employed

Table 2.8

Option Format	Interpretation
BIAS = a number	Used in conjunction with the AMPLITUDE option to produce a train of positive and negative pulses
WIDTH = a number (in secs)	Specifies the duration of a pulse
CYCLE = a number (in secs)	Specifies the time of one cycle

To illustrate how some of the pulse options are combined, consider the plot statement

```
PLOT(TYPE=PULSE/AMPLITUDE=10/BIAS=-2/WIDTH
=10/CYCLE=15/TIME=30)
```

The program interprets this statement to mean a pulse train of two pulses (TIME = 30 seconds) where each pulse alternates between +8 and -2 (from AMPLITUDE = 10, BIAS = -2) and each pulse duration is 15 seconds (CYCLE = 15) with a change from a positive to negative amplitude 10 seconds from the beginning of the cycle (WIDTH = 10). Figure 2.12 illustrates the waveform involved.

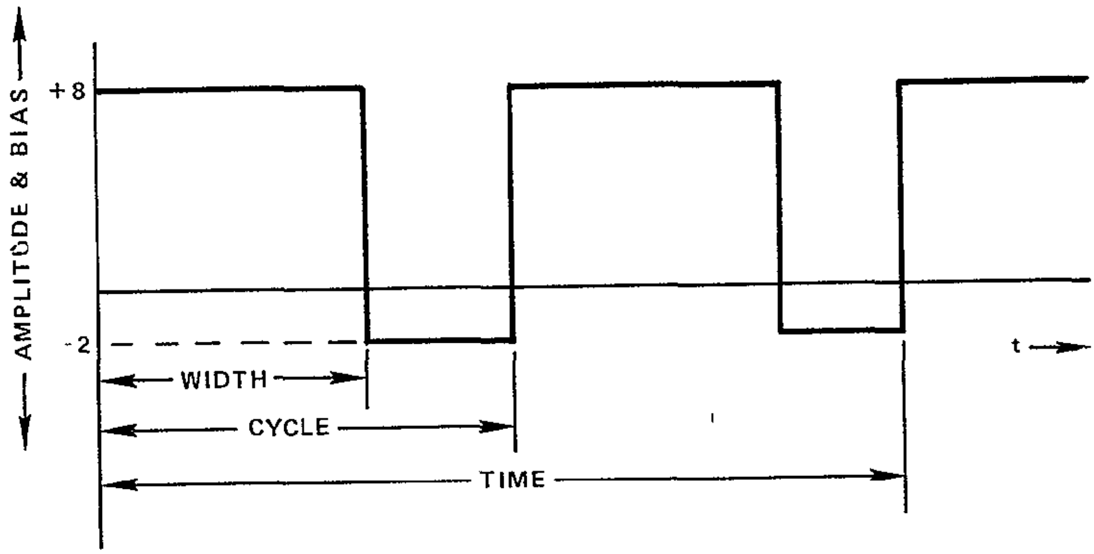


Figure 2 12 Pulse Wave Options

Options associated with other type inputs are as follows·

Table 2.9

TYPE OPTION	INTERPRETATION
CONSTANT = a number	Used only with EXPONENTIAL input. It represents the a in Ke^{at}
FREQUENCY = a number (in Hz/sec)	Used only with the SINE input to specify the frequency of the sine wave

Table 2 10 summarizes all the type input options and their permissable function and printer graph parameters. Only the first two letters of each word are needed. For instance, AM may be used to abbreviate AMPLITUDE, IM for IMPULSE, etc.

The following default conditions apply

Table 2. 10

TYPE	TRANSIENT OPTIONS								
	AMPLITUDE	BIAS	CONSTANT	FREQUENCY[1]	DENSITY	TIME	WIDTH	CYCLE	STEP[2]
IMPULSE	X				X	X			X
STEP[2]	X				X	X			X
SINE	X			X	X	X			X
EXPON	X		X		X	X			X
PULSE (PULSE-TRAIN)	X	X			X	X	X	X	X

1. The option "frequency" is used here in conjunction with the TYPE option SINE, it is also used as a TYPE option in generating a frequency response (see Table 2. 12).
2. Step is used in three senses as a TYPE option to denote a step function, as an increment between plot points (in secs) for transient responses, and as an increment between plot points (in CPS) as given in Table 2. 12.

Table 2.11

OPTION	DEFAULT
TYPE	IMPULSE
TIME	1 second
STEP	.01
AMPLITUDE	1
DENSITY	1
CONSTANT	1
FREQUENCY	1

Inconsistent requests will be ignored and are defaulted from left to right, e.g.,

PLOT(TYPE = IMPULSE/FREQ = 2) yields an impulse response.

The printer plot capability can also be employed to produce a frequency plot (Bode) of a transfer function and sensitivity functions. The PLOT instruction include the appropriate codes shown below

Table 2.12

FORMAT	INTERPRETATION
TYPE = FREQUENCY	Plots the magnitude and phase of the transfer function
TYPE = SENSITIVITY	Plots the magnitude and phase of the sensitivity function
TYPE = WORST	Plots the magnitude of the square worst case tolerance function
EL = Element Designator	Specifies to the program which particular sensitivity function request should be plotted (from a 'SENSITIVITY' or 'WORST' card)
FROM = a number (in CPS)	Specifies the beginning of a frequency, sensitivity, or worst case response plot
TO = a number (in CPS)	Specifies the end of a frequency, sensitivity, or worst case response plot
STEP = a number (in CPS)	Indicates the frequency step increment between individual calculated points

Note that the keyword STEP is used in a different context here than in the transient response case.

To illustrate how the PLOT options can be used, consider the plot statements

PLOT(TY = FR/FR = 10/TO =1E3/ST = 1.122)

PLOT(TY = SE/EL = C3/FR = 10/TO = 1ES/ST = 1.22)

taken from the output request statements in Figure 2.7. The first statement requests a frequency plot of the transfer function VR4/VV1, to encompass the frequency range from 10 to 1000 cps in increments of 1.122 cps. The second statement requests a plot of the sensitivity function of the transfer function VR4/VV1 with respect to the element C3. The frequency range is the same as the first plot statement. Figures 2.13 and 2.14 are plot and table outputs of the two PLOT requests.

Note that the PLOT requests always refer to the previous transfer function request, if more than one such request is made. The sensitivity plot request must include an element option since more than one sensitivity request can be called for in conjunction with a transfer function request. Both the magnitude and phase are printed out.

Control Card #3 can have either the word EXECUTE or END on it.

As another example problem describing PLOT requests, consider the circuit shown in Figure 2.15 taken from the report by Moe and Schwartz.[7] The input code is shown in Figure 2.16 where the transfer function of the voltage across C7 to the drawing voltage source V1 is requested. There are three transients plots requested (1) an impulse response of .5 sec duration, (2) a step response of .5 sec duration, and (3) a 4 cps sine wave response, outputted every .005 second until 9 seconds is reached.

The output plots and tabular data are shown in Figures 2.17 through 2.22.

Figure 2.23 shows a typical deck setup for multiple circuit problems for one computer run. The three different coding schemes are illustrated.

Complete input and output listing of another sample problem can be found in Appendix C. Note only one STOP appears – at the end of the deck.

PLOT(TY=FR/FR=10/TO=1E3/ST=1.122)

THE TRANSFORM OF THE RESPONSE IS ---

EXP. OF S	NUMER. COEFFS.	DENOM. COEFFS.
0	-4.53398E 02	3.10779E 03
1	-2.17638E 00	5.50359E 00
2	-2.61157E-03	1.90822E-02
3	4.57058E-10	3.40184E-05

FREQ	MAG	FREQ	MAG	FREQ	MAG	FREQ	MAG
1.000E 01	1.50113E-01	3.162E 01	1.54649E-01	9.996E 01	2.93322E-01	3.161E 02	4.17079E-02
1.122E 01	1.51226E-01	3.547E 01	2.10189E-01	1.122E 02	2.12767E-01	3.546E 02	3.65928E-02
1.259E 01	1.52640E-01	3.980E 01	2.31598E-01	1.258E 02	1.62737E-01	3.979E 02	3.22108E-02
1.412E 01	1.54439E-01	4.466E 01	2.63725E-01	1.412E 02	1.29291E-01	4.464E 02	2.84271E-02
1.585E 01	1.56736E-01	5.011E 01	3.12164E-01	1.584E 02	1.05569E-01	5.009E 02	2.51393E-02
1.778E 01	1.59679E-01	5.622E 01	3.90674E-01	1.777E 02	8.79515E-02	5.620E 02	2.22677E-02
1.995E 01	1.63469E-01	6.308E 01	5.22660E-01	1.994E 02	7.43875E-02	6.305E 02	1.97493E-02
2.238E 01	1.68381E-01	7.077E 01	6.91435E-01	2.238E 02	6.36414E-02	7.075E 02	1.75335E-02
2.512E 01	1.74800E-01	7.941E 01	6.40991E-01	2.511E 02	5.49304E-02	7.938E 02	1.55789E-02
2.818E 01	1.83280E-01	8.909E 01	4.32003E-01	2.817E 02	4.77371E-02	8.906E 02	1.38509E-02

GREATEST VALUE = 6.91435E-01 LOWEST VALUE = 0.0 INTERVAL = 6.01248E-03

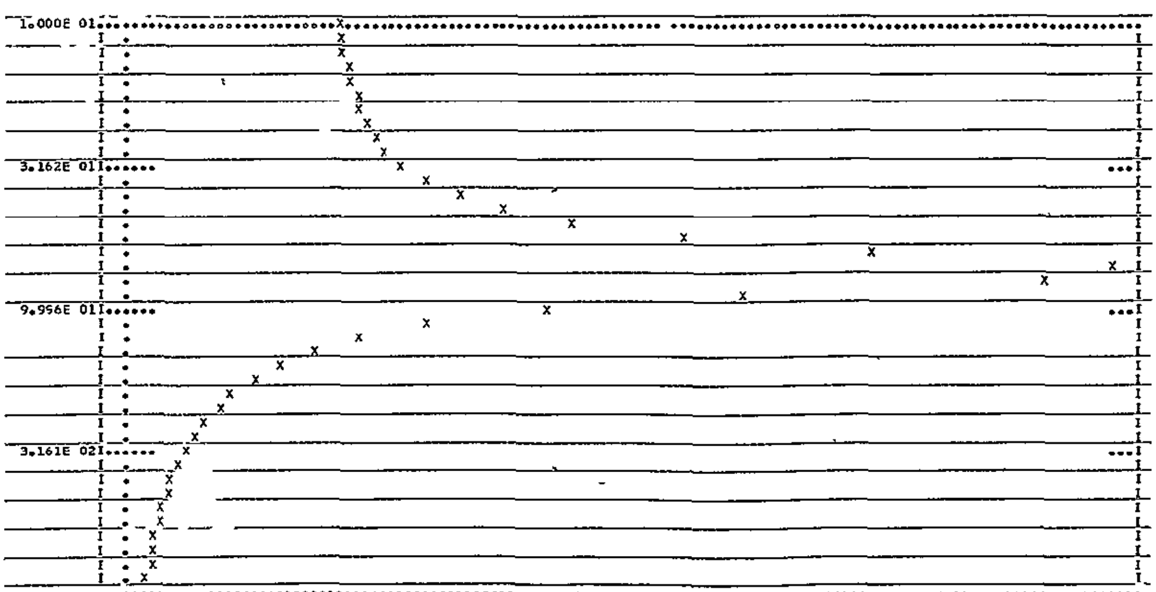


Figure 2 13a

FREQ	PHA	FREQ	PHA	FREQ	PHA	FREQ	PHA
1.000E 01	-1.73835E 02	3.162E 01	-1.63565E 02	9.996E 01	8.37798E 01	3.161E 02	8.25309E 01
1.122E 01	-1.73114E 02	3.547E 01	-1.62592E 02	1.122E 02	7.95327E 01	3.546E 02	8.32624E 01
1.259E 01	-1.72 17E 02	3.980E 01	-1.61992E 02	1.258E 02	7.79037E 01	3.979E 02	8.39364E 01
1.412E 01	-1.71441E 02	4.466E 01	-1.62112E 02	1.412E 02	7.75396E 01	4.464E 02	8.45524E 01
1.585E 01	-1.70484E 02	5.011E 01	-1.63635E 02	1.584E 02	7.78226E 01	5.009E 02	8 51117E 01
1.778E 01	-1 69446E 02	5.622E 01	-1.68051E 02	1.777E 02	7.84394E 01	5.620E 02	8.56173E 01
1.995E 01	-1.68333E 02	6.308E 01	-1.79034E 02	1.994E 02	7.92191E 01	6.305E 02	8.60725E 01
2.238E 01	-1.67155E 02	7.077E 01	1.55400E 02	2.238E 02	8.00642E 01	7.075E 02	8.64810E 01
2.512E 01	-1.65937E 02	7.941E 01	1.17726E 02	2.511E 02	8.09175E 01	7.938E 02	8.68468E 01
2.818E 01	-1.64718E 02	8.909E 01	9.39690E 01	2.817E 02	8.17460E 01	8.906E 02	8.71734E 01

GREATEST VALUE = 1.55400E 02 LOWEST VALUE = -1.79034E 02 INTERVAL = 2.90813E 00

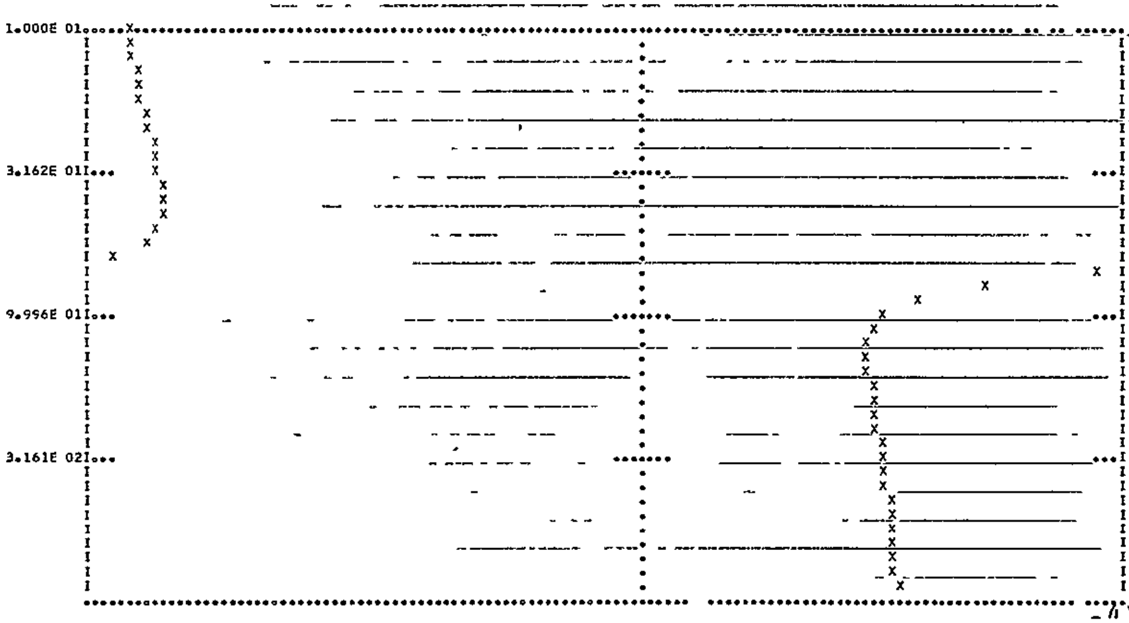


Figure 2 13b

PLOT(TY*SE/EL=C3/FR=10/TO*1E3/ST*1.122)

SENSITIVITY WITH RESPECT TO C3

THE TRANSFORM OF THE RESPONSE IS ---

EXP. OF S	NUMER. COEFFS.	DENOM. COEFFS.
0	0.0	1.40906E 06
1	1.05305E 02	1.10726E 04
2	7.05150E 00	3 14515E 01
3	3.97402E-02	8 17718E-02
4	7.47259E-05	1.23867E-04
5	4.44249E-08	8.88326E-08
6	0.0	-1.55484E-14

FREQ	MAG	FREQ	MAG	FREQ	MAG	FREQ	MAG
1.000E 01	1.87546E-02	3.162E 01	2.01618E-01	9 996E 01	1.38362E 00	3.161E 02	5.60164E-01
1.122E 01	2.35103E-02	3.547E 01	2.63168E-01	1.122E 02	1.12001E 00	3.546E 02	5.47629E-01
1.259E 01	2.95297E-02	3.980E 01	3.48541E-01	1.258E 02	9.52435E-01	3.979E 02	5.37739E-01
1.412E 01	3.71615E-02	4.466E 01	4 71535E-01	1.412E 02	8.39270E-01	4.464E 02	5.29920E-01
1.585E 01	4.68584E-02	5.011E 01	6 58415E-01	1.584E 02	7.59117E-01	5.009E 02	5 23731E-01
1.778E 01	5.92140E-02	5.622E 01	9.63080E-01	1.777E 02	7.00355E-01	5.620E 02	5 18827E-01
1 995E 01	7.50181E-02	6.308E 01	1.491E4E 00	1.994E 02	6.56213E-01	6.305E 02	5.14940E-01
2.238E 01	9.53375E-02	7.077E 01	2.26433E 00	2.238E 02	6.22488E-01	7.075E 02	5.11856E-01
2.512E 01	1.21646E-01	7.941E 01	2.38786E 00	2.511E 02	5.96415E-01	7.938E 02	5 09407E-01
2.818E 01	1.56040E-01	8.909E 01	1.81677E 00	2.817E 02	5.76093E-01	8.906E 02	5.07465E-01

GREATEST VALUE = 2.38786E 00 LOWEST VALUE = 0 0 INTERVAL = 2.07640E-02

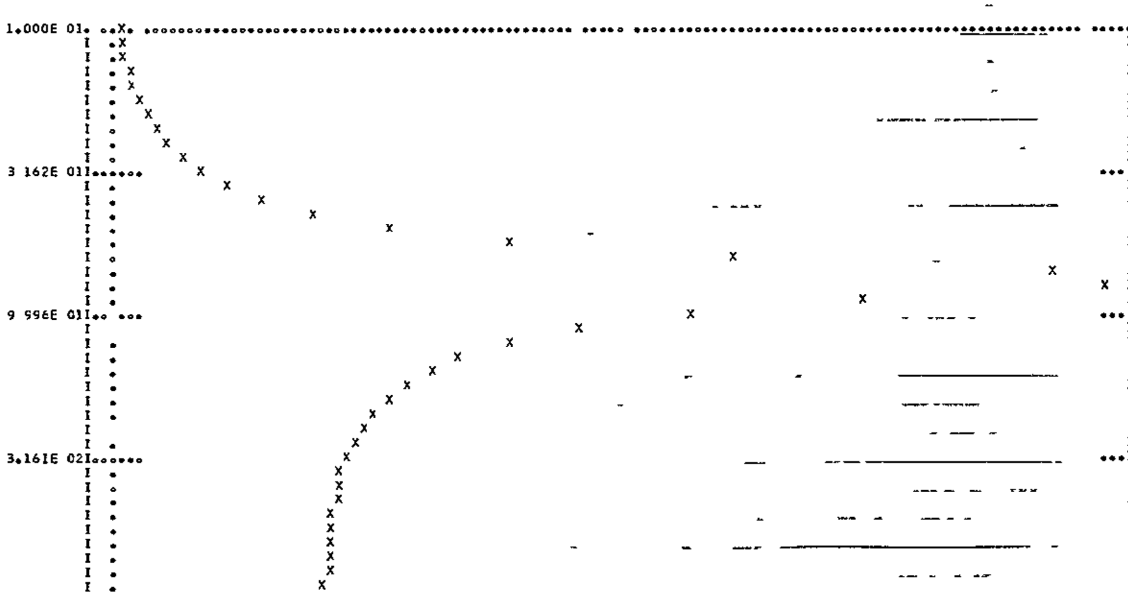


Figure 2 14a

FREQ	PHA	FREQ	PHA	FREQ	PHA	FREQ	PHA
1.000E 01	1.58658E 02	3.162E 01	1.54154E 02	9.996E 01	6.44658E 00	3.161E 02	-6.94653E 00
1.122E 01	1.59421E 02	3.547E 01	1.51900E 02	1.122E 02	-3.52648E-01	3.546E 02	-6.41708E 00
1.259E 01	1.558[illegible]6E 02	3.980E 01	1.45032E 02	1.258E 02	-4.17048E 00	3.979E 02	-5.88182E 00
1.412E 01	1.60123E 02	4.466E 01	1.45255E 02	1.412E 02	-6.36671E 00	4.464E 02	-5.35784E 00
1.585E 01	1.60100E 02	5.011E 01	1.399[illegible]2E 02	1.584E 02	-7.58162E 00	5.009E 02	-4.85635E 00
1.778E 01	1.59824E 02	5.622E 01	1.31705E 02	1.777E 02	-8.16203E 00	5.620E 02	-4.38394E 00
1.995E 01	1.59287E 02	6.308E 01	1.16924E 02	1.994E 02	-8.31873E 00	6.305E 02	-3.94417E 00
2.238E 01	1.58477E 02	7.077E 01	8.76770E 01	2.238E 02	-8.19118E 00	7.075E 02	-3.53841E 00
2.512E 01	1.57375E 02	7.941E 01	4.65167E 01	2.511E 02	-7.87680E 00	7.938E 02	-3.16645E 00
2.818E 01	1.55951E 02	8.905E 01	1.55387E 01	2.817E 02	-7.44515E 00	8 906E 02	-2.82700E 00

GREATEST VALUE = 1.60123E 02 LOWEST VALUE = -8.31873E 00 INTERVAL = 1.46471E 00

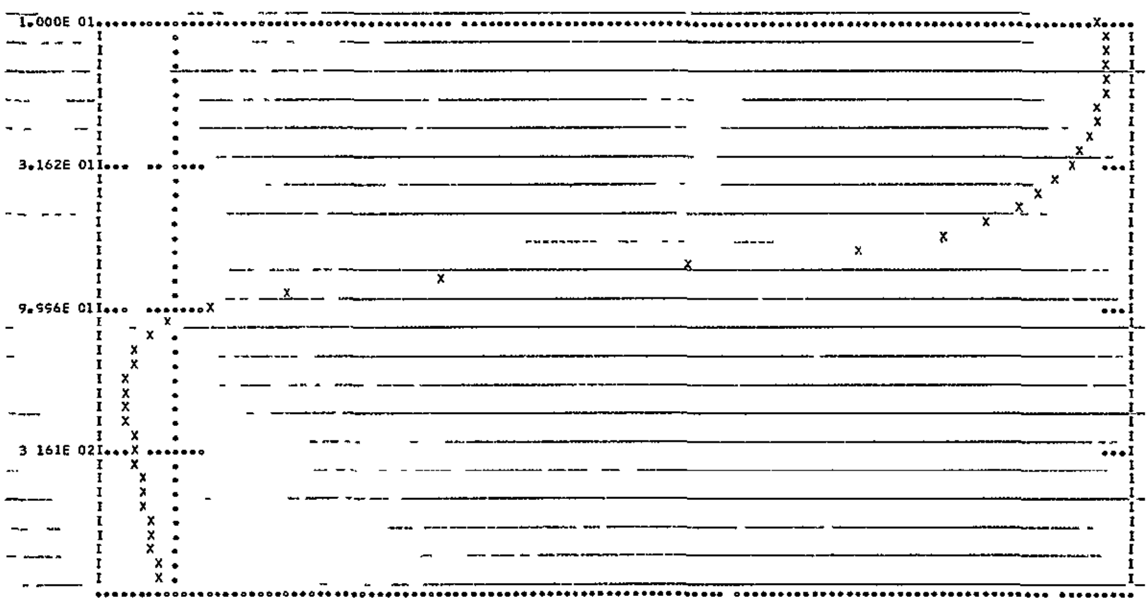


Figure 2 14b

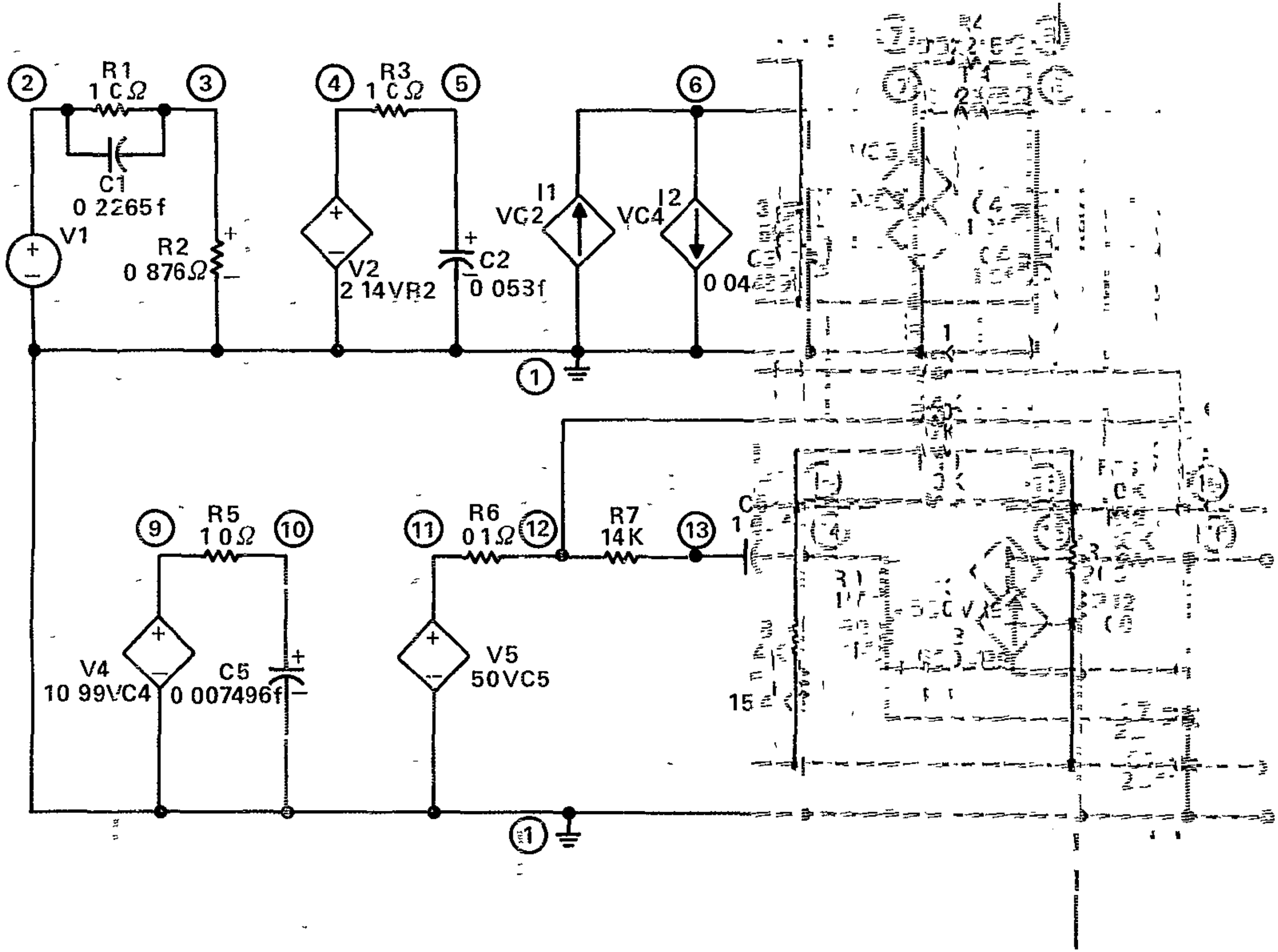


Figure 2 15 Complete Simplified Circuit for Anal
Complete EOG System with a Fourt

NASAP (SIMPLIFIED EDG SYSTEM WITH BLINK INPUT)

```
V1 1 2  0.1166
R1 2 3  1.0
C1 2 3  0.2265F
R2 3 1 0.876
V2 1 4 2.14    VR2
R3 4 5 1.0
C2 5 1 0.0580F
I 1 1 6  1 VC2
I 2 6 1  1 VC4
C3 6 1 0.04459F
V3 1 7 1.0 VC3
R4 7 8  0.02275
C4 8 1 1.0F
V4 1 9 10.99 VC4
R5 9 10 1.0
C5 10 1 0.007496F
V5 1 11 50 VC5
R6 11 12 0.1
R7 12 13 14K
C6 13 14 1.6UF
R8 14 1 15.4K
R9 16 14 1M
R10 14 15 10K
R11 12 16 14K,
R12 15 1 200
I 3 1 15 500 VR9
R13 15 16 10K
C7 16 1 2.0UF
OUTPUT
VC7/VV1
PLOT(TY=IM/TI=0.5)
PLOT(TY=ST/TI=0.5)
PLOT(TY=SI/FR=4/TI=1.0/ST=.005)
EXECUTE
STOP
```

Figure 2 16 NASAP – 70 Input Coding

TRANSFER FUNCTION

$$\frac{-8.181E\ 06 - 1.853E\ 06S - 1.000E\ 00S^{2} + 0.0 + 0.0 + 0.0 + 0.0 + 0.0}{3.108E\ 09 + 9.319E\ 08S + 1.275E\ 08S^{2} + 1.051E\ 07S^{3} + 5.159E\ 05S^{4} + 1.438E\ 04S^{5} + 2.162E\ 02S^{6} + 1.000E\ 00S^{7}}$$

THE FUNCTION FACTOR = -2.294E 06

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	1.32969E 19	3.98672E 18	5.45604E 17	4.49549E 16	2.20682E 15	6.15044E 13
9.25033E 11	4.27796E 09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	-8.02895E 22	-1.81856E 22	-9.81459E 15	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

Figure 2 17

```
PLOT(TYPE=IMPULSE/TIME=0.5)

          SPECIFIED INPUT IS ---
                    IMPULSE, STRENGTH =  1.00000E 00

    THE TRANSFORM OF THE RESPONSE IS ---
```

EXP. OF S	NUMER. COEFFS.	DENOM. COEFFS.
0	8.02895E 22	1.32969E 19
1	1.81556E 22	3.98672E 18
2	9.91459E 15	5.49604E 17
3	0.0	4.49549E 16
4	0.0	2.20682E 15
5	0.0	6.15044E 13
6	0.0	9.25033E 11
7	0.0	4.27796E 09

SIGMA = 0.0

TIME	MAG	TIME	MAG	TIME	MAG	TIME	MAG
0.0	0.0	1.600E-01	4.56956E 04	3.200E-01	1.23072E 04	4.800E-01	-1.56420E 04
5.000E-03	9.31171E-02	1.650E-01	4.6[illegible]454E 04	3.250E-01	1.04229E 04	4.850E-01	-1.55119E 04
1.000E-02	2.53035E 00	1.700E-01	4.79564E 04	3.300E-01	8.58927E 03	4.900E-01	-1.53425E 04
1.500E-02	1.62587E 01	1.750E-01	4.88441E 04	3.350E-01	6.80616E 03	4.950E-01	-1.51365E 04
2.000E-02	5.84426E 01	1.800E-01	4.94865E 04	3.400E-01	5.07798E 03	5.000E-01	-1.48965E 04
2.500E-02	1.52772E 02	1.850E-01	4.99230E 04	3.450E-01	3.40863E 03	5.050E-01	-1.46249E 04
3.000E-02	3.26806E 02	1.900E-01	5.01555E 04	3.500E-01	1.80149E 03	5.100E-01	-1.43244E 04
3.500E-02	6.09139E 02	1.950E-01	5.01871E 04	3.550E-01	2.59541E 02	5.150E-01	-1.39974E 04
4.000E-02	1.02700E 03	2.000E-01	5.00229E 04	3.600E-01	-1.21471E 03	5.200E-01	-1.36463E 04
4.500E-02	1.60434E 03	2.050E-01	4.96690E 04	3.650E-01	-2.61918E 03	5.250E-01	-1.32736E 04
5.000E-02	2.36043E 03	2.100E-01	4.91229E 04	3.700E-01	-3.95218E 03	5.300E-01	-1.28816E 04
5.500E-02	3.30901E 03	2.150E-01	4.84231E 04	3.750E-01	-5.21241E 03	5.350E-01	-1.24726E 04
6.000E-02	4.45793E 03	2.200E-01	4.75488E 04	3.800E-01	-6.39895E 03	5.400E-01	-1.20488E 04
6.500E-02	5.80893E 03	2.250E-01	4.65203E 04	3.850E-01	-7.51119E 03	5.450E-01	-1.16124E 04
7.000E-02	7.35800E 03	2.300E-01	4.53481E 04	3.900E-01	-8.54890E 03	5.500E-01	-1.11653E 04
7.500E-02	9.09571E 03	2.350E-01	4.40432E 04	3.950E-01	-9.51212E 03	5.550E-01	-1.07098E 04
8.000E-02	1.10078E 04	2.400E-01	4.26172E 04	4.000E-01	-1.04012E 04	5.600E-01	-1.02475E 04
8.500E-02	1.30760E 04	2.450E-01	4.10814E 04	4.050E-01	-1.12167E 04	5.650E-01	-9.78046E 03
9.000E-02	1.52783E 04	2.500E-01	3.94477E 04	4.100E-01	-1.19595E 04	5.700E-01	-9.31032E 03
9.500E-02	1.75904E 04	2.550E-01	3.77274E 04	4.150E-01	-1.26307E 04	5.750E-01	-8.83880E 03
1.000E-01	1.99860E 04	2.600E-01	3.59322E 04	4.200E-01	-1.37316E 04	5.800E-01	-8.36744E 03
1.050E-01	2.24375E 04	2.650E-01	3.40734E 04	4.250E-01	-1.37637E 04	5.850E-01	-7.89777E 03
1.100E-01	2.49165E 04	2.700E-01	3.21619E 04	4.300E-01	-1.42286E 04	5.900E-01	-7.43118E 03
1.150E-01	2.73961E 04	2.750E-01	3.02086E 04	4.350E-01	-1.46282E 04	5.950E-01	-6.96901E 03
1.200E-01	2.98479E 04	2.800E-01	2.82236E 04	4.400E-01	-1.49644E 04	6.000E-01	-6.51245E 03
1.250E-01	3.22459E 04	2.850E-01	2.62171E 04	4.450E-01	-1.52393E 04	6.050E-01	-6.06268E 03
1.300E-01	3.45653E 04	2.900E-01	2.41985E 04	4.500E-01	-1.54551E 04	6.100E-01	-5.62074E 03
1.350E-01	3.67831E 04	2.950E-01	2.21768E 04	4.550E-01	-1.56140E 04	6.150E-01	-5.18762E 03
1.400E-01	3.88783E 04	3.000E-01	2.01605E 04	4.600E-01	-1.57186E 04	6.200E-01	-4.76420E 03
1.450E-01	4.08324E 04	3.050E-01	1.81577E 04	4.650E-01	-1.57712E 04	6.250E-01	-4.35129E 03
1.500E-01	4.26289E 04	3.100E-01	1.61758E 04	4.700E-01	-1.57742E 04	6.300E-01	-3.94964E 03
1.550E-01	4.42537E 04	3.150E-01	1.42219E 04	4.750E-01	-1.57303E 04	6.350E-01	-3.55987E 03

Figure 2 18

FOLDOUT FRAME 1

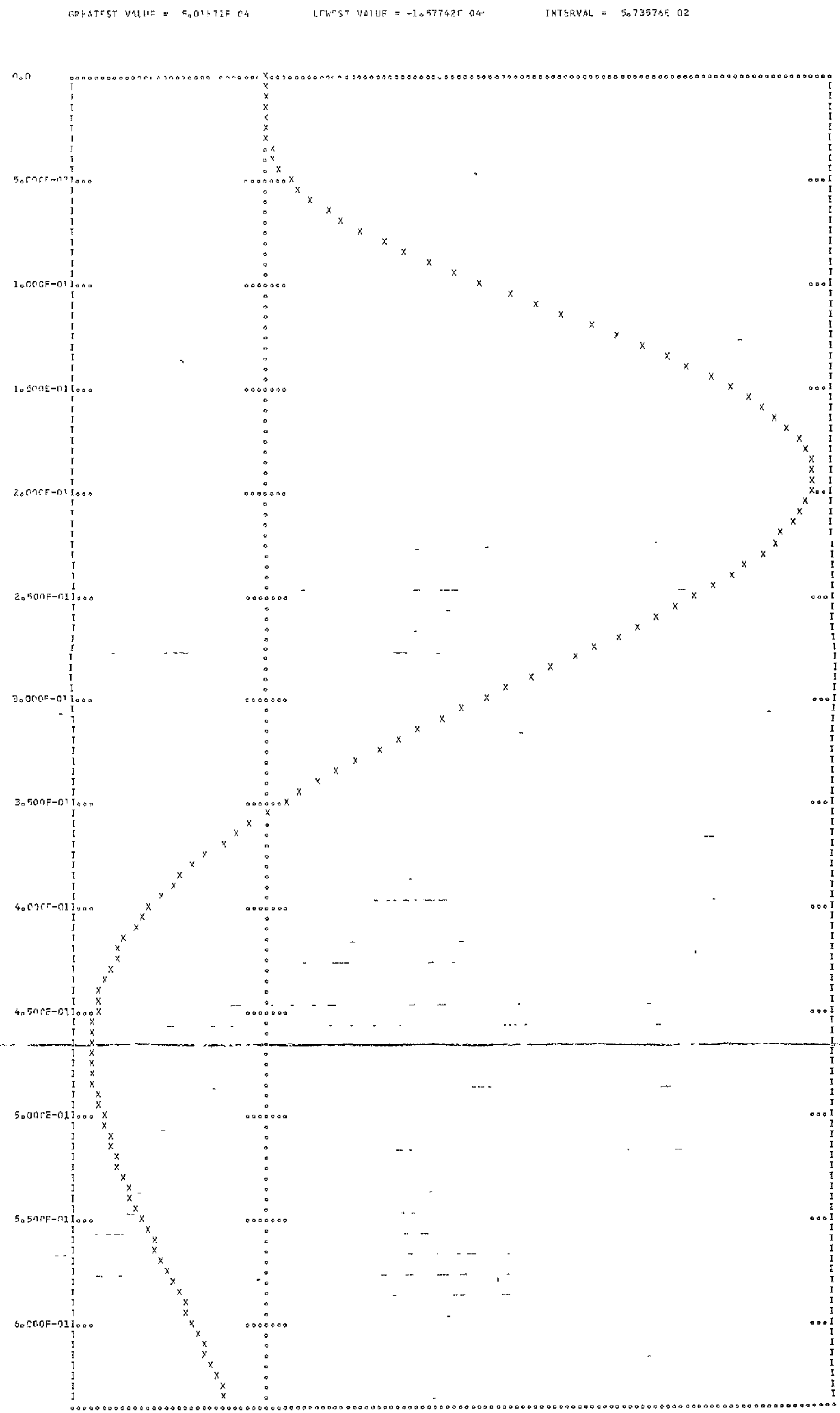


Figure 2 19

PRECEDING PAGE BLANK NOT FILMED

FOLDOUT FRAME 2

PLOT(TYPE=STEP/TIME=0.5)

SPECIFIED INPUT IS ---
STEP, MAGNITUDE = 1.00000E 00

THE TRANSFORM OF THE RESPONSE IS ---

EXP. OF S	NUMER. COEFFS.	DENOM. COEFFS.
0	8.02895E 22	0.0
1	1.81856E 22	1.32969E 19
2	9.81459E 15	3.98672E 18
3	0.0	5.45604E 17
4	0.0	4.49549E 16
5	0.0	2.20682E 15
6	0.0	6.15044E 13
7	0.0	9.25033E 11
8	0.0	4.27796E 09

SIGMA = 5.60000E 00

TIME	MAG	TIME	MAG	TIME	MAG	TIME	MAG
0.0	0.0	1.600E-01	2.55324E 03	3.200E-01	8.63470E 03	4.800E-01	7.45086E 03
5.000E-03	7.95470E-05	1.650E-01	2.78492E 03	3.250E-01	8.69150E 03	4.850E-01	7.37296E 03
1.000E-02	4.40618E-03	1.700E-01	3.02236E 03	3.300E-01	8.73902E 03	4.900E-01	7.29580E 03
1.500E-02	4.71056E-02	1.750E-01	3.26454E 03	3.350E-01	8.77748E 03	4.950E-01	7.21960E 03
2.000E-02	2.17228E-01	1.800E-01	3.51045E 03	3.400E-01	8.80717E 03	5.000E-01	7.14451E 03
2.500E-02	7.17766E-01	1.850E-01	3.75906E 03	3.450E-01	8.82836E 03	5.050E-01	7.07069E 03
3.000E-02	1.87731E 00	1.900E-01	4.00934E 03	3.500E-01	8.84136E 03	5.100E-01	6.99830E 03
3.500E-02	4.16672E 00	1.950E-01	4.26028E 03	3.550E-01	8.84648E 03	5.150E-01	6.92748E 03
4.000E-02	8.19458E 00	2.000E-01	4.51089E 03	3.600E-01	8.84406E 03	5.200E-01	6.85837E 03
4.500E-02	1.47021E 01	2.050E-01	4.76020E 03	3.650E-01	8.83444E 03	5.250E-01	6.79106E 03
5.000E-02	2.45357E 01	2.100E-01	5.00727E 03	3.700E-01	8.81799E 03	5.300E-01	6.72566E 03
5.500E-02	3.86268E 01	2.150E-01	5.25123E 03	3.750E-01	8.79504E 03	5.350E-01	6.66227E 03
6.000E-02	5.79600E 01	2.200E-01	5.49123E 03	3.800E-01	8.76598E 03	5.400E-01	6.60096E 03
6.500E-02	8.35431E 01	2.250E-01	5.72646E 03	3.850E-01	8.73118E 03	5.450E-01	6.54180E 03
7.000E-02	1.16379E 02	2.300E-01	5.95619E 03	3.900E-01	8.69100E 03	5.500E-01	6.48487E 03
7.500E-02	1.57438E 02	2.350E-01	6.17972E 03	3.950E-01	8.64581E 03	5.550E-01	6.43017E 03
8.000E-02	2.07627E 02	2.400E-01	6.39642E 03	4.000E-01	8.59599E 03	5.600E-01	6.37778E 03
8.500E-02	2.67775E 02	2.450E-01	6.60571E 03	4.050E-01	8.54191E 03	5.650E-01	6.32770E 03
9.000E-02	3.38609E 02	2.500E-01	6.80707E 03	4.100E-01	8.48395E 03	5.700E-01	6.27996E 03
9.500E-02	4.20741E 02	2.550E-01	7.00004E 03	4.150E-01	8.42245E 03	5.750E-01	6.23461E 03
1.000E-01	5.14653E 02	2.600E-01	7.18422E 03	4.200E-01	8.35777E 03	5.800E-01	6.19158E 03
1.050E-01	6.20693E 02	2.650E-01	7.35926E 03	4.250E-01	8.29025E 03	5.850E-01	6.15093E 03
1.100E-01	7.39073E 02	2.700E-01	7.52486E 03	4.300E-01	8.22024E 03	5.900E-01	6.11261E 03
1.150E-01	8.69862E 02	2.750E-01	7.68080E 03	4.350E-01	8.14807E 03	5.950E-01	6.07661E 03
1.200E-01	1.01299E 03	2.800E-01	7.82689E 03	4.400E-01	8.07406E 03	6.000E-01	6.04289E 03
1.250E-01	1.16825E 03	2.850E-01	7.96300E 03	4.450E-01	7.99853E 03	6.050E-01	6.01146E 03
1.300E-01	1.33532E 03	2.900E-01	8.08905E 03	4.500E-01	7.92176E 03	6.100E-01	5.98225E 03
1.350E-01	1.51373E 03	2.950E-01	8.20499E 03	4.550E-01	7.84408E 03	6.150E-01	5.95525E 03
1.400E-01	1.70294E 03	3.000E-01	8.31082E 03	4.600E-01	7.76572E 03	6.200E-01	5.93038E 03
1.450E-01	1.90228E 03	3.050E-01	8.40661E 03	4.650E-01	7.68690E 03	6.250E-01	5.90760E 03
1.500E-01	2.11100E 03	3.100E-01	8.49243E 03	4.700E-01	7.60809E 03	6.300E-01	5.88683E 03
1.550E-01	2.32829E 03	3.150E-01	8.56841E 03	4.750E-01	7.52931E 03	6.350E-01	5.86806E 03

Figure 2 20

PRECEDING PAGE BLANK NOT FILMED

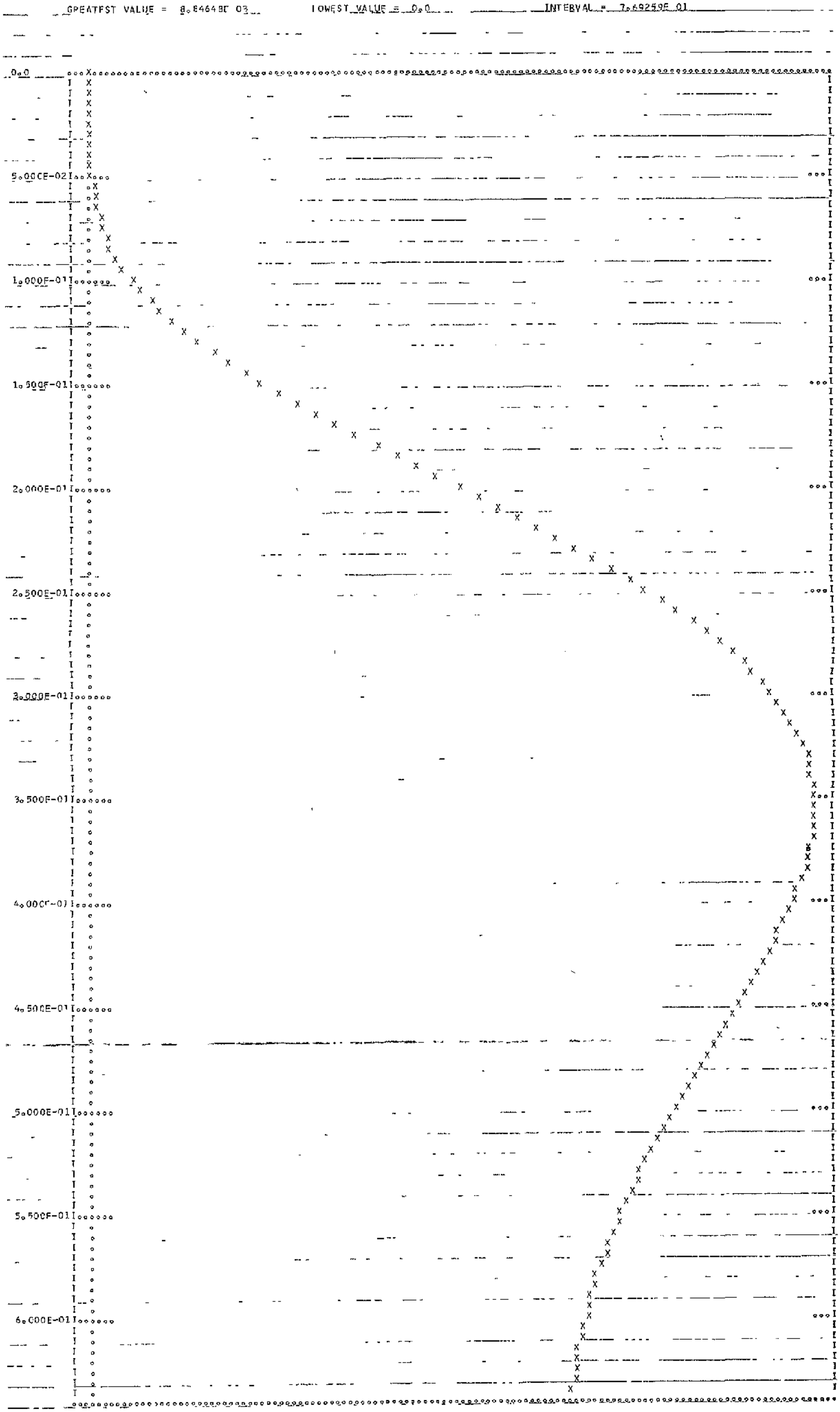


Figure 2.21

FOLDOUT FRAME 1

PRECEDING PAGE BLANK NOT FILMED

FOLDOUT FRAME 2

PLOT(TYPE=SINE/FREQUENCY=4/TIME=1.0/STEP=.005)

SPECIFIED INPUT IS ---
SINE, MAGNITUDE = 1.00000E 00 FREQUENCY = 4.00000E 00 HZ.

THE TRANSFORM OF THE RESPONSE IS ---

EXP. OF S	NUMER. COEFFS.	DENOM. COEFFS.
0	8.02855E 22	8.39907E 21
1	1.81856E 22	2.51873E 21
2	9.81459E 15	3.57930E 20
3	0.0	3.23827E 19
4	0.0	1.93955E 18
5	0.0	8.38044E 16
6	0.0	2.79112E 15
7	0.0	6.42066E 13
8	0.0	9.25033E 11
9	0.0	4.27756E 09

SIGMA = 2.30000E 00

TIME	MAG	TIME	MAG	TIME	MAG	TIME	MAG
0.0	0.0	3.200E-01	6.48115E 00	6.400E-01	-5.92769E 01	9.600E-01	2.62566E 01
5.000E-03	5.78735E-08	3.250E-01	-4.12480E 00	6.450E-01	-5.62316E 01	9.650E-01	3.24994E 01
1.000E-02	6.52122E-06	3.300E-01	-1.44053E 01	6.500E-01	-5.23699E 01	9.700E-01	3.82334E 01
1.500E-02	7.50872E-05	3.350E-01	-2.42441E 01	6.550E-01	-4.77439E 01	9.750E-01	4.33667E 01
2.000E-02	6.30756E-04	3.400E-01	-3.35306E 01	6.600E-01	-4.24182E 01	9.800E-01	4.78178E 01
2.500E-02	2.74332E-03	3.450E-01	-4.21613E 01	6.650E-01	-3.64687E 01	9.850E-01	5.15155E 01
3.000E-02	8.79142E-03	3.500E-01	-5.00418E 01	6.700E-01	-2.99814E 01	9.900E-01	5.44003E 01
3.500E-02	2.30886E-02	3.550E-01	-5.70982E 01	6.750E-01	-2.30515E 01	9.950E-01	5.64261E 01
4.000E-02	5.25084E-02	3.600E-01	-6.[illegible]2275E 01	6.800E-01	-1.57811E 01	1.000E 00	5.75598E 01
4.500E-02	1.07081E-01	3.650E-01	-6.83999E 01	6.850E-01	-8.27830E 00	1.005E 00	5.77832E 01
5.000E-02	2.00387E-01	3.700E-01	-7.25588E 01	6.900E-01	-6.55190E-01	1.010E 00	5.70920E 01
5.500E-02	3.49879E-01	3.750E-01	-7.56720E 01	6.950E-01	6.97385E 00	1.015E 00	5.54961E 01
6.000E-02	5.76874E-01	3.800E-01	-7.77217E 01	7.000E-01	1.44942E 01	1.020E 00	5.30207E 01
6.500E-02	9.06491E-01	3.850E-01	-7.87054E 01	7.050E-01	2.17923E 01	1.025E 00	4.97037E 01
7.000E-02	1.36727E 00	3.900E-01	-7.86352E 01	7.100E-01	2.87578E 01	1.030E 00	4.55973E 01
7.500E-02	1.99058E 00	3.950E-01	-7.75382E 01	7.150E-01	3.52855E 01	1.035E 00	4.07654E 01
8.000E-02	2.80956E 00	4.000E-01	-7.54556E 01	7.200E-01	4.12765E 01	1.040E 00	3.52843E 01
8.500E-02	3.86018E 00	4.050E-01	-7.24426E 01	7.250E-01	4.66400E 01	1.045E 00	2.92398E 01
9.000E-02	5.17628E 00	4.100E-01	-6.85671E 01	7.300E-01	5.12951E 01	1.050E 00	2.27269E 01
9.500E-02	6.79242E 00	4.150E-01	-6.39087E 01	7.350E-01	5.51712E 01	1.055E 00	1.58484E 01
1.000E-01	8.74080E 00	4.200E-01	-5.85575E 01	7.400E-01	5.82103E 01	1.060E 00	8.71202E 00
1.050E-01	1.10504E 01	4.250E-01	-5.26131E 01	7.450E-01	6.03669E 01	1.065E 00	1.43040E 00
1.100E-01	1.37461E 01	4.300E-01	-4.61825E 01	7.500E-01	6.16091E 01	1.070E 00	-5.88185E 00
1.150E-01	1.68470E 01	4.350E-01	-3.93786E 01	7.550E-01	6.19193E 01	1.075E 00	-1.31094E 01
1.200E-01	2.03662E 01	4.400E-01	-3.23188E 01	7.600E-01	6.12944E 01	1.080E 00	-2.01384E 01
1.250E-01	2.43093E 01	4.450E-01	-2.51228E 01	7.650E-01	5.97456E 01	1.085E 00	-2.68581E 01
1.300E-01	2.86740E 01	4.500E-01	-1.79109E 01	7.700E-01	5.72985E 01	1.090E 00	-3.31625E 01
1.350E-01	3.34493E 01	4.550E-01	-1.08023E 01	7.750E-01	5.39926E 01	1.095E 00	-3.89523E 01
1.400E-01	3.86151E 01	4.600E-01	-3.91318E 00	7.800E-01	4.98808E 01	1.100E 00	-4.41357E 01
1.450E-01	4.41424E 01	4.650E-01	2.64542E 00	7.850E-01	4.50283E 01	1.105E 00	-4.86314E 01
1.500E-01	4.99926E 01	4.700E-01	8.76864E 00	7.900E-01	3.95120E 01	1.110E 00	-5.23682E 01
1.550E-01	5.61184E 01	4.750E-01	1.43598E 01	7.950E-01	3.34191E 01	1.115E 00	-5.52870E 01
1.600E-01	6.24638E 01	4.800E-01	1.93319E 01	8.000E-01	2.68454E 01	1.120E 00	-5.73416E 01
1.650E-01	6.89644E 01	4.850E-01	2.36086E 01	8.050E-01	1.98943E 01	1.125E 00	-5.84997E 01
1.700E-01	7.55492E 01	4.900E-01	2.71259E 01	8.100E-01	1.26753E 01	1.130E 00	-5.87428E 01
1.750E-01	8.21404E 01	4.950E-01	2.98323E 01	8.150E-01	5.30148E 00	1.135E 00	-5.80667E 01
1.800E-01	8.86552E 01	5.000E-01	3.16904E 01	8.200E-01	-2.11170E 00	1.140E 00	-5.64819E 01
1.850E-01	9.50071E 01	5.050E-01	3.26769E 01	8.250E-01	-9.44775E 00	1.145E 00	-5.40131E 01
1.900E-01	1.01107E 02	5.100E-01	3.27830E 01	8.300E-01	-1.65924E 01	1.150E 00	-5.06993E 01
1.950E-01	1.0686[illegible]E 02	5.150E-01	3.20145E 01	8.350E-01	-2.34338E 01	1.155E 00	-4.65923E 01
2.000E-01	1.1[illegible]30E 02	5.200E-01	3.03916E 01	8.400E-01	-2.98650E 01	1.160E 00	-4.17566E 01
2.050E-01	1.16991E 02	5.250E-01	2.79488E 01	8.450E-01	-3.57860E 01	1.165E 00	-3.62683E 01
2.100E-01	1.21189E 02	5.300E-01	2.47338E 01	8.500E-01	-4.11048E 01	1.170E 00	-3.02140E 01
2.150E-01	1.24702E 02	5.350E-01	2.08073E 01	8.550E-01	-4.57386E 01	1.175E 00	-2.36883E 01
2.200E-01	1.27458E 02	5.400E-01	1.62412E 01	8.600E-01	-4.96158E 01	1.180E 00	-1.67942E 01
2.250E-01	1.29390E 02	5.450E-01	1.11181E 01	8.650E-01	-5.26770E 01	1.185E 00	-9.64023E 00
2.300E-01	1.30442E 02	5.500E-01	5.52999E 00	8.700E-01	-5.48749E 01	1.190E 00	-2.33877E 00
2.350E-01	1.30570E 02	5.550E-01	-4.24237E-01	8.750E-01	-5.61766E 01	1.195E 00	4.99527E 00
2.400E-01	1.29737E 02	5.600E-01	-6.63928E 00	8.800E-01	-5.65634E 01	1.200E 00	1.22462E 01
2.450E-01	1.27923E 02	5.650E-01	-1.30054E 01	8.850E-01	-5.60303E 01	1.205E 00	1.93003E 01
2.500E-01	1.25116E 02	5.700E-01	-1.94105E 01	8.900E-01	-5.45877E 01	1.210E 00	2.60468E 01
2.550E-01	1.21321E 02	5.750E-01	-2.57420E 01	8.950E-01	-5.22596E 01	1.215E 00	3.23789E 01
2.600E-01	1.16555E 02	5.800E-01	-3.18882E 01	9.000E-01	-4.90845E 01	1.220E 00	3.81974E 01
2.650E-01	1.10847E 02	5.850E-01	-3.77404E 01	9.050E-01	-4.51139E 01	1.225E 00	4.34106E 01
2.700E-01	1.04242E 02	5.900E-01	-4.31946E 01	9.100E-01	-4.04121E 01	1.230E 00	4.79368E 01
2.750E-01	9.67559E 01	5.950E-01	-4.81533E 01	9.150E-01	-3.50548E 01	1.235E 00	5.17048E 01
2.800E-01	8.85772E 01	6.000E-01	-5.2[illegible]264E 01	9.200E-01	-2.91278E 01	1.240E 00	5.46549E 01
2.850E-01	7.96663E 01	6.050E-01	-5.62339E 01	9.250E-01	-2.27262E 01	1.245E 00	5.67415E 01
2.900E-01	7.01535E 01	6.100E-01	-5.92058E 01	9.300E-01	-1.59523E 01	1.250E 00	5.79315E 01
2.950E-01	6.01386E 01	6.150E-01	-6.13844E 01	9.350E-01	-8.91448E 00	1.255E 00	5.82065E 01
3.000E-01	4.97289E 01	6.200E-01	-6.27245E 01	9.400E-01	-1.72511E 00	1.260E 00	5.75626E 01
3.050E-01	3.90382E 01	6.250E-01	-6.31544E 01	9.450E-01	5.50112E 00	1.265E 00	5.60096E 01
3.100E-01	2.81856E 01	6.300E-01	-6.27763E 01	9.500E-01	1.26489E 01	1.270E 00	5.35723E 01
3.150E-01	1.72922E 01	6.350E-01	-6.14668E 01	9.550E-01	1.96042E 01	1.275E 00	5.02900E 01

Figure 2 22a

PRECEDING PAGE BLANK NOT FILMED

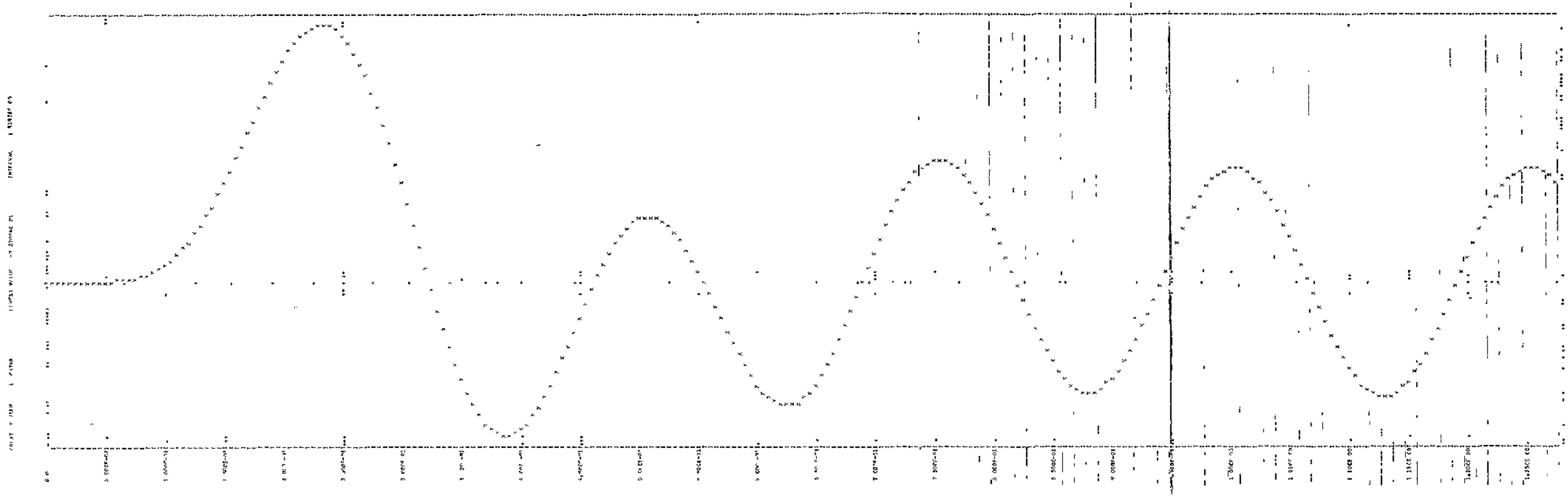


Figure 2 22b

FOLDOUT FRAME 1

FOLDOUT FRAME

PRECEDING PAGE BLANK NOT FILMED

FOLDOUT FRAME 2

```
                  TITLE CARD
CONTROL CARD →    NASAP
                  VINPUT 1 2 1
                  IOUTPUT 3 4 1 IR1
                  R1 2 4 .37M
                  C23 2 3 2E10 PF
                  LO 1 4 33
CONTROL CARD →    OUTPUT                                        1st Problem
                  WORST CASE
                  TOL=LO= 50
                  V I OUTPUT/VVINPUT
                  ROOTS, POLES
                  PLOT (TYPE = IMPULSE/TIME = 10)
                  PLOT (TYPE = FR/FR = .01/TO = 1E2)
                  PLOT (TY=WORST/FR= .01/TO = 1E2)
                  VR←/VVINPUT
CONTROL CARD →    EXECUTE
                  ANOTHER OPTIONAL TITLE CARD
CONTROL CARD →    TREE
                  E1 (1-2) = 1
                  DJ2/IRE1 (3-4) = 1
                  RE1 (2-4) = .37M
                  CE, (2-3) = 2E10PF
                  LJ, (1-4) = 33
CONTROL CARD →    END
                  SENS=RE1
                  SENS=LJ                                         2nd Proble
                  VLJ/VE1
                  PLOT (TY = SI/AM = 2/FR = 10)
                  PLOT (TY=SE/FR=1E-3/TO=1E6/EL=RE1)
                  END
```

PRECEDING PAGE BLANK NOT FILMED

```
                          CIRC (5E6)
                          E1 (1-2) = 1
                          R2 (2-3) = 2
                          L3 (3-1) = 1
                          C4 (3-1) = 4
CONTROL CARD →            END                    3rd Problem
                          IC4/VE1
                          ROOTS
CONTROL CARD →            END
                          STOP
```

Figure 2 23   Sample Data Decks for NASAP–70

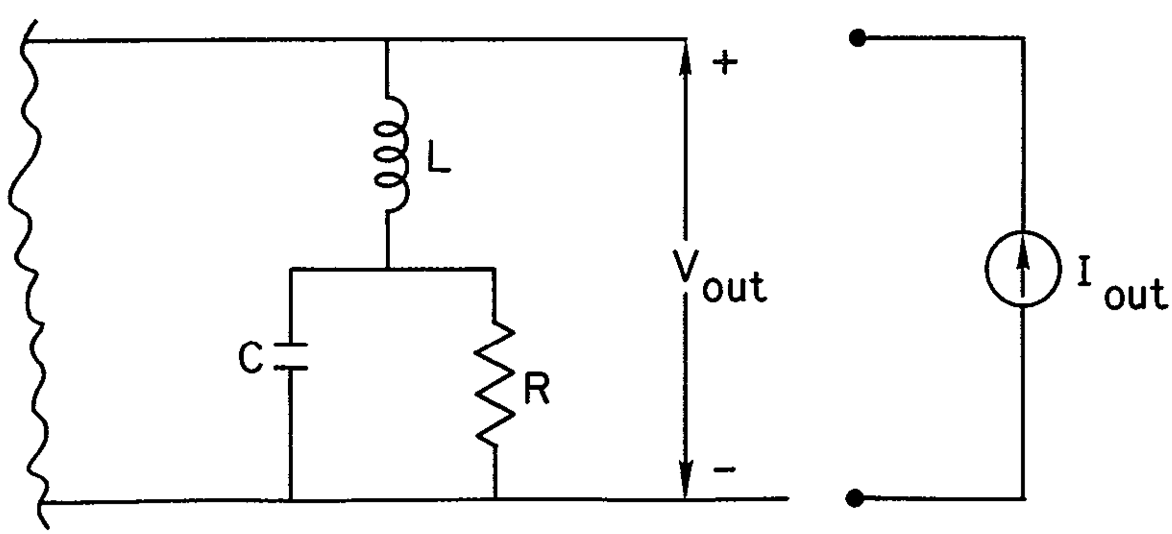


Figure 2 24   Example of Outputs Taken Across Two or More Circuit Elements

2.4 Coding Tips

As is usually the case with any application program, a number of design tricks will evolve as the circuit designers become familiar with its operation. Some of these tips are described here.

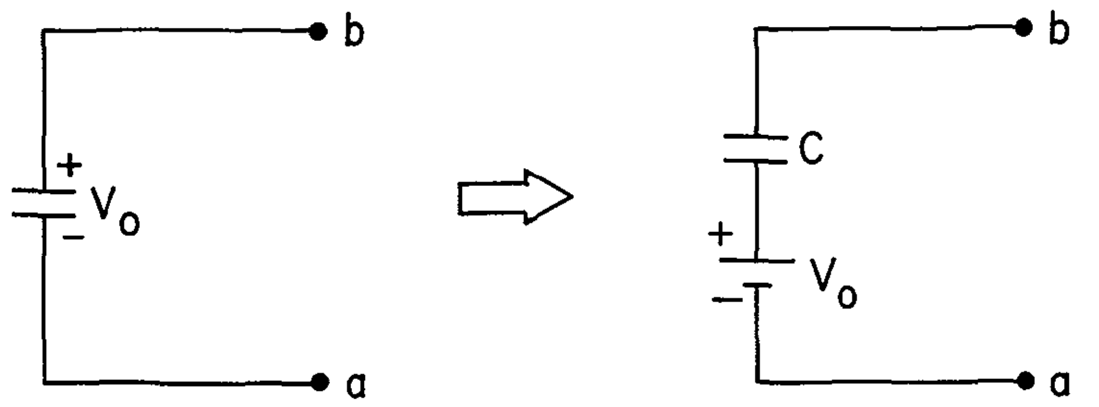
Outputs Taken Across Two or More Circuit Elements

Quite often a circuit designer needs the transfer function expression taken across more than one circuit element, and with the capabilities of NASAP-70, as many computer runs as there are circuit output elements will be required. To circumvent this difficulty, an extra current source can be inserted across the desired output terminals and the output voltage measured across it (see for example Figure 2.24). This element can be viewed as an ideal voltmeter, that is, a device that extracts a constant value (which is always zero) of current from the circuit. By adding the output current source (I = 0), provision has been made for a general method of solution.

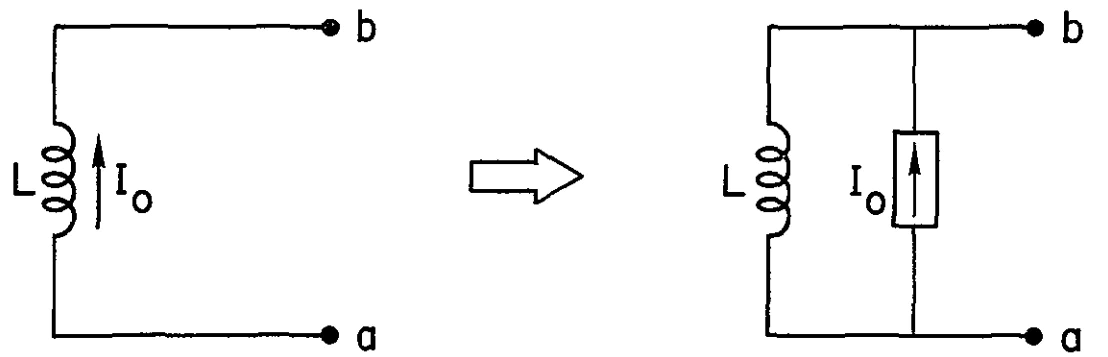
(As an added note, an ideal ammeter which is a short circuit is a voltage source whose voltage is a constant value of zero.)

Total Output Response

This design tip is really a reminder to the circuit designer of how the output response of a circuit should be handled when a number of active devices and energy storage elements are present. For such cases NASAP-70 requires that as many transfer functions as there are active devices and elements with energy stored in them be computed, relating these devices to the output requested. The transient response associated for each transfer function is then computed and summed all together to form the total output response. It is important to note that any charged capacitor with V_0 volts has to be replaced by an uncharged capacitor in series with a battery of V_0 volts



Also, any inductor L with an initial current of I_o through it has to be replaced by an inductor placed in parallel with a current source I_o



These sources are considered like any other active device, and thus their presence must be reflected in the desired output response.

There are approximation-oriented guidelines by which a tree selection procedure should be implemented. For low frequency approximation, capacitors should be chosen as links while inductors should be chosen branches (as much as possible while still retaining a tree structure.) Conversely, high frequency approximation results in inductors chosen as links while capacitors are chosen as branches. The practical underlying reason is that while a zero is well represented and well behaved on a computer, $\infty \leftarrow 1/0$ causes error and havoc on the machine and thus will not be accepted by the algorithm This means that if one requires low as well as high frequency approximations for the same problem, this problem should be submitted as two problems with two different trees Similarly, elements with relatively large impedances compared with the circuit (for example, grid-plate tube impedance) should always be chosen links. (To set $I \leftarrow 0$ which means grid current equal zero.) On the other hand, elements with relatively small input impedances (r_i in some transistors) should always be chosen branches (To set $V \leftarrow 0$).

In an approximation, branches can be collapsed or links can be opened without disturbing the tree structure. This fact can be verified by observing the example problems. Hence, by carefully selecting elements to be links or branches, foresight in design can be incorporated into circuit description and a great deal of flexibility in calculations can be achieved.

2.5 NASAP Analysis of Nonlinear DC Circuits*

The basic procedure for analyzing four classes of nonlinear dependencies is described (1) Nonlinear controlled source dependent on current or voltage in the circuit, (2) Nonlinear controlled source dependent on a parameter external to the circuit such as temperature, (3) Nonlinear resistor dependent on current or voltage associated with its terminals, and (4) Nonlinear resistor dependent on a parameter external to the circuit.

A method for obtaining models for nonlinear elements is presented and is used to establish a model for an exponential nonlinearity This model is used in the analysis of transistor bias circuits.

Also included are the analysis procedures for power supply regulator circuits using the real s evaluation option. This analysis produces results related to the load and line regulation properties of the regulator.

---

*This section has been taken from the work presented in references 9 and 10.

2.5.1 Modeling Circuit Element Variations

Linear and nonlinear circuit element variations in a resistive circuit can be simulated with the use of the PLOT(TYPE=RE) capabilities of the NASAP program which results in a real variable evaluation of the transfer function. Consider a variable circuit element A(x) where A may be a resistor, a current source, or a voltage source and x may be a current, a voltage, or some external parameter such as temperature. The function A(x) is approximated by a ratio of polynominals using either approximation methods or trial and error.

$$A(x) = \frac{P(x)}{Q(x)} \tag{2.1}$$

The variable x is replaced by s and a circuit is synthesized which has a transfer function or an input immitance given by P(s)/Q(s). When the TYPE=REAL option is used, this circuit will simulate the variation of the circuit element A(x) if the range of s is the same as the expected range of x. It should be noted that since the circuit is used for simulation purposes only, it does not have to be physically realizable, i.e., controlled sources or negative circuit elements may be used.

The method for using this model in computer analysis of a circuit depends on the circuit element and variable which A and x represent. In Cases I and II of the discussion which follows it is assumed that P(s)/Q(s) is realized as an input impedance. If the realization is an input admittance or a transfer function, relatively minor modifications are required.

Case I

A is a controlled voltage or a current source and x is a voltage or a current in the circuit.

The dependency function A(s) is synthesized as an impedance and a current generator of value unity is connected to the impedance. This auxilliary network is hinged to the network which is to be analyzed. The nonlinear dependent source in the network is replaced by a source which is dependent on the voltage across the auxilliary network. The transfer function x/independent source is specified and the TYPE=REAL option is used. The solution for the circuit is obtained from the tabular printout or the graphical plot as the value where x = s. If x is not the desired output variable, the circuit is recoded with A(x) replaced by its known value from the above solution. The auxilliary network is eliminated and the desired output variable is specified.

This analysis is applicable when the circuit contains no more than one variable source.

To illustrate the method, consider the circuit of Figure 2.25. Let R = 2 and assume that V is a function of I_1 and is adequately approximated by

$$V(I_1) = \frac{I_1^2 + I_1 - 5}{I_1 + 1} \quad 0 \leq I_1 \leq 1 \tag{2.2}$$

A pencil and paper solution results in

$$I_1 = .921 \quad , \quad V_o = .0795 \tag{2.3}$$

For computer analysis an impedance is synthesized for which

$$Z(s) = \frac{s^2 + s - 5}{s + 1} = s - \frac{5}{s + 1} \tag{2.4}$$

A circuit realization of this impedance is shown in Figure 2.26. The auxilliary network is hinged to the basic circuit with R2 = 2, I1 = IR7 and V2 = VR6. Note that the R5, R6 resistance combination is used for voltage measurement only and could be replaced by a single, large valued resistance. The transfer function IR1/VV1 is specified and a frequency of .915 to .925 in steps of .001 is specified. The frequency range is chosen on the basis of the expected "ballpark" value of the current IR1. If a good estimate of IR1 cannot be made, a larger frequency range is specified with larger

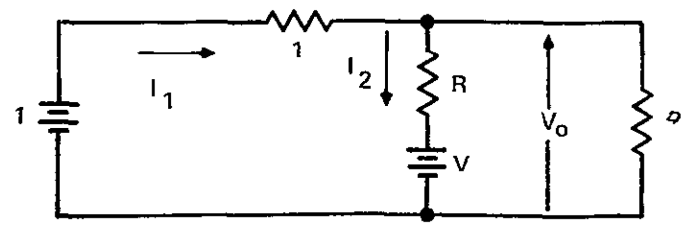


Figure 2 25 Circuit for Example

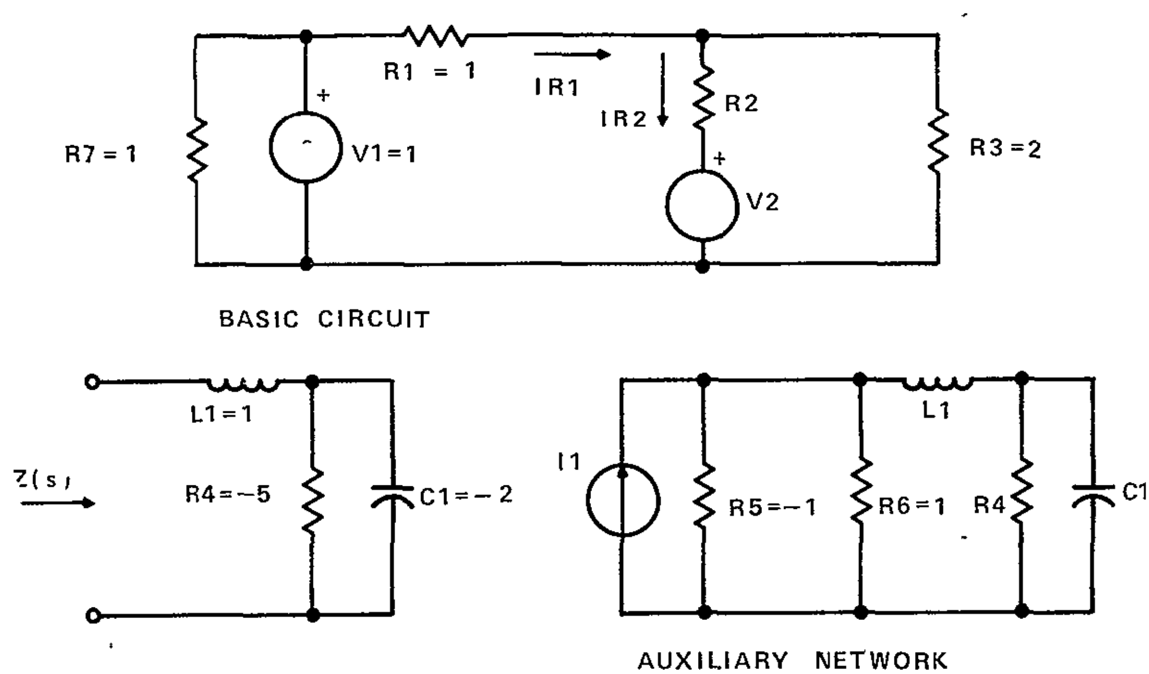


Figure 2 26 Computer Analysis Models

increments and more than one pass on the computer may be necessary to obtain IR1 to the desired degree of accuracy.

The computer analysis results in

$$\frac{IR1}{VV1} = -.25 \ \frac{-7 - s + s^2}{1 + s} \tag{2.5}$$

and the printout table gives IR1 = s for s = .921. The value of V2 is then obtained as V2 = -1.682. The basic circuit is then coded using this value of V2 and the result is VR3/VV1 = .0795.

Case II

A is a controlled voltage or current source and x is an external parameter which is known to vary over a specified range.

In this case, the controlled source is generally dependent on a current or voltage in the circuit (denoted by y in this discussion) and the dependency factor is a function of an external parameter x. The $h_{FE}I_B$ generator in the transistor dc equivalent circuit where h_{FE} is temperature dependent is an example of this situation. The circuit is coded with yA(x) replaced by a generator, V(s), which is dependent on the voltage across the auxilliary network. The desired transfer function is specified and the SIGMA option is used. The computer solution gives the relation between the output variable and x in functional (replace s by x), tabular and graphical forms. This method is applicable for any number of variable sources in the circuit which depend on the same external parameter x.

The circuit of Figure 2.25 is used to illustrate the procedure with R = 2 and V dependent on I_1 and temperature according to

$$V = I_1 \frac{T^2 + T - 5}{T + 1} \qquad 10 \leq T \leq 50 \tag{2.6}$$

A hand solution results in

$$V_o = \frac{T^2 + 3T - 3}{T^2 + 5T - 1} \tag{2.7}$$

For computer analysis the nonlinear dependency has the same functional form as in Case I and therefore Z(s) of Figure 2.26 simulates the temperature dependency. The auxilliary network is hinged to the basic circuit with I1 = IR1 and V2 = VR6. The transfer function VR3/VV1 is requested and a frequency range of 10 to 50 in steps of 1 is specified. The result is

$$\frac{VR3}{VV1} = \frac{-3 + 3s + s^2}{-1 + 5s + s^2} \tag{2.8}$$

A table is printed out and a graph of the function over the range $s = 10$ to $s = 50$ is plotted.

Case III

A is a resistor and x is the voltage across or the current through the resistor.

The circuit is coded with R(x) replaced by Z(s). The transfer function x/independent source is specified and the TYPE=REAL option is used. The solution for the circuit is obtained from the tabular printout or the graphical plot as the value where $x = s$. If x is not the desired output variable, the circuit is recoded with R(x) replaced by its known value and the desired transfer function is specified. This analysis is applicable when the circuit contains no more than one nonlinear resistance.

In the circuit of Figure 2.25, let $V = -2$ and R be dependent on I_2.

$$R(I_2) = \frac{I_2^2 + I_2 - 5}{I_2 + 1} \quad 2 \le I_2 \le 4 \tag{2.9}$$

The solution is

$$I_2 = 2.155 \ , \quad V_o = -.771 \tag{2.10}$$

For computer analysis, the basic circuit of Figure 2.26 is used with R2 replaced by Z(s) and V2 = -2 IR7. The transfer function IL1/VV1 is requested and a frequency range of 2.15 to 2.16 in steps of .001 is specified. As in Case I, it may require more than one pass on the computer to determine the proper frequency range and increment which will result in the desired degree of accuracy for IL1. The computer analysis gives

$$\frac{IL1}{VV1} = 2.67 \ \frac{1 + s}{-4.333 + 1.667s + s^2} \tag{2.11}$$

for which IL1 = s at s = 2.155. The value of $R(I_2)$ is evaluated as $R(I_2) = .5702$. The basic circuit is recoded using this value of R and the above value for V. The analysis results in VR3/VV1 = -.771.

Case IV

A is a resistor and x is an external parameter which is known to vary over a specified range.

The circuit is coded with R(x) replaced by Z(s) and the desired transfer function is specified. The computer solution gives the relation between the output variable and x. This method is applicable for any number of variable resistances in the circuit which depend on the same external parameter x.

As an example, let V = 2 and R be a function of temperature in the circuit of Figure 2.25.

$$R(T) = \frac{T^2 + T - 5}{T + 1} \qquad 10 \le T \le 50 \tag{2.12}$$

A hand solution gives

$$V_o = .667 \frac{T^2 + 3T - 3}{T^2 + 1.667T - 4.333} \tag{2.13}$$

In the computer analysis, the basic circuit of Figure 2.26 is used with R2 replaced by Z(s) and V2 = 2IR7. The frequency range 10 to 50 in steps of 1 is specified and the computer output is

$$\frac{VR3}{VV1} = .667 \frac{-3 + 3s + s^2}{-4.333 + 1.667s + s^2} \tag{2.14}$$

A table is printed out and a graph of the function over the range s = 10 to s = 50 is plotted.

2.5.2 Modeling Temperature Variations of Transistor Parameters

The parameters of the transistor dc equivalent circuit, h_{FE}, V_{BE} and I_{co}, are temperature dependent. The functional form of the dependence may vary depending on the construction of the transistor. Generally, graphical plots of the dependencies are available from the manufacturer on request.

Analytical investigations predict the following nominal temperature variations.

$$h_{FE}(T) = h_{FEo}(1 - 25\ B + BT) \tag{2.15}$$

where h_{FEo} is the value of h_{FE} aat $25^{\circ}C$

T is expressed in $^{\circ}C$

B = .02 for Ge and .013 for Si

$$V_{BE}(T) = V_{BEo} + 25D - DT \tag{2.16}$$

where V_{BEo} is the value of V_{BE} at $25^{\circ}C$

D = .002 to .0025

$$I_{co}(T) = I_{coo} 2^{\frac{T-25}{F}} \tag{2.17}$$

where I_{coo} is the value of I_{co} at $25^{\circ}C$

F = 10 for Ge and 7 for Si

The model for temperature variation of h_{FE} requires

$$Z(s) = 1 - 25B + Bs \tag{2.18}$$

while the V_{BE} model must satisfy

$$Z(s) = V_{BEo} + 25D - Ds \tag{2.19}$$

Simple series RL circuits will satisfy these relations with R = 1 - 25B and L = B for the h_{FE} model and R = V_{BEo} + 25D and L = - D for the V_{BE} model. The current generators used to complete the auxilliary networks will have value $h_{FEo}\ I_B$ for the h_{FE} model and unity for the V_{BE} model.

The circuit to simulate I_{co} variation is somewhat more complicated than the h_{FE} and V_{BE} models. As indicated by Equation (2.17), I_{co} doubles every $10^{\circ}C$ for germanium and every $7^{\circ}C$ for silicon. A trial and error method results in the following approximating functions for the temperature range $0^{\circ}C$ to $75^{\circ}C$.

$$I_{coGe}(T) = I_{coo}\ 128.485\ \frac{T + .471698}{T^2 - 159.394T + 6632.58} \tag{2.20}$$

$$I_{coSi}(T) = I_{coo}\ 202.650\ \frac{T - 12.5480}{T^2 - 149.500T + 5690.00} \tag{2.21}$$

Figure 2.27 shows a possible synthesis of the normalized approximating function $I_{co}(s)/I_{coo}$. The values of the elements are listed below.

	Ge	Si
C	.007783	.0049346
R_1	-.8037	-1.4797
R_2	.0090349	-.640268
L	.019154	.0510255

Although these circuits model temperature variation of I_{co}, they do not satisfy the transistor model since $(h_{FE}+1)I_{co}$ is required and h_{FE} is also temperature dependent. This product can be modeled with the configuration shown in Figure 2.28 If $\frac{1}{Y_h(s)} >> Z(s)$ for the real s range of interest, then

$$I = h_{FEo}\ Y_h(s)\ Z_{I_{co}}(s) \tag{2.22}$$

The control variable which is required for the transistor model is

$$I = \left[h_{FE}(s) + 1\right] I_{co}(s) \tag{2.23}$$

since $Z_{I_{co}}(s) = I_{co}(s)/I_{coo}$, the correspondence is

$$h_{FEo} Y_h(s) = \left[h_{FE}(s) + 1\right]\ I_{coo}$$

or

$$Y_h(s) = \frac{\left[h_{FE}\ (1 - 25B + Bs) + 1\right] I_{coo}}{h_{FE}} \tag{2.24}$$

A summary of the models is presented in Figure 2.29 where a complete transistor circuit is modeled. The V1, R1 combination is included

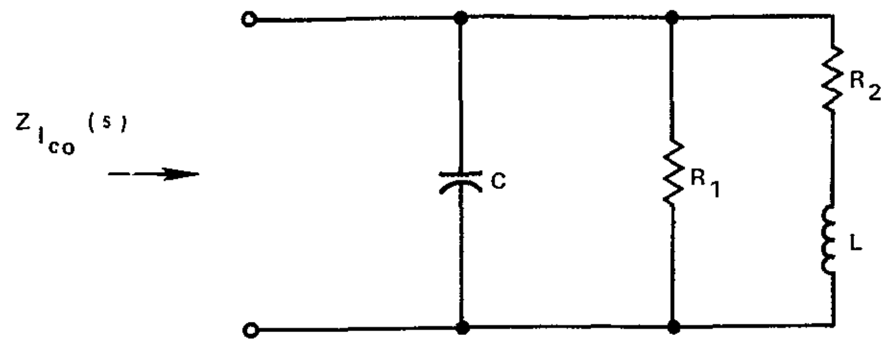


Figure 2 27 Model for I_{co} Temperature Variation

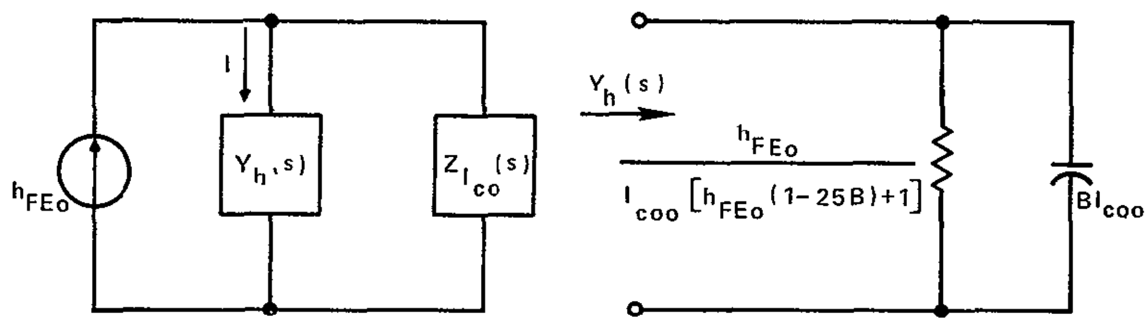
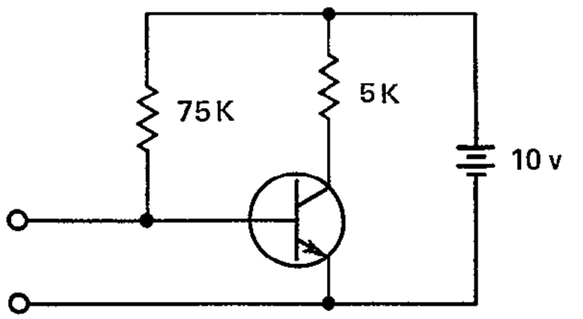


Figure 2 28 Configuration to Obtain Product Relation

in this model so that V2, I4 and I5 can be constant amplitude controlled sources. If a plot of the collector current is required, the transfer function IR5/VV1 is specified. A typical printout for the circuit shown in Figure 2.29 using the (TYPE=REAL) option is included in Appendix C.

2.5.3 Regulation Curves for Power Supply Regulators

The TYPE=REAL option of the NASA-70 program is useful in obtaining power supply regulation curves. In the circuit of Figure 2.30, R2 is a large valued resistor which is used for measuring voltage across the auxilliary network. The voltage across R1 is constant, the voltage across L1 varies linearly with s, and the voltage across R2 is the sum of VR1 and VL1. For



GERMANIUM TRANSISTOR

h_{ie} = 1K

h_{oe} = 25 x 10^{6} mho

h_{FEo} = 50

V_{BEo} = 2 v

I_{coo} = 1 μa

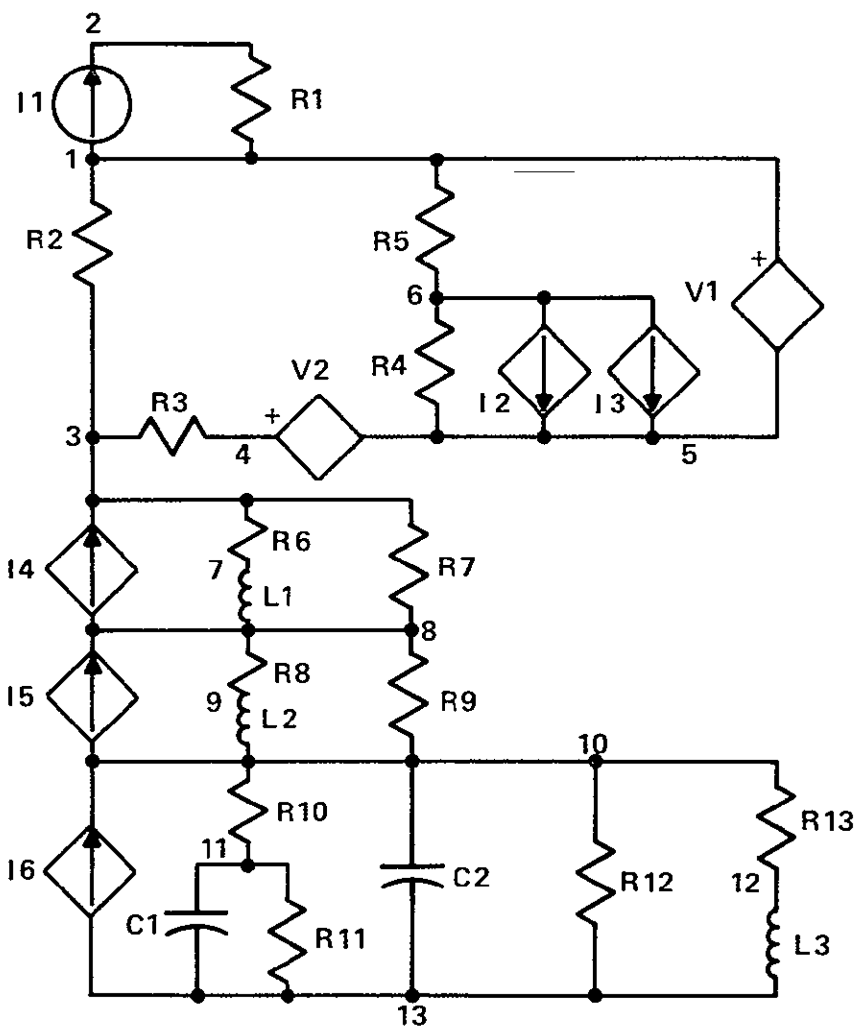


Figure 2 29 Model for Simulation of Temperature Variation Effects in a Transistor Circuit
*Resistors Included in Model for Voltage Measurement
**Resistor Included in Model for Current Measurement

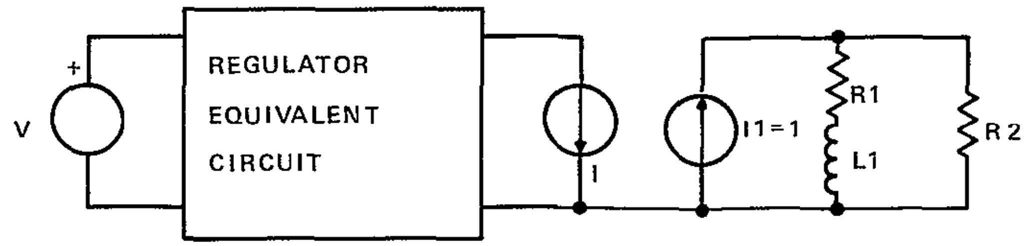


Figure 2 30 Power Supply Regulator Circuit Analysis

load regulation data, V is made dependent on VR1 and I is made dependent on VL1. Line regulation information is obtained if V depends on VR2 and I depends on VR1. If V depends on VR2 and I depends on VL1, a regulation curve is obtained for simultaneous variation of line voltage and load current.

2.6 Network Partitioning Schemes for NASAP-70[11]

Assume a general network is partitioned into several smaller subnetworks as shown in Figure 2.31. The input and output variables, X_1 and X_2, can be located arbitrarily. Note that the partitioning conveniently lends itself to transfer function formulations Subnetwork 1 ports are identified by X_1, 1_2, and 1_3, the other subnetwork ports are numbered similarly.

2.6.1 N-Port Interconnection Methods

Some interconnection methods are briefly described here to make the discussion self-contained. A detailed description can be found in reference 12.

Laemmel Method The Laemmel Method[13] is employed to connect two subnetworks in cascade as shown in Figure 2.31. This technique utilizes the open circuit impedances of the network. The input and output ports of N_1 and N_2 are in groups of a and b, respectively. The network N_c obtained by cascading N_1 and N_2 has its input and output ports in group a of N_1 and group b of N_2, respectively.

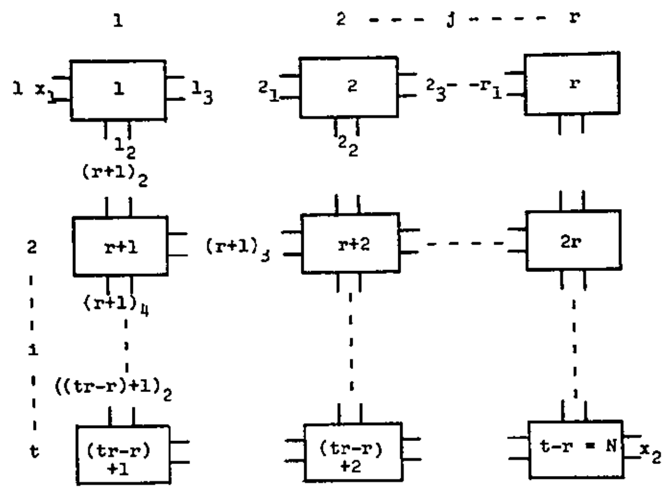


Figure 2 31 A Complex Network Partitioned into Several Smaller Networks

The impedance matrix of subnetwork N_1 is defined as

$$U = \begin{bmatrix} U^{aa} & U^{ab} \\ U^{ba} & U^{bb} \end{bmatrix} \tag{2.25}$$

where

$$U^{aa} = (u_{ij}\ i=1\ ---q,\ j=1\ ---q),\quad U^{ab} = (u_{ij}\ i=1\ ---q,\ j=q+1\ ---r).$$

$$U^{ba} = (u_{ij}\ i=q+1\ ---r,\ j=1\ ---q),\quad U^{bb} = (u_{ij}\ i=q+1\ ---r,\ j=q+1\ ---r)$$

are the open circuit impedances relating ports of the subnetworks. Note that the U matrix, and succeeding matrices, are partitioned according to the grouping of the input and output ports.

In a similar manner, the open loop impedance matrix for N_2 is

$$T = \begin{bmatrix} T^{aa} & T^{ab} \\ T^{ba} & T^{bb} \end{bmatrix} \tag{2.26}$$

where

$$T^{aa} = (t_{ij}\ i=1\ ---s,\ j=1\ ---s),\quad T^{ab} = (t_{ij}\ i=1\ ---s,\ j=s+1\ ---w).$$

$$T^{ba} = (t_{ij}\ i=s+1 \text{ ---} w,\ j=1 \text{ ---} s),\quad T^{bb} = (t_{ij}\ i=s+1 \text{ ---} w,\ j=s+1 \text{ ---} w)$$

are open circuit impedances relating ports in N_2.

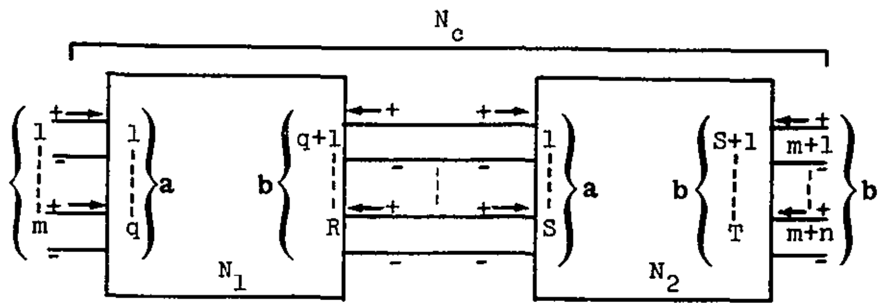


Figure 2.32 Connecting Two N Port Subnetworks in Cascade

The overall open circuit impedance of N_c, obtained by interconnecting N_1 and N_2 (Figure 2.32), can be computed by Laemmel's relations

$$Z^{aa} = U^{aa} - U^{ab}\,(U^{bb} + T^{aa})^{-1}\,U^{ba} \tag{2.27}$$

$$Z^{ab} = U^{ab}\,(U^{bb} + T^{aa})^{-1}\,T^{ab} \tag{2.28}$$

$$Z^{ba} = T^{ba}\,(U^{bb} + T^{aa})^{-1}\,U^{ba} \tag{2.29}$$

$$Z^{bb} = -\,T^{ba}\,(U^{bb} + T^{aa})^{-1}\,T^{ab} + T^{bb} \tag{2.30}$$

where

$$T = \begin{bmatrix} Z^{aa} & Z^{ab} \\ Z^{ba} & Z^{bb} \end{bmatrix} \tag{2.31}$$

Laemmel's relations can be applied repeatedly to interconnect subnetworks of the form shown in Figure 2.31. For example, consider the first three subnetworks of row 1. Subnetworks 1 and 2 are combined to form subnetwork 1, 2 as in Figure 2.33(b). Subnetwork 1, 2 is rearranged and connected with subnetwork 3 to form subnetwork 1, 2, 3 of Figure 2.33(d). This process is continued until all subnetworks of row 1 have been interconnected. In an identical manner, the subnetworks of rows 2 are interconnected, producing t subnetworks. This is depicted in Figure 2.34 The subnetworks are interconnected to reconstruct the original network.

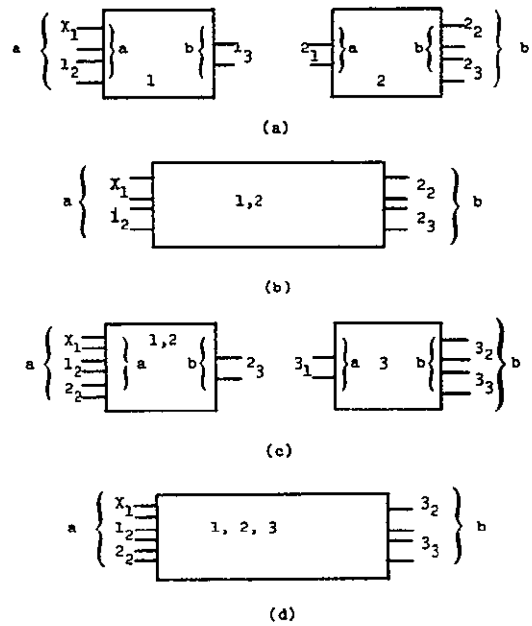


Figure 2 33 Interconnection of Subnetworks

The total number of times the Laemmel method is applied in forming the original network from the group of subnetworks given in Figure 2 31 is tr-1, where t is number of rows and r is the number of columns of the subnetworks.

The following guidelines should be adhered to in subdividing a network for an analysis by the Laemmel Method

1. No mutual coupling should exist between components of different subnetworks,
2. A dependency relationship must be contained within a subnetwork,
3. The port of a subnetwork must be taken across an element which is part of a closed mesh, such as shown in Figure 2. 35. (This restriction is due to NASAP-70 rather than to the methods themselves).

Cascade Parameter Method When a network can be partitioned into cascade subnetworks as shown in Figure 2. 36, the cascade parameter method[14] is usually employed to compute transfer functions rather than the Laemmel method because of the reduction in computing time achieved. The cascade parameters relate the terminal voltages and currents for the overall network by the expression

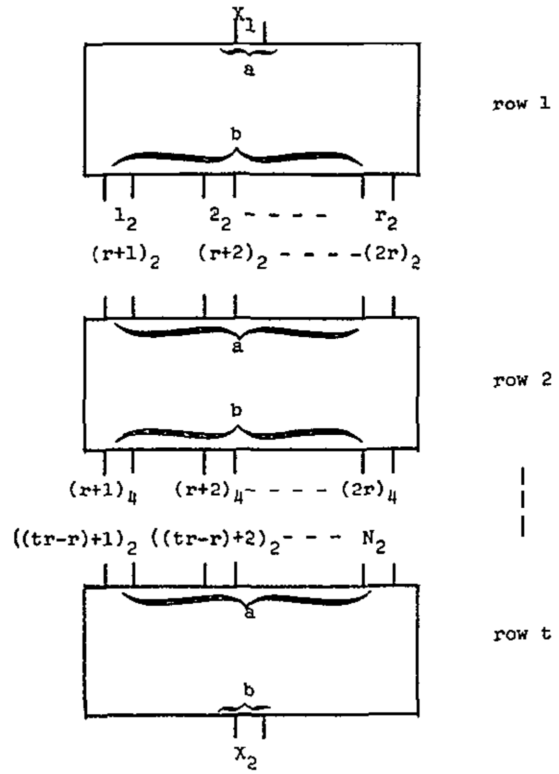


Figure 2 34 Subnetworks of the Overall Network

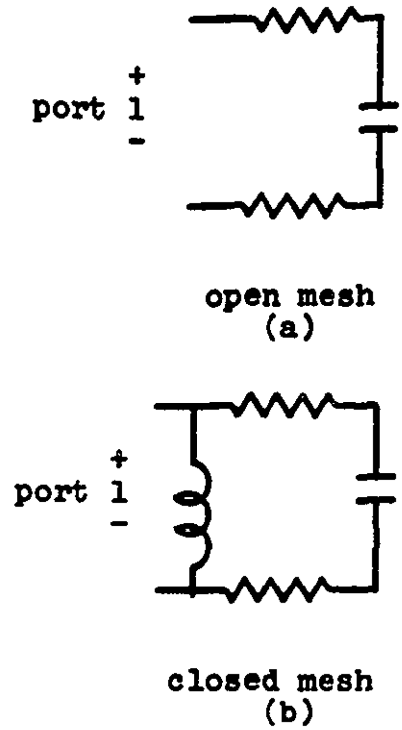


Figure 2 35 A Mesh of a Subnetwork

$$\begin{vmatrix} V_I \\ I_I \end{vmatrix} = \begin{vmatrix} A & B \\ C & D \end{vmatrix} \begin{vmatrix} V_o \\ -I_o \end{vmatrix} \qquad (2.32)$$

where

$$\begin{vmatrix} A & B \\ C & D \end{vmatrix} = \begin{vmatrix} A_1 & B_1 \\ C_1 & D_1 \end{vmatrix} \begin{vmatrix} A_2 & B_2 \\ C_2 & D_2 \end{vmatrix} \cdots \begin{vmatrix} A_r & B_r \\ C_r & D_r \end{vmatrix} , \qquad (2.33)$$

The transmission parameters A_1, B_1, C_1, and D_1 for each subnetwork are the inverse voltage transfer ratio, negative inverse of the transfer admittance, the inverse transfer impedance, and the negative inverse current transfer ratio, respectively, of the input port to the output port. A_1 and C_1 are calculated with the output port open, B_1 and D_1 are calculated with the output port shorted. The transmission parameters can be computed by NASAP-70 and are in general ratios of polynomials in S.

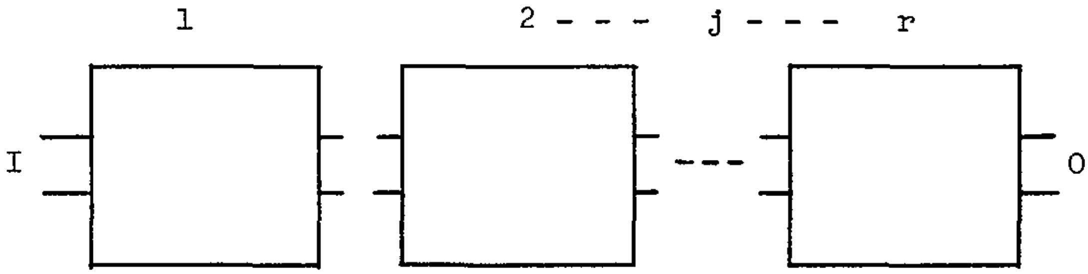


Figure 2 36 Cascade Connection of Two Port Subnetworks

Parallel Interconnection Method Another means of network interconnection is the paralleling of ports of networks as shown in Figure 2.37. Murti and Thulasiraman[15] show that two n-port networks described by short circuit admittances and having no internal vertices, can be combined in parallel if their edge and port orientations are identical, and also if their modified cut set matrices with identical row and column order are equal.

Let the short circuit admittance matrices of networks N_1 and N_2 be

$$Y_1 = \begin{vmatrix} Y_1^{aa} & Y_1^{ab} \\ Y_1^{ba} & Y_1^{bb} \end{vmatrix} \tag{2.34}$$

and

$$Y_2 = \begin{vmatrix} Y_2^{aa} & Y_2^{ab} \\ Y_2^{ba} & Y_2^{bb} \end{vmatrix} \tag{2.35}$$

The individual entries of these matrices are defined similar to the Laemmel matrices except that short-circuit admittances are involved.

Assuming the interconnection criteria are satisfied, the short circuit admittance matrix of the combined network is

$$Y_t = Y_1 + Y_2 = \begin{vmatrix} Y_1^{aa} + Y_2^{aa} & Y_1^{ab} + Y_2^{ab} \\ Y_1^{ba} + Y_2^{ba} & Y_1^{bb} + Y_2^{bb} \end{vmatrix} \quad (2.36)$$

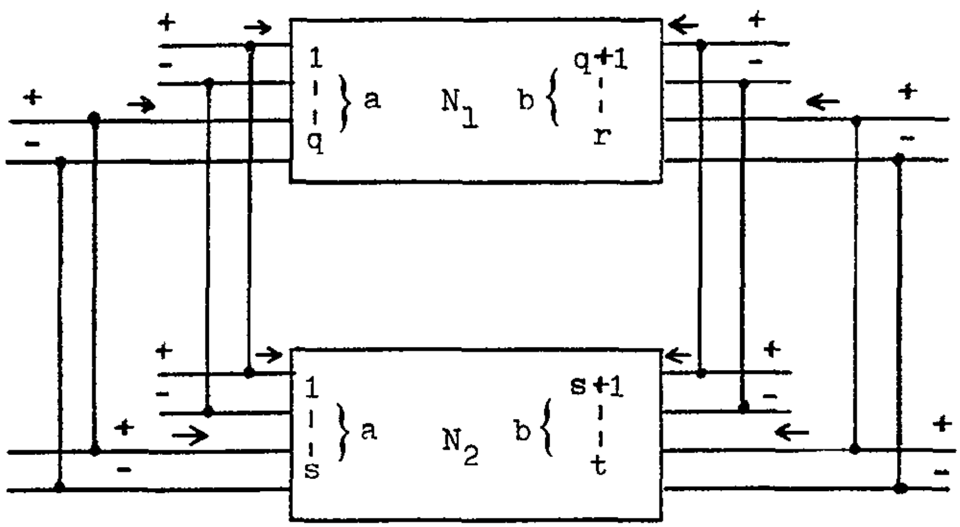


Figure 2.37 Parallel Connection of Two N Port Networks

2.6.2 Network Interconnection Examples

The network shown in Figure 2.38 was arbitrarily constructed to illustrate Laemmel's Method. Assume that V_2/I_1 is required. Let this network be subdivided into two subnetworks as shown in Figure 2.39

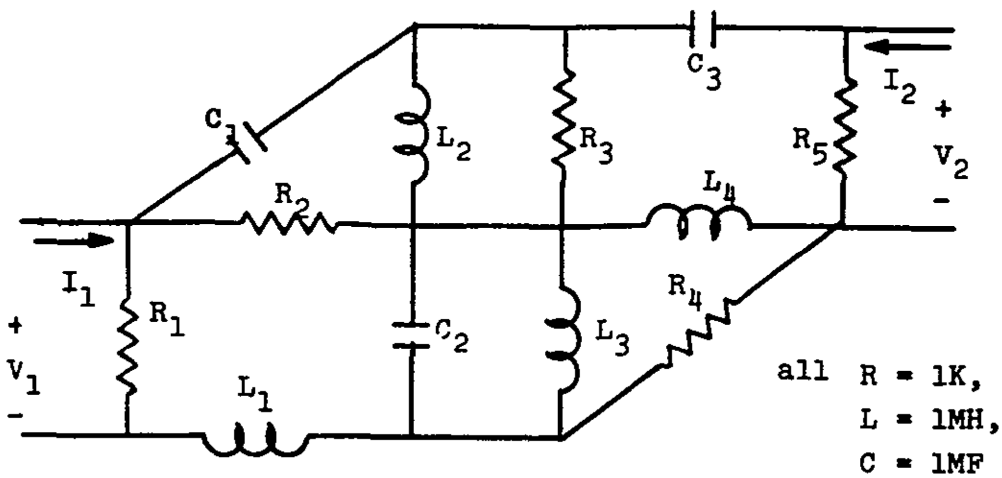


Figure 2.38

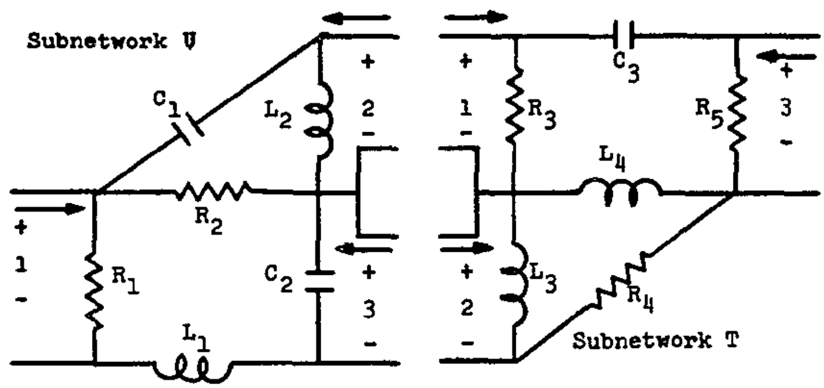


Figure 2 39

The port equations for subnetwork U are

$$\begin{vmatrix} V_a^U \\ V_b^U \end{vmatrix} = \begin{vmatrix} U^{aa} & U^{ab} \\ U^{ba} & U^{bb} \end{vmatrix} \begin{vmatrix} I_a^U \\ I_b^U \end{vmatrix} \qquad (2.37)$$

where

$$U^{aa} = (u_{ij}\ i = 1, j=1),\ U^{ab} = (u_{ij}\ i = 1, j = 2, 3),$$

$$U^{ba} = (u_{ij}\ i = 2, 3, j = 1),\ U^{bb} = (u_{ij}\ i = 2, 3, j = 2, 3),$$

$$V_a^U = V_1^U,\ V_b^U = V_2^U,\ V_3^U\ I_a^U = I_1^U,\ I_b^U = I_2^U\ I_3^U$$

The port equations for subnetwork T are

$$\begin{vmatrix} V_a^T \\ V_b^T \end{vmatrix} = \begin{vmatrix} T^{aa} & T^{ab} \\ T^{ba} & T^{bb} \end{vmatrix} \begin{vmatrix} I_a^T \\ I_b^T \end{vmatrix} \qquad (2.38)$$

and those for network Z are

$$\begin{vmatrix} V_a^Z \\ V_b^Z \end{vmatrix} = \begin{vmatrix} Z^{aa} & Z^{ab} \\ Z^{ba} & Z^{bb} \end{vmatrix} \begin{vmatrix} I_a^Z \\ I_b^Z \end{vmatrix} \qquad (2.39)$$

The matrices of (2.38) and (2.39) can readily be defined by an examination of Figures 2.38 and 2.39.

The desired transfer function is

$$V_2/I_1\Big|_{I_2=0} = Z_{21} = Z_{ba}, \tag{2.40}$$

From Equation (2.29) $Z^{ba} = T^{ba}(U^{bb}+T^{aa})^{-1}\,U_{ba}$. Therefore the following impedance parameters are computed utilizing NASAP-70 $t_{31}, t_{32}, t_{11}, t_{12}, t_{21}, t_{22}, u_{22}, u_{32}, u_{33}, u_{21}$, and u_{31}. Since the subnetworks are lumped, linear and passive, then $t_{12} = t_{21}$ and $u_{23} = u_{32}$, hence t_{21} and u_{32} need not be computed. The parameters computed are listed.

$$u_{12} = \frac{+6.872E+10S + 1.000E+09S^3}{+1.000E+18 + 3.000E+15S + 1.002E+12S^2 + 3.000E+06S^3 + 1.000E+00S^4} \tag{2.41}$$

Let the denominator of u_{12} be equal to u_{den} then

$$u_{12} = \frac{+6.872E+10S + 1.000E+09S^3}{u_{den}} \tag{2.42}$$

$$u_{13} = U_{31} = \frac{1.000E+21 + 1.000E+18S + 1.000E+12S^2}{u_{den}}$$

$$u_{22} = \frac{1.000E+15S + 3.000E+12S^2 + 1.001E+09S^3 + 1.000E+3S^4}{u_{den}} \tag{2.43}$$

$$u_{23} = U_{32} = \frac{-1.000E + 12S^2}{u_{den}} \tag{2.44}$$

$$u_{33} = \frac{2.000E+21 + 1.001E+18S + 3.000E+12S^2 + 1.000E+06S^3}{u_{den}} \tag{2.45}$$

$$t_{11} = \frac{1.000E+06 + 1.002E+03S + 3.000E-03S^2 + 1.000E-09S^3}{1.000E+03 + 2.002E+00S + 5.000E-06S^2 + 1.000E-12S^3}$$

Let the denominator of t_{11} to be equal to t_{den} then

$$t_{11} = \frac{1.000E{+}06 + 1.002E{+}03S + 3.000E{-}03S^2 + 1.000E{-}09S^3}{t_{den}} \tag{2.46}$$

$$t_{12} = T_{21} = \frac{-1.000E - 09S^3}{t_{den}} \tag{2.47}$$

$$t_{13} = \frac{1.000E{+}03S + 2.000E{-}03S^2}{t_{den}} \tag{2.48}$$

$$t_{22} = \frac{1.000E{+}00S + 2.00E{-}03S^2 + 3.000E{-}09S^3}{t_{den}} \tag{2.49}$$

$$t_{23} = t_{32} = \frac{1.000E{-}09S^3}{t_{den}} \tag{2.50}$$

Substituting the appropriate matrices elements of (2.37) and (2.38) into (2.29) the transfer impedance Z_{21} is computed

$$Z_{21} = Z_{ba} = \begin{vmatrix} T_{31} & T_{32} \end{vmatrix} \left[\begin{vmatrix} U_{22} & U_{23} \\ U_{32} & U_{33} \end{vmatrix} + \begin{vmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{vmatrix} \right]^{-1} \begin{vmatrix} U_{21} \\ U_{31} \end{vmatrix} \tag{2.51}$$

A computer program was written in Fortran IV to solve Equation (2.51), the result is

$$Z_{21} = \frac{+1.3{+}4E{+}38S^2 + 1.001E{+}39S^3 + 5.010E{+}36S^4 ++.036E{+}33S^5}{+2.000E{+}48S^0 + 1.301E{+}46S^1 + 3.009E{+}43S^2 + 3.026E{+}40S^3 + 1.333E{+}3{+}S^4}$$

$$\frac{+2.052E{+}30S^6 + 2.414E{+}25S^+ + 2.086E{+}21S^8 + 1.313E{+}16S^9}{+2.189E{+}33S^5 + 4.298E{+}28S^6 + 2.299E24S^+ + 2.59{+}E{+}19S^8 + 1.206E{+}14S^9}$$

$$\frac{+2.908E{+}10S^{10} + 2.602E{+}04S^{11} + 9.002E{-}03S^{12} + 1.000E{-}09S^{13}}{+2.{+}{+}3E{+}08S^{10} + 3.286E{+}02S^{11} + 1.931E{-}04S^{12} + 5.201E{-}11S^{13} + 5.000E{-}18S^{14}} \tag{2.52}$$

Also Z_{21} was computed directly for network Z by means of NASAP-70. This result is

$$Z_{21} = \frac{-7.885\mathrm{E}04S^2 + 1.001\mathrm{E}09S^3 + 3.001\mathrm{E}03S^4}{2.000\mathrm{E}18 + 3.008\mathrm{E}15S + 1.021\mathrm{E}12S^2 + 1.204\mathrm{E}07S^3} \;\; \frac{+1.002\mathrm{E}00S^5 + 1.000\mathrm{E}{-06}S^6}{+1.037\mathrm{E}03S^4 + 7.031\mathrm{E}{-03}S^5 + 1.201\mathrm{E}{-08}S^6 + 5.000\mathrm{E}{-15}S^7} \quad (2.53)$$

The frequency response of Equations (2.52) and (2.53) was plotted as shown in Figure 2.40. A comparison of the two curves of Figure 2.40 gives an indication of the relative error between the two methods for computing Z_{21}. The peak at the upper frequency range of the curve for (2.52) is a result of solving each of the two networks separately and then combining these results to obtain the solution for the interconnected networks. One possible explanation for this is loss of significance error which can occur when two numbers of similar magnitudes are added.

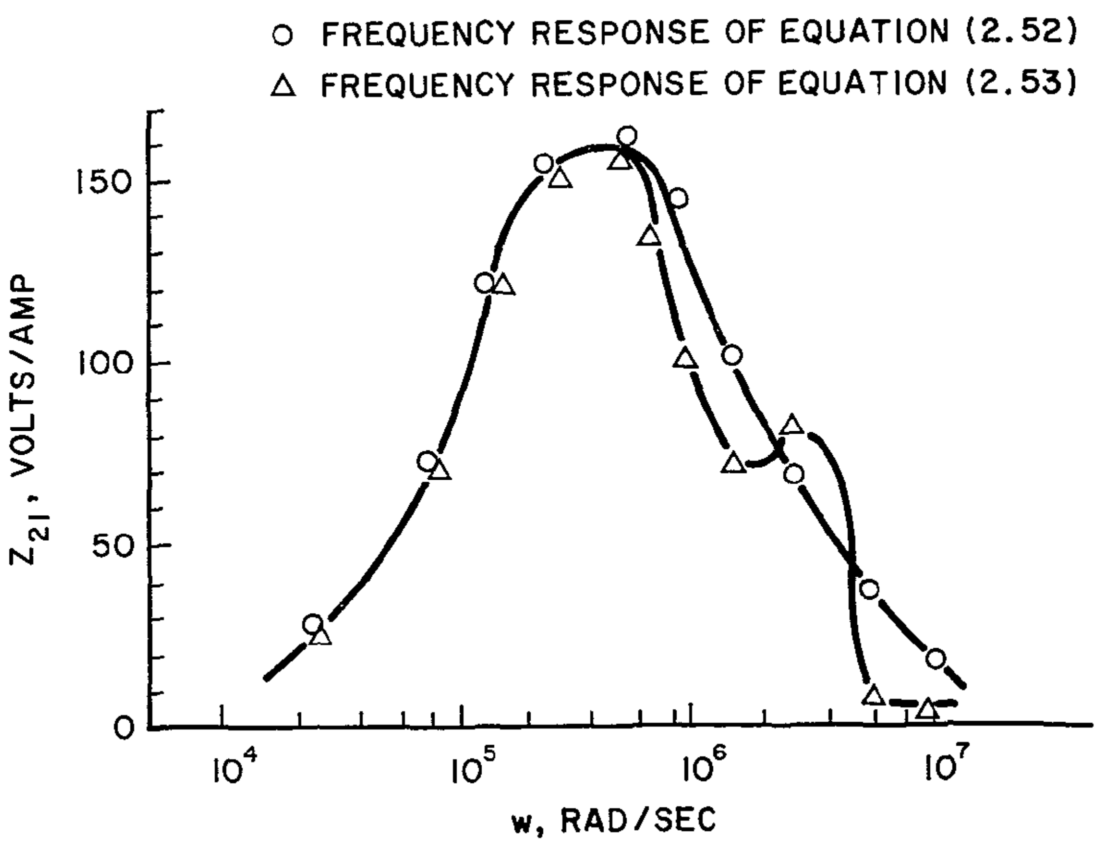


Figure 2 40 Frequency Response of Laemmel

A cascade connection example illustrates how the computation time increases as the number of cascaded stages increases. The circuit for each stage is shown in Figure 2.41.

The desired transfer function, I_{out}/I_{in}, can be calculated using NASAP-70. This was done first for one stage, then an identical stage was connected in cascade with this first stage to form a two stage amplifier. A coupling capacitor was placed between the collector of the first stage and the base of the second stage to provide DC isolation. NASAP-70 was again used to compute the transfer function I_{out}/I_{in}, where I_{out} is the current through the collector resistor of the last stage. This process was continued until five stages had been cascaded. The results, which were calculated by an IBM 360 computer are plotted in Figure 2.42.

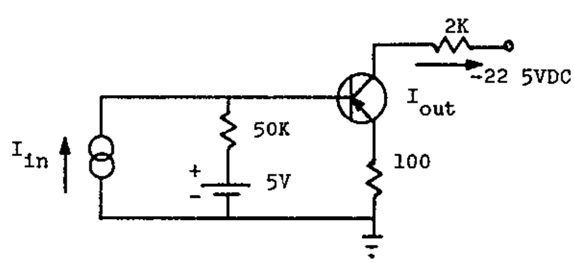


Figure 2 41 A Single Stage Amplifier

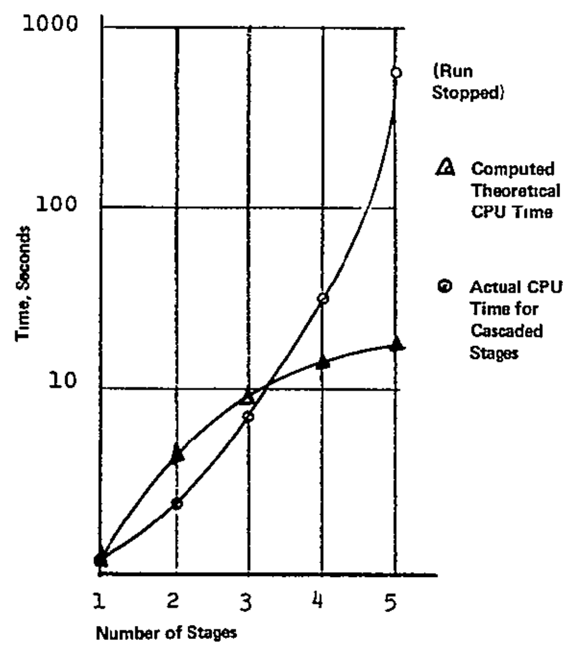


Figure 2 42 Results of Cascading Amplifier Stages

From this curve it can be seen that the computer time required increases rapidly as the number of cascaded stages is increased.

Calculations were made to predict the approximate theoretical time required to compute the desired transfer function when the cascaded stages are interconnected by the Cascade Parameter method. It was assumed that 1.1 seconds of CPU time was required to compute each parameter. The CPU time required to multiply the matrices of the individual stages was neglected.

If two stages were cascaded the resultant overall matrix would be

$$\begin{vmatrix} A & B \\ C & D \end{vmatrix} = \begin{vmatrix} A_1 & B_1 \\ C_1 & D_1 \end{vmatrix} \begin{vmatrix} A_2 & B_2 \\ C_2 & D_2 \end{vmatrix}$$

The desired parameter is, $D = B_2C_1 + D_1D_2$ To compute D, only B_2, C_1, D_1, and D_2 need to be calculated. Hence, the total assumed time required to compute D is 4X1.1 = 4.4 seconds. The D parameter for three, four and five cascaded stages was computed in a similar manner. These theoretical computation times are plotted in Figure 2.42. It is evident that if more than four stages are cascaded, then a definite time savings will result if the desired transfer function is computed by the Cascade Parameter method. In the preceding example, the transfer function I_{out}/I_{in} was computed, whereas the parameter D is defined as, $D = -I_{in}/I_{out} \big|_{V_{out} = 0}$. However, the parameter D can be computed and then inverted to give the desired transfer function.

2.7 Computational Errors

This section is concerned with the computational errors associated with NASAP-70. The impact and effect of errors at individual stages of computation are given, and in some cases recommendations regarding their minimization are also included.

2.7.1 Flowgraph Algorithm Errors

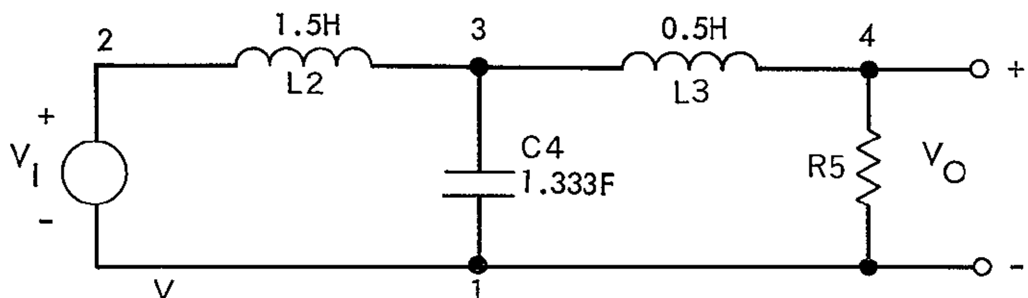
If the flowgraph algorithm and application of the Shannon-Happ formula were implemented utilizing a symbolic, or tagging approach no errors would be introduced into the derived transfer function. The NASAP-70 program discussed here performs all operations on numerical data. Thus, the constraint of digital computing introduces errors. Definition of these errors and how they relate to NASAP-70 follows.

Input/Output Errors[16]

Circuit parameters are input to NASAP-70 in decimal floating point notation. Binary arithmetic units require that the input data be converted to floating point notation compatible with the machine. Errors are introduced in program working data because there is no one-to-one correspondence between fractions of different radicies when operations are restricted to a finite wordlength. The magnitude of the error is dependent on machine hardware and the algorithm used to perform the conversion.

After required calculations are completed, the internal binary results are converted back to decimal notation. The conversion effects only the last significant digit of each coefficient and is quite minor. Also, conversion is not performed when further operations are carried out in the transfer function. For these reasons, no attempt is made here to analyze these errors.

An example of actual input/output errors (IBM 360/75) is shown in Figure 2.43.



$$T = \frac{V_O}{V_F}$$

$L_2 = 1\ 49999905 \times 10^{0}$
$L_3 = 4.99999821 \times 10^{-1}$
$C_4 = 1\ 33299828 \times 10^{0}$
$R_5 = 1\ 00000000 \times 10^{0}$

Figure 2 43

Round-off Error

Round-off errors result directly from the finite wordlength of the machine Errors can occur after both additive and multiplicative operations. The severe additive type errors are called significance loss, and will be described later

Multiplying two numbers, each with n significant digits yields a product significant to 2n places. To store the product it is rounded to n significant digits. Wilkinson[17] has shown that in a binary rounding system the floating point computation, $F\ell\,(X_1, X_2\ \ldots X_n)$, produces the error bound E,

$$|E| \leq (n-1)2^{-t} + \frac{(n-1)(n-2)2^{-2t}}{2!} + \ldots \tag{2-54}$$

where

$$F\ell(X_1, X_2, \ldots X_n) = X_1 \cdot X_2 \ldots X_n(1+E)$$

When computing high order loops, the flowgraph technique requires a number of consecutive multiplications. The round-off errors introduced can contribute significantly to the noise contained in the coefficients of a computed transfer functions. Relation (2-54) shows that increasing the word-length, t, results in a decreasing error ratio, E. It is proposed that double precision arithmetic be utilized in carrying out all required flowgraph multiplications. The round-off noise is located in the lower order digits. In most cases, results of multiplicative operations can be guaranteed to the significance of input data.

An example of round-off error is shown in Figure 2.44. The errors are totally attributed to round-off error because (1) Component values were selected such that the binary equivalent of the input data was exact, (2) Number of calculations and element values indicate no significant loss could have taken place.

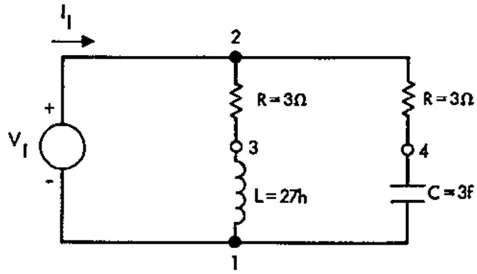


Figure 2 44

Output requested. $I_I/V_I = T$

$T_{True} = I_I/V_I = 1/3$

$$T_{NASAP} = \frac{1}{3 + \left[\dfrac{2 \times 10^{-5}}{3S^2 + 0.667S} + 3\ 709 \times 10^{-2}\right]}$$

Loss of Significance Errors

Loss of significance errors occur during floating point addition operations involving two numbers of opposite sign. The maximum error, E, resulting from the floating point addition X + Y is bounded by

$$|E| \leq \left(1 \frac{1}{2}\right) 2^{-t}$$

where

$$F\ell(X + Y) = (X + Y)\,(1 + E)$$

The error impact is largely dependent on the magnitude and polarity of the parameters involved, and the order in which computations are carried out. Two situations demonstrate the extent of significance loss. First consider a simple floating point addition of two numbers of opposite sign but comparable magnitudes. The resulting sum is located in the low order positions, the same locations affected by conversion and round-off errors. When the sum is left-shift normalized, the true precision of the result is disguised. The second case involves a set of repeated additions where the order of calculation can introduce loss of significance.

Significance loss errors can occur in the calculations associated with the flowgraph technique. Circuits containing elements with values covering a wide range are especially prone. The table below presents three ways in which significance loss can be detected, controlled or eliminated

METHOD	PURPOSE
Compare results of normalized and non-normalized arithmetic	Detection
Time scaling	Minimization or Elimination
Optimum tree	Minimization or Elimination

The first approach employs a two pass system with the first pass utilizing regular floating point arithmetic.[18] During the second pass, the arithmetic is changed such that sums are not left-shift normalized. When the number is subsequently required in another calculation, the appropriate number of zeros (necessary to align the two operands) are shifted into the least significant bit positions. This scheme defines a lower bound for the calculation. The results of the two passes are examined and significance loss has occurred if these numbers are not equivalent. The significance of the calculation is considered to be equal to the number of digits which are identical. The calculation of Figure 2.43 is not prone to significance loss,

$$T = \left(\frac{V_1}{V_{R_5}}\right)_{\text{normalized}} = \frac{1.00025}{1.00000S^3 + 2.00000S^2 + 2.00050S + 1.00025}$$

$$T = \left(\frac{V_1}{V_{R_5}}\right)_{\text{unnormalized}} = \frac{1.\overline{0025}}{1.00000S^3 + 2.00000S^2 + 2.00050S + 1.00025}$$

Figure 2.45 is a problem vulnerable to significance loss as is demonstrated below

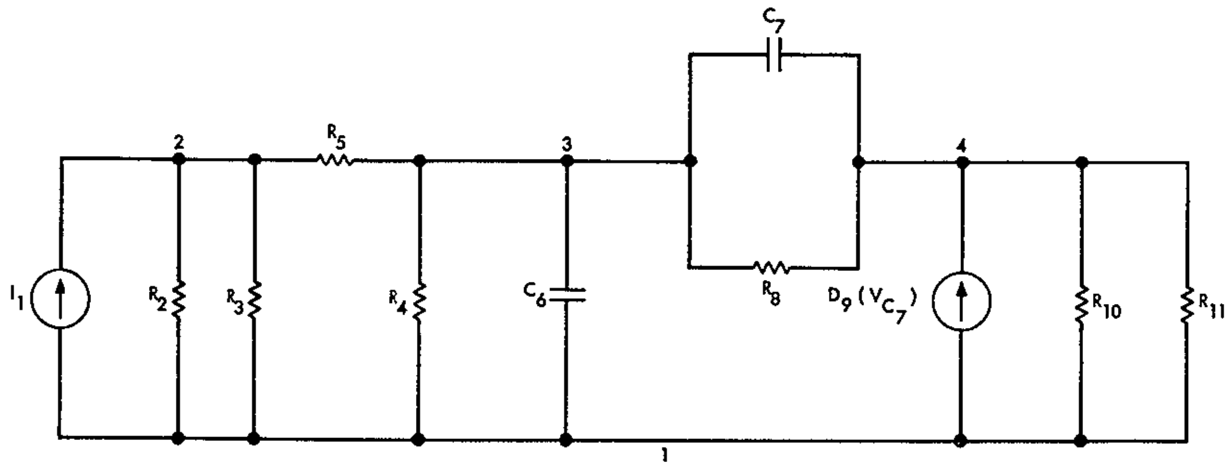
$$T = \left(\frac{V_I}{I_I}\right)_{\text{normalized}}$$

$$= \frac{2.04000 \times 10^{-11}S^2 + 1.10709 \times 10^{-1}S + 7.19903 \times 10^7}{2.04000 \times 10^{-13}S^2 + 7.67740 \times 10^{-4}S + 1.16400 \times 10^4}$$

$$T = \left(\frac{V_I}{I_I}\right)_{\text{unnormalized}}$$

$$= \frac{2.04000 \times 10^{-11}S^2 + 1.10709 \times 10^{-1}S + 7.19903 \times 10^7}{2.04000 \times 10^{-13}S^2 + 7.67827 \times 10^{-4}S + 1.16640 \times 10^4}$$

Figure 2.46 shows how the span of element magnitudes in Figure 2.45 changes as the time scale is decreased. The particular transfer function was computed and checked for significance loss at the various time scales. The results of the technique although inconclusive, tend to show that time scaling in one technique to combat significance loss.[16]



$I_1 = 1$ a

$R_2 = 1 \Omega$

$R_3 = -1 \Omega$

$R_4 = 100 \Omega$

$R_5 = 1 \times 10^6 \Omega$

$C_6 = 6$ pf

$C_7 = 200$ pf

$R_8 = 150 \Omega$

$D_9 = 0.4$

$R_{10} = 1\ 7 \times 10^4 \Omega$

$R_{11} = 100 \Omega$

Compute $T = \dfrac{V_{R2}}{I_1}$

Figure 2 45

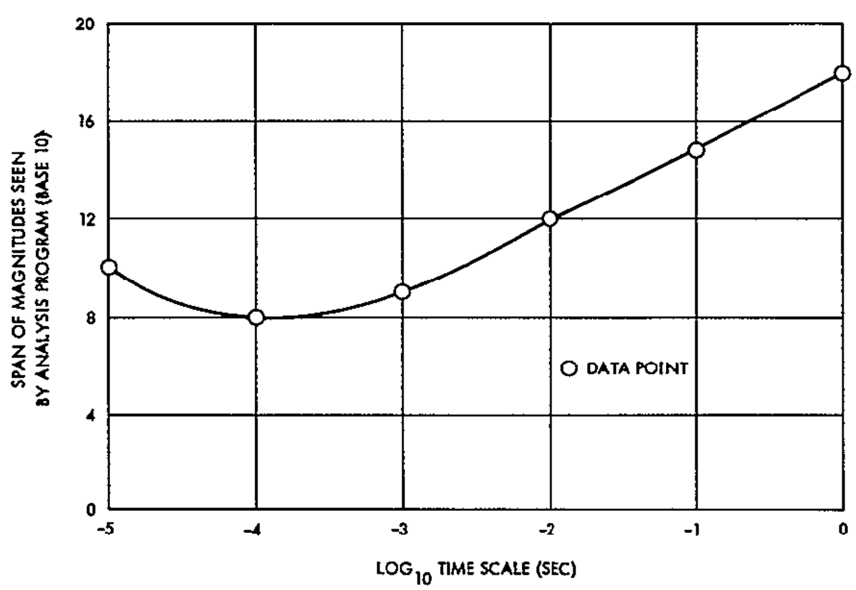


Figure 2 46

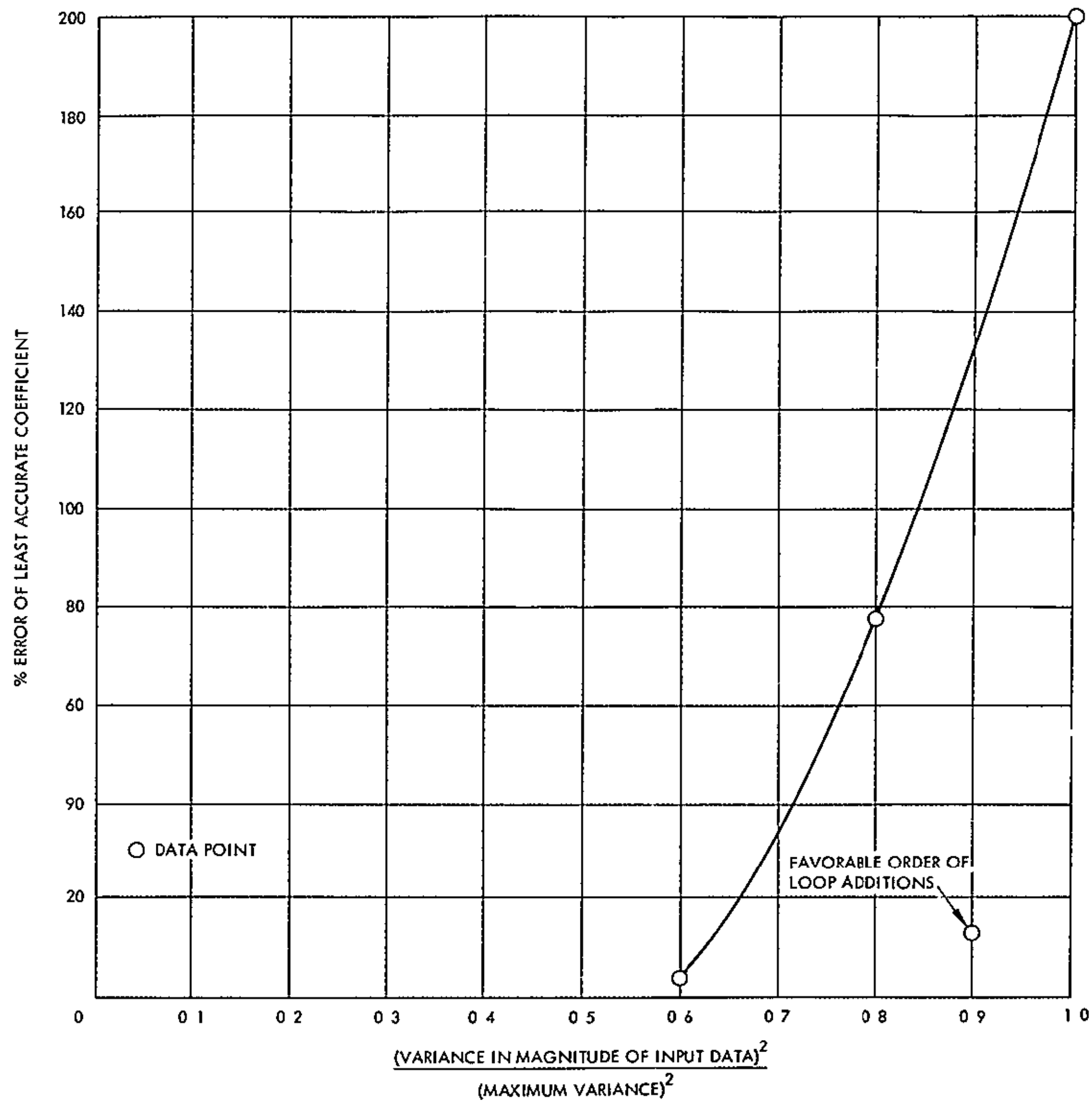


Figure 2 47

Another way of reducing significance loss in the computation of Figure 2.45 is to select an optimal tree. An optimal tree is defined as the tree which, if selected, displays to the program the minimum variance between the magnitudes of the input data. The results of optimal tree selection are plotted above.

Different input trees generally result in varying computational orders. The above definition can be used as a means for reducing computational errors, although the method itself is not foolproof The third data point in Figure 2.47

is assumed to result from a favorable computational order for the particular tree selected and, contradictory to the definition of an optimum tree, a comparably good transfer function was calculated.

Error Prediction[16]

It was shown that the errors associated with transfer function coefficients are related directly to the number of loops of all orders in the flowgraph and the number and range of computations required for each loop. It is desirable to obtain a measure of these parameters before an attempt is made to derive a particular transfer function. Given the flowgraph model described by Figure 2.48, it can be shown that there exists, $F_n = 1/\sqrt{5}\,[(1+\sqrt{5/2})^{n+2} - (1-\sqrt{5/2})]^{n+2}$ loops of all orders, where $n = 2m - 1$ (the number of passive elements minus one) and F_n = Fibonacci number of order n.

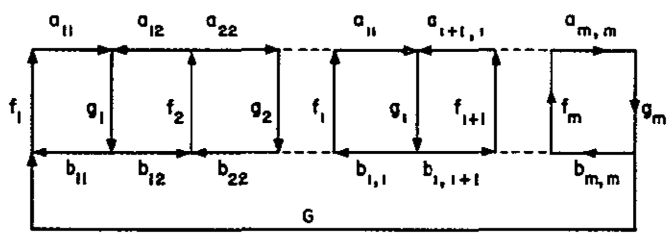


Figure 2.48

If flowgraph loops of all orders are assumed to have numerical values of the same magnitude, then the equation below is an estimate of the maximum round-off error contained in a calculated transfer function coefficient

$$\frac{|\Delta\Sigma(n)|}{|\Sigma(n)|} \approx \frac{2^{-t}}{F_n}\left[\sum_{N=1}^{N_{max}} \frac{(n-N+1)!}{N!(n-2N+1)!}\,(NK-1)\right]$$

where

1. $\Sigma(n)$ is the sum of all loop values of all orders
2. K is the number of circuit elements in a particular first order loop (=2 for the specific model)
3. $N_{max} = n+1/2$ if n is odd
   $N_{max} = n/2$ if n is even.

In an attempt to predict errors for those circuits which do not exactly correspond to the flowgraph model, Figure 2.49 has been drawn for various values of K. Experience has shown that for practical problems, a good average value of K lies between 2 and 3.

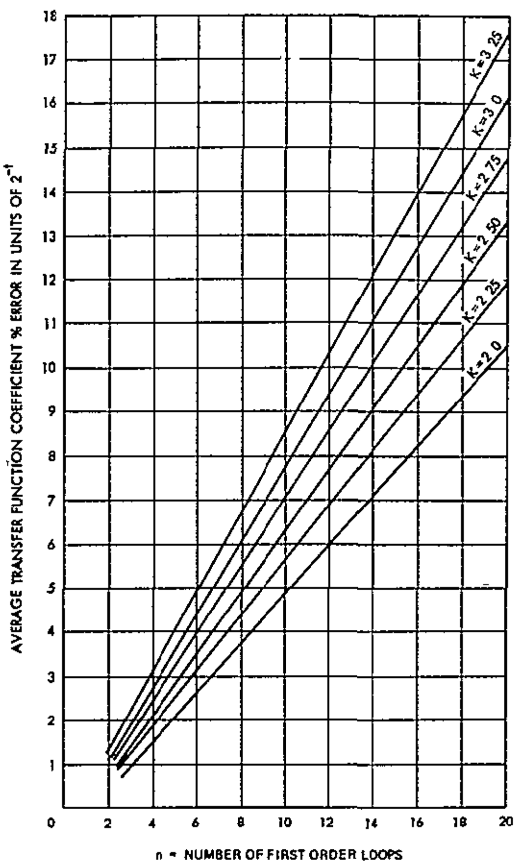


Figure 2.49

2.7.2 Impact of Coefficient Errors on AC Analysis

The NASAP version utilizes a straight forward technique in deriving the AC Bode plot. The programs produce a frequency response of the form

$$F(s) = \frac{N(s)}{D(s)} = \frac{\sum_{i=0}^{p} a_i s^i}{\sum_{j=0}^{q} b_j s^j}$$

The computed coefficients, a_i and b_j, can be expressed as,

$$a_i = a_i^T + \Delta a_i$$

$$b_j = b_j^T + \Delta b_j$$

where a_i and b_j represent the accumulated coefficient errors a_i^T and b_i^T are the true coefficients, and F(s) as

$$F(s) = F^T(s) + \Delta F(s) = \frac{N^T(s)}{D^T(s)} + \Delta F(s) \quad .$$

If F(s) is assumed to be a function of its coefficients a, b for any s, then since[19]

$$f(a^T + \Delta a, b^T + \Delta b) \approx f(a^T, b^T) + \left.\frac{\partial f}{\partial a}\right|_{f^T} \Delta a + \left.\frac{\partial f}{\partial b}\right|_{f^T} \Delta b$$

$$\text{for} \quad \Delta a \ll a$$

$$\Delta b \ll b \quad ,$$

it can be shown that

$$\Delta F(s) = \sum_{i=0}^{p} \frac{\Delta a_i}{D^T(s)} s^i - \sum_{j=0}^{p} \frac{F^T(s)}{D^T(s)} \Delta b_j s^j \quad .$$

The computational errors in F(s) are approximated from an initial estimate of coefficient errors by substituting D(s) and F(s) for $D^T(s)$ and $F^T(s)$. These substitutions become quite laborious for functions which are a ratio of two high order polynomials. An alternate approach is taken.

Coefficient errors are not likely to exceed 15 percent of the true coefficient values (this statement is predicated on the assumption that any "large" problems have been partitioned and the steps have been taken to avoid severe significance loss). By randomly perturbating the transfer function coefficient not more than ± 15 percent and plotting the frequency response curves, the results of the above can be simulated. A band which most probably contains the true response curve will result. For example, the frequency response curve of Figure 2.50 can be obtained from the circuit shown in Figure 2.43.

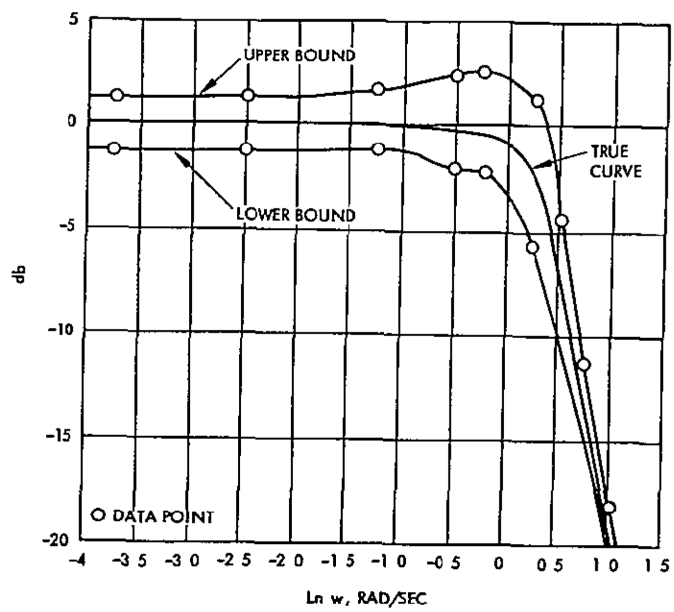


Figure 2 50

2 7 3 Transient Response Errors

Errors contained in a final time domain solution after application of the Fast Fourier Transform can be attributed to original coefficient errors and algorithm errors.

Given the results of Section 2.7.2

$$f(t) = \mathfrak{F}[F(s)] = \mathfrak{F}[F^T(s) + \Delta F(s)]$$
$$= \mathfrak{F}[F^T(s)] + \mathfrak{F}[\Delta F(s)]$$
$$= f^T(t) + \Delta f(t)$$

Since the errors due to coefficient deviations are simply an additive function to the time response, an initial approximation of these errors can be transformed directly into an estimate of time response errors.

The digital implementation of the Fast Fourier Transform introduces round-off, discretization and band limiting errors.

Gentleman and Sande[20] have described the Fast Fourier Transform round-off error found in a digital environment in terms of the ratio of the Euclidian norm of the error sequence to the Euclidian norm of the input data sequence. Using the Fast Fourier Transform yield an upper bound for the associated Euclidian norm of

$$R \leqq 1.06 \sqrt{N} \ K(2n)^{3/2} \ 2^{-b}$$

where N is the number of discrete steps taken, $n^k = N$ (for NASAP-70 n=2, k=12), and b equals the number of bits in the calculation mantissa.

Figure 2.51 shows how discretization and band limiting errors affect the results of a Fast Fourier Transformation. For each curve, the function $F(s) = s/s + \omega$, was inverted. Discretization errors are demonstrated by the more noisey 'o' curve. Band limiting errors are shown by the poor initial points of both curves.

It should be noted that the errors resulting from coefficient errors dominate those introduced by the Fast Fourier algorithm.

Based upon the error analysis of a typical circuit analysis program such as NASAP-70 the user must be aware of the computational aspects of the algorithms before the results can be considered reliable.

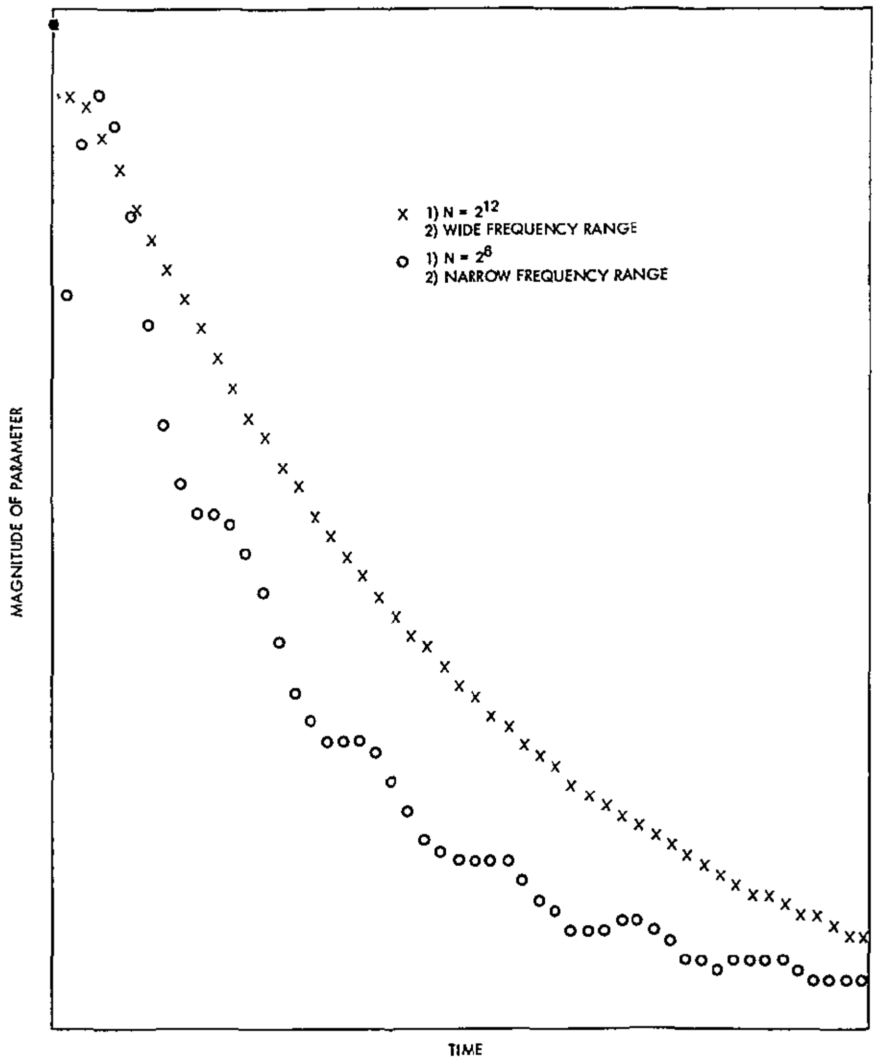

1) N = 2^12
2) WIDE FREQUENCY RANGE
1) N = 2^6
2) NARROW FREQUENCY RANGE
MAGNITUDE OF PARAMETER
TIME

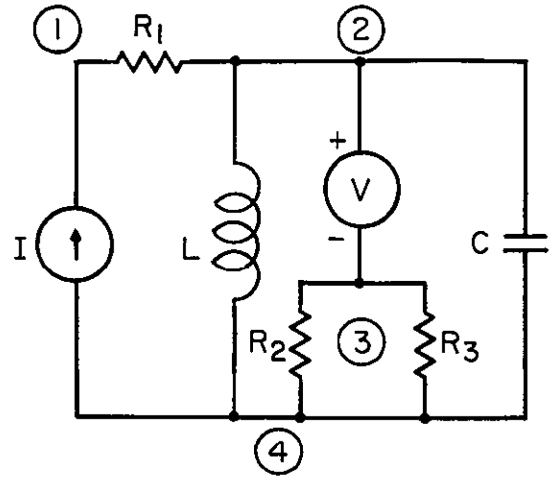
Figure 2 51

a graph represents a physical connection or junction point in the system and is usually represented by a solid dot In Figure 3 2(a) a linear circuit of seven elements is shown with its corresponding linear graph in Figure 3 2(b) Note that the seven circuit elements and four junction points result in seven branches and four nodes in the linear graph Hence a linear graph of a physical system can be defined as a mathematical model that consists of an interconnection of branches and nodes

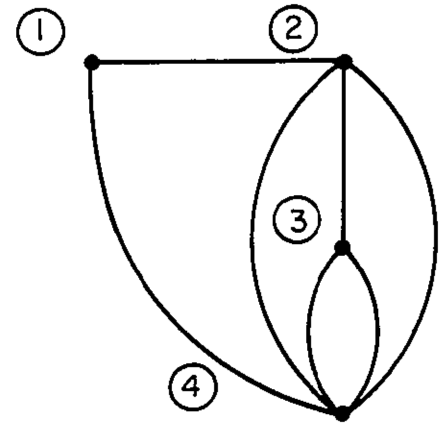
If a graph contains branches which indicate a direction (usually by an arrow), the graph is called oriented In circuit theory, the direction associated with a linear branch is interpreted as current flow As it now stands, the graph of Figure 3 2(c), an oriented and physically interpreted linear graph results Note that the current direction of the current source is retained and the current assignment for the voltage source element is from minus to plus in the direction of voltage rise The current directions of the passive elements can be arbitrary although dependent sources require a knowledge of the nature of the dependency

Given a graph G, a graph G' is said to be a subgraph of G if G' contains only branches and nodes of the original graph G. A graph G is connected if there is at least one path (neglecting branch directions) between any two nodes in the graph If a graph is not connected, it is said to be separated The connected parts of a separated graph are subgraphs

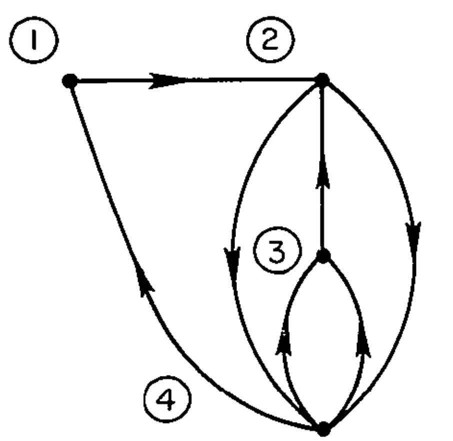
The basic properties of linear graph theory which pertain directly to circuit design are the concepts of cut-set, tree, and loop A cut-set is a group of branches which when removed from the graph divides the graph into two separate parts and at the same time permits the graph to be connected when the removal of all but <u>any</u> one branch of the cut-set remains intact Note that this definition implies that when the branches are removed from the parent graph G, the terminal nodes incident with the removed branches are retained in G A branch is incident with a node if it either enters or leaves the node An example of a cut-set is shown in Figure 3 3 Note also



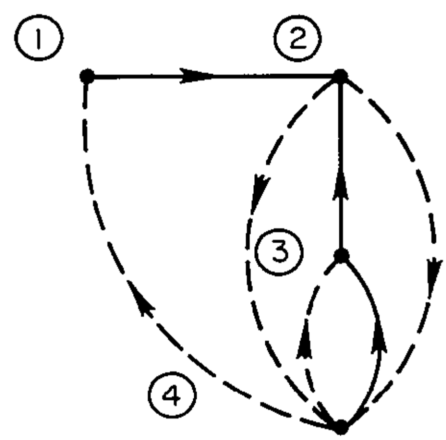
(a)
CIRCUIT DIAGRAM



(b)
LINEAR GRAPH



(c)
ORIENTED GRAPH (G)



(d)
TREE (T)

Figure 3 2

REFERENCES

1. Desoer, C., Kuh, E. S., "Basic Circuit Theory," McGraw-Hill, 1966.
2. Seshu, S., Reed, M. B., Linear Graphs and Electrical Networks, Addison - Wesley, 1961.
3. DiStefano, J., Stubberud, A., Williams, I., Theory and Problems of Feedback and Control Systems, Schaum Publishing Co. 1967.
4. Potash, H., McNamee, L. P., "Application of Unilateral and Graph Techniques to Analysis of Linear Circuits," Proceedings of 23rd ACM National Conference, Las Vegas, Nevada, 1968, pp. 367-378.
5. McNamee, L. P., and Potash, J., "A User's Guide and Programmers Manual for NASAP," Report No. 68-38, August 1968.
6. Gaertner Research Incorporated, "Coding Instructions for NASAP 69/I," Revision No. 1, Contract No. NAS 12-663, Stamford, Conn.
7. Moe, M. L., Schwartz, John T., "The Application of NASAP to the Design of Biomedical Instrumentation Circuits," Denver Research Institute Report DRI #2535, University of Denver, January 1970.
8. Carpenter, R. M., "Computer-Oriented Sensitivity and Tabrance Techniques," Course Notes Automated Circuit Analysis, UCLA, April 3-7, 1967.
9. Haag, K. W., Weber, E. W., "Designers Manual for Computer-Aided Design of Communication Circuits," Department of Electrical Engineering, Illinois Institute of Technology, December 1969.
10. Haag, K. W., "NASAP Analysis of Nonlinear DC Circuits," Seventh Allerton Conference on Circuit and System Theory, University of Illinois, October 8-10, 1969, pp. 905-914.
11. Wilson, M., McNamee, L., "Considerations for Solving Large Scale Circuit Problems Utilizing NASAP-70," Midwest Circuit Theory Symposium, Minneapolis, Minnesota, May 1970.
12. Wilson, M. L., "An Investigation of Methods for the Interconnection of n-Port Networks," Master's Thesis, UCLA, 1969.
13. Laemmel, A. E., "Scattering Matrix Formulation of Microwave Networks," Proceedings of the Symposium on Modern Network Synthesis (Audio to Microwaves), April 1952, pp. 259-276.
14. Huelsman, L. Q., Circuits, Matrices, and Linear Vector Spaces, McGraw-Hill, 1963.
15. Murti, V. G. K., and K. Thalasiraman, "Parallel Connection of n-Port Networks," IEEE Proceedings, Vol. 55, No. 7, July 1967, pp. 1216-1217.

16. Sesar, D., "On the Errors Associated with a Digital Implementation of the Flowgraph Approach to Circuit Design/Analysis," M.S. Thesis, University of California, Los Angeles, June 1969.

17. Wilkinson, J. H., Rounding Errors in Algebraic Processes, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1963.

18. Ashenhurst, R. L. and Metropolis, N., "Unnormalized Floating Point Arithmetic," Journal of the Association for Computing Machinery, Vol. 6, No. 3, July 1959, pp. 415-428.

19. Fischer, M. M., "The Effect of Instrumentation Errors on Basic Oxygen Furnace Control," Presented at Instrument Society of America, 17th Annual Conference, October 1967.

20. Gentleman, W. M and Sande, G , "Fast Fourier Transforms for Fun and Profit," Proceeding AFIPS Fall Joint Computer Conference, Volume 29, Washington, D.C., 1966, pp. 563-578.

CHAPTER 3

THEORETICAL FOUNDATIONS

3 0 Introduction

Fundamental to any meaningful discussion of a computer-aided circuit analysis/design program is the systematic treatment of the basic concepts behind it As shown in Figure 3 1, the theoretical treatment for NASAP can be separated into the disciplines of linear graph theory, flowgraph theory, sensitivity analysis, and the relevant numerical techniques and supporting computer algorithms.

Each discipline plays a distinctive role in the operation of the program Linear graph theory is employed as a convenient mathematical tool to bridge the gap between the circuit designer and the computer A computer subroutine then transforms the linear graph description of a circuit into flowgraph terms forming the basis of the computer algorithms for subsequent output requests The transfer function, the principal output request of NASAP-70, is computed from the flowgraph description by means of the Shannon-Happ formula, the other user requests are based upon the transfer function formulation.

As shown in Figure 3 1, the description of each discipline in NASAP is taken in the order of implementation thus the sections are numbered accordingly

3 1 Linear Graph Theory

The initial step in the formulation of the flowgraph relations is the identification of circuit elements in topological terms. Since this requires some background in linear graph theory, a few definitions and some basic concepts of how they pertain to circuit theory should be stated first A more elaborate treatment of graph theory can be found in a number of textbooks A complete exposition can be found in Kuh and Desoer [1]

The essential constituents of a linear graph are branches and nodes Each branch, symbolized by a line, represents a physical element of the system from which it is abstracted with its electrical properties A node in

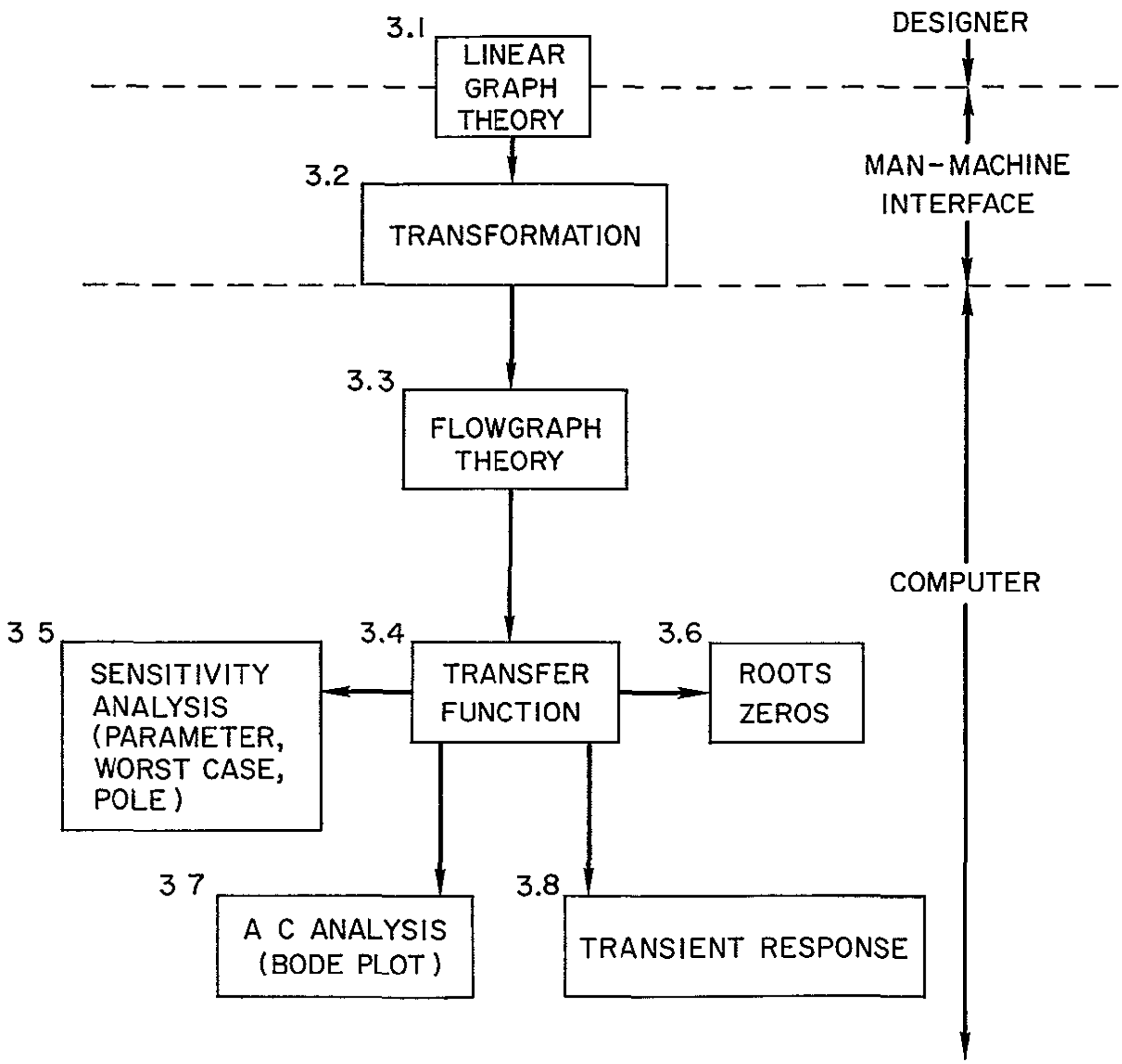

3.1
LINEAR GRAPH THEORY
DESIGNER
MAN-MACHINE INTERFACE
3.2
TRANSFORMATION
3.3
FLOWGRAPH THEORY
COMPUTER
3 5
SENSITIVITY ANALYSIS (PARAMETER, WORST CASE, POLE)
3.4
TRANSFER FUNCTION
3.6
ROOTS ZEROS
3 7
A C ANALYSIS (BODE PLOT)
3.8
TRANSIENT RESPONSE

Figure 3 1

that a hinged linear graph of Figure 3 4 violates the cut-set definition at the "hinge" and thus circuits of this type should be avoided or separated into two problems

In simple terms a loop is defined as a collection of branches which form a closed path (ignoring the direction of flow in the branches) A more formal definition can be stated· a set of n nodes $A(A_1 \text{-----} A_n)$ and a set of n branches $B(B_1 \text{---} B_n)$ which are connected constitutes a loop in a graph G if A is in G and B is in G, and each branch in B is incident to two nodes in A and each node in A has two branches in B incident to it The branches 1-2, 3-2, 4-3, 4-1 form a loop as shown in Figure 3 5 Note that a loop is defined independent of the direction of the arrows.

If a subgraph G' of a graph G can be selected such that it consists of only branches and all nodes of G, is connected, and contains no loops, then the subgraph G' is called a tree (G' = T) (See Figure 3 2(d)) The branches of a tree are called tree branches or just branches, and those branches in G but not in T are name links

A number of fundamental properties concerning trees are now stated without proof [1,2,3] If a connected graph G has n nodes and b branches, then

1. The Tree has (n-1) tree branches and (b-n+1) links,
2. A unique path exists along the tree between any two nodes (neglecting the direction of the arrows if it is an oriented graph),
3. Fundamental loops (tie sets) can be formed by taking each link with the unique path in the tree that exists between the two nodes of the link,
4. Fundamental cut sets can be formed by grouping the set of connected links about a tree branch thus forming (n-1) cut sets,
5. The cut set relations form a complementary set to the tie set relation (in matrix notation one is the transpose of the other),
6. Cut set and tie set relations can be expressed unilaterally. Branch currents can be expressed as a function of link currents and link voltages as a function of branch voltages

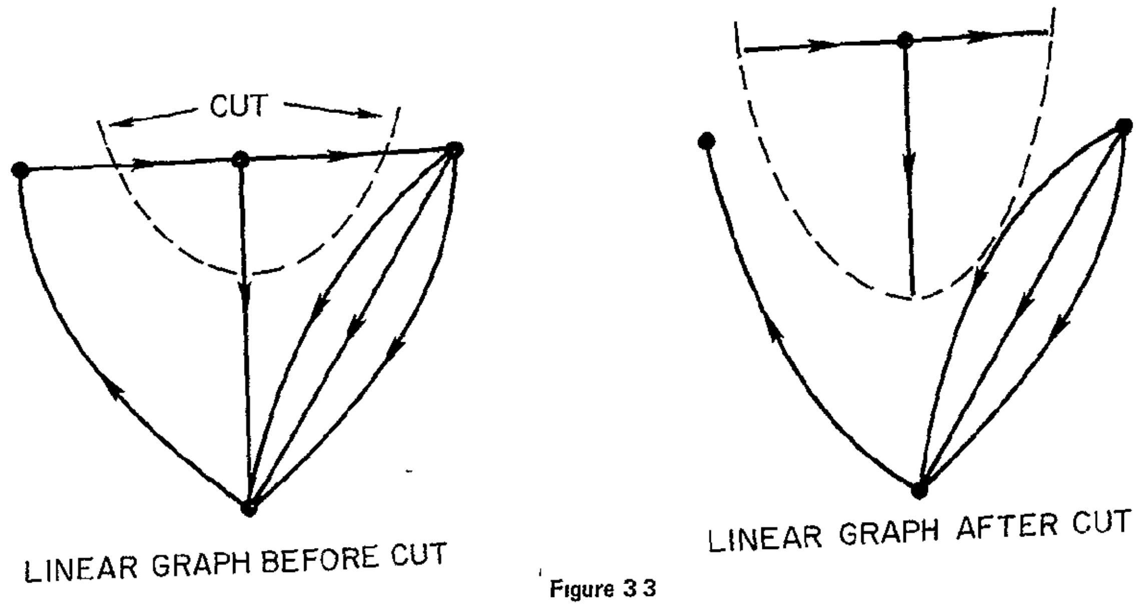


Figure 3 3

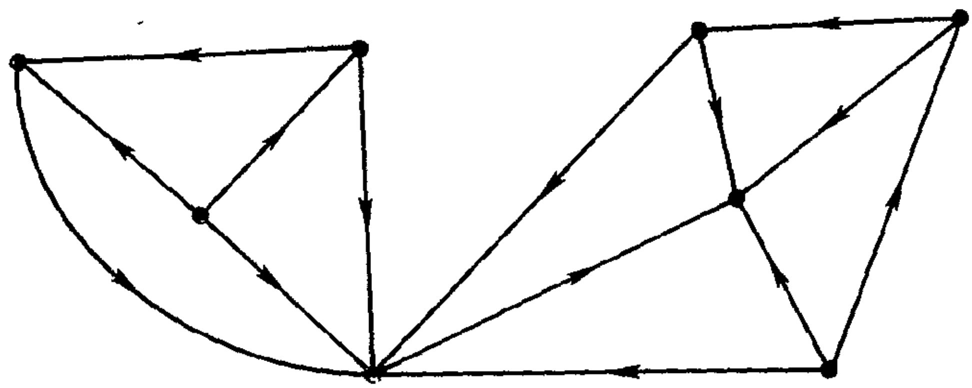


Figure 3 4

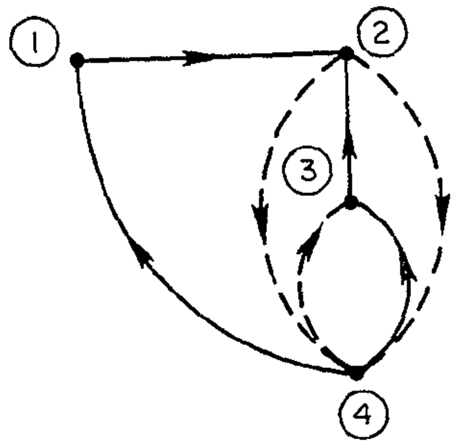


Figure 3 5

3.2 Transformation of a Linear Graph to a Flowgraph — Disjoint Current and Voltage Relationships

Algorithms must provide for the computer a means of producing a consistent set of determinable variables. To a circuit designer this usually means either applying Kirchhoff's voltage law in conjunction with the impedance relationships (Ohm's law) or Kirchhoff's current law in conjunction with the admittance relationships. Another approach has been advocated (originally by Kirchhoff himself) and is used in this manual.[4, 5] Basically it requires that Kirchhoff's current law be applied first, then Kirchhoff's voltage law, and finally Ohm's law and dependencies.

The first step in the procedure requires that all circuit elements, active or passive, dependent or independent, be divided into two groups of current-type or voltage-type elements. Naturally, a voltage source will remain a voltage type element, and a current source will remain a current-type element. Passive elements, which are bilateral in nature, will be replaced by either a voltage-type or current-type element (i.e., replace Ohm's law $I = V/Z$ by either $I \leftarrow V/A$ or $V \leftarrow I*Z$), the choice being somewhat arbitrary and subject to the conditions that Kirchhoff's two laws have to be applied independently. (Some guidelines will be presented in the example problem of section 3.3) It should be emphasized that by assigning a voltage or current type element in place of a passive element, the bilateral characteristic is replaced by a unilateral relation.

In the following selection process, Kirchhoff's laws will be applied to two sets of disjoint circuit relations. Tree-link partitioning is accomplished by using a tree structure T in a graph G, where the voltage-current equations are separated into two complimentary sets. The graph G contains as its branches all the circuit elements. The arrow direction corresponds to the direction of assigned current flow. The branches of a tree T (Figure 3.2(d)) are selected to be the voltage elements. The remaining elements (links in T) are chosen to be current elements. For this selection, a voltage at any point can be calculated as a function of voltage-type elements only. Currents and impedances need not be considered. Similarly, the current at any element can be determined as a function of current-type elements only, and therefore is independent of voltage-type elements and impedances.

Note that the tree-link partition does not restrict one from determining the voltage at a voltage-type element as a function of currents, voltages, and Ohm's law. It is particularly important to realize that expressing a voltage as a function of voltage-type elements does not render a voltage across these elements constant. It does mean, however, that a consistent set of Kirchhoff's equations can be written as a function of any voltage in terms of voltage type elements. A similar statement can be made concerning the current type elements. Namely, the current expressed as a function of current-type elements does not render the current through current-type elements constant.

Consider R_1 as a voltage element. The voltage at R_1 (V_{R_1}) is set as a known variable and the current as an unknown variable, then the applications of Kirchhoff's laws will result in determining [illegible] as a function of other currents in the circuit. The subsequent application of Ohm's law in the form $V = RI$ is made only to substitute for the original use of V_{R_1} as a known variable. Thus the known variables in applying Kirchhoff's laws are either constants or [illegible] relations. The unknown variables in application of Kirchhoff's laws are the known relations of Ohm's law and the dependency relationships.

In summary, the NASAP algorithms make use of the following

1. A tree T in graph G divides the elements into two disjoint sets,
2. A tree T in graph G allows writing of (n-1) independent current equations of one set (voltage) in terms of the other (current) by writing current equations for any (n-1) out of (n) nodes,
3. By Gaussian elimination[6] one obtains (n-1) cut set equations. These equations can be expressed as unilateral current relations, (this will be elaborated in the example),
4. Since a cut set and tie set are complementary, one obtains Kirchhoff's voltage equations in a unilateral form as a complement of the unilateral current relations,

5 Current-voltage relations can be made functionally dependent by a set of unilateral elements provided that for each element one of the quantities (current or voltage) is known and the second quantity (voltage or current) is unknown These are exactly the conditions that the two disjoint current-voltage sets provide.

3.3 Flowgraph Theory Shannon-Happ Formula

For a set of linear relations that satisfy consistency criteria (to be explained later) there exists a formula by which the solution can be obtained in a noniterative substitution and does not require stability criteria for each relation This formula was originally developed by Claude Shannon while investigating the functional operation of an analog computer [7] (Essentially the same formula was rediscovered by S. Mason in 1952 [8])* Shannon's Formula is an analytic expression for calculating the gain of an interconnected set of amplifiers in an analog computing network W. W. Happ generalized Shannon's work for topologically closed systems.[9] The Shannon-Happ Formula is valid in deriving transfer functions, sensitivities, and error functions For the particular use intended for this formula, the sensitivity factor H is set equal to zero

Flowgraph Notation

Before the Shannon-Happ Formula can be defined explicitly, it is necessary to define the basic concepts of flowgraph for which the equation was derived [10, 11] The fundamental terms of a flowgraph, namely, node, transmittance, and loop, and how they interact can best be described by example Consider the equations,

$$a_{11}x_1 + a_{22}x_2 + x_1 = x_1$$

$$a_{21}x_1 + a_{22}x_2 + c_2 = x_2$$

and the corresponding flowgraph shown in Figure 3 6

*During World War II Claude Shannon developed his formula Because of war-time restrictions, his work was not published, and was virtually unknown to the academic community In 1952, Samual Mason rediscovered the same formula.

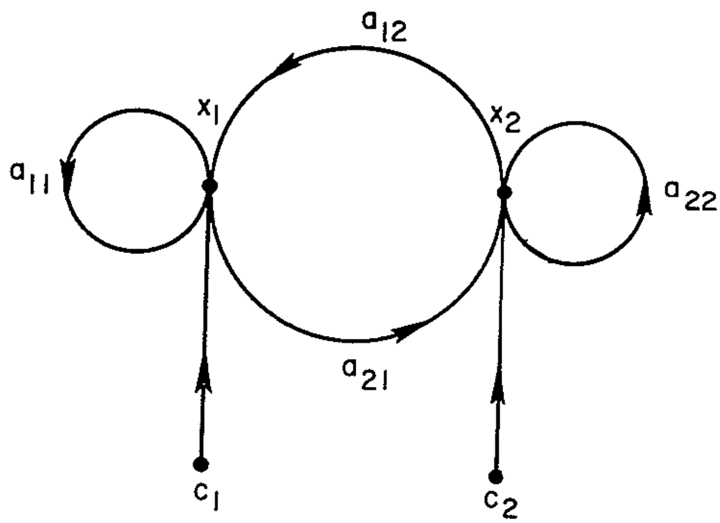


Figure 3 6

Each node represents a variable, thus x_1 and x_2 are variables and c_1 and c_2 are considered constant inputs The nodes c_1 and c_2 are called independent nodes. If the arrows leading from c_1 or c_2 were reversed, these nodes would be dependent nodes

A transmittance is a value assigned to the directed path between two nodes The node at which the directed line begins is called the origin node, whereas the node at the receiving end is the target node The transmittance relates the origin node variable Therefore, in reading the flowgraph, the functional relationship of a variable is the summation of the product of respective incoming transmittances with their node variables This can be verified by writing out the equation associated with the flowgraph (Figure 3 6) and the original equations Note that a flowgraph transmittance function such as

$$A \bullet \xleftarrow{\quad b \quad} \bullet\, C$$

will be written as A ← b * C on a programming sheet

A directed loop is defined as a closed path consisting of a sequence of transmittances taken in the direction of the arrow The sequence must be taken such that no node is traversed more than once in the closed path. The value of a directed loop is the product of the transmittances forming the directed loop In the example, three directed loops exist, (a_{12}, a_{21}), (a_{11}), (a_{22}). (Note that loop in a flowgraph does depend upon direction whereas in linear graph theory this was not required)

Distinct types of loops must be defined in the Shannon-Happ Formula

(a) a first-order loop is a simple directed loop as defined above

(b) an Nth-order loop is composed of N disjoint first order loops (none of the nodes are common between loops) The value of an Nth order loop is the product of the N first-order loop values comprising the Nth-order loop In the example, only one second-order loop combination exists, namely, (a_{11} a_{22})

(c) a zeroth-order loop is by definition equal to a value one and has no flowgraph significance It is a mathematical convenience employed in the Shannon-Happ Formula

Shannon-Happ Formula

If L(0) is the value of the zeroth-order loop (always equals one) and L(N) is the sum of the values of all the loops whose order is N, the Shannon-Happ formula can be expressed as

$$H = L(0) - L(1) + \cdots + (-1)^N L(N) = 0$$

which can be written in expanded form[12]

$$H = 1 + \sum_{r=1}^{N} \sum_{i=1}^{R} L_i(r)(-1)^r = 0$$

where R = the number of r-order loops in the system and N = the order of the highest loop in the system.

3.4 Transfer Function Formulation

The circuit shown in Figure 3.7 is used to illustrate the circuit analysis techniques. The AC model of this circuit using only linear elements is shown in Figure 3.8. Note that an extra current source has been inserted at the output (across L). This element can be viewed as a voltmeter as described in section 2.

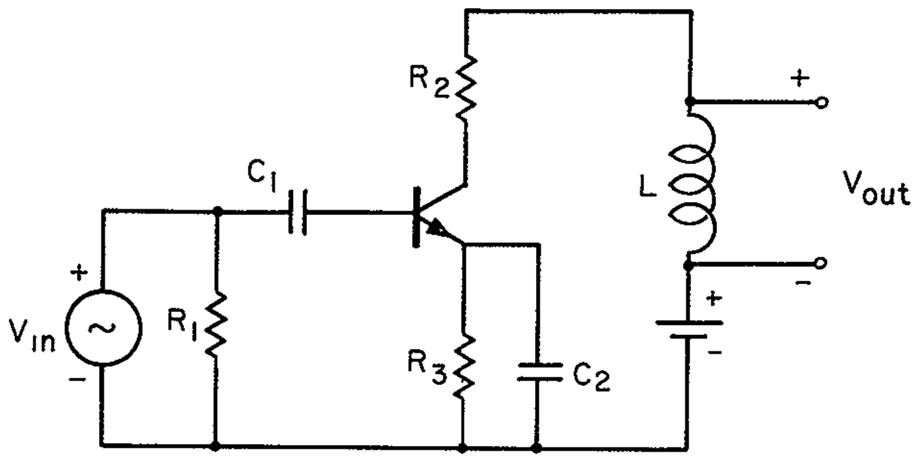


Figure 3 7

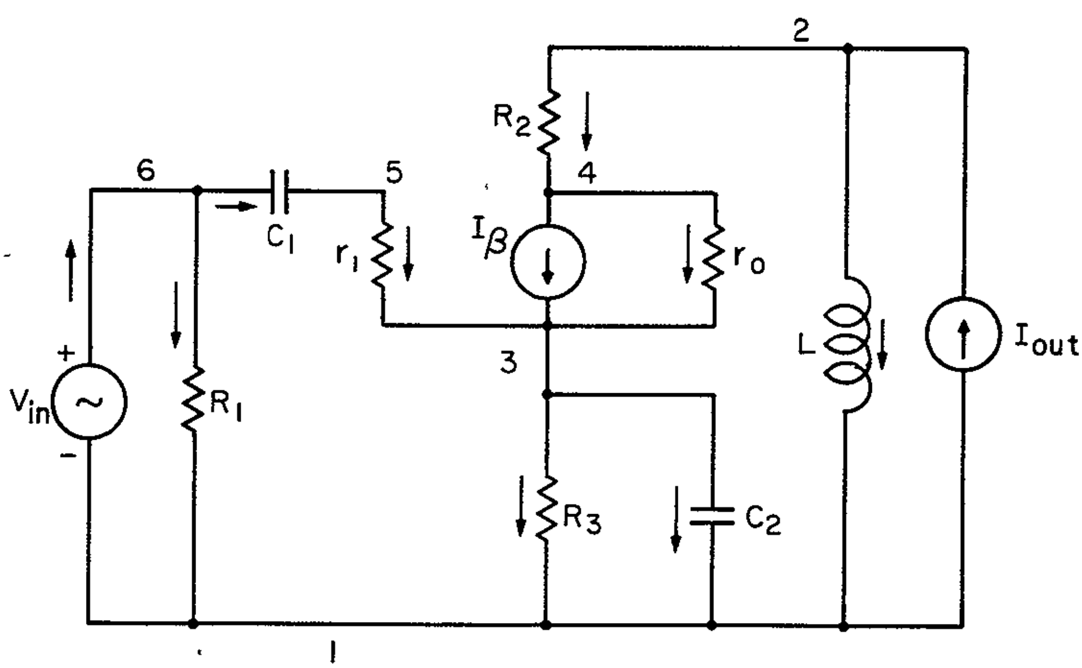


Figure 3 8 (All Transistors Modeling Parameters Are Represented by Lower Case Letters)

For the circuit in Figure 3 7, the output voltage is taken across the inductor L Since the voltage across L is one of the variables, the output voltage, V_{out}, can be expressed directly, that is, $V_{out} \leftarrow V_L$ By adding the output current source ($I \leftarrow 0$) in Figure 3 8, provision has been made for a general method of solution

After the equivalent circuit is constructed, its corresponding directed linear graph must be drawn with the assigned current direction in the circuit elements In the linear graph G one selects a tree structure T which must contain all voltage sources (dependent or independent) If the voltage sources form a loop, the circuit's definition is ambiguous and hence cannot be solved Assuming that the voltage sources do not form a loop, passive elements must be added to complete the tree To this end, some of the passive devices are made voltage-type elements and thus provide the remaining constituents for the tree (See Figure 3 9). Finally, the current sources and the rest of the

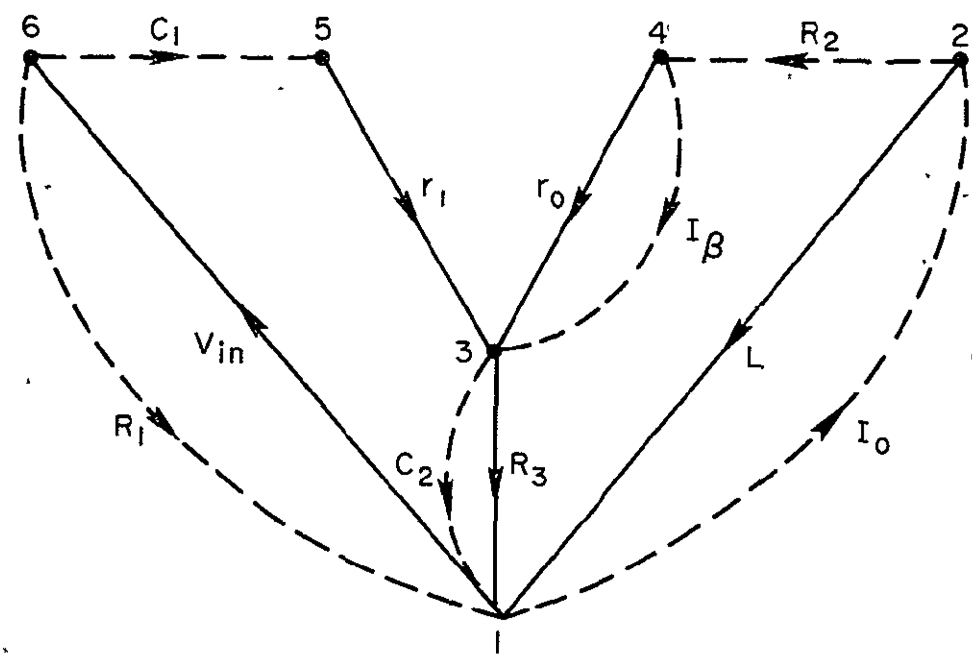


Figure 3 9.

passive elements form links (This operation constitutes the particular selection of either a voltage or current relation for each passive element That is $v \leftarrow i*z$ or $i \leftarrow v/z$ The next step requires one to obtain the current at the branches (voltage-type elements) as a function of the currents in the links (current-type elements). This entails writing the current equations for (n-1) out of n nodes of T. Then, the current in each branch is found as a function of link currents only by the Gaussian elimination process. By finding branch current as a function of link currents one forms the (n-1) cut sets.

The current equations are written by setting the branch currents on the left-hand side of the equals sign and the link currents on the right-hand side For Figure 3 9, the node current equations can be written as

Node 6	$-I_{V_{in}}$	=	$-I_{R_1} - I_{C_1}$
Node 5	I_{r_1}	=	I_{C_1}
Node 4	I_{r_0}	=	$-I_{\beta} + I_{R_2}$
Node 3	$-I_{r_1} - I_{r_0}\ I_{R_3}$	=	$I_{\beta}\ -I_{C_2}$
Node 2	I_L	=	$I_0\ -I_{R_2}$

In compact matrix notation

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I_{V_{in}} \\ I_{r_1} \\ I_{r_0} \\ I_{R_3} \\ I_L \end{bmatrix} = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} I_{r_1} \\ I_{C_1} \\ I_{\beta} \\ I_{C_2} \\ I_0 \\ I_{R_2} \end{bmatrix}$$

or $A*I_T = B*I_L$ where

A is an (n-1)*(n-1) matrix $\big((n-1)$ is the number of branches of the tree$\big)$,

I_T is a column vector of n branch currents,

B is an (n-1*(m) matrix (m=b-m+1 is the number of links),

I_L is a column vector of m link currents

After matrix A is diagonalized, performing the same operations on B as an A, the current equations take the form

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I_{V_{in}} \\ I_{r_1} \\ I_{r_0} \\ I_{r_3} \\ I_L \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} I_{r_1} \\ I_{C_1} \\ I_\beta \\ I_{C_2} \\ I_0 \\ I_{R_2} \end{bmatrix}$$

Note that the Gaussian elimination is performed on a set of integers, (+1, 0, -1) and will not entail any loss of significance or introduce any trancation errors

Following the Gaussian elimination that transforms the current equations into a unilateral set, the flowgraph can be constructed The first step is forming the current flowgraph relations as shown in the lower part of Figure 3 10 In the diagram (Figure 3. 10), two flowgraph nodes are assigned to each element The bottom nodes correspond to the currents in the elements and the top nodes correspond the voltage across the elements

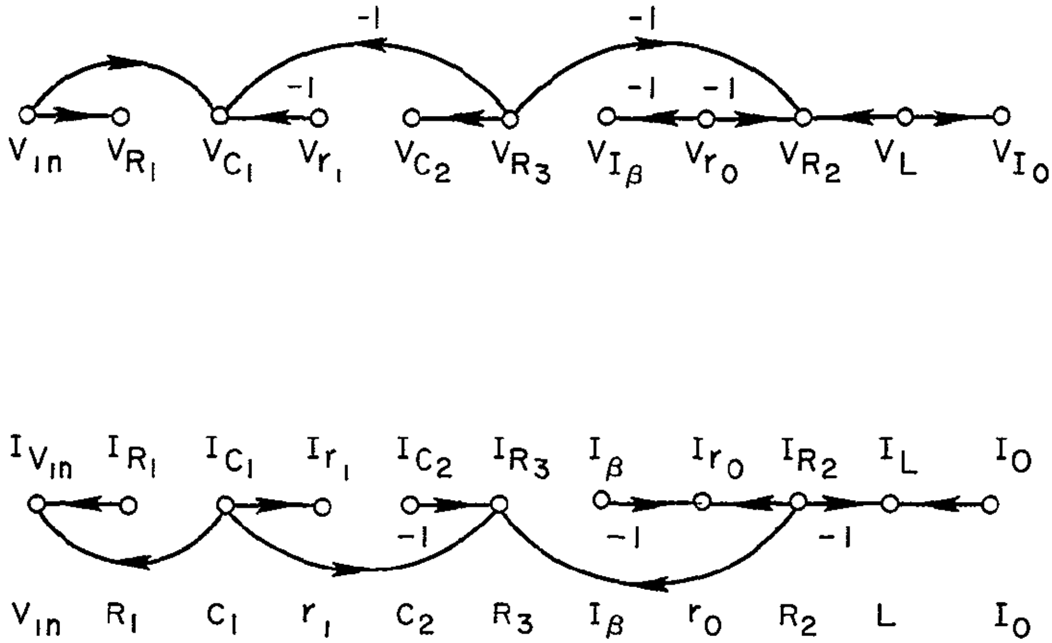


Figure 3 10

The flowgraph voltage relations are constructed by an imaging method based on the current relations The rules for this technique are

1 Consider two elements for which there exists current transmittance connecting two current nodes Form a transmittance connecting the two corresponding voltage nodes The direction of the voltage transmittance arrow is the reverse of the corresponding current transmittance

2 If one element is passive and the other is active, the sign attached to the voltage transmittance is the same as the corresponding current transmittance The sign is reversed if both elements are the same type, that is, both passive or both active The use of the imaging technique is a consequence of the fact that disjoint sets of current and voltage also form complementary sets since they are complementary cut and tie sets (The difference in the sign reversal between active and passive elements is due to the fact that conventionally used Kirchhoff' s equations are not written in a complementary form, that is, currents are summed to zero ($I_1 + I_2 + \quad I_n = 0$) while voltages are summed to a nonzero quantity ($V_1 + V_2 + \ . \ V_m = E$) Also Kirchhoff originally stated them in a complementary form

After Kirchhoff' s current law and Kirchhoff' s voltage law have been applied, the current-voltage relations and dependencies and requested transfer function are determined by making the input dependent on the output by the relation [Input ← P * Output]. This forms a closed flowgraph. The forward transfer function $T = 1/P$ can then be arrived at since the total gain $P*T = 1$ as evaluated by use of the Shannon-Happ Formula (for P unknown) This evaluation can be done by separation of products containing P and those terms which do not contain P since no higher powers of P than P^1 can occur.

A transfer function can be evaluated only as a function of a free variable (it cannot be found as a function of a variable which was already set as a determinable variable) The free variables are defined as. (1) the current in the independent current source and (2) the voltage across an independent voltage source. The transmittance P will cause one of the free variables to become a determinable variable (that is, it can be included in the evaluation of the Shannon Happ Formula) The rest of the free variables will be set to zero for the evaluation of the function A comparison between Figures 3 10 and 3.11 will indicate how this is accomplished Note that the current-voltage relations in passive elements are taken as impedances for branches and admittances for links

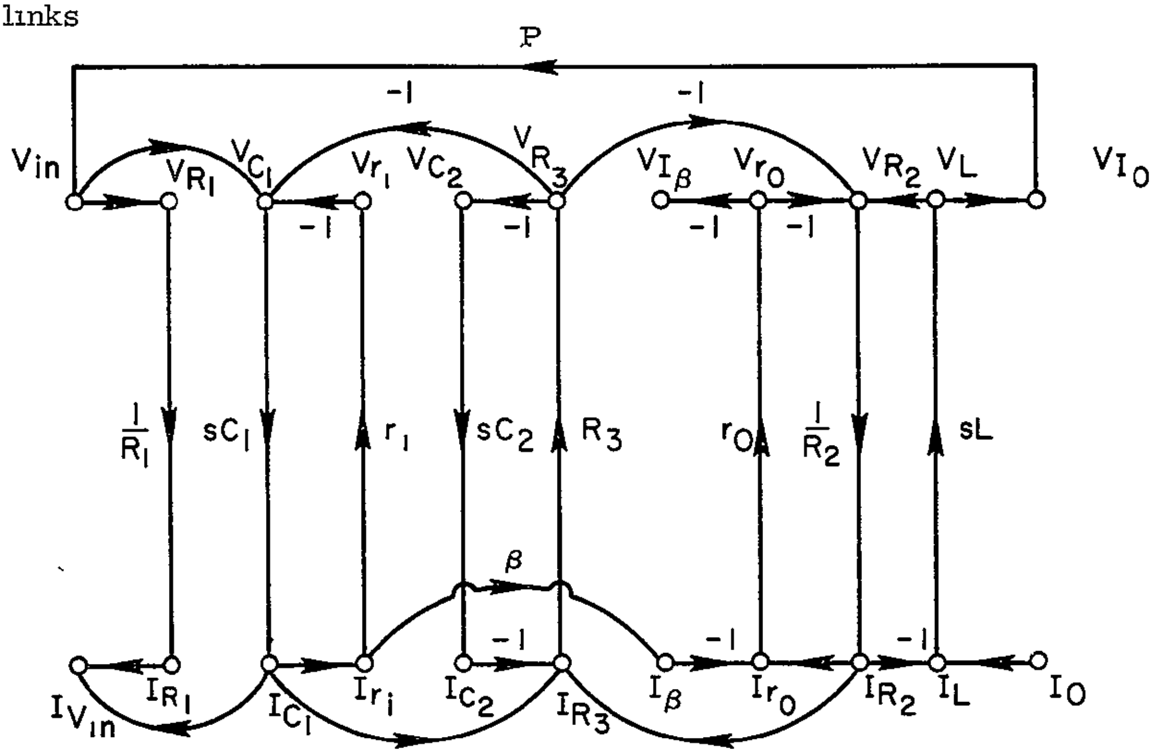


Figure 3 11

It is important to emphasize the uniqueness of transformations achieved by the algorithm presented herein and its advantages over techniques which are based on solutions of a set of equations This means that from the representations such as that presented in Figure 3 11 one can reconstruct the original system, thus showing that the algorithmic procedure described above does not incur information loss *

The transfer function V_{out}/V_{in} is calculated by use of the Shannon-Happ Formula In the present example there are nine first-order loops, which are summarized with their node sequence and loop transmittances in Table 3 1

The higher-order loops are unions of first-order loops that do not intersect and the loop value is the product of the corresponding first-order loop values. Loop substitution into the Shannon-Happ Formula, and the setting of H = 0, followed by solving for T = 1/P yields the requested transfer function,

$$\frac{V_{out}}{V_{in}} = T \leftarrow \left\{\frac{L_8}{P}\left(1 - L_2\right) + \frac{L_9}{P}\right\} \Big/ \left\{1 - \left(L_1 + L_2 + L_3 + L_4 + L_5 + L_6 + L_7\right)\right.$$
$$+ \left(L_1L_2 + L_1L_4 + L_1L_5 + L_1L_6 + L_2L_5 + L_2L_6 + L_3L_5 + L_3L_6\right)$$
$$\left. - \left(L_1L_2L_5 + L_1L_2L_6\right)\right\}$$
$$= - \left\{\left(\frac{\beta C_1 L r_o C_2 R_3}{R_2}\right) S^3 + \left(\frac{\beta C_1 L r_o}{R_2} - \frac{C_1 L R_3}{R_2}\right) S^2\right\} \Big/$$
$$\left\{\frac{C_1 C_2 L r_1 R_3}{R_2} S^3 + \left(C_1 C_2 r_1 R_3 + \frac{C_1 L r_1}{R_2} + \frac{C_2 L R_3}{R_2} + \frac{C_1 L R_3}{R_2}\right.\right.$$
$$\left. + \frac{C_1 C_2 r_1 R_3 r_o}{R_2}\right) S^2 + \left(C_1 r_1 + C_2 R_3 + C_1 R_3 + \frac{L}{R_2} - \frac{\beta C_1 r_o R_3}{R_2}\right.$$
$$\left.\left. + \frac{C_1 r_1 R_3}{R_2} + \frac{C_1 r_1 r_o}{R_2} + \frac{C_2 r_o R_3}{R_2} + \frac{C_1 r_o R_3}{R_2}\right) S + \left(1 + \frac{R_3}{R_2} + \frac{r_o}{R_2}\right)\right\}$$

---

*Although the proof is rather involved (thus not presented here), note that the information necessary to reconstruct the original circuit is given by the flowgraph and by preserving the I_T and I_L vectors in the arrangement of the flowgraph nodes

The solution $T = 1/P$ is obtained by placing the loops containing P and their products in the numerator and those devoid of P in the denominator

Since the only arithmetic in the entire algorithm is used in calculating the transfer function performed at the end of the algorithm and essentially is the sum of products, one can control the error and make the solution as accurate as desired Furthermore, comparison between values of loops is indicative of the significance of each element in the transfer function and is helpful in omitting the elements which have a negligible effect on the final transfer function

TABLE 3.1

Loop Number	Node Sequence	Transmission
L_1	$V_{C_1}\ I_{C_1}\ I_{r_1}\ V_{r_1}$	$-C_1 r_1 S$
L_2	$V_{C_2}\ I_{C_2}\ I_{R_3}\ V_{R_3}$	$-C_2 R_3 S$
L_3	$V_{C_1}\ I_{C_1}\ I_{R_3}\ V_{R_3}$	$-C_1 R_3 S$
L_4	$V_{R_3}\ V_{R_2}\ I_{R_2}\ I_{R_3}$	$-R_3/R_2$
L_5	$V_{r_0}\ V_{R_2}\ I_{R_2}\ I_{r_0}$	$\frac{-r_0}{R_2}$
L_6	$V_{R_2}\ I_{R_2}\ I_L\ V_L$	$-\frac{LS}{R_2}$
L_7	$I_{r_1}\ I_\beta\ I_{r_0}\ V_{r_0}\ V_{r_2}\ I_{r_2}\ I_{r_3}\ V_{r_3}\ V_{C_1}\ I_{C_1}$	$\frac{-\beta C_1 r_0 R_3 S}{R_2}$
L_8	$V_{in} V_{C_1}\ I_{C_1}\ I_{r_1}\ I_\beta\ I_{r_0} V_{r_0} V_{R_2}\ I_{R_2}\ I_L\ V_L V_{I_0}$	$-P\left(\frac{\beta C_1 L r_0}{R_2} S^2\right)$
L_9	$V_{in}\ V_{C_1}\ I_{C_1}\ I_{R_3}\ V_{R_3}\ V_{R_2}\ I_{R_2}\ I_L V_L V_{I_0}$	$P\left(\frac{C_1 L R_3}{R_2} S^2\right)$

3.5 Sensitivities Concepts

There are three basic sensitivity algorithms employed in NASAP-70. These are parameter sensitivity, worst case (tolerance) analysis, and pole sensitivity. A description of each follows.

3.5.1 Parameter Sensitivity[13]

In section 3.4, the transfer function $T = \frac{1}{P}$ was obtained by keeping track of the loops that contained P and those devoid of P, and forming the ratio of the two expressions. As shown below, the somewhat similar tagging technique in the computer algorithms can produce the sensitivity of the transfer function to changes in a circuit parameter

Consider the system of Figure 3 12 where X is the input or excitation and Y is the output or response If the input and output are connected by a transmittance P, then the transfer function can be expressed in terms of P since $P = \frac{x}{y} = \frac{1}{T}$ This represents a closed system for which the input and output are dependent upon each other The Shannon-Happ Formula states that the topology equation H(T) must equal zero, thus

$$H(T) = H(\overline{T}) + TH(T') = 0$$

where $H(T') = \frac{dH}{dT}$ = the portion of H containing T linearly and indicated in the computer algorithm by a tag of T = 1, and

$$H(\overline{T}) + H(T{=}0) = \text{the portion of H devoid of T}$$

and indicated in the computer algorithm by a tag of T = 0

To compute the sensitivity function, consider a circuit that contains a component Q and sustains a desired performance specification T, the transfer function It is always possible to formulate a closed system containing the two parameters T and Q linearly The topology equation H = 0 is a constraint on the system from which the unknown T is calculated

If the topology equation H is a function of two parameters T and Q, then

$$H(T, Q) = H(\bar{T}, \bar{Q}) + TH(T', \bar{Q}) + QH(\bar{T}, Q') + TQH(T', Q')$$

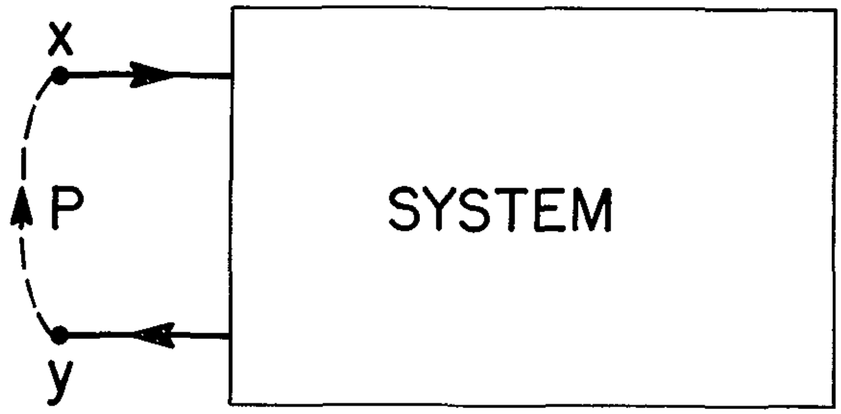


Figure 3 12

where the coefficients of T and Q are defined as Taylor series coefficients and the terms of the topology equation H(T, Q) are defined as follows

$$H(\bar{T}, \bar{Q}) = H(T=0,\ Q=0) = H(0, 0)$$

$$H(T', \bar{Q}) = H(T=1,\ Q=0) = H(1, 0)$$

$$H(\bar{T}, Q') = H(T=0,\ Q=1) = H(0, 1)$$

$$H(T', Q') = H(T=1, Q=1) = H(1, 1)$$

All of these quantities refer to partial sums of the topology equation which can be tagged and identified in the computer program.

A topological derivation of sensitivity is obtained by taking the total derivation of H(T, Q) above.

$$dH(T, Q) = \left| H(T', \bar{Q}) + QH(T', Q') \right| dT + \left| H(\bar{T}, Q') + TH(T', Q') \right| dQ = 0$$

or

$$\frac{dT}{dQ} = -\frac{H(\bar{T}, Q') + TH(T', Q')}{H(T', \bar{Q}) + QH(T', Q')}$$

since

$$H(T, Q) = \left\{H(\bar{T}, \bar{Q}) + H(\bar{T}, Q')\right\} + T\left\{H(T', \bar{Q}) + H(T', Q')\right\} = 0$$

and

$$H(T, Q) = \left\{H(\bar{T}, \bar{Q}) + H(T', \bar{Q})\right\} + Q\left\{H(\bar{T}, Q') + H(T', Q')\right\} = 0$$

therefore,

$$\frac{Q}{T} = \frac{[H(\bar{T}, \bar{Q}) + H(T', \bar{Q})]\,[H(T', \bar{Q}) + H(T', Q')]}{[H(\bar{T}, \bar{Q}) + H(\bar{T}, Q')]\,[H(\bar{T}, Q') + H(T', Q')]}$$

The parameter sensitivity is defined by

$$S_Q^T = \frac{d \log T}{d \log Q} = \frac{\frac{dT}{T}}{\frac{dQ}{Q}}$$

or,

$$= -\frac{H(T', \bar{Q}) + H(\bar{T}, \bar{Q})}{H(\bar{T}, \bar{Q}) + H(\bar{T}, Q')}$$

3.5.2 Worst Case (Tolerance) Analysis[13]

The sensitivity techniques outlined in section 3.5.1 may be used to perform a worst case tolerance analysis on all components of the circuit. Assume that a given circuit specification P can be specified as a function of n parameters Q_1, Q_2 --- Q_n,

$$P = f(Q_1, Q_2, \ldots Q_n)$$

Then for small changes in the parameter, the statistical tolerance T_P of P is defined as

$$T_P = \left[\left(\frac{\partial P}{\partial Q_1} Q_1\right)^2 + \cdots + \left(\frac{\partial P}{\partial Q_n} Q_n\right)^2\right]^{1/2}$$

T_P is a measure of the deviation of P from its mean value due to deviations of the components from their respective mean values.

Using the definition of sensitivity T_P can be expressed in a more suitable form for utilization of the tagging technique as

$$\frac{T_P}{P} = \sum_{i=1}^{n} \left[S_{Q_n}^{P} \frac{dQ_n}{Q_n} \right]^2$$

where $\frac{T_P}{P}$ is the functional tolerance.

The sensitivity of P with respect to each parameter Q may be calculated as before, then substituted into the $\frac{T_P}{P}$ expression to derive the tolerance.

3.5.3 Root Sensitivity[14]

When a parameter of a system changes, the locations of the poles will also change. Root sensitivity is defined as a measure of the amount of change a root undergoes, given a percent change in some system parameter.

The following analysis assumes that (1) the system is linear and time invarient, and (2) the transfer function is obtainable and can be expressed by the form

$$T(S) = \frac{A(S) + KB(S)}{C(S) + KD(S)}$$

Here, A(S), B(S), C(S), and D(S) are polynomial in S and K is a system parameters for which the sensitivity is desired.

For differential changes in the parameter K, there will be differential changes in the roots of the characteristic equation, C(S) + KD(S). The definition of root sensitivity employed in NASAP-70 considers one system parameter, K, and one pole of the transfer function S_i, and is defined as

$$\mathcal{S}_{K}^{S_i} = \frac{dS_i}{dK} \quad K$$

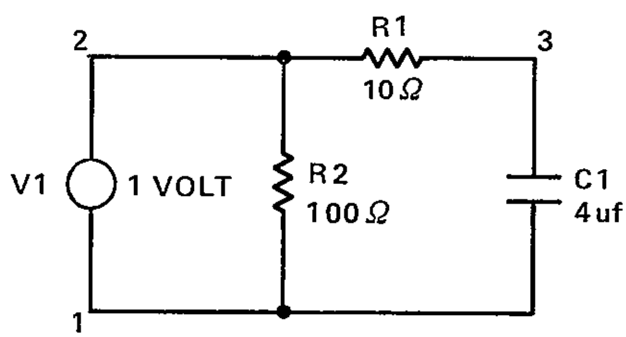
If the transfer function of the system is given above for T(S), then the root sensitivity can be found by evaluating the residue of the transfer function at the poles of the transfer function, or

$$\mathcal{S}_{K}^{S_1} = \left. \frac{-KD(S)}{\frac{\partial}{\partial S}\left[C(S) + KD(S)\right]} \right|_{S=S_1}$$

Here S_1 is a pole described in the unperturbated system. This operation is performed in the program with simple tagging technqiue, just as in the parameter sensitivity case.

3.5.4 Sample Calculations

Given the following circuit, the following parameter sensitivity, worst case, and pole sensitive calculations are made.



NASAP

V1	1	2	1
R1	2	3	10
C1	3	1	4UF
R2	2	1	100

END

VC1/VV1

LOOP 1 contains R1, C1, P has value of $-2.5 \times 10^4 S^{-1}$

LOOP 2 contains R1, C1 has value $2.5 \times 10^4 S^{-1}$

ZERO ORDER LOOP has value 1

Parameter Sensitivity

$H(P') = -2.5 \times 10^4 S^{-1}$

$H(\bar{P}) = 2.5 \times 10^4 S^{-1} + 1$

$H(\bar{P}, \overline{R1}) = 1$ $\quad H(P', \bar{C}_1) = 0$

$H(\bar{P}, \overline{C1}) = 1$ $\quad H(P', \bar{R}_1) = 0$

$H(\bar{P}, \overline{R2}) = 2.5 \times 10^4 S^{-1} + 1$ $\quad H(P', \overline{R2}) = -2.5 \times 10^4 S^{-1}$

$$S_Q^P = \frac{H(P', \bar{Q})}{H(P')} - \frac{H(\bar{P}, \bar{Q})}{H(\bar{Q})}$$

$$S_{R1}^P = \frac{0}{-2.5 \times 10^4 S^{-1}} - \frac{1}{2.5 \times 10^4 S^{-1}+1} = - \frac{1}{2.5 \times 10^4 S^{-1}+1}$$

$$S_{C1}^P = \frac{0}{-2.5 \times 10^4 S^{-1}} - \frac{1}{2.5 \times 10^4 S^{-1}+1} = \frac{-1}{2.5 \times 10^4 S^{-1}+1}$$

$$S_{R2}^P = \frac{-2.5 \times 10^4 S^{-1}}{-2.5 \times 10^4 S^{-1}} - \frac{2.5 \times 10^4 S^{-1}+1}{2.5 \times 10^4 S^{-1}+1} =$$

$$\frac{-6.25 \times 10^8 S^{-2} - 2.5 \times 10^4 S^{-1} + 6.25 \times 10^8 S^{-2} + 2.5 \times 10^4 S^{-1}}{-6.25 \times 10^8 S^{-2} - 2.5 \times 10^4 S^{-1}} =$$

$$\frac{0}{-6.25 \times 10^8 S^{-2} - 2.5 \times 10^4 S^{-1}}$$

$$\text{TRANSFER FUNCTION} = \frac{-H(P')}{H(\bar{P})} = \frac{2.5 \times 10^4 S^{-1}}{2.5 \times 10^4 S^{-1}+1}$$

Pole Sensitivity

POLE $2.5 \times 10^4 + S = 0$

$S_1 = - 2.5 \times 10^4$

$$\text{Pole Sensitivity} = \left. \frac{-H(\bar{P}, Q')}{\frac{\partial}{\partial S}[H(\bar{P})]} \right|_{S = S_1}$$

$$\frac{\partial}{\partial S}[H(\bar{P})] = -2.5 \times 10^4 S^{-2}$$

$H(\bar{P}, R1') = 2.5 \times 10^4 S^{-1}$

$H(\bar{P}, C1') = 2.5 \times 10^4 S^{-1}$

$H(\bar{P}, R2') = 0$

$$\text{Pole Sensitivity to R1} = \frac{-2.5 \times 10^4 S_1^{-1}}{-2.5 \times 10^4 S_1^{-2}} = S_1 = -2.5 \times 10^4$$

$$\text{Pole Sensitivity to C1} = \frac{-2.5 \times 10^{4} S_1^{-1}}{-2.5 \times 10^{4} S_1^{-2}} = S_1 = -2.5 \times 10^{4}$$

WORST CASE Analysis

$$(TP/P)^2 = \sum_{i=1}^{n} \left(S^{P}_{Qi} \frac{dQi}{Qi} \right)^2$$

$$\frac{dR1}{R1} = \frac{dC1}{C1} = \frac{dC2}{C2} = .1$$

$$(TP/P)^2 = \frac{(-1)^2 (.01)}{(2.5 \times 10^{4} S^{-1}+1)^2} + \frac{(-1)^2 (.01)}{(2.5 \times 10^{4} S^{-1}+1)^2} +$$

$$+ \frac{(0)^2 (.01)}{(-6.25 \times 10^{8} S^{-2} - 2.5 \times 10^{4} S^{-1})^2}$$

$$= \frac{.02}{(2.5 \times 10^{4} S^{-1}+1)^2} + \frac{0}{(2.5 \times 10^{4} S^{-1}+1)^2 (-2.5 \times 10^{4} S^{-1})} =$$

$$= \frac{.02(-2.5 \times 10^{4} S^{-1})^2 + 0}{(2.5 \times 10^{4} S^{-1}+1)^2 (-2.5 \times 10^{4} S^{-1})^2} =$$

$$= \frac{1.25 \times 10^{7} S^{-2}}{3.90625 \times 10^{17} S^{-4} + 3.125 \times 10^{13} S^{-3} + 6\ 25 \times 10^{8} S^{-2}}$$

3.6 The Svoboda Method for Computing Roots of Polynomials

Overview of the Algorithm The flow of control among the various logical routines of the algorithm is depicted in Figure 3.13. After obtaining the input data, the Scanning Routine systematically steps over the Unit Circle evaluating the reduced polynomial at five points on each step. When a criterion for the possible existence of a root at any step is fulfilled, a transfer is made to the Home-in-Routine. The Home-in-Routine approximates the root as closely as possible and then transfers to the Root Examination Routine. Using the Round-off Routine, the Root Examination Routine

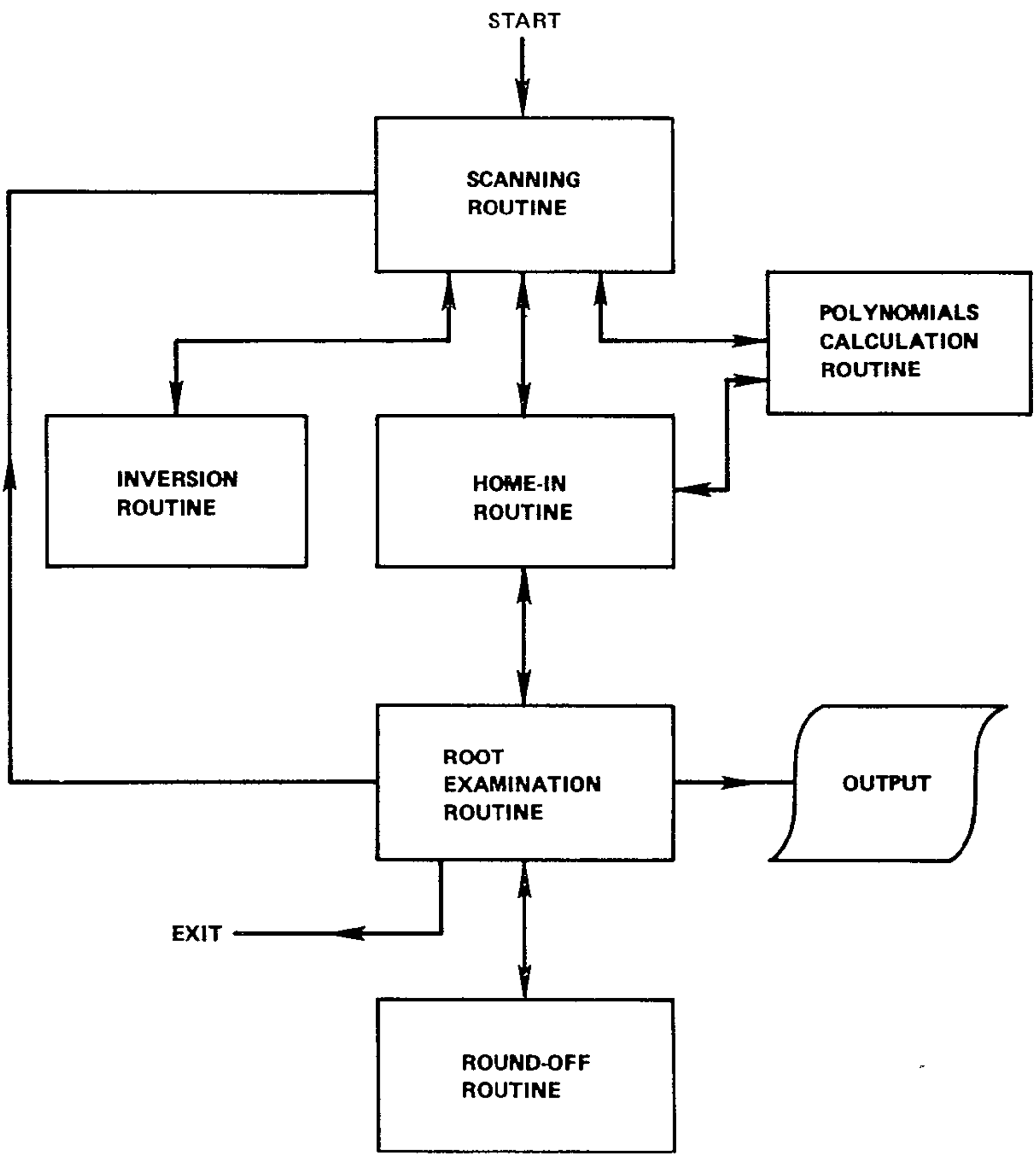


Figure 3 13 Flow of Control in Svoboda Algorithm either rejects the root or refines it (by means of the Home-in-Routine again) and outputs it. Control then passes back to the Scanning Routine which either continues scanning or stops because the required number of roots has been obtained. When a scan is completed, if the maximum number of scans has not yet been reached, the Inversion Routine gains control to "invert" the polynomial, before returning to the Scanning Routine which continues to scan. A detailed discussion of each element of the algorithm follows.

The Scanning Routine The main routine in the Algorithm is the Scanning Routine which moves across the Unit Circle as shown in Figure 3. 14. The real and imaginery axes between the ranges (-1, 0), (1, 0) and (0, -i), (0, i) are divided into sixteenths. The Scanning Routine uses the Polynomial

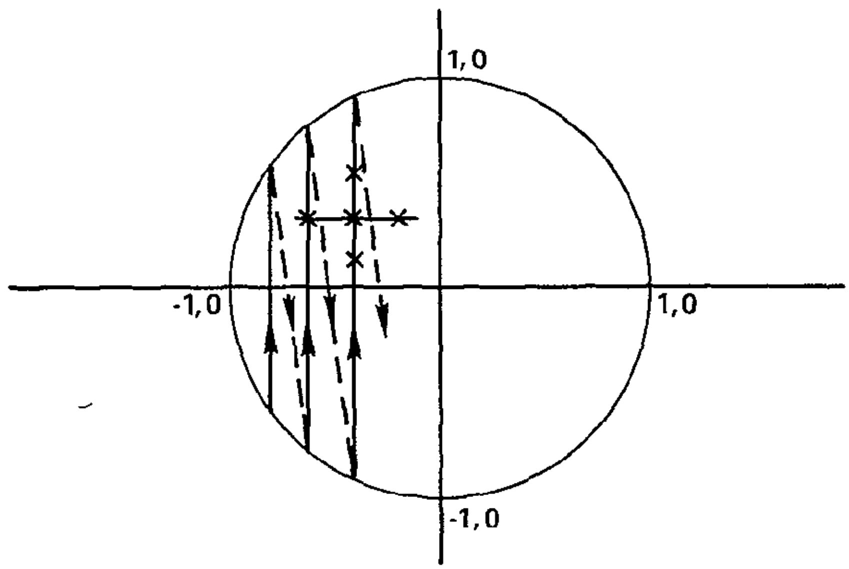


Figure 3 14 The Five Point Scan

Evaluation Routine to find five points at each step in the scan, the central point and four points one sixteenth above, below, to the left and to the right of the central point. A test is then performed to see if this point may be near a root. The criteria for transferring to the Home-in-Routine to examine a possible root more closely are

1. The value at the central point must be less than the value of any of the four surrounding points.
2. The five points must not be equal in value.

If these criteria are not met, the scan is incremented by one sixteenth and the scanning process is resumed. Scanning is performed from left to right and from bottom to top. When a scan has been completed, that is, when the point (1, 0) has been evaluated, a check is performed to see if the number of scans has exceeded an upper limit. If not, a transfer is made to the Inversion Routine (described below) which performs the inversion of the polynomial and then returns control to the Scanning Routine.

The maximum number of scans allowable is twice the number of roots in the polynomial. Since at least one root must be obtained after each pass of both the "real" polynomial and the "inverted" polynomial (requiring a total of two scans), a number of scans no more than twice the number of roots should be required for the algorithm to work properly. (Usually, considerably fewer scans are required because several roots may be found on any one pass). If this number is exceeded, however, the routine stops automatically.

The problem of evaluating the circumference of the Unit Circle twice (once on a "real" scan and once on an "inverted" scan) is solved by taking two precautions. Firstly, all points evaluated in a "real" scan must be within the Unit Circle, while an "inverted" scan is permitted to violate the boundary of the circumference slightly. This precaution alone, however, is not sufficient to avoid a duplication of roots which lie close to the boundary, therefore, in the Root Examination Routine a check is performed to see if the root under consideration has appeared previously in another scan. This second precaution prevents roots which have been found once by a "real" scan and once by an "inverted" scan from appearing twice.

Polynomial Calculation The Polynomial Calculation Routine uses Horner's Technique to evaluate the polynomial complex value from the coefficients. Basically, Horner's Technique is the iterative evaluation of the expression

$$F_{i+1} = (F_i + a_i) \cdot x \qquad 1 \leqq i \leqq n$$

with $F_1 = (0,0)$ and $F_{n+1} = F_n + a_{n+1}$ the final value, and a_i is the i'th coefficient in the polynomial

$$P(x) = a_1 x^n + a_2 x^{n-1} + \cdots + a_n x + a_{n+1}.$$

Thus the final value is cumulatively built up in F. If necessary, the complex value F is divided by previously found roots to form a reduced polynomial value

$$F = \frac{P(x)}{(R_1 - x)(R_2 - x) \cdots (R_m - x)}$$

where m is the number of roots found previously. In order to avoid overflow, the absolute value of any of the factors $(R_i - x)$ is not allowed to be less than 10^{-60}. The final step is to obtain the residual (the complex absolute value) of F.

$$H = \sqrt{([\mathrm{Re}(F)]^2 + [\mathrm{Im}(F)]^2}$$

The residual is computed for four or five points (the Home-in-Routine does not require the central point to be evaluated each time) on each entry to the Calculation Routine. It is these four or five values (H) which are used by the Scanning and Home-in-Routines as the polynomial values at the test points.

Polynomial Inversion In order to obtain all the roots of the polynomial

$$P(x) = a_1 x^n + a_2 x^{n-1} + \cdots + a_n x + a_{n+1}$$

by scanning the complex plane within the Unit Circle, at some point a new polynomial is formed from the original one by reversing the order of the coefficients. The roots of this new polynomial are the reciprocals of the roots of the original polynomial, thus effectively bringing inside the Unit Circle all the roots which were previously outside. The Inverted Polynomial is

$$P'(y) = a_{n+1}\, y^n + a_n y^{n-1} + \cdots + a_2 y + a_1 \,.$$

If $P(r) = 0$, it is easy to show that

$$P(r) = 1/r^n P(r) = P'(1/r)$$

Therefore $y = 1/r$ is a root of the new polynomial p'.

Thus, in order to obtain all the roots of a polynomial, the program must deal effectively with two kinds of roots, "real" and "inverted", and must obtain the reciprocal of "inverted" roots before writing them on the output. The Polynomial Inversion routine, simply reverses the order of the coefficients, sets a program variable to point to either the "real" or "inverted" techniques, and returns control to the main program.

Homeing-In Once five points have been obtained which fulfill the criterion for root examination, the homing-in procedure subdivides the distance between the central point and one of the outside points into sixteenths and, using the central points as a starting value, employs the Polynomial Evaluation Routine to compute four new outside test points. These new four points and the central point are then examined for the one having the smallest residual which becomes the central point for a new five point test. If this new central point is the same as the previous central point, the scale is subdivided into sixteenths as before so that the five point test becomes progressively more refined. At each subdivision, a new level of significance is obtained (each significant digit being a Hexadecimal digit because the scale factor is 16). This process of "homeing-in" is continued until either

1. The required number of digits of accuracy is reached or,
2. The residuals of the five points do not differ by a significant amount.

In case 2, the resolution of the computing method and system has been reached and the remaining significant digits, if any, are filled with zeroes. This represents a Stopping Criterion which is independent of any formula for found-off error,[15] but depends only on the previously mentioned resolution.

Each time a new step is taken in the Home-in-Routine, three checks are made. Firstly, if the central point should happen to be the origin (0, 0), the scale is expanded (subdivided by sixteenths), this allows for roots which may range over large orders of magnitude. Secondly, if more than sixteen steps should be taken in one direction at the same level of significance the level of significance is dropped so that larger (coarser) steps may be taken. Finally, a check is performed to ensure that the routine does not stray too far out of the Unit Circle. This safeguards against the possibility that the original location presented to the Home-in-Routine was not in the vicinity of a root, in which case it could easily wander outside the Unit Circle. If this condition should be detected, the Home-in is aborted and control is returned to the Scanning Routine.

If either case 1 or case 2 occurs, the Home-in-Routine has reached a successful completion and transfers control to the Root Examination Routine.

Root Examination Procedures The Root Examination Routine first checks to see if the root just found came from the original polynomial coefficients or from a reduced polynomial. If the polynomial was a reduced one, the value of the root can be regarded merely as an approximation. The Home-in-Routine is therefore called again to repeat the latter part of the home-in procedure this time using the original polynomial and using the approximate root as a starting value. This technique ensures that every root is found from the original polynomial and that its accuracy does not depend on the accuracy with which previous roots were found.

After the final root value is obtained, a second check is made, this time on its residual. If the residual is greater than a certain value, the root is considered unreliable and is discarded. If this should occur, control is transferred back to the Scanning Routine. The maximum value of the residual is chosen arbitrarily to be one half the value of a_{n+1}, the constant coefficient in the polynomial.

Having passed these tests, the root value is rounded-off in the Round-off Routine (which is not discussed here because it has no bearing on the actual rootfinding algorithm). If the root is "inverted", its reciprocal is found using floating point division, and this value, after being rounded-off again, becomes the root value. A third check is then performed on the root to see if it has been found on a previous pass. (This check was mentioned previously in connection with the Unit Circle Boundary). If both real and imaginary parts fall within one significant digit of a root found on a different type of pass ("real"), it is rejected as a root but is nevertheless entered, in the same way as acceptable roots, in the list of roots used to form a reduced polynomial. (This prevents its being found again). If the root is not rejected, it is converted form its internal hexadecimal representation to decimal, rounded-off again, and is written on the output.

A final check is made to see if the required number of roots has been obtained. If not, control is returned to the Scanning Routine, otherwise the algorithm terminates.

NOTES

1. In writing this algorithm, it was intended to produce a technique for finding successive roots from the original polynomial and thus eliminate the mutual interdependence of these roots and convergence problems which are commonly found in other methods, (e. g., Newton-Raphson, Muller).[16-19] However, in the course of investigating the behavior of the algorithm it was discovered that even widely dispersed roots affected the behavior of the polynomial "surface" over a wide range, and it therefore became necessary to use a reduced polynomial as a first approximation to eliminate this interference. Nevertheless, since the final root is always obtained from the original unreduced coefficients, the original algorithm has been effectively retained.

2. Since most major computing machines in this country are based on some multiple of binary arithmetic (and, in particular, hexadecimal arithmetic), a step size of sixteenths was used as the basis for the algorithm. On decimal based machines, it is likely that the algorithm would prove more effective if tenths were to be used.

3. Although no organized test of the algorithm has been performed as yet, it has performed well over a wide class of applications. In particular, 34 of the 36 roots of the 36th degree polynomial given in reference 19 were found to 5 significant digits in Single Precision Arithmetic within 60 seconds on an IBM 360/91.

3.7 A C. Analysis-Bode Plot[20-21]

At this stage of the program, with the transfer function available as

$$T = \frac{b_o S^m + b_1 S^{m-1} + \text{------} + b_m}{a_o S^m + a_1 S^{m-1} + \text{------} + a_n} ,$$

where all the coefficients, a_i and b_i, are known, the A C response is calculated by setting $S = j\omega$ and simplifying the expression to a linear combination of real and imaginary terms

$$T = A(\omega) + jB(\omega)$$

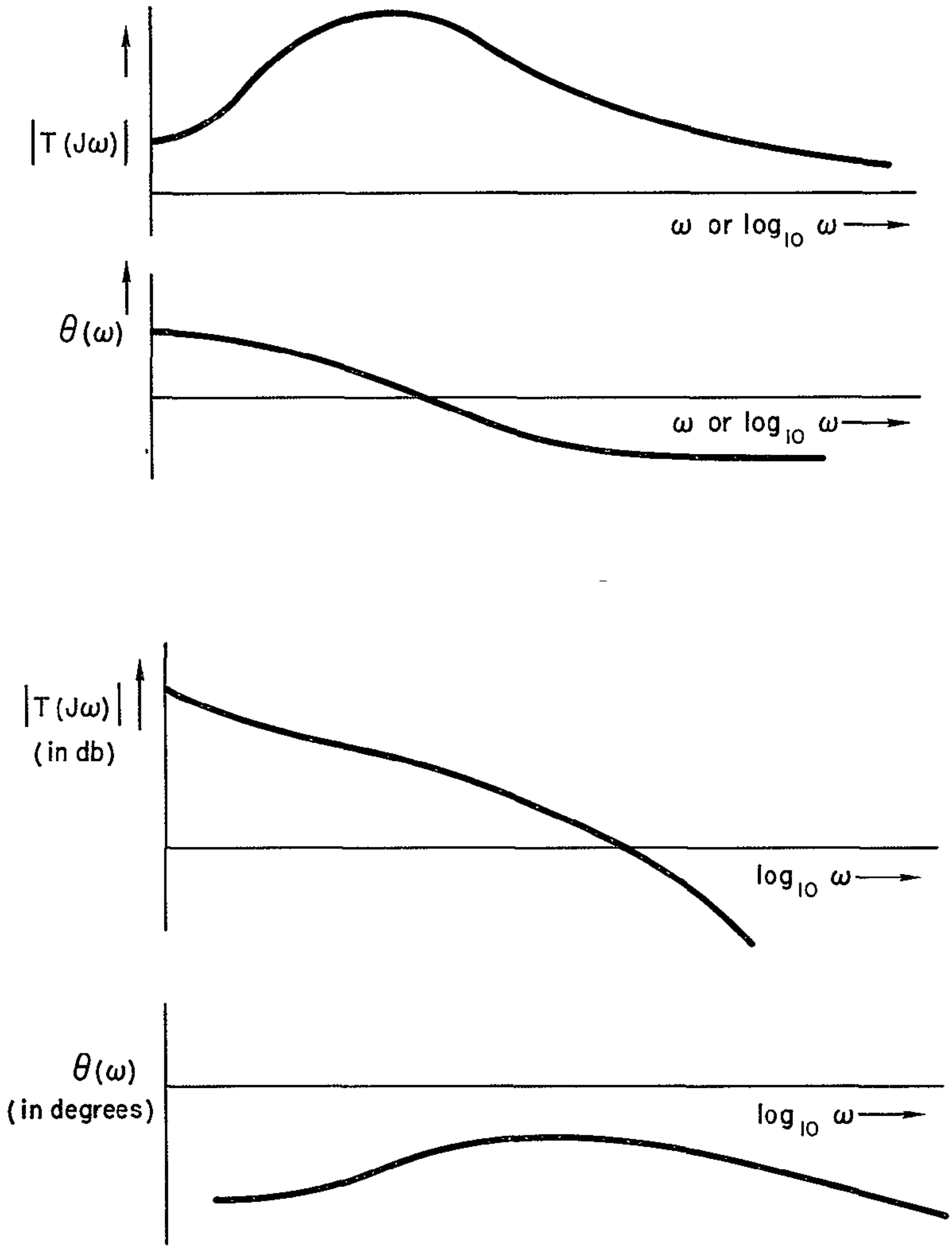
The magnitude of T and the angle $\theta(\omega)$ are computed according to the equations,

$$|T(j\omega)| = \sqrt{A^2(\omega) + B^2(\omega)} ,$$

and

$$\theta(\omega) = \tan^{-1} \frac{B(\omega)}{A(\omega)} .$$

Now if ω is made to vary, then for each value of ω the $|T(j\omega)|$ and $\theta(\omega)$ can be obtained over the frequency range of interest and thus can be made available for plotting. One commonly employed plot is $|T(j\omega)|$ and $\theta(\omega)$ versus either $\log_{10}\omega$ or ω of Figure 3.15. Another graph, a Bode plot, (see Figure 3 16) consists of the $|T(j\omega)|$ in decibel units and $\theta(\omega)$ in degrees versus the $\log_{10}\omega$, taken over the frequency range specified by the user



|T (Jω)|
ω or log10 ω
θ(ω)
ω or log10 ω
|T (Jω)|
(in db)
log10 ω
θ(ω)
(in degrees)
log10 ω

Figure 3 16

With the complex arithmetic capability of FORTRAN IV, the desired computation can easily be carried out in NASAP-70.

3.8 Transient Response

The transient response routine in NASAP-70 derives its theoretical basis from Fast Fourier Transform methods for the numerical inversion of Laplace Transform. Although the techniques described here follow along the approach presented by Dubner and Abate,[22] it was, however, developed independently at UCLA.

Theory The inverse f(t) of a Laplace Transform F(s) is given by the formula

$$f(t) = \frac{1}{2\pi j} \int_{\sigma - j\infty}^{\sigma + j\infty} F(s)\, e^{st}\, dS \tag{3.1}$$

If F(s) is a quotient of polynomials, with all coefficients real, i.e.,

$$F(s) = \frac{\sum_{i=0}^{K} a_i s^i}{\sum_{i=0}^{n} b_i S^i} \tag{3.2}$$

then f(t) is real, and

$$f(t) = \frac{e^{\sigma t}}{\pi} \int_0^{\infty} F(\sigma + j\omega) e^{j\omega t}\, d\omega \tag{3.3}$$

For the F(s) which are rational, i.e., have K > n in Equation (3.2), it is apparent that Equation (3.3) can be evaluated by a gradrature rule for a specific t, since $\lim_{S \to \infty} F(s) = 0$. However, if K > n, a long division can be performed on F(s), to yield

$$F(S) = C_{K-\ell} S^{K-\ell} + C_{K-\ell-1} S^{K-\ell-1} + \cdots + C_o + R(S) \tag{3.4}$$

where

$$R(S) = \frac{\sum\limits_{i=0} d_i S^i}{\sum\limits_{j=0} b_j S^j} \tag{3. 5}$$

All terms of (3. 4) represent impulses of varying degrees and can easily be inverted. The only concern is the inversion of R(S).

Inversion of Equation (3. 3) by the trapezoidal rules, gives

$$f(t) \approx \Delta w \left\{ \frac{\operatorname{Re}\{F(\sigma)\}}{2\pi} + \frac{e^{\sigma t}}{\pi} \sum_{K=1}^{\infty} F(\sigma + jK\Delta w) e^{jk\Delta\omega t} \right\} \tag{3.6}$$

where ω is an appropriately chosen sampling interval, and σ is chosen so that the integration path is to the right of all singularities. Since $\lim\limits_{S\to\infty} F(S)=0$, an upper limit ν can be chosen for the summation such that $F(\sigma+j\Delta\omega k)$ is small for $K > n$. Thus

$$f(t) \approx \Delta\omega \left\{ \frac{\operatorname{Re}\{F(\sigma)\}}{2\pi} + \frac{e^{\sigma t}}{\pi} \sum_{K=1}^{n} F(\sigma + jk\Delta\omega) e^{jk\Delta\omega t} \right\} \tag{3 7}$$

(It should be noted that Equation (3 7) is the same expression obtained by Dubner and Abate,[22] except that the present derivation did not consider that when $f(t)$ is known to be real and nonzero for $t \geqq 0$ only, then

$$f(t) = \frac{2e^{\sigma t}}{\pi} \int_0^{\infty} \operatorname{Re}\{F(\sigma t)\omega)\} \cos(\omega t) d\omega \tag{3. 8}$$

Note that if $f(t)$ is obtained from the trapezoidal rule for m different values of t, a total of mN calculations (evaluations of the term appearing under the summation) are required. For small m, no difficulty exists, but if m is large, the machine time could be excessive.

This difficulty is overcome by the use of the Fast Fourier Transform (FFT).[23, 24] The discrete Fourier Transform pair

$$X(K) = \Delta t \sum_{n=0}^{N-1} x(n) e^{-j2\pi \frac{nK}{N}} \tag{3.8a}$$

$$x(n) = \frac{\Delta\omega}{2\pi} \sum_{K=0}^{N-1} X(K)\, e^{\frac{2\pi nK}{N}} \tag{3.8b}$$

can be evaluated using the FFT with less than $N \log_2 N$ calculations if N is chosen so that $N = 2^{\delta}$, where δ is a positive integer.

In Equation (3.7), if the following definitions are adopted

$$Z_K = \begin{cases} \dfrac{\mathrm{Re}\{F(\sigma)\}}{2} & K = 0 \\ \mathrm{Re}\{F(\sigma + j\Delta\omega K\} & K > 0 \end{cases}$$

and the summation limit is raised to N-1, where N is the smallest integral power of 2 greater than ν, then,

$$f(t) \approx \frac{\Delta\omega\, e^{\sigma t}}{\pi} \sum_{K=0}^{N-1} Z_K e^{jK\Delta\omega t} \tag{3.9}$$

If it is required that f(t) is evaluated only at integral multiples of some time interval Δt, then f(t) can be defined by.

$$f(n\Delta t) \approx \frac{\Delta\omega\, e^{\sigma n\Delta t}}{\pi} \sum_{K=0}^{N-1} Z_K e^{jK\Delta\omega\, n\Delta t} \tag{3.10}$$

for $n = 0, 1, 2, \ldots$

Now, if $\Delta t \Delta\omega = \dfrac{2\pi}{N}$ is introduced, Equation (3.10) becomes

$$f(n\Delta t) \approx \frac{\Delta\omega\, e^{\sigma n\Delta t}}{\pi} \sum_{K=0}^{N-1} Z_K e^{j2\pi \frac{nK}{N}} \tag{3.11}$$

It is apparent that the form of Equation (3.11) is compatble with Equation (3 8b). It should be noted that the summation in (3.11), considered as a function of n, is periodic, and that $f(n\Delta t)$ given by (3.11) is an oscillating exponential with period $N\Delta t$, since

$$f[(Ntn)\Delta t] \approx \frac{\Delta\omega\, e^{\sigma(N+n)\Delta t}}{\pi} \sum_{K=0}^{N-1} Z_K e^{j2\pi \frac{(n+N)K}{N}} \tag{3.12}$$

but,

$$e^{j2\pi \frac{(n+N)K}{N}} = e^{j2\pi K} e^{j2\pi \frac{nK}{N}}$$

$$= e^{j2\pi \frac{nK}{N}}$$

$$f[(N+n)\Delta t] = e^{\sigma N\Delta t} f(n\Delta t) \tag{3 13}$$

Since (3.13) cannot, in general, be true a check of the validity of Equation (3.11) must be made.

If the substitutions $n\Delta t = t$ and $K\Delta\omega = \omega$ are made in the inverse transforn relation of Equation (3 3) and the upper limit of integration is reduced to $\omega_o = N\Delta\omega$, then

$$f(n\Delta t) = \frac{e^{\sigma n\Delta t}}{\pi} \int_o^N F(\sigma + jk\Delta\omega)\, e^{jK\Delta\omega t} \Delta\omega\, dK \tag{3.14}$$

Inserting the condition $\Delta t\, \Delta\omega = \frac{2\pi}{N}$ yields

$$f(n\Delta t) = \frac{\Delta\omega\, e^{\sigma n\Delta t}}{\pi} \int_o^\infty F(\sigma + jK\Delta\omega)\, e^{\frac{j2\pi nK}{N}} dK \tag{3 15}$$

The trapezoidal approximation of Equation (3 15) is

$$f(n\Delta t) \approx \frac{\Delta\omega\, e^{\sigma n\Delta t}}{\pi} \left\{ \frac{1}{2} \operatorname{Re}\{F(\sigma)\} + \sum_{K=1}^{N-1} F(\sigma + jK\Delta\omega)\, e^{j2\pi \frac{nK}{N}} + \frac{1}{2} \operatorname{Re}\{F(\sigma + jN\Delta\omega)\} \right\} \tag{3.16}$$

Since it is assumed that $F(\sigma + jN\Delta\omega)$ is small, this term can be eliminated from Equation (3.16), thus making it identical with Equation (3 11), as expected. Additionally, the validity of the approximation of Equation (3 15) by Equation (3.16) should be checked. It is a well known rule in quadrative evaluation of integrals that the summation rapidly becomes inaccurate when the sampling interval is made larger than one half the wavelength of the highest frequency component of the integrand. If it is assumed $F(\sigma + j\omega)$ has a frequency spectrum with all its components having a wavelength much greater than $\Delta\omega$, then the highest frequency in the integrand of Equation (3.15) will be $\frac{n}{N}$. The sampling interval in Equation (3.11) = 1, thus the wavelength must be $\frac{N}{n} \geq 2$ and forcing the range of $0 \leq n \leq \frac{N}{2}$ on Equation (3 11). This is the same result used by Dubner and Abate.[22] It will be shown later that it is necessary to further restrict n so that $0 \leq n < \frac{N}{4}$.

It is necessary to find a value of the parameter and for which (3.11) will be valid. The method employed in NASAP-70 involved repeated application of the Routh Stability criterion[21] followed by a translation along the real axis.

The Routh criterion involved the construction of a matrix for the transfer function,

$$F(S) = \frac{\sum_{i=0}^{K} a_i S^i}{\sum_{i=0}^{\ell} b_i S^i}$$

The corresponding matrix is:

$$\begin{array}{llll}
b_{\ell} & b_{\ell-2} & b_{\ell-4} & \cdots \\
b_{\ell-1} & b_{\ell-3} & b_{\ell-5} & \cdots \\
d_{31} & d_{32} & d_{33} & \cdots \\
d_{41} & d_{42} & d_{43} & \cdots \\
\vdots & \vdots & \vdots &
\end{array}$$

The first two rows are made up of the coefficients of the denominator polynomial. If ℓ is odd, the final element in the first row is b_1, the second row, b_0. If ℓ is even, the final element is b_0 in the first row, and zero in the second. In either case, the rows have length $\left[\frac{\ell}{2}\right] + 1$. The d_{ij} are evaluted by

$$d_{ij} = \frac{d_{i-1,1}\, d_{i-2,j+1} - d_{i-2,1}\, d_{i-1,j+1}}{d_{i-1,1}} \tag{3.17}$$

The meaning of d_{ij} for $i = 1$ or 2 is obvious. The length of nonzero elements in each row decreases as the row number increases, until in the $(\ell + 1)$ row, only $d_{\ell+1}$ is nonzero.

After the matrix is constructed, the first element in each row is examined. If all of them are positive, no singularities are to the right of the imaginary axis. If one or more is zero or negative, then there are one or more singularities with a positive real value. To test a particular σ_o to determine whether it is to the right of all the poles, F(S) is shifted by σ_o,

$$F(S+\sigma_o) = \frac{\sum_{i=0}^{K} A_i (S+\sigma_o)^i}{\sum_{i=0}^{\ell} b_i (S+\sigma_o)^i} \;, \tag{3.18}$$

where

$$\sum_{i=0}^{\ell} b_i (S+\sigma_o)^i = \sum_{i=0}^{\ell} C_i S^i \tag{3.19}$$

and

$$C_i = \sum_{j=1}^{\ell} b_i \, \sigma_o^{j-1} \binom{j}{i}$$

The Routh criterion can now be applied to C_i, and if it is stable, then σ_o can be used in Equation (3.11).

Although the theory is valid at this point, a number of practical problems remain. One of these appears when F(S) contains a high order polynomial which does not go to zero until ω is very large. In these cases, it is possible for some of the calculations involved in obtaining sample values to fall outside the range of numbers representable in a particular machine. To avoid this difficulty, the frequency scaling property of the Laplace transform is used,

$$f(a\,t) = \frac{1}{a} \; \mathcal{L}^{-1}\left\{F\left(\frac{S}{a}\right)\right\} \tag{3.20}$$

There also exists the inconvenience for users, resulting from the fact that the choice of $\Delta\omega$ files Δt. It would be much more convenient if Δt could be set to any desired value. To overcome this difficulty, a parameter selection procedure is used. Always setting $\Delta t = 1$, gives $\Delta\omega = \frac{2\pi}{N}$. If a user desires a time step different from one second, the transform is frequency scaled according to Equation (3.20) by the time interval desired, i.e., $a = \Delta t$.

With these changes, Equation (3.15) becomes

$$f(na) = \frac{2e^{n\sigma a}}{Na} \sum_{K=0}^{N-1} Z_K^1 e^{j 2\pi \frac{nK}{N}} \tag{3.21}$$

$$\text{where } Z_K^1 = \left\{ \begin{array}{l} \frac{1}{2} \operatorname{Re}\{F^1(\sigma)\}, \; K = 0 \\ F^1(\sigma + j\, 2\pi \frac{K}{N} 0, \; K > 0 \end{array} \right\}$$

$$F^1(S) = (S/a)$$

This change introduces ω complication, in that the setting of $\Delta w = \frac{2\pi}{N}$ means F(S) is sampled until $w = \frac{2\pi}{a}$, regardless if it may be non-zero at this or higher values of w.

(Note that the choice of N no longer has any effect on the upper limit of the sampling. Increasing N merely causes more samples to be taken with the sampling interval, which is set once a is chosen.

The effect of this high frequency cut-off will vary, depending upon the particular transform being inverted. The cases where the error is the greatest, however, occur near those points at which f(t) is discontinuous. In rational transforms, f(t) is discontinuous only at the origin, if at all. (If the numerator polynomial is of one degree less than the denominator, there is a discontinuity at f(0). If the difference in degree is greater than one, f(t) is continuous for all finite t. The usual effect of the high frequency cutoff error is the "rounding of the corners" at f(0).

A correction procedure can be applied when this error occurs. By carrying out a long division of F(S), one obtains

$$F(S) = \sum_{i=1}^{q} \frac{C_i}{S^i} + R(S) \tag{3.22}$$

Inversion of Equation (3 22) yields,

$$f(t) = \sum_{i=1}^{q} C_i \frac{t^{i-1}}{(i-1)!} + \mathcal{L}^{-1}\{R(S)\} \tag{3.23}$$

Note Equation (3 23) is a power series expansion for F(t) taken at $t = 0$. In NASAP-70, c_i is evaluated with $q = 50$ in Equation (3 22), following by Equation (3 23) to evaluate the first five time points. During the evaluation of Equation (3.23) a check is made to determine that a smooth series fit is achieved (if the higher order terms in the summation are insignificantly small, then $\mathcal{L}^{-1}\{R(S)\}$ is not significant because it is $O(t^{q+1})$ near the origin). Also checked is the possibility that some of the terms of Equation (3 23) are $\gg f(t)$, if this is the case, there will be a loss of significance, and the error correction procedure is by-passed. If it turns out that the series is well behaved, it is used in cases where f(t) is continuous at the origin.

The theoretical foundations of NASAP just described cannot be considered extensive. But hopefully enough of the concepts have been presented to explain the underlying theme of NASAP. For those persons interested in more of an elaboration of the principle involved, they are encouraged to read the references shown at the end of the chapter.

REFERENCES

1. Kuh, E., Deoser, C., Basic Circuit Theory, preliminary edition, McGraw-Hill, 1966.

2. Busacker, R., Saaty, T., Finite Graphs and Networks An Introduction with Applications, McGraw-Hill, 1965.

3. Branin, Jr., F., "Machine Analysis of Networks and its Applications," IBM Data Systems Technical Report TR00855, March 30, 1962.

4. Kron, G., "A Set of Principles to Interconnect the Solutions of Physical Systems," Journal Applied Physics, Vol. 24, pp. 965-980, 1953.

5. Kirchhoff, G., "Uber die Auflosung der Gleichungen, auf welche man bei der Untersuchungen der Linearen Verteilung Galvanishcer Strome gefuhrt wird," Poggenforf Ann. Physik, Vol. 72, pp. 497-508, 1847, In English, Transcations Institute of Radio Engineers, Vol. CT-5, pp. 4-7, March 1958.

6. Fröberg, C. E., Introduction to Numerical Analysis, Addison-Wesley, 1965.

7. Shannon, C. E., "The Theory and Design of Linear Differential Equation Machines," National Defense Research Committee, OSRD Report 411, January 1942.

8. Mason, S. J., "Feedback Theory Further Properties of Signal Flow Graphs," Proceedings of IRE, Vol. 44, No. 7, pp. 920-926, July 1956.

9. Happ, W. W., "Flowgraph Techniques for Closed Systems," IEEE Transactions on Aerospace and Electronic System, Vol. AES-2, No. 3, pp. 252-264, May 1966.

10. Lorens, C. S., Flowgraphs, McGraw-Hill, 1964.

11. Robechaud, L. P., et al., Signal Flow Graphs and Applications, Prentice-Hall, 1962.

12. Russell, E. C., Okrent, H., McNamee, L. P., "Instrumentation of a NASAP Subroutine," IEEE Transactions on Education, Vol. E-12, pp. 243-250, December 1969.

13. Carpenter, R. M., "Computer-Oriented Sensitivity and Tolerance Techniques," Automated Circuit Analysis Course Notes, UCLA, April 3-7, 1967.

14. Vattuone, E. S., Dorf, R., "Root Sensitivity as a Design Criterion," First Asilomar Conference, pp. 287-290, November 1967.

15. Adams, A., "A Stopping Criterion for Polynomial Root Finding," Communications of the ACM, pp. 655-658, 1967.

16. Ralston, A., Wilf, H.S., Mathematical Methods for Digital Computers, Vol. 1, Wiley, 1962.

17. Muller, D.E., "A Method for Solving Algebraic Equations Using an Automatic Computer," Mathematics of Computation, formerly Mathematical Tables and Other Aids to Computation, pp. 208-215, 1956.

18. Nattemeier, A., "Roots of Polynomials by a Root-Squaring and Resultant Routine," Communications of the ACM, pp. 779-782, 1968.

19. Henrici, P., Watkins, B.O., "Finding Zeros of a Polynomial by the Q-D Algorithms," Communications of the ACM, pp. 570-574, 1965.

20. Nilsson, J.W., Introduction to Circuits, Instruments and Electronics, Harcourt, Brace, and World, 1968.

21. DiStefano III, J., Stubberud, A., Williams, I., Theory and Problems of Feedback and Control Systems, Schaum, 1967.

22. Dubner, H., Abote, J., "Numerical Inversion of LaPlace Transforms and the Finite Cosine Transform," JACM, Vol. 15, No. 1, pp. 115-123, January 1968.

23. Cooley, J.W., Tukey, J.W., "An Algorithm for the Machine Calculation of Complex Fourier Series," Mathematics of Computation, pp. 297-301, April 1965.

24. Brigham, E.O. Morrow, R.E., "The Fast Fourier Transform," Spectrum, Vol. 4, No. 12, pp. 63-70, December 1967.

PART II A PROGRAMMER'S MANUAL

The Programmer's Guide consists of Chapters 4 and 5 and Appendix A. Chapter 4 describes the algorithms used in NASAP-70 and the general flow of control among the NASAP-70 routines. A user intending to make any modifications to his copy of NASAP should become familiar with that chapter. The Dictionaries in Appendix A are provided to aid in easy identification of the more important names in NASAP-70.

Chapter 5 is designed as a reference for the general user. In particular, the user's attention is directed to Sections 5.4 and 5.5 which should be read before any NASAP-70 problems are run on a computer which is not an IBM 360. These sections describe some important modifications which may be required to make NASAP-70 operate properly on other computers.

Finally, Appendix A contains a program listing of NASAP-70 and Appendix C contains the output generated by sample problems.

CHAPTER 4

NASAP-70 PROGRAMMER'S GUIDE

4.0 Program Organization

4.1 General Description

The NASAP Circuit Analysis program is written entirely in Fortran IV-H.[1] It is built around a basic program which produces a circuit tree from the user-supplied circuit data, and then uses this tree to evaluate the circuit Transfer Function. Additional modules are supplied which accept a user-defined tree (in a different data format), perform Sensitivity and Worst Case Analysis, produce Transient and Frequency Responses, find Poles and Zeroes of the Transfer Function, and perform automatic scaling on the input data.

Since NASAP-70 is written in Fortran IV, a user familiar with this language may add his own routines.

The flow of control in NASAP is shown in Figure 4.1. The Main Program first calls one of the three forms of Circuit Description Analysis. Each of these makes use of the Card Scanning Utilities. Then the Main Program can call (optionally) a Sensitivity Analysis routine if requested by the user. The Automatic Scaler may also be called at this point. The Transfer Function section is called to build the Flowgraph and compute the Transfer Function. The remaining sections, Sensitivities, Plotter, Rootfinder and the Transfer Function section again, may be called optionally. If more than one circuit description is supplied, any one of the Circuit Description Analysis routines may be called to restart the whole process.

Only one Circuit Description Analysis routine and the Transfer Function Section are absolutely required to be called during a NASAP-70 run. The remaining sections are called depending on the particular options selected by the user in his circuit data cards.

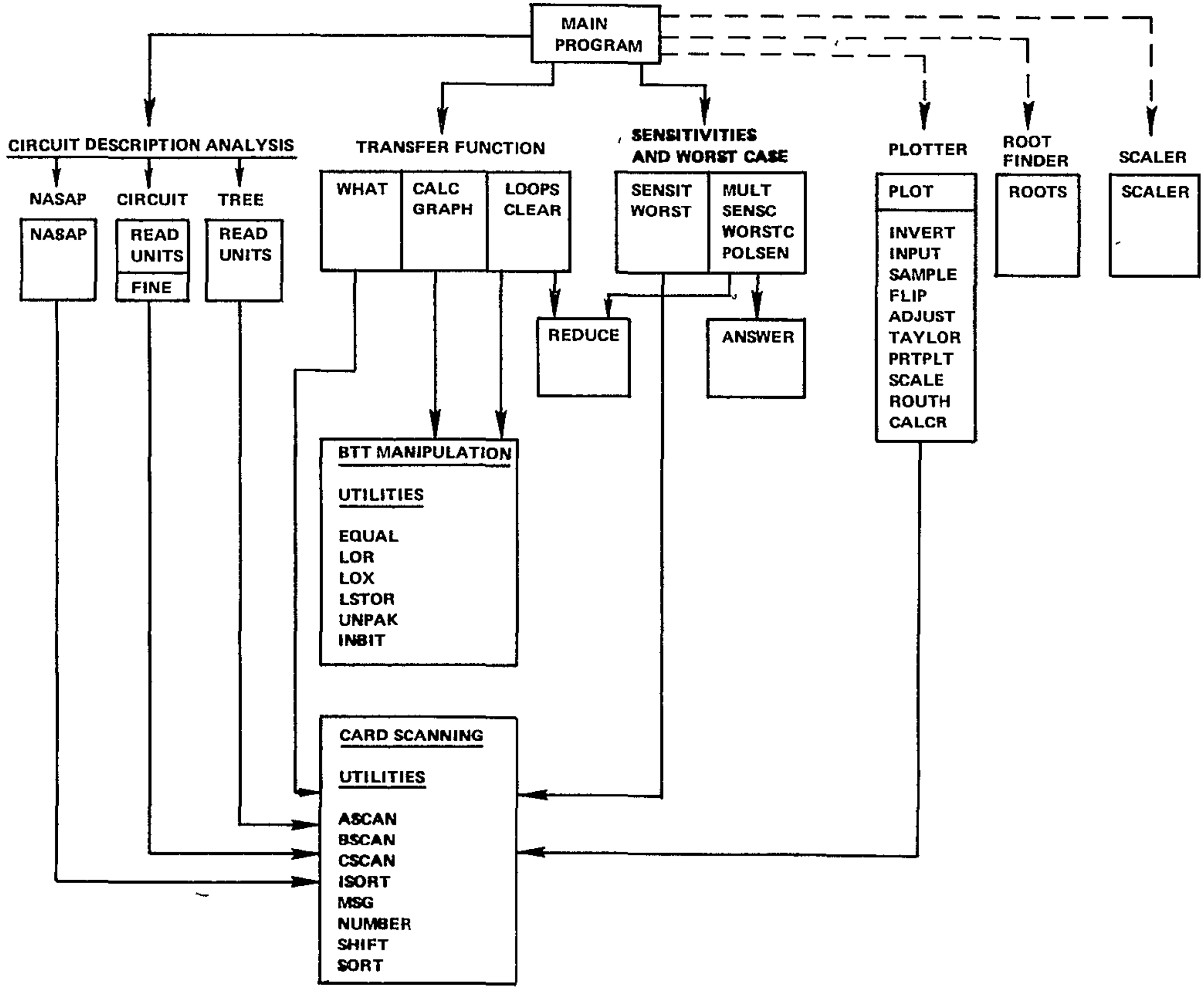


Figure 4 1 Flow of Control in NASAP–70

4.2 Algorithms and Dictionaries

In this section, the general algorithm of each subprogram is described. Instead of a statement by statement analysis of each routine, the algorithm is divided into a number of logical steps for the purpose of description. It is best to refer to a listing of the routine being studied while reading the algorithm description. (See Appendix A) The listing should be used to identify the source statements with the logical steps in the description.

For quick identification of the more important names in NASAP, a dictionary is supplied in Appendix A.

4.2.1 Main Program

The Main Program and Block Data subprogram serve to declare the more important program variables and to provide the flow of control among the modules of NASAP-70 based on the reading of certain NASAP-70 control cards from the Input Data Set.

1. In the Block Data subprogram, three Common Blocks are initialized. Common Block GAG contains an array of 20 variables which is initialized to the Sensitivity tags. These tag values are used in the Sensitivity analysis to identify flowgraph loops and the resulting Sensitivity functions with their corresponding circuit elements. Common block DATA contains an array of 32 variables which is initialized to 32 alphameric characters. Common block X contains an array of ten variables which is initialized to ten numeric characters. Common blocks DATA and X are used by the card scanning utilities (ASCAN, BSCAN, etc.) and various other routines which examine the NASAP data cards.

2. Upon entry to the Main program the important program control variables are reset beginning with NE (Number of Circuit Elements) and ending with the array SENS (The 50 Sensitivity tags). The page heading is then printed.

3. A card is read and printed, Subroutine SHIFT is called to squeeze out embedded blanks and commas, NE (which currently serves as a heading card counter) is incremented, and the first four characters on the card are checked as follows.

- If they are NASA, NE is reset and Subroutine NASAP is called, followed by a transfer to step 4 below.
- If they are TREE, NE is reset and Subroutine READ is called, followed by a transfer to step 4 below.
- If they are CIRC, NE is reset, ITREE is set (to signal that a tree is to be built by Subroutine FINE) and Subroutine READ is called. Then, provided ERR, the terminal error flag, has not been set, Subroutine FINE is called, followed by a transfer to step 4 below.
- If they are STOP, control is returned to the system monitor.
- If none of these sets of characters appear, the card is assumed to be a title card. If NE is more than ten, a warning is printed, NE is reset and control returns to step 2 above to read another card.

4. Statement 3 checks the ERR flag. If it is set, an error message is printed and control returns to Step 2 above where NASAP-70 is reinitiatized. If it is reset, the checklist of elements is printed out so that the user may verify that his data were read correctly. Subroutine CALC is called to check tree legality and to create the current equations. If Subroutine CALC sets the ERR flag (e.g., due to an invalid tree), the error message is printed as before and NASAP-70 is reinitialized. Otherwise, Subroutine SCALER is called to determine if the circuit element values require scaling. If FACTOR is nonzero, then scaling was performed and the scale factor is printed out.

5. If TAG is set, then the user omitted an END card after the circuit description and therefore another card need not be read. Otherwise, a new card is read, Subroutine SHIFT is called and the following tests are made on the first four characters of this card

If they are SENS, Subroutine SENSIT is called (to analyze Sensitivity cards) and the next statement is skipped so that Subroutine WORST is not called.

If they are WORS, Subroutine WORST is called (to analyze Worst Case tolerance cards).

Subroutine GRAPH is then called to build the flowgraph (except for the unknown transmittance) which is then printed out. Subroutine REDUCE is called for each flowgraph transmittance to handle Sensitivity tags. As a final step, preparatory to servicing Output requests, NPATH (the number of flowgraph transmittances) is incremented to include the unknown transmittance, VPATH(NPATH) (its value) is set to 1.0, S(NPATH) (its tag) is set to 1000, NTIMES (the number of words per block in the BITS array) is set based on 30 bits per word, INP (the Transfer Function request flag) is reset, finally ERR is reset. By transferring to statement 10, step 6, which reads a new Output Request card, is avoided. Control goes to step 7.

6. The INP and ERR flags are reset, a new Output Request is read and Subroutine SHIFT is called.

7. Statement 10 prints the latest Output Request, calls and then analyzes the Output Request as follows

If the first three letters are END or the first two letters are EX, control goes to step 2 where NASAP is reinitialized.

If the first three letters are CIRC, TREE or NASA, it is assumed that the user omitted the END card after his Output Requests. Therefore control goes to statement 12 which prints an error message, the NASAP-70 heading, and then returns control to step 2 at a point just after a new card is read (because this card has already been read by accident).

If INP is 0, no Transfer Function request has yet been read so the next five tests are not performed.

If the first letter is V or I, this is a Transfer Function request, therefore control goes to step 8 where the Transfer Function is formed.

If the first four letters are PLOT and ERR has not been set, Subroutine PLOT is called, and if ERR is still not set, Subroutine INVERT is called to perform the plotting. Finally ERR is reset.

If the first four letters are ROOT and ERR has not been set, Subroutine ROOTS is called. Subroutine SHIFT is then called so that the succeeding test for a Pole Sensitivity Request will operate on a card of the form ROOTS, POLES.

If the first four letters are POLE and ERR has not been set, Subroutine POLSEN is called.

Control now returns to step 6 after an error message is printed for the cards beginning with SENS, WORS, or TOL.

At this point INP = 0, which means that there has been no previous Transfer Function request. Hence, cards beginning with PLOT, ROOT, SENS, TOL or WORS are meaningless and if the current card is one of these, an error message is printed and control returns to step 6.

8. The card is now considered to be a Transfer Function request, consequently INP is set to signal that a Transfer Function request has been encountered. The number stored in NTIMES is printed and then Subroutine WHAT is called to analyze the Transfer Function request. If the ERR flag has been set, control goes to step 6 to search for another Output Request. Otherwise Subroutine INBIT is called to enter the unknown transmittance into the BITS array and then Subroutine LOOPS is called to solve for the Transfer (and Sensitivity) Functions.

9. The BITS, SMAX1 and SMIN1 arrays are reset before the Sensitivity Functions are entered. Subroutine ANSWER is called to print out the Transfer Function. If NSEN is 0, this means that no Sensitivity Functions are requested. Control then returns to step 6 (except INP and ERR are

not reset), otherwise subroutine SENSC is called. If NWORST is not 0, Subroutine WORSTC is called to evaluate the Worst Case Function, and Subroutine ANSWER is called to print it. Finally, control returns to step 6 (except INP and ERR are not reset).

4.2.2 Circuit Description Analysis

NASAP-70 offers three techniques of circuit description analysis. The technique selected is determined by the heading card read by the Main program. All the techniques make use of the card scanning utilities which perform various operations on the card image stored in an 80 word array in Common block A.

The User's Circuit Description may be supplied to NASAP-70 in one of three ways

1. As a circuit description with a tree[2] to be defined by the program for minimum computation time. This is the standard form of NASAP Input Data used in previous versions of the program.[3] It is initiated by a card having the work NASAP.

2. As a circuit description with a tree to be defined by the program for optimum accuracy at a given frequency. This form of input data is new to NASAP. It is initiated by a card having the word CIRCUIT followed by a value in parentheses which represents a frequency in cycles per second.

3. As a circuit description with a user-defined tree. This option, like the CIRCUIT option, uses a new form of input data. It is initiated by a card having the word TREE.

The method of analyzing the Circuit Description Data associated with each of these forms of input is described below.

Card Scanning Utilities

There are eight routines which constitute the card scanning utilities. Subroutines ASCAN (B, I, J), BSCAN(B, I, J) and CSCAN(B, I, J) scan the card image in Common block A starting at column I until a special character (described below) is encountered. The character is placed in B and the number of the column just before this character is placed in J.

- Subroutine ASCAN scans for (/ -) = blank.
- Subroutine BSCAN scans for U blank M K P O V I.
- Subroutine CSCAN scans for . E + - blank.
- Function ISORT(A) returns the fixed point numeric value of the character stored in A. If A is not numeric, FLAG is set.
- Subroutine MSG(A, I) prints a diagnostic.
- Subroutine NUMBER(N1, N2) stores the floating point value into VPATH(NE) of the number between columns N1 and N2 inclusive, irrespective of the format in which this number is written. If there is an error in the number (e.g., an invalid character), FLAG is set and the value entered into VPATH(NE) is unreliable. For instance, the following numbers are acceptable:

  -1 .73-7 1E2 +0.0009

  but these are not acceptable

  -1A 7..3 1B2 ++.0009

The algorithm of Subroutine NUMBER is now described in more detail·

1. Upon entry, K (the column pointer) is set to N1, and VPATH(NE) is set to 0. Provided K is less than N2, control passes to step 3 to begin the analysis.

2. This is the error return. When this step gains control, a diagnostic is printed and control is returned to the calling routine.

3. A test is performed on the first character in the field. If it is a -, TAG is set. Subroutine CSCAN is called and if I already points to a special character this number must be less than 1, so control passes to step 5.

4. The digits which form the part of the number greater than 1 are contained between columns I and J inclusive. This step (DO 3) enters them

into VPATH(NE). If FLAG is set for any digit, the error return (step 2) is given control.

5. If the end of the number field has been reached, control passes to step 7. If the next character is not the decimal point, control passes to step 6 which checks for an exponent. Otherwise the digits of the fractional part of the number are entered into VPATH(NE) (DO 7). If FLAG is set, step 2 gets control.

6. If the next character is the E of the exponent field, the column pointer is incremented. If a - is found in the exponent field, EXPO is set. Subroutine BSCAN is called to obtain the limits of the exponent field. A diagnostic is printed if this field contains more than two digits. The rightmost two digits are used to enter the exponent value into POW If EXPO has been set, the exponent is negative. Lastly, the value in VPATH(NE) is modified according to the value of the exponent.

7. If TAG has been set, the number was negative so the sign is reversed. Control then returns to the calling program.

- Subroutine SHIFT left adjusts all characters in Common Block A removing blanks and commas.
- Function SORT is the same as ISORT but returns a floating point value.

Subroutine NASAP

Subroutine NASAP decodes Circuit Description cards which follow a NASAP heading card. If no error is detected before the END or OUTPUT card is reached, a tree is constructed (subject to the constraint of minimum First Order Loops). Each Circuit Description Card is in the following format

NAME ORIGIN TARGET VALUE UNITS(optional) DEPENDENCY(optional)

The Data cards are described in more detail in Sec. 4.2. The algorithm details of Subroutine NASAP follow

1. On entry, NE (the number of elements) is reset to 0.

2. A new card is read into Common block A and Subroutine ASCAN is called repeatedly until the first non-blank character is found. If this character is E or 0, this is the end of the circuit description, so control passes to step 7 where the tree is built. Otherwise NE is incremented, the element name is padded with blanks on the right and entered into the CARD array (which contains the names of all the circuit elements).

3. Subroutine ASCAN is called repeatedly to find the next non-blank character which is the start of the ORIGIN field. Function ISORT is used to enter the value of the origin node into the ORIGIN array. If FLAG is set, control transfers to step 6 which is the error return. A similar procedure is used to enter the value of the TARGET node into the TARGET array.

4. Subroutine BSCAN is called repeatedly to find the next blank character or the first character of the UNITS field, followed by a call to Subroutine NUMBER to enter the value of the VALUE field into VPATH(NE). This value is multiplied by a factor depending on the next one or two non-blank characters (which constitute the UNITS field):

U	10^{-6}	P	10^{-12}
K	10^{3}	M	10^{6}
MH	10^{-3}		

If the character is a V or I, the DEPENDENCY field is present so Subroutine ASCAN is used to extract the name of the dependency that is then entered into the DEP array (which contains the names of controlling elements). If a DEPENDENCY field is not present, a string of blank characters is entered into the DEP array.

5. GENER(NE) and TYPE(NE) are reset. If the first character of the element name is V or I, the element is active requiring that GENER(NE) be set. If it is V, the element is a voltage source and is immediately required

to be in the tree, therefore TYPE(NE) is set. Finally, before transferring back to step 2 to read the next card, NNODES and INODE (the maximum and minimum node numbers appearing so far) are set.

6. (The error return). A diagnostic is printed, ERR is set and control is returned to the calling program.

7. Before building the tree, an array N is set up which initially contains the number of elements connected to each node N(I). When N(I) is 0, node I has been connected to the tree. When N(I) is negative, node I is currently being considered for entry into the tree.

8. The next free node having the highest number of elements connected to it is selected for insertion into the tree. Its N entry is reset to 0 Then, all the nodes connected to this one via voltage sources are put under consideration, i. e., N(I) is made negative (DO 23). As long as M is 1 additional nodes have to be considered, consequently step 8 is repeated until M is 0.

9. Each passive element connected to this node is entered into the tree when the other node to which it is connected is not yet under consideration. At the same time, all the nodes connected to this new node via voltage sources are given consideration. (DO 26 and DO 31).

10. The N array is searched for the least value. If this is 0, all possible nodes have been entered into the tree, control returns to the calling program. Otherwise the node is tagged as being processed by subtracting 1000 from its N entry. Then control passes to step 8.

Subroutine NASAP does not perform any checks on the circuit described or the tree it builds (i. e., to see if all nodes are connected to the same tree). If the circuit supplied is valid, then the tree constructed from it will be valid also. If the circuit is invalid, the resulting tree is meaningless. All checking on both program and user constructed trees is performed by Subroutine CALC while solving the current equations. If any invalidity is found, Subroutine CALC prints the appropriate diagnostics.

In general, the tree constructed by Subroutine NASAP produces the minimum number of First Order Loops, and therefore produces a transfer function in minimum computation time. This tree may be described briefly as the "bushiest" tree. For instance, for the circuit shown, two trees are given (solid lines represent the tree) both of which are valid. However, tree 2 is the "bushiest" and would probably result in fewest First Order Loops.

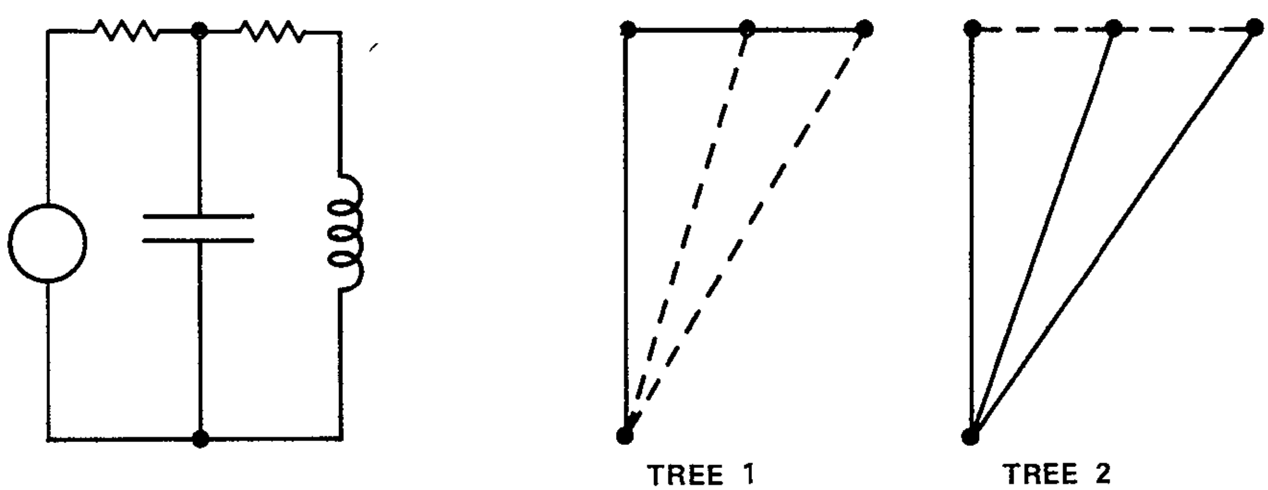


Figure 4 2

Subroutine READ and Subroutine FINE

In response to reading a heading card beginning with either the word CIRCUIT or the word TREE, Subroutine READ is called to analyze Circuit Description Data. If the heading card was a CIRCUIT card, Subroutine FINE is called to build a tree which results in optimum accuracy at the frequency shown on the CIRCUIT card. If the heading card was a TREE card, Subroutine FINE need not be called because it is assumed that the user has supplied the tree. The format of the circuit description data accepted by Subroutine READ is considerably different from that accepted by Subroutine NASAP. This is because Subroutine READ is designed for a user who has some knowledge of flowgraph theory while Subroutine NASAP is designed for the user interested primarily in the Transfer Function rather than the method of analysis.

CIRCUIT cards have the form:

CIRCUIT (Frequency)

CIRCUIT and TREE circuit description cards have the form

NAME/DEPENDENCY (ORIGIN-TARGET) = VALUE UNITS

(note that DEPENDENCY and UNITS are optional quantities)

Circuit Description Data is described in detail in Sec. 4.2.

1. The first step in Subroutine READ is to reset a number of control variables. Then if this is a CIRCUIT card (TREE=1), the default frequency is set to 1.0, and Subroutines ASCAN and NUMBER are called to scan the CIRCUIT card for the frequency enclosed in the parentheses, if the frequency cannot be found, the appropriate diagnostic is printed.

2. The next card is read and Subroutine SHIFT is called to eliminate blanks and commas. If the card is completely blank or if it starts with the letters PL, the appropriate diagnostic is printed and step 2 is performed again. If the card starts with END, control returns to the calling program. If there are more than 50 elements, the appropriate diagnostic is printed. The TYPE and GENER entries for this element are reset to 0 and the CARD and DEP entries are filled with blanks.

3. The following tests are performed on the first 2 characters in the card buffer (Common block A)

a. If the first character is E or J, an active source is assumed. Its GENER entry is set to 1. If the first character is E, its TYPE entry is set to 1 also to indicate that it is part of the tree.

b. If the first character is R, L, C or D, and if the second letter is E, the TYPE entry is set to 1. If the first letter is D, the GENER entry is set to 1 (to show this is a dependent source). Unless TREE is 1 (i.e., unless no user tree is supplied) and if the second letter is not E or J (branch or link) a diagnostic is printed and the second letter is set to J.

c. If the first letter is V or I, it is assumed that because the user omitted the END card from the circuit description the first Output Request was read by mistake. Thus, TAG is set to 1 to show that the first Output Request is already in the buffer. A diagnostic is printed and control returns to the calling program.

d. If the first letter is P, it is assumed that a PLOT card has mistakenly been placed in the circuit description. Therefore, a diagnostic is printed and control passes to step 2 to read a new card.

If none of these tests is true, the first letter is set to R and processing proceeds as if test 2 were true.

4. Subroutine ASCAN is called to find the element name. If it is more than 12 characters long, a diagnostic is printed. In the case of an element whose first letter is not D, if the first character after the name is not a left parenthesis, a diagnostic is printed and step 2 gains control to read a new card. In the case of an element having D as the first letter, a diagnostic is printed if a slash is not found after the element name. The element name is then entered into the CARD array. A check is performed of this new name against all the previous names in the CARD list (DO 17). If a match is found, a diagnostic is printed and the program attempts to replace the latest name with another one based on the value of the counter NR. Up to 10 duplications of a name are allowed. If more than 10 have occurred, the ERR flag is set.

5. If this element is not a dependent element (i.e., the first character is not D), step 5 is not performed. Otherwise Subroutine ASCAN is called to extract the Dependency Name. If the character following the Dependency Name is not a left parenthesis, a diagnostic is printed and control passes to step 2 where a new card is read. If the first letter of the Dependency Name is not V or I, a default is assigned and a diagnostic is printed. The Dependency Name is then entered into the DEP array.

6. Subroutine ASCAN is called once to extract the Origin number which is then entered into the ORIGIN array and then again to obtain the Target number which is entered into the TARGET array. In either case, if an

error is detected (FLAG = 1) or if there are more than two digits in the Origin or Target fields, a diagnostic is printed and step 2 gains control to read a new card. If the Origin or Target number is 0, a diagnostic is printed and the number is changed to 99. If the appropriate characters are not found after the Origin or Target nodes, diagnostics are printed.

7. Having reached this point an element is now entered permanently into the list of elements by incrementing NE, the number of elements. NNODES and INODE, the maximum and minimum node numbers, respectively, are set. Subroutine BSCAN and NUMBER are then called to obtain the element value. The characters following the Value field are checked to determine if a Units field is present. If it is not the one (that is, if the next character is blank), then the value in VPATH is not modified. Otherwise the value is multiplied according to the following units

PF:	10^{-12}	K:	10^{3}	MF	10^{-6}
M	10^{6}	MH:	10^{-3}	MMF:	10^{-12}
MMH.	10^{-6}				

Subroutine UNITS is called to write out the units detected. Then control is transferred to step 2 to read another card.

In general, Subroutine READ is fairly tolerant of mistakes in circuit description data. Diagnostics are printed whenever reasonable and defaults are supplied whenever possible.

After Subroutine READ returns to the Main program, Subroutine FINE is called to build a tree, if the heading card was a CIRCUIT card (TREE = 1). Essentially, the logic of building this tree is to take the user supplied frequency (written on the CIRCUIT card) and to evaluate the impedance of inductive and capacitive elements according to,

$$X_c = \left| \frac{1}{j\omega c} \right| \quad X_L = \left| j\omega L \right| \quad .$$

All elements are then ordered by their impedance magnitudes and the tree is built from elements having the lowest impedance, since these elements have the closest approximation to the impedance of a voltage source (i. e., 0 ohms). The value 9.9×10^{30} is used to represent ∞ (i. e., the impedance of a current source).

1. The first step in Subroutine FINE is to evaluate the impedance of each element (DO 1) and to store it in the array Z. The elements are then reordered by increasing values of Z. (DO 4). The number of nodes in the circuit is found, taking into account the possibility that they may not be numbered consecutively (DO 40).

2. All the TAGs are initially reset to 0 (to exclude all elements from the tree). The first element is then entered into the tree immediately. Then the elements are added to the tree one by-one, each time checking that the added branches do not conform to closed loops (i. e., that it always remains a tree). As soon as all the nodes are connected, the ordered list of elements is printed and control returns to the calling program.

In this fashion, as many low impedance elements as possible are included in the tree according to the operating frequency defined by the user. This tree is more likely to result in a larger computation time and a greater number of First Order Loops in the Flowgraph when the Transfer Function is calculated. This is offset, however, by increased accuracy in the Transfer Function.

4.2.3 The Transfer Function[4]

Evaluation of the Transfer Function involves calculation of the current equations for the circuit described employing the tree which has been supplied either by the user or derived by the program. From the current equations calculated by Subroutine CALC, Subroutine GRAPH builds the Flowgraph except for the closing transmittance. Each time a Transfer Function Request appears, Subroutine WHAT is called to obtain the closing transmittance and Subroutine LOOPS is called to evaluate the Transfer Function from the loops in the flowgraph according to the Shannon-Happ formula. Finally, Subroutine ANSWER is called to normalize the Transfer Function and to print it.

The process of evaluating the Transfer Function involves the manipulation of individual bits in a word. The bit manipulation words are stored in an array of 3000 words in Common block BITS. In addition, these words are handled in blocks of varying size depending on the current value of NTIMES. For instance, if NTIMES is three, the blocks consist of three words each. Furthermore, the various routines involved in calculating the Transfer Function address the array in BITS differently. In order to facilitate the addressing and manipulation of these words, a set of six Bit Manipulation Utilities are provided.

1. Subroutine EQUAL (I, J, N, NTIMES) performs operation N (see LOR, LOX, LSTOR) on the blocks of bits identified by I and J and puts the result into block I. The size of the blocks is given by NTIMES (the number of words per block) while different addressing techniques are used depending on the signs of I, J and NTIMES

a. If NTIMES is positive and I, J are positive, then they refer to the I^{th} and J^{th} blocks in the BITS array.

b. If NTIMES is positive and I, J are negative, then they refer to the $I + 100^{th}$ and $J + 100^{th}$ blocks.

c. If NTIMES is negative and I, J are positive, then they refer to the I^{th} and J^{th} blocks from the high end of the BITS array.

d. If NTIMES is negative and I, J are negative, then they refer to the I^{th} and J^{th} blocks in the BITS array as in case a.

2. Function LOR(I, J) performs a full-word logical OR on I and J and puts the result in I.

3. Function LOX (I, J) performs a full word logical AND on I and J and subtracts the result from both I and J.

4. Function LSTOR (I, J) sets I equal to J.

5. Subroutine UNPAK(I, J, NN) takes the block of bits identified by I (according to the same scheme as Subroutine EQUAL) and searches them starting at bit 1 and proceeding as far as bit NN. When the first 1 bit is found, Function LOX is used to reset it to 0 and the position of this bit is returned in J. If no 1 bits are found, J gets the value 0.

6. Subroutine INBIT(I, J, NTIMES) takes the block of bits identified by I (according to the same scheme as Subroutine EQUAL) and sets the bit in position J to 1. If J is 0, all the bits in block I are reset to 0.

All the Bit Manipulation Utilities use only the low order 30 bits of a word. Thus on the IBM 360, the settings of the two high order bits are ignored throughout the bit manipulations. Subroutine EQUAL uses Functions LOR, LOX and LSTOR to perform certain operations. For further information on Functions LOR and LOX see Sec. 4.4.

Subroutine CALC is called by the Main program after the Circuit Description Data has been successfully analyzed. CALC calculates the current equations using Kirchhoff's equations. At the same time the validity of the tree is checked. If the tree should prove to be invalid, a diagnostic is printed, the ERR flag is set, and control returns to the Main program.

1. Upon entry to Subroutine CALC a page is skipped and NTIMES is set according to the number of elements in the circuit (NE). NTIMES is increased by 1 for every 30 elements, i.e., each bit represents 1 element. The arrays NEL (Number of elements at node I), NBR (number of branches at node I) and the LINKS array (the Bit manipulation array in Common block BITS) are reset to 0. Then the NEL array is filled with the number of elements at each node, the NBR array is filled with the number of (tree) branches at each node, NB gets the number of branches in the tree, and by means of Subroutine INBIT, the LINKS array is set up as follows

(DO 1) Each block in the LINKS array represents a node. Each bit in the block represents an element not in the tree (i. e., a link). Blocks in the lower half of the array represent links leaving the node corresponding to that block. Blocks in the upper half of the LINKS array represent links arriving at the node corresponding to that block. Thus, each node has two entries in the LINKS array, one entry in the lower half of the array containing links directed away from the node, and one entry in the upper half of the array containing links directed towards the node. For example, if node 2 of a circuit has elements 1 and 2 directed away, element 1 being a voltage source (i. e., part of the tree), and element 3 directed towards, then block 2 of the LINKS array has bit 2 set while block (2+100=102) has bit 3 set.

2. The next step is to set up the NS and NQ pointers for the BRANCH list while checking the validity of the tree (DO 2). The BRANCH list[5] contains the branches connected to each node, positive if the branch is directed towards, negative if the branch is directed away. NQ(I) points to the beginning of the list of branches for node I. For example, if node 1 has three branches and node 2 has two branches, then NQ(1) = 1, NQ(2) = 4, NQ(3) = 6. The number of nodes, NN, is also counted at this time. If any node has no elements connected to it, the node numbering was not sequential so a warning is printed. If any node has only one element connected to it, the ERR flag is set.

3. The number of branches, NB, is checked against NN, the number of nodes. For a legal tree NB must be exactly equal to NN-1. If this is not true, an appropriate diagnostic is printed and the ERR flag is set. At this point, a test of the ERR flag is performed, if it has been set, control returns to the calling program. Otherwise, the BRANCH list is filled (DO 7) as described in step 2, using NS as a set of temporary pointers.

4. This step performs the actual solution of the current equations which are now contained in the BRANCH and LINKS arrays. Each node

in the circuit has associated with it a current equation. The object is to express the current in each branch in terms of the current in the circuit links. Since there is one less branch than the number of nodes, there is an independent equation for each node except one. Thus, the equations of nodes with more than one branch are gradually reduced until each equation has only one branch. The solution is then complete.

The first step is to find a node with one branch (i. e., the first cut set) and the first node with more than one branch.

If none can be found, all the equations that could be reduced have already been reduced. If this is not every equation, the ERR flag is set, a diagnostic listing the nodes remaining to be reduced is printed, and control returns to the calling program. Otherwise, the remaining multi-branch nodes are checked against the single branch node just found and Subroutine EQUAL is called to either add or subtract the relevant equations (DO 26). The branch is removed from the list of branches at the multi-branch node and when all reductions have been performed, the single node associated with this branch is reset (i. e., NBR is reset to 0) to show that all possible reductions with this branch have been performed. Step 4 is repeated until every equation has been reduced (NBR(I) are all less than or equal to 1).

5. The current equations are printed by searching the BRANCH list for non-empty entries (i. e., where $NQ(I) \neq NQ(I+1)$). When one is found, the branch is printed out along with the words from the LINKS array which constitute the current equation. For instance, the equation:

$$I_3 = I_1 - I_4$$

would be printed as

3 2 16

This list is essentially for debugging purposes. After the current equations are printed, control is returned to the calling program.

The next step in Transfer Function evaluation is the construction of the Flowgraph. This is performed by Subroutine GRAPH which stores the Flowgraph is Common Block PATHS.

a. On receiving control, Subroutine GRAPH initially resets NPATH, the number of paths (or transmittances) in the flowgraph, to 0.

b. A Flowgraph transmittance is inserted for each element which is not an independent source (DO 100). If the element is passive, then the transmittance represents an impedance (if the element is a branch) or an admittance (if the element is a link). NPATH is incremented, the S entry is reset and LPATH is assigned. LPATH(I, 1) is the origin node of transmittance I, LPATH(I, 2) is its target node. Nodes with a number less than or equal to NE are the current nodes, nodes between NE+1 and 2xNE are voltages nodes. Thus if element 3 is an impedance, in a 10 element circuit its transmittance goes from node 3 to node 13 (i.e., the voltage node depends on the current node since V = IR) and there are 20 nodes in the Flowgraph. Admittances go from voltage to current nodes. A series of tests is performed on the element·

a. If it is a resistor branch, its VPATH value is unchanged, and its S value is 0 (Since it is not frequency dependent).

b. If the element is a resistor link, its admittance is $\frac{1}{\text{VPATH}}$.

c. If it is an inductor branch or capacitor link, its S value is 1 and its VPATH value is unchanged since $X_c = sL$ and $Yc = sC$.

d. If it is an inductor link or capacitor branch, its S value is -1 and its VPATH value is inverted since $Y_L = \frac{1}{sL}$ and $X_c = \frac{1}{sC}$.

3. If the element is a dependent source, a search of the list of element names in CARD is performed (DO 17) to find the name of the controlling element in DEP. If the element is not found, a diagnostic is printed and the element is treated as an independent source. Otherwise a transmittance is inserted from the voltage or current node of the controlling element (according to whether voltage or current dependence has been requested) to the voltage or current node of the dependent source (according to whether it is a voltage or current source).

4. Having completed the transmittances due to the elements, step 4 (DO 1) inserts the transmittances due to current and voltage equations. Although only the current relationships have been calculated, the voltage relationships may be easily obtained from them by a simple rule If there is a current transmittance between two current nodes there is a corresponding voltage transmittance between the two corresponding voltage nodes. The direction of the voltage transmittance is opposite to that of the current transmittance. If it connects the nodes of two passive elements or two sources, its sign is opposite to that of the current transmittance.

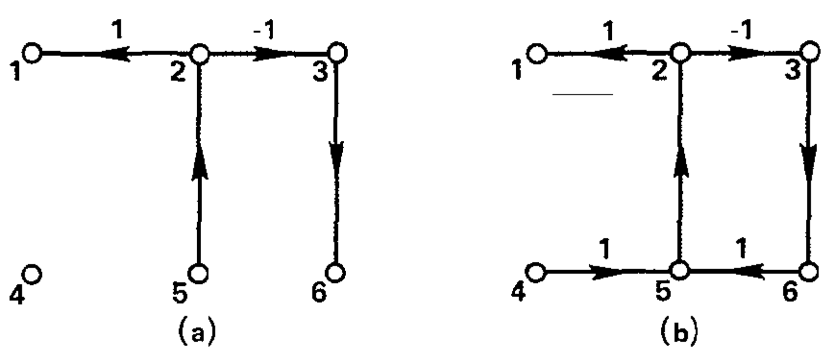


Figure 4 3

For example, given the current relationships of Flowgraph (Figure 4. 3) the voltage relationships may be easily determined as shown in (b). These Flowgraphs represent the current equations

$$I_1 = I_2 \qquad I_3 = - I_2$$

This method is implemented by searching the BRANCH list for non-empty entries. For each non-empty entry, which represents a branch, Subroutine UNPAK is used to search the corresponding LINKS (alias LBITS) entries for 1 bits. The lower half is searched first for positive links, and when UNPAK returns 0 indicating there are no more 1 bits in the word, the search is switched to the upper half of LINKS which contains negative links.

The transmittances are inserted two at a time (i. e., both current and voltage transmittances are added simultaneously). The S value of these relationships is 0 since they are obviously not frequency dependent. The VPATH values are either +1 or -1 determined as described earlier.

5. Having completed the Flowgraph (except for the unknown transmittance) control returns to the calling program.

The Flowgraph resides in Common block PATHS until a Transfer Function Request is read. Subroutine WHAT is called by the Main program to obtain the closing (unknown) transmittance by analyzing the Transfer Function Request which is in the form

TYPE NAME 1 / TYPE NAME 2

Subroutine WHAT makes use of the Card Scanning Utilities described in Sec. 4.2.2.

1. Upon receiving control, Subroutine WHAT copies the Flowgraph from its permanent storage in Common block PATHS into an array in Common block BITS (DO 18). Subroutine ASCAN is then called to find the slash which divides the Output field from the Input field. If none is found, the ERR flag is set, a diagnostic is printed and control returns to the calling program. Otherwise the CARD array is searched (DO 2) for the name in the Output field of the Transfer Function Request. If it is not found, a diagnostic is printed, the ERR flag is set, and control returns to the calling program.

2. If the first letter in the Output field (TYPE NAME 1) is not V or I, a diagnostic is printed and defaults are assumed if the element is a branch the default is I (since a branch has independent current), for a link, V is assumed. The unknown transmittance is connected from the voltage or current node (according to the V or I) of the element in the Output field.

3. In a procedure similar to steps 1 and 2, Subroutine ASCAN is called to find the blank at the end of the Input field. If the blank cannot be

found, the error return is taken. Otherwise a search is made for the element name (DO 10), defaults are used for V or I, if necessary, and the transmittance is connected to the voltage or current node of the element in TYPE NAME 2. Subroutine WHAT then returns control to the calling program.

Subroutine WHAT makes no check on the validity of the Transfer Function Request. Only syntax and the legality of the element names are checked. For example,

VR2/VV7 could be a valid request but

VR2/IV7 is not, because the voltage across V7 is an independent variable but the current is not. The Transfer Function resulting from the second example will be 0 or meaningless.

After Subroutine WHAT has returned control and the Main program has inserted the Flowgraph in bit representation in Common block BITS using Subroutine INBIT (DO 18), Subroutine LOOPS[6] gains control to perform the actual calculation of the Transfer Function from the flowgraph according to the Shannon-Happ formula:

$$1 - \Sigma L_1 + \Sigma L_2 - \Sigma L_3 + \cdots = 0$$

where the L_i represent the sums of the values of the loops of order i. A loop of order 1 (First Order Loop) is a Simple Directed Loop defined as a closed path consisting of a sequence of transmittances taken in the direction of the arrow. The sequence must be taken such that no node is traversed more than once in the closed path. The value of the directed loop is the product of the transmittances forming the directed loop. An N^{th} order loop is composed of N disjoint First Order Loops, that is, none of which have any nodes in common. The value of an N^{th} order loop is the product of the N First-Order Loop values comprising the N^{th} order loop.

In order to illustrate the operation of Subroutine LOOPS the example Flowgraph shown in Figure 4.4 is employed.

4

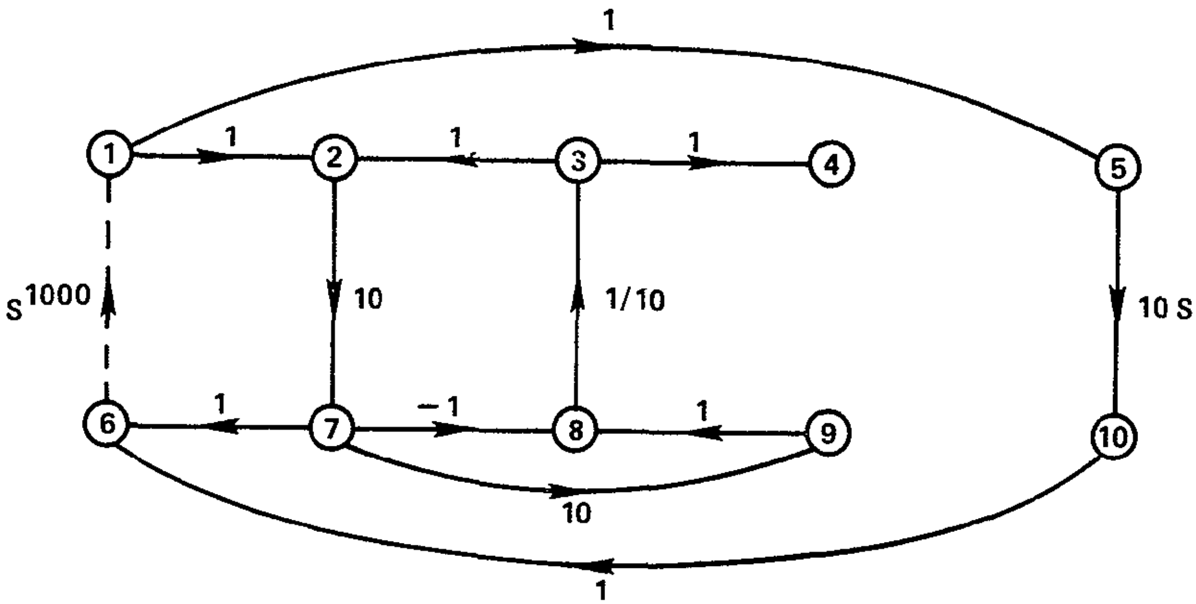


Figure 4 4

Note that the transmittances joining nodes 1 and 2, and nodes 6 and 7 have the same sign. This flowgraph is represented in the LPATH, VPATH and S arrays of Common block BITS, as shown

I	LPATH(I, 1)	LPATH(I, 2)	VPATH(I)	S(I)
1	2	7	10	0
2	8	3	0.1	0
3	7	9	10	0
4	5	10	10	1
5	1	2	1	0
6	7	6	1	0
7	3	2	1	0
8	7	8	-1	0
9	3	4	1	0
10	9	8	1	0
11	1	5	1	0
12	10	6	1	0
13	6	1	1	1000

NPATH = 13

In addition, the information stored in the LPATH array is also stored in Common block BITS (array LOOP) in a different form.

I	LOOP(I)	
3000	$2^2 + 2^7 = 132$	(Since NTIMES = 1, each
2999	$2^8 + 2^3 = 264$	transmittance and each loop
2998	$2^7 + 2^9 = 640$	is contained in only 1 word.)
2997	$2^5 + 2^{10} = 1056$	
2996	$2^1 + 2^2 = 6$	
2995	$2^7 + 2^6 = 192$	
2994	$2^3 + 2^2 = 12$	
2993	$2^7 + 2^8 = 384$	
2992	$2^3 + 2^4 = 24$	
2991	$2^9 + 2^8 = 768$	
2990	$2^1 + 2^5 = 34$	
2989	$2^{10} + 2^6 = 1088$	
2988	$2^6 + 2^1 = 66$	

Subroutine LOOPS receives these data as input and using the Shannon-Happ formula, the Transfer Function is calculated and stored in the VN array in Common block CIRCIT. VN(I, 1) is the value of the coefficient of the (I - 51th) power of S in the denominator. VN(I, 2) is the negative of the value of the coefficient of the (I - 51th) power of S in the numerator. SMAX(J) and SMIN(J) are the highest and lowest powers of S in VN(I, J). For example, a Transfer Function such as $\frac{1}{1+2S}$ would be stored as

SMAX(1) = 1 SMIN(1) = DO VN(51, 1) = 1 VN(52, 1) = 2

SMAX(2) = O SMIN(2) = 0 VN(51, 2) = -1

The LOOP array is used in two ways: the low end is used to store First Order Loops and Higher Order Loops during evaluation; the high end is used to store the Flowgraph Transmittances.[5] Thus, as data are removed from the high end and transformed, it is reinserted into the low end. Other important variables used in Subroutine LOOPS are the arrays V and LS which contain the values and powers of S of flowgraph loops. These arrays are in Common block SPEED which is used elsewhere as workspace. Subroutine CLEAR is used to remove transmittances which have been marked deleted (i. e., LPATH (I, 1) = 0) from the list of paths.

1. The first step in Subroutine LOOPS performs initialization by resetting important program variables to 0. NN, the number of flowgraph nodes is set to 2 x NE (the number of elements) since each element has 2 nodes (voltage and current) in the flowgraph. NLOOP, the number of loops is reset to 0. VN(51, 1) is set to 1; this is one 1 of the Shannon-Happ formula. The NODE(J) and COUNT(J) arrays (which contain the number of transmittances arriving at and leaving node J) are set up (DO 1).

2. The first step in the algorithm is the elimination of all nodes having only one transmittance directed away (i. e., COUNT = 1) since every transmittance entering this node must ultimately exit through the one leaving transmittance. This transmittance is attached to the end of each entering transmittance (DO 3) by changing the terminating node (LPATH(I, 2)) and calling Subroutine EQUAL to 'OR' in the bits common to both paths. The leaving transmittance is then deleted (LPATH(I, 1) = 0). At this point the data for the example flowgraph would appear as follows

I	LPATH(I, 1)	LPATH(I, 2)	VPATH(I)	S(I)	LOOP(3001-I)
1	0	7	10	0	$2^2 + 2^7$
2	0	3	0. 1	0	$2^3 + 2^8$
3	7	3	1	0	$2^3 + 2^7 + 2^8 + 2^9$
4	0	10	10	1	$2^5 + 2^{10}$
5	1	7	10	0	$2^1 + 2^2 + 2^7$
6	7	1	1	1000	$2^1 + 2^6 + 2^7$
7	3	7	10	0	$2^2 + 2^3 + 2^7$
8	7	3	-0. 1	0	$2^3 + 2^7 + 2^8$

I	LPATH(I, 1)	LPATH(I, 2)	VPATH(I)	S(I)	LOOP(3000 -I)
9	3	4	1	0	$2^3 + 2^4$
10	0	3	0.1	0	$2^3 + 2^8 + 2^9$
11	1	1	10	1001	$2^1 + 2^5 + 2^6 + 2^{10}$
12	0	6	1	0	$2^6 + 2^{10}$
13	0	1	1	1000	$2^1 + 2^6$

NPATH = 13 NLOOP = 0.

3. A check is now performed (DO 6) for First Order Loops that may have been created. These are recognized by having identical LPATH entries (i.e., LPATH(I, 1) = LPATH(I, 2), e.g., entry 11 above). Each time one is found, it is stored in the lower half of the LOOP array by means of Subroutine EQUAL and its value is stored in V and LS. NLOOP is incremented. The LPATH entry is marked deleted and the loop is printed, (as a debugging aid). The DO 11 then marks all nodes which have only leaving or only arriving transmittances as deleted since they obviously cannot be contained in any loops. (Entry 9 in the example). Subroutine CLEAR is called to remove deleted entries. Now the data looks like this:

I	LPATH(I, 1)	LPATH(I, 2)	VPATH(I)	S(I)	LOOP(3001 -I)
1	7	3	1	0	$2^3 + 2^7 + 2^8 + 2^9$
2	1	7	10	0	$2^1 + 2^2 + 2^7$
3	7	1	1	1000	$2^1 + 2^6 + 2^7$
4	3	7	10	0	$2^2 + 2^3 + 2^7$
5	7	3	-0.1	0	$2^3 + 2^7 + 2^8$

I	LOOP(I)	V (I)	LS(I)
1	$2^1+2^5+2^6+2^{10}$	10	1001

NPATH = 5 NLOOP = 1

4. If the number of paths remaining (NPATH) is 0, this step is not performed. Otherwise this step combines the remaining paths to generate more First Order Loops. The method is to combine path 1 with path 2 if possible. If this combination is a legal loop, it is stored in the low end of

LOOP. If it is a legal path, it is combined with path 3 and testing continues. If it is not a legal path, the combination 1 and 3 is tested. After the combination of 1 and NPATH has been tested, the procedure begins again with 2 and 3, 2 and 4 and so on until NPATH - 1 and NPATH is reached. During this process all paths are kept in LPATH and the high end of LOOP ORDER(1) is the number of the current transmittance which is being tested against ORDER(2). The ORDER(I) contains the numbers of all the paths from 1 to I being tested. The testing consists of the following:

a. The beginning node of the first path is checked against the ending node of the second. If they are not the same, they cannot be combined (e.g., entries 1 and 3 in the example), so the test fails.

b. The two blocks in LOOP are ANDed and Subroutine UNPAK is called to find the nodes that are common to both.

c. If the ending node of the first path is not the same as the beginning node of the second, this cannot be a loop but it might be a new legal path so test is performed.

d. If N3 is non-zero, there are more than two nodes in common so this cannot be a legal combination - the test fails.

e. If the nodes in common (N1, N2) are the beginning and ending nodes of the two paths, this is a new First Order Loop. It is inserted by incrementing NLOOP and calling Subroutine EQUAL.

f. If more than one node is in common (N2, N3 $\neq$ 0), this cannot be a new legal path so the test fails. Otherwise Subroutine EQUAL is called to insert the new path and I is incremented so that testing will continue with this latest path.

This step ends when ORDER(1), the pointer to the lowest level path, is pointing to NPATH, the last path.

In the example the following results are obtained from applying these tests.

a. 1 and 2 from a new legal path. This cannot be combined with 3 because node 7 is traversed twice. 1 and 4 however produce a new First Order Loop.

b. 2 and 3 form a new First Order Loop.

c. 3 and 4 form a new path. This new path cannot be combined with 5 because node 7 is traversed twice.

d. 4 and 5 form a new First Order Loop.

Now the data has been changed to First Order Loops.

I	LOOP(I)	V(I)	LS(I)	NLOOP = 4
1	$2^1+2^5+2^6+2^{10}$	10	1001	
2	$2^2+2^3+2^7+2^8+2^9$	10	0	
3	$2^1+2^2+2^6+2^7$	10	1000	
4	$2^2+2^3+2^7+2^8$	-1	0	

These First Order Loops may be easily verified by examining the Flowgraph. Notice that the order of the nodes in the loops is irrelevant for the purposes of the program.

5. The values of the First Order Loops are now entered (DO 25) into the Transfer Function array VN according to the Shannon-Happ formula. Tagged values (LS > 500) have the tag of 1000 subtracted and are then entered into the numerator VN(I, 2) instead of the denominator VN(I, 1). The number of First Order Loops is then printed and the calculation of Higher Order Loops is ready to begin.

6. The procedure for the generation of Higher Order Loops is essentially the same as that for generating First Order Loops described in step 4; that is, loop 1 is compared with loop 2. If this creates a new Higher Order Loop, then the value of the new loop is entered into the VN array and it is checked against loop 3 for the possibility of a still higher order loop. When all combinations with loops 1 and 2 have been tried, combinations of 1 and 3 are tried, then 1 and 4 and so on until 1 and NLOOP. Processing then continues with 2 and 3, 2 and 4 and so on. As before, ORDER(1) points to the lowest level loop so that each time it is incremented it is written out, thus serving as a guide to how much

processing remains to be done. (Since this step is often the most time consuming part of the entire NASAP program, this information is very useful). As soon as ORDER(1) is equal to NLOOP, processing is complete and control returns to the calling program with the Transfer Function now in array VN. In the example, Higher Order Loop evaluation would proceed as follows

a. 1 and 2 form a Second Order Loop, its value is entered ($100S^{1001}$) and it is checked against 3 (failure, i.e., $M1 \neq 0$ because nodes 1, 2, 6, 7 are common to both) and 4 (failure).

Then 1 is checked against 3 (failure) and 4 (success, value $-10s^{1001}$).

b. 2 and 3 fails, 2 and 4 fails.

c. 3 and 4 fails.

In several places, instead of manipulating entries in LOOP by means of Subroutine EQUAL, the manipulations are performed directly in Subroutine LOOPS by means of the Equivalence array LOGIC (See Sec. 4.4). This is because in frequently executed statements the time required for execution of the subroutine linkage becomes significant. This is especially true in the ANDing of Higher Order Loops which can increase rapidly with the number of First Order Loops.

With the Transfer Function now in array VN, Subroutine ANSWER is called to format the raw form in VN, normalize it and print it out.

1. Upon entry Subroutine ANSWER writes the page heading on the output. Then VM, the array containing the Function after normalization, is created (DO 2).

2. The arrays NEXP (containing powers of S), AS and ASIGN (containing arithmetic signs) are created (DO 40).

3. The arrays are written out (except for zero coefficients) and the line dividing numerator and denominator is printed to the proper length.

4. Finally the factor after normalization is printed. (Also, as a debugging aid, the original VN array is printed). Control then returns to the calling program. This completes the description of the basic NASAP-70 Transfer Function algorithm. The routines described may be modified by the addition of other options (particularly the Sensitivity and Worst Case option). Nevertheless the basic algorithm remains the same.

4.2.4 Sensitivity and Worst Case

The Sensitivity and Worst Case option involves additions to the basic algorithm for finding the Transfer Function. The basic formulas used for evaluation of the Sensitivity Function, Worst Case Tolerance and Pole Sensitivity are·

FORMULAS USED[7]

SENSITIVITY FUNCTION

$$S_Q^P = \frac{d(\text{Ln}P)}{d(\text{Ln}Q)} = \frac{H(P',\bar{Q})}{H(P')} - \frac{H(\bar{P},\bar{Q})}{H(\bar{P})}$$

$$= \frac{H(\bar{P})\,H(P',\bar{Q}) - (H(P')\,H(\bar{P},\bar{Q})}{H(P')\,H(\bar{P})}$$

WORST CASE TOLERANCE

$$T_P = \left[\left(\frac{\partial P}{\partial Q_1}\ Q_1\right)^2 + \cdots + \left(\frac{\partial P}{\partial Q_n}\ Q_n\right)\right]^{2\ 1/2} \qquad Q = \text{circuit element value}$$

or

$$TP/P = \left[\sum_{i=1}^{n}\left(S_{Q_i}^P\ \frac{dQ_i}{Q_i}\right)^2\right]^{1/2} \qquad P = \text{dummy transmittance}$$

POLE SENSITIVITY[8]

$$\frac{dS_i}{dQ} Q = \left. \frac{-KD(S)}{\frac{\partial}{\partial S}[C(S)+KD(S)]} \right|_{S=S_i}$$

where Transfer $F_n = \frac{A(S)+KB(S)}{C(S)+KD(S)}$ or , K=circuit element value, and S_i = a pole

$$= \left. \frac{-Q\, H(\bar{P},Q')}{\frac{\partial}{\partial S}[H(P)]} \right|_{S=S_i}$$

In response to reading a card following the Circuit Description and before the first Output Requests section, beginning with the letters SENS, Subroutine SENSIT is called. If the card begins with WORS Subroutine WORST is called. Subroutine SENSIT and WORST make use of the Card Scanning Utilities described in Sec. 1.2.2.

Subroutine SENSIT processes Sensitivity Request cards which have the form:

SENS[ITIVITY] = Element Name

1. Upon entry Subroutine SENSIT calls Subroutine ASCAN to find the equals sign. If one is not found, a diagnostic is printed and step 5 gains control to read a new card.

2. A search of the CARD array is performed (DO 4) in order to find the Element Name. If it is not found, a diagnostic is printed and step 5 gains control.

3. Otherwise the element is rejected with a diagnostic if it is an independent source, if more than 20 Sensitivity Requests have been made, or if it is another Sensitivity Request for the same element, and step 5 gains control.

4. Having passed all these tests the element is inserted in the list of Sensitivity Requests by incrementing NSEN, the number of Sensitivity

Requests and by assigning a TAG value to the SENS entry for this element. NSEN, SENS and TAG are in Common block GAG. SENS(I) contains the value of the sensitivity tag assigned to element I from the list of 20 tags, TAG. These tag values are used later to identify various loops during the evaluation of Higher Order Loops in Subroutine LOOPS, and Subroutine REDUCE.

5. A new card is read and Subroutine SHIFT is called. If it is not a Sensitivity Request card control returns to the calling program. Otherwise the card is printed and step 1 regains control to continue processing.

If Subroutine WORST is called, Sensitivity Requests are inserted for each valid request. Thus, if more than 20 Sensitivity Requests results NFLAG is set and the Worst Case Request is ignored.

1. Initially Sensitivity Requests are inserted for all valid elements (i.e., not independent sources) in the DO 1. This is done by inserting a TAG value in SENS(I) for element I.

2. If the number of Sensitivity Requests resulting is greater than 20, NFLAG, an error flag internal to Subroutine WORST, is set and a diagnostic is printed. Otherwise, NWORST, the flag which indicates the Worst Case option has been requested, is set, and the TOL(I) array which contains the Tolerance value for element I is initialized to 0.1.

3. This step looks for cards of the form

TOL = Element Name = Value.

A new card is read and Subroutine SHIFT is called. If the card begins with TOL and NFLAG $\neq$ 1 (which would indicate that Worst Case analysis has been aborted) then Subroutine ASCAN is called to find the element name which is checked against the list in CARD. If there is no matching element name a diagnostic is printed and step 3 is repeated. Otherwise Subroutine ASCAN and NUMBER are called to determine the Tolerance Value which is inserted in TOL(I) for element I. Then step 3 is repeated.

If NFLAG is 1, the Tolerance Cards are read in step 3 but no other action is taken.

4. If step 3 encounters a card not beginning with TOL and NFLAG is not 1, the table of Tolerance Values is printed. Control then returns to the calling program.

Once the Sensitivity Tags have been inserted in the SENS list then, as the Flowgraph is built by Subroutine GRAPH, the tags are inserted in the S array for each Flowgraph transmittance. Then, in order to write out the correct power of S for each Flowgraph transmittance, these tags must be temporarily removed. This is done by calling Subroutine REDUCE for each transmittance before it is printed by the Main program.

Subroutine REDUCE (I, J, K) removes the tags from the S or LS array. I is the entry in the S or LS array. J is the true power of S returned by Subroutine REDUCE after all the tags have been removed. K is set to 1 if the entry does not contain the dummy transmittance and to 2 if it does.

1. Upon entering Subroutine REDUCE, K is set to 1 and J is set to I. If no Sensitivity Requests were made (NSEN = 0) and step 3 gains control.

2. If Sensitivity Requests were made, these tags are removed in the DO 1 loop. Since the Sensitivity Tags start from 2000 and go up, and since it is necessary to remove the largest tags first, the DO 1 loop is executed from L = 21 - NSEN to 20 and 21-L is used as the index of the TAG array. For example, if two elements are tagged M=21-2=19 then L goes from 19 to 20, i.e., 21-L goes from 2 ro 1. An entry contains a tag if the value (with all higher tags removed) is greater than the tag less 500. If it is greater, the tag is subtracted and the search continues.

3. J is checked for the presence of the unknown transmittance tag (1000). Thus if J is greater than 500, 1000 is subtracted from J and K is set to 2.

Processing continues as normal until the steps in Subroutine LOOPS which insert values of First and Higher Order Loops into various arrays. The method of inserting values into VN, the Transfer Function array, has already been described (Sec. 4.2.3). The Sensitivity Functions now involve three additional arrays: VN00, VN01 and VN10 which represent $-H(\bar{P}, \bar{Q})$, $H(P,\bar{Q})$ and $-H(\bar{P}, Q)$ in the formulas described previously in this section. The insertion of these values is performed by DO 979 and DO 799 in Subroutine LOOPS. Subroutine REDUCE is called to extract the pertinent tags and the values are inserted in the correct Sensitivity Function arrays based on these tag values. Thus, when Subroutine LOOPS returns control, not only has the Transfer Function been inserted in VN, but also the Sensitivity Functions have been entered into VN00, VN01 andVN10.

After Subroutine LOOPS has returned and Subroutine ANSWER has printed out the Transfer Function, if NSEN is non-zero (indicating the presence of Sensitivity Functions) then Subroutine SENSC is called.

Subroutine SENSC forms the Sensitivity Functions from polynomials produced by Subroutine LOOPS. The Sensitivity Function is equal to

$$\frac{H(\bar{P})\ H(P,\bar{Q}) - H(P)\ H(\bar{P},\bar{Q})}{H(P)\ H(\bar{P})}$$

where the H functions are stored as follows

$H(\bar{P})$. VN(I, 1) $\quad$ $H(P, \bar{Q})$: VN01(I, J)

$H(P)$: VN(I, 2) $\quad$ $-H(\bar{P},\bar{Q})$ VN00(I, J)

J is the element whose sensitivity is being found.

In many cases, $H(P, \bar{Q})$ (VN01) is 0 in which case the Sensitivity Function reduces to

$$\frac{-H(\bar{P},\bar{Q})}{H(\bar{P})}$$

1. Upon entry to Subroutine SENSC, the DO 305 loop is executed for each circuit element checking to see if it was tagged for sensitivity

(SENS(I)≠0). If it was tagged the following steps are performed, otherwise the succeeding element is checked. When all the elements have been checked control returns to the calling program.

2. The DO 306 loop determines the sensitivity number of the element (II). Then the DO 701 loop checks the VN01 array. If it is completely 0, the simplified formula above may be used so VNSEN, the array containing the Sensitivity Function is filled with the appropriate coefficients (DO 703) and Subroutine ANSWER is called to print out VNSEN.

3. If VN01 is not completely 0, then the standard formula must be used to compute VNSEN. Thus Subroutine MULT is called once to multiply VN(I, 1) and VN01(I, II) and again to multiply VN(I, 2) and VN00(I, II) and the results are added, (DO 702) to form the Sensitivity Function numerator. Then Subroutine MULT is called again to multiply VN(I, 1) and VN(I, 2) to form the denominator and then Subroutine ANSWER is called to print VNSEN.

Thus, upon termination of Subroutine SENSC, the information in VN, VN00, VN01 and VN10 has been converted to a set of Sensitivity Functions stored in VNSEN and VN00, VN01 and VN10 are no longer needed. Subroutine SENSC (and also Subroutine WORSTC below) make use of Subroutine MULT(A, B, C, MINA, MAXA, MINB, MAXB). This Subroutine multiplies the two polynomials A, B and puts the result in C. MINA, MAXA and MINB, MAXB are the minimum and maximum powers in polynomials A and B. A, B, C are arrays of length 101 and the index is 51 plus the power of the term. Array C is reset to 0 before multiplication occurs.

After Subroutine SENSC returns control to the Main program, the information in VNSEN is copied from Common block BITS into Common block POLY. This is to release the area in BITS for use as a workspace for use in later routines (e.g., the Plotter). If NWORST is non-zero, a Worst Case has been requested so the Main program calls Subroutine WORSTC.

Subroutine WORSTC computes the square of the Worst Case Tolerance function from the Sensitivity Functions and Tolerances for the elements. This function becomes

$$T_{P/P} = \left[\sum_{i=1}^{n} \left(S^{P}_{Q_i} \; \frac{dQ_i}{Q_i} \right)^2 \right]^{1/2}$$

The Sensitivity Function for an element can be in one of two forms,

$$S^{P}_{Q_i} = \frac{A_i(s)}{H(\overline{P})} \quad \text{or} \quad S^{P}_{Q_i} = \frac{B_i(s)}{H(P)\, H(\overline{P})}$$

where A(s) and B(s) are polynomials in S. Therefore two separate sums are kept, one of $\left(A_i(s) \; \frac{dQ_i}{Q_i}\right)^2$ and one of $\left(B_i(s) \; \frac{dQ_i}{Q_i}\right)^2$ and these are combined and put with the proper denominator to produce the square of the function.

1. Upon entry Subroutine WORSTC checks each element to see if a Sensitivity Function exists for that element (DO 10). If a Sensitivity Function exists (SENS (I) $\neq$ 0, then the Tolerance for that element (TOL) is squared and Subroutine MULT is called to square the Sensitivity Function numerator of that element (VNSEN(I, 2, J).

2. The denominator of the Sensitivity Function is now checked (DO 20) to see if it is the same as the Transfer Function denominator. If it is, the squared tolerance (TOL1) and the squared Sensitivity Function numerator (VWORKN) need only be multiplied and added to VW1. If it is not, NDEM is set to 1 and the result of the product is put in VW2.

3. If NDEM is 0, the simplified formula·

$$\frac{VW1}{VN(I, 1)^2}$$

may be used, so VW1 is put into the numerator of VW(DO 40) and Subroutine MULT is called to square VN(I, 1) and enter it in the denominator, VW(I, 1). If NDEM is non-zero, the standard formula

$$\frac{VW2 + VW1 \times VN(I, 2)^2}{VN(I, 2)^2 \times VN(I, 1)^2}$$

must be used. Thus Subroutine MULT is called once to square VN(I, 2) into VWORKN and again to multiply VWORKN and VW1 into the numerator of the Worst Case Function VW(I, 2). Then VW2 is added into the numerator (DO 31). Finally Subroutine MULT is called to multiply VN(I, 1) and VN(I, 2) into VWOKRN and again to square VWORKN into VW(I, 1), the Worst Case Function denominator.

4. The maximum and minimum powers of S in the Worst Case Functions are determined and control then returns to the calling program.

When Subroutine WORSTC returns control, the Main program calls Subroutine ANSWER to print out the Worst Case Function.

The last subprogram of the Sensitivity and Worst Case option is Subroutine POLSEN. This subroutine evaluates the sensitivities of the Poles of the Transfer Function (determined by Subroutine ROOTS described in Sec. 4.2.6) to variations in a circuit parameter. Subroutine POLSEN is called by the Main program when a card of the form

ROOTS, POLES

is read. Subroutine POLSEN then analyzes the roots found by the preceding call to Subroutine ROOTS.

1. If the Poles have not yet been found (NPOLES=0) or there is no Sensitivity Request (NSEN=0), Subroutine POLSEN prints a diagnostic and returns to the calling program.

2. The derivative of the denominator of the Transfer Function is found (DO 5) and stored in VDERIV. This is evaluated at the Real and Imaginary parts of the Poles and is stored in C(Real) and D(Imaginary).

3. For each element, tagged for Sensitivity Analysis, the numerator of the Root Sensitivity Function, $-QH(\bar{P},Q)$ (VN10) is evaluated at each Pole and then Real and Imaginary points are stored in X and Y. The Real and Imaginary parts of the Pole Sensitivity X/C(L) and Y/D(L) are evaluated and printed.

4. Subroutine Polsen then returns control to the calling program.

Two other routines, Subroutines PLOT and INVERT, of the Plotter also contain some code in connection with the Sensitivity and Worst Case option. These subprograms are described in the following Section 4.2.5.

4.2.5 The Plotter

NASAP-70 is equipped with extensive plotting capabilities for analysis of the Transfer Function and Sensitivity Functions. When a card beginning PLOT is encountered among the Output Requests, the Main program calls Subroutine PLOT to analyze this card and providing that the analysis of this card was successful (ERR $\neq$ 0), Subroutine INVERT is called to initiate the plotting. If the plot card TYPE option is FREQ or REAL,[9] the calculation of the points to be plotted is handled entirely within Subroutine INVERT. Any other TYPE option constitutes a call to the Fast Fourier Transform routines. After the points have been calculated, Subroutine PRTPLT writes out the plotted points. (BMOD and AMOD below are fixed and floating point names for the internal plotter option list).

1. Subroutine PLOT initially sets the BMOD list (the internal list of options) to -1 which indicates that the option did not appear on the PLOT card. Subroutine ASCAN is called to find the Left Parenthesis of the PLOT card which should be of the form:

PLOT(Option = Value/Option = Value/Option = Value...)

If a Left Parenthesis is not found Subroutine MSG is called to print a diagnostic. Subroutine ASCAN is called repeatedly until a Left Parenthesis or a Slash is found. If a Blank is found step 6 gains control to fill in options which are required by the Plotter but were not specified by the user.

2. Subroutine ASCAN is called to find the equal sign. If an equal sign is not found, Subroutine MSG is called to print a diagnostic. If a Blank or a Right Parenthesis is found, step 6 gains control. If a Slash is detected, step 2 is repeated. At this point, the Option field has been found.

3. Subroutine ASCAN is called to find the Value field. If the first 2 letters in the Option field are TY then BMOD(3) (known elsewhere as NTYPE), the internal code which tells the plotter which type of plot has been requested, is set according to the first 2 letters in the Value field·

IM· 0	ST. 1	EX 2	SI: 3
PU 4	FR· 6	SE 7	WO·8
RE· 9			

If this is the end of the PLOT card (B is a Blank or a Right Parenthesis), then step 6 gains control. Otherwise step 2 regains control to get the next Option.

4. If the first 2 letters of the Option field are EL, this option is connected with the Sensitivity and Worst Case Option. Thus, the CARD and SENS arrays are searched (DO 31 and DO 37) to match the element name in the Value field and obtain the Sensitivity Tag associated with it. If such a match cannot be obtained, a diagnostic is printed, the ERR flag is set and control returns to the calling program. Otherwise, if this is the end of the PLOT card then step 6 gains control, else step 2 regains control.

5. If this step is reached, the first 2 letters of the Option field are not TY or EL so the Value field contains a number. Hence, Subroutine NUMBER is called to find the number in the Value field. This number is inserted into the correct variable according to the first 2 letters in the Option field

ST STEP (AMOD(1), the interval between successive plot points)

TI TOTEE (AMOD(2), the interval between the first and last points plotted)

AM : AMP1 (AMOD(4), the positive portion of a pulse train)

BI AMP2 (AMOD(5), the negative portion of a pulse train)

CO FREQ1 (AMOD(6), a in e^{at})

CY FREQ1 (AMOD(6), f in Sin $2\pi ft$)

FR FREQ1 (AMOD(6), starting point for frequency and real plots)

TO FREQ2 (AMOD(7), ending point for frequency and real plots)

WI · FREQ2 (AMOD(7), pulse width for a pulse train)

DE · NPLOT (BMOD(8), number of calculated points per printed point)

Then, if the end of the PLOT card has not been reached control returns to step 2.

6. This step completes user options which were not specified, with default values. If an option has not been specified, BMOD is still -1. If BMOD(3) is more than 5 (i. e., not a Time Response) then these defaults are used:

FREQ1 (AMOD(6)) 1

FREQ2 (AMOD(7)) . 10^6

STEP (AMOD(1)) . $\left(\frac{\text{FREQ2. 01}}{\text{FREQ1}}\right)$

If BMOD(3) (NTYPE) is less than 6 and then

If STEP and TOTEE (AMOD(1) and AMOD(2)) are both unspecified,

STEP .01 TOTEE 1.0

STEP (AMOD(1)) TOTEE/100

TOTEE (AMOD(2)) . STEP X 100

NTYPE (BMOD(3)) · 0 (Impulse Response).

AMP1 (AMOD(4)) 1.0

If AMP2 (AMOD(5)) was specified, AMP1 is biased accordingly.

FREQ1 (AMOD(6)) : 1.0

If FREQ2 (AMOD(7)) was specified, NTYPE is set to 5 to indicate a Pulse Train.

7. In all cases, if DEN (BMOD(8)) was unspecified, it is set to 1. If BMOD(3) is 7 (Sensitivity Plot), the arrays containing the numerator (A1) and denominator (B1) of the function to be plotted are filled with the correct Sensitivity Function (DO 50) and a heading is printed. If BMOD(3) is 8 (Worst Case Plot), the arrays are filled with the Worst Case Function and a heading is printed. For all other values of BMOD(3), the arrays A1 and B1 are filled with the Transfer Function. Subroutine PLOT then returns control to the calling program.

Assuming the analysis of the PLOT card was successful (ERR $\neq$ 1), Subroutine INVERT is called to initiate the plotting process·

1. Upon entry, Subroutine INVERT resets NFLAG (the remainder indicator) to 0, sets STEP to 1 if it was negative, and evaluates NUMTI, the number of points to be plotted. Then, if a Time Response has been requested (NTYPE less than 6), Subroutine INPUT is called to evaluate the transform of the response. Then, in the case of a Time Response, the function to be plotted in A and B is checked. If it is too simple, a diagnostic is printed and the plot is aborted. If it is not rational, the numerator is divided by the denominator and the remainder of this division is used as the function to be plotted. A message to this effect is printed and NFLAG is set to 1.

2. The function in A and B is printed and if NTYPE is less than 6 (a Time Response), the plot axes (ABSC and ORD) are marked 'TIME' and 'MAG' and Subroutine SCALE, ROUTH, SAMPLE, FLIP and ADJUST are called to perform the Fast Fourier Transform. Control then returns to the calling program.

3. In cases where a Time Response has not been requested (NTYPE greater than 5) a further test is made. If NTYPE is not 9 (a REAL plot), NUMTI is calculated for a logarithmic plot instead of a linear plot. Then the points are calculated (DO 4) and entered into the TEE, FCTN (and PH when NTYPE is not 9) arrays. For a REAL plot, the real part of Z,

(i. e., σ) is incremented in linear steps. Otherwise, the imaginary part of Z (i. e., jw) is incremented in logarithmic steps).

4. After the points have been calculated, ABSC and ORD are labelled 'FREQ' and 'MAG, and Subroutine PRTPLT is called to output the points. In the case of a REAL plot, the Phase must always be 0 so if NTYPE is 9, Subroutine INVERT returns control here. Otherwise, ORD is labelled 'PHA', FCTN is filled with the points from PH and Subroutine PRTPLT is called again before control returns to the calling program.

In the case of Time Response plots (NTYPE less than 6) Subroutine INPUT is called by Subroutine INVERT to evaluate the transform of the response.

1. Upon entry, Subroutine INPUT prints out the quantities associated with each type of plot according to NTYPE.

2. The Transform of the response is calculated as follows:

NTYPE = 0 Impulse Response. No change is required since F(jw) = 1.

1 Step Response. $F(jw) = \frac{1}{jw}$ so B, the denominator, is multiplied by jw.

2 Exponential. $F(jw) = \frac{1}{jw\text{-FREQ1}}$, so B is multiplied by (jw-FREQ1)

3 Sine Wave. $F(jw) = \frac{2\pi x\text{FREQ1}}{(jw)^2 + (2\pi x\text{FREQ1})^2}$ so B is multiplied by $(jw)^2 + (2\pi x\text{FREQ1})^2$ and SCMAG is multiplied by (2πxFREQ1).

4 Single Pulse. $F(jw) = \frac{1}{jw}(1-e^{-jw\text{FREQ2}})$ so B is multiplied by jw and the tags ISN and NTAY are set to indicate that the factor $(1-e^{-jw\text{FREQ2}})$ is to be included.

5 Pulse Train. $F(jw) = \frac{1}{jw}(1-e^{-jw\text{FREQ2}})\frac{(e^{-jw\text{FREQ1}})}{1-e^{-jw\text{FREQ1}}}$ so B is multiplied by 1/jw and the tags NTAY and ISN are set to show that the additional factors must be included later.

3. After the correct transform has been calculated, Subroutine INPUT returns control to the calling program.

Later on, Subroutine INVERT begins the Fast Fourier Transform, by calling Subroutine SCALE:

> Subroutine SCALE simply readjusts the values of the Fourier Transform coefficients (in A and B) to avoid Floating Point Overflow. Control then returns to the calling program.

The next step in the Fast Fourier Transform is to call Subroutine ROUTH. Subroutine ROUTH uses Subroutine CALCR to find a path along which the integral

$$\frac{1}{2\pi}\int_0^{2\pi} F(\sigma + jw)\,e^{(\sigma + jw)t}\,dw$$

is to be evaluated. This path must lie to the right of any poles in the complex plane. As shown in Figure 4.5, Path 1 is not acceptable while Path 2 is acceptable. This is due to the fact that the Real Part of S ($=\sigma$) must be negative in $e^{(\sigma+jw)t}$

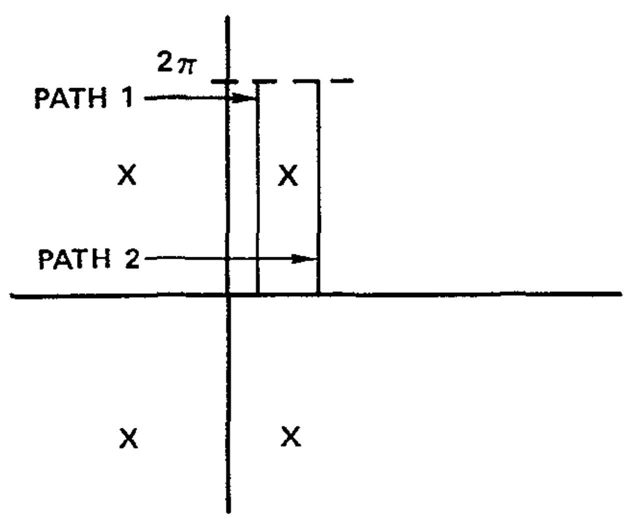


Figure 4 5

in order for the integral to converge. The upper limit of 2π for the integral is sufficient because after Subroutine SCALE has readjusted the Transform coefficients, all the significant information has been pushed into this range.

1. Subroutine ROUTH uses an initial SIGMA value of 0.0 and calls Subroutine CALCR(LOGI) to see if all the poles lie to the left of the SIGMA value. Subroutine CALCR returns a LOG1 value of True if all the poles lie to the left of SIGMA, otherwise, it returns False. Subroutine ROUTH increments the value of SIGMA until Subroutine CALCR returns True. Having found an acceptable value of SIGMA, control returns to the calling program.

2. Subroutine CALCR (LOG1) uses the Routh Stability Criterion to determine whether all poles lie to the left of the current value of SIGMA. If they do, the value returned in logical variable LOG1 is True, otherwise it is False. Upon entry, D(I, J), the Routh Table, is reset to 0(DO 242).

3. The Routh Table is then computed, and the first column (D(1, J)) is checked (DO 245). If any zero or negative values are found, LOG1 gets a value of False and control returns to the calling program.

4. Otherwise LOG1 gets the value True, and control returns to the calling program.

Having found an acceptable SIGMA value, Subroutine INVERT calls Subrouting SAMPLE.

1. Subroutine SAMPLE computes the quantities $F(\sigma + jw)$ and $e^{(\sigma + jw)}$ which are stored in RSPNS(I) and EXCMP(I), respectively, (DO 130). As each point is evaluated, the flag ISN is tested to see if the transform requires any modification. (ISN was set by Subroutine INPUT as the Fourier Transform was calculated). Any modification required is performed by means of complex variable ZZ.

2. Subroutine SAMPLE then returns control to the calling program.

With the sample points now in RSPNS and EXCMP, Subroutine INVERT calls Subroutine FLIP, the heart of the Fast Fourier Transform.

1. Upon entry to Subroutine FLIP, the coefficients in the RSPNS array are reordered as prescribed by the Fast Fourier Transform technique (DO 106).[10]

2. The integral

$$\int_0^{2\pi} F(\sigma + jw) e^{jwt} \, dw$$

is computed (DO 110 and DO 108) thus producing the output response samples in RSPNS.

Subroutine ADJUST is called to re-evaluate the first 5 sample points by means of a Taylor Series. The Taylor Series coefficients are computed by Subroutine TAYLOR and entered into the TERM array.

1. Upon entry Subroutine ADJUST enters the Time points into the TEE array and the output sample points into the FCTN array (DO 150).

2. If SIGMA is non-zero, each of the sample points in FCTN is multiplied by the $e^{\sigma t}$ factor. (DO 153).

3. Subroutine TAYLOR is called to enter the Taylor Series coefficients into the TERM array.

4. If NTAY is non-zero, the Taylor Series coefficients are used to enter the first 5 output points into the FCTN array. If the series fails to converge at any one of the points (FNP(I) greater than 1000) the remaining points of the original 5 are not altered.

5. Subroutine PRTPLT is called to print and plot the data points. Control then returns to the calling program.

Subroutine TAYLOR uses the Transfer Function coefficients A(I) B(I) to obtain a Taylor Series.

1. The Taylor Series is computed term by term and entered into the TERM and POW array. (DO 214). If any term becomes too large (greater than 10^{60}), the Taylor Series is halted and NTAY is reset to zero.

2. Control returns to the calling program.

Subroutine PRTPLT prints and plots the points in the TEE and FCTN arrays.

1. Upon entry Subroutine PRTPLT writes out the points in TEE and FCTN in three columns. If NPLOT, the number of points to be plotted from the total number computed (the plot density), is zero, control returns to the calling program.

2. The maximum and minimum plot points are found (DHI, DLO). The points are plotted into the P array and printed out one at a time (DO 182) along with axis labelling for the TEE axis. Control then returns to the calling program.

4.2.6 Rootfinding

The Poles and Zeroes of the Transfer Function are evaluated by means of Subroutine ROOTS. This routine is called by the Main Program in response to reading an Output Request Card containing the word ROOTS.

Subroutine ROOTS is essentially a modified version of the Svoboda Polynomial Rootfinder. In this algorithm, the complex plane is searched for zeroes using a five point test. As a root is detected, the search step size is repeatedly decreased to 1/16 of its previous value until the five points in the five point test do not yield significantly different results. Thus, the root is found to the maximum resolution available from the computer. The original polynomial is used for the final test while the reduced polynomial, having previously found roots divided out, is used for the coarse testing. This technique combines the ability to handle multiple roots while retaining the accuracy of the original unreduced polynomial. The scan is performed initially within the unit circle to obtain roots less than 1.0. The polynomial is then inverted to cause roots which were greater than 1.0 to fall within the unit circle. The five point test used by the algorithm is neither necessary nor sufficient to find all the roots of the polynomial, yet in practice, the algorithm has yielded remarkable results despite its non-rigorous mathematical foundation. The roots produced by this algorithm represent not only the root values themselves but also the accuracy to which the roots were found, since every non-zero digit in the root is guaranteed correct within the accuracy of the computer. This information is of considerable value when high order polynomials are being evaluated.

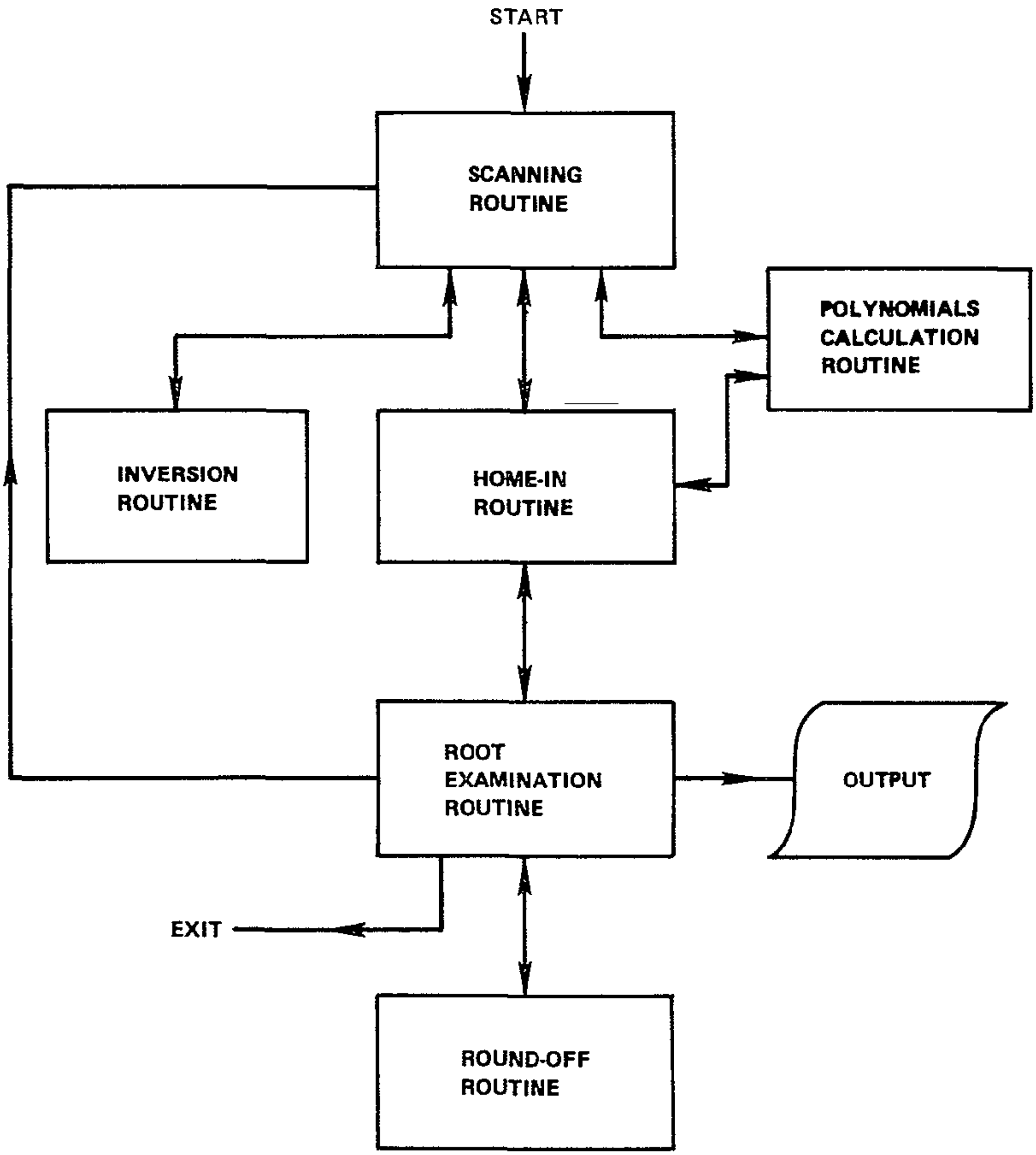


Figure 4 6 Flow of Control in Svoboda Algorithm

The flow of control among the various logical routines of the algorithm is depicted in Figure 4.6. After obtaining the input data, the Scanning Routine systematically steps over the Unit Circle evaluating the reduced polynomial (i.e., one having previously found roots divided out) at five points on each step. When a criterion for the possible existence of a root at any step is fulfilled, a transfer is made to the Home-in-Routine. Mathematically, this criterion is neither necessary nor sufficient for the existence of a root. The Home-in-Routine approximates the root as closely as possible and then transfers to the Root Examination Routine. Using the Round-off Routine, the Root Examination Routine either rejects the root or refines it (by means of the Home-in-Routine again) and outputs it. Since refinement always uses the original unreduced polynomial, and since it proceeds by successive significant digits, every non-zero digit printed is guaranteed correct within the accuracy of the computing system. Control then passes back to the Scanning Routine which either continues scanning or stops because the required number of roots has been obtained. When a scan is completed, if the maximum number of scans has not yet been reached, the Inversion Routine gains control to "invert" the polynomial, before returning to the Scanning Routine which continues to scan. A detailed discussion of each element of the algorithm follows.

1. <u>The Scanning Routine</u>

The main routine in the Algorithm is the Scanning Routine which moves across the Unit Circle as shown in Figure 4.7. The real and imaginary axes between the ranges (-1, 0), (1, 0) and (0, -i), (0, i) are divided into sixteenths. The Scanning Routine uses the Polynomial Evaluation Routine to find five points at each step in the scan, the central point and four points one sixteenth above, below, to the left and to the right of the central point. A test is then performed to see if this point is near a root. The criteria for transferring to the Home-in-Routine to examine a possible root more closely are.

1. The value at the central point must be less than the value of any of the four surrounding points.

2. The five points must not be equal in value.

Note that the criteria are mathematically insufficient to determine the existence of a root.

If these criteria are not met, the scan is incremented by one sixteenth and the scanning process is resumed. Scanning is performed from left to right and from bottom to top.

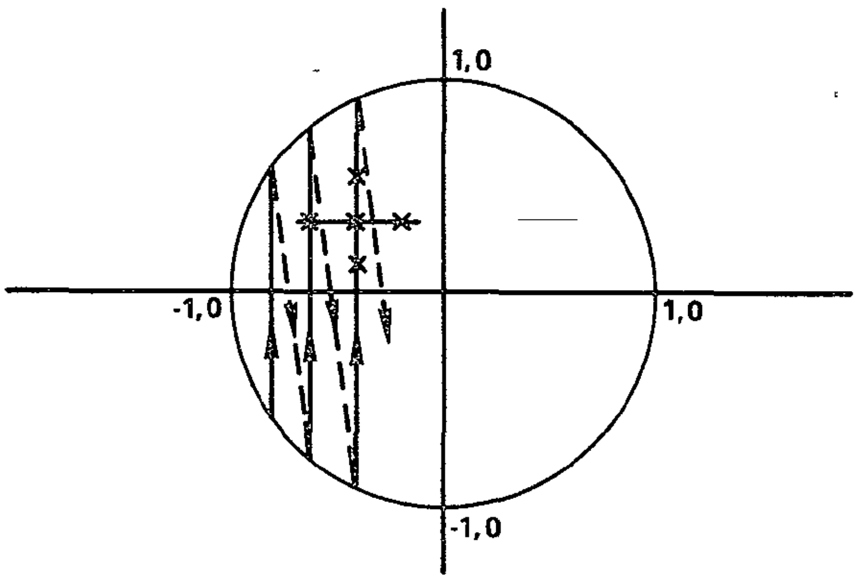


Figure 4 7 The Five Point Scan

When a scan has been completed, that is, when the point (1, 0) has been evaluated, a check is performed to see if the number of scans has exceeded an upper limit. If not, a transfer is made to the Inversion Routine (described below) which performs the inversion of the polynomial and then returns control to the Scanning Routine.

The maximum number of scans allowable is twice the number of roots in the polynomial. Since at least one root must be obtained after each pass of both the "real" polynomial and the "inverted" polynomial (requiring a total of two scans), a number of scans no more than twice the number of roots not yet found should be required for the algorithm to work properly. (Usually, considerably fewer scans are required because several roots may be found on one pass). If this number is exceeded, however, the routine stops automatically.

The problem of evaluating the circumference of the Unit Circle twice (once on a "real" scan and once on an "inverted" scan) is solved by taking two precautions. Firstly, all points evaluated in a "real" scan must be within the Unit Circle, while an "inverted" scan is permitted to violate the boundary of the circumference slightly. This precaution alone, however, is not sufficient to avoid a duplication of roots which lie close to the boundary, therefore, in the Root Examination Routine a check is performed to see if the root under consideration has appeared previously in another scan. This second precaution prevents roots which have been found once by a "real" scan and once by an "inverted" scan from appearing twice.

2. Polynomial Calculation

The Polynomial Calculation Routine uses Horner's Technique to evaluate the polynomial complex value from the coefficients. Basically, Horner's Technique is the iterative evaluation of the expression

$$F_{i+1} = (F_i + a_i) \cdot x \quad i \text{ goes from } 1 \text{ to } n$$

with $F_1 = (0, 0)$ and $F_{n+1} = F_n + a_{n+1}$ the final value. a_i is the i^{th} coefficient in the polynomial.

$$P(x) = a_1 x^n + a_2 x^{n-1} + \cdots + a_n x + a_{n+1}.$$

Thus the final value is cumulatively built up in F. If necessary, the complex value (F) is divided by previously found roots to form a reduced polynomial value

$$F = \frac{P(x)}{(R_1 - x)(R_2 - x) \ldots (R_m - x)}$$

where m is the number of roots found previously. In order to avoid overflow, the absolute value of any of the factors $(R_i - x)$ is not allowed to be less than 10^{-60}. The final step is to obtain the residual (the complex absolute value) of F.

$$H = \sqrt{([Re(F)]^2 + [Im(F)]^2}$$

The residual is computed for four or five points (the Home-in-Routine does not require the central point to be evaluated each time) on each entry to the Calculation Routine. It is these four or five values (H) which are used by the Scanning and Home-in-Routines as the polynomial values at the test points.

3. Polynomial Inversion

In order to obtain all the roots of the polynomial

$$P(x) = a_1 x^n + a_2 x^{n-1} + \cdots + a_n x + a_{n+1}$$

by scanning the complex plane within the Unit Circle, at some point a new polynomial is formed from the original one by reversing the order of the coefficients. The roots of this new polynomial are the reciprocals of the roots of the original polynomial, thus effectively bringing inside the Unit Circle all the roots which were previously outside. The Inverted Polynomial is

$$P'(y) = a_{n+1} y^n + a_n y^{n-1} + \cdots + a_2 y + a_1 .$$

If $P(r) = 0$, it is easy to show that

$$P(r) = 1/r^n P(r) = P'(1/r)$$

Therefore, $y = 1/r$ is a root of the new polynomial p'.

Thus, in order to obtain all the roots of a polynomial, the program must deal effectively with two kinds of roots, "real" and "inverted," and must obtain the reciprocal of "inverted" roots before writing them on the output. The polynomial Inversion routine, simply reverses the order of the coefficients, sets a program variable to point to either the "real" or "inverted" techniques, and returns control to the main program.

4. Home-ing In

Once five points have been obtained which fulfill the criterion for root examination, the homing-in procedure subdivides the distance between the central point and one of the outside points into sixteenths and, using the central points as a starting value, employs the Polynomial Evaluation Routine to compute four new outside test points. These new four points and the central point are then examined for the one having the smallest residual which becomes the central point for a new five point test. If this new central point is the same as the previous central point, the scale is subdivided into sixteenths as before so that the five point test becomes progressively more refined. At each subdivision, a new level of significance is obtained (each significant digit being a Hexadecimal digit because the scale factor is 16). This process of "home-ing in" is continued until either

1. The required number of digits of accuracy is reached or,

2. The residuals of the five points do not differ by a significant amount.

In case 2, the resolution of the computing method and system has been reached and the remaining significant digits, if any, are filled with zeroes. This represents a Stopping Criterion which is independent of any formula for round-off error,[11, 12] but depends only on the previously mentioned resolution.

Each time a new step is taken in the Home-in-Routine, three checks are made. Firstly, if the central point should happen to be the origin (0, 0), the scale is expanded (subdivided by sixteenths), this allows for roots which may range over large orders of magnitude. Secondly, if more than sixteen steps should be taken in one direction at the same level of significance the level of significance is dropped so that larger (coarser) steps may be taken. Finally, a check is performed to ensure that the routine does not stray too far out of the Unit Circle. This safeguards against the possibility that the original location presented to the Home-in-Routine was not in the vicinity of a root, in which case it could easily wander outside the Unit Circle. If this condition should be detected, the Home-in is aborted and control is returned to the Scanning Routine.

If either case 1 or case 2 occurs, the Home-in-Routine has reached a successful completion and transfers control to the Root Examination Routine.

5. Root Examination Procedures

The Root Examination Routine first checks to see if the root just found came from the original polynomial coefficients or from a reduced polynomial. If the polynomial was a reduced one, the value of the root can be regarded merely as an approximation. The Home-in-Routine is therefore called again to repeat the latter part of the home-in procedure this time using the original polynomial and using the approximate root as a starting value. This technique ensures that every root is found from the original polynomial and that its accuracy does not depend on the accuracy with which previous roots were found.

After the final root value is obtained, a second check is made, this time on its residual. If the residual is greater than a certain value, the root is considered unreliable and is discarded. If this should occur, control is transferred back to the Scanning Routine. The maximum value of the residual is chosen arbitrarily to be one half the value of a_{n+1}, the constant coefficient in the polynomial.

Having passed these tests, the root value is rounded-off in the Round-off Routine (which is not discussed here because it has no bearing on the actual rootfinding algorithm). If the root is "inverted" it's reciprocal is found using floating point division, and this value, after being rounded-off again, becomes the root value. A third check is then performed on the root to see if it has been found on a previous pass. (This check was mentioned previously in connection with the Unit Circle Boundary). If both real and imaginary parts fall within one significant digit of a root found on the other type of pass ("real") it is rejected as a root but is nevertheless entered, in the same way as acceptable roots, in the list of roots used to form a reduced polynomial. (This prevents its being found again). If the root is not rejected, it is converted from its internal hexadecimal representation to decimal, rounded-off again, and is written on the output.

A final check is made to see if the required number of roots has been obtained. If not, control is returned to the Scanning Routine, otherwise the algorithm terminates.

Because the algorithm lacks a rigorous mathematical foundation, it cannot be guaranteed to find all the roots. However, every root detected is guaranteed accurate within the accuracy of the computing system, it is independent of the technique used to find the roots.

4.2.7 The Automatic Scaler

The Main program always calls the Automatic Scaler, Subroutine SCALER, before the Flowgraph is constructed. Subroutine SCALER examines the element values in the VPATH list and if the range of exponents of Inductors and Capacitors is too large, the values are multiplied by 10^{FACTOR} where FACTOR is an integer which is determined so as to minimize the range of exponents. If no such FACTOR can be found, a warning is printed.

Since the element values are modified after scaling, the 'S' symbol which appears in the Transfer Function must be reinterpreted. For example, if FACTOR is determined to be 3, a capacitor of value $(10^{-3}\,s)^{-1}$ becomes

$$(10^{-3}\,s \times 10^{3})^{-1} = (s)^{-1}$$

thus the new s is 10^{-3} times the true s. Therefore, wherever 's' appears in the Transfer Function $10^{3}s$ must be substituted. Similarly, all the plots and roots obtained by operating on this scaled transfer function must be modified.

1. Upon entry to Subroutine SCALER, all elements containing s (i. e., inductors and capacitors) are checked (DO 1). The sum of their exponents is entered into D. At the same time, the sum of their exponents scaled by 10^{-1} is entered into DL and the sum of their exponents scaled by 10 is entered into DR.

2. If D is less than 60 and FACTOR is 0, no scaling need be done so Subroutine SCALER returns. Otherwise, D, DL, and DR are compared to see whether FACTOR should be incremented or decremented for minimum D. If D is not already minimum, FACTOR is adjusted and then control passes back to step 1 for another iteration. If D is already minimum, FACTOR is checked to see if it is still 0. If it is, no scaling could be performed so a warning is printed and control returns to the calling program.

3. If FACTOR is non-zero, all the inductors and capacitors are multiplied by 10^{FACTOR} (DO 5). Control then returns to the calling program.

4.3 Modular Organization

NASAP-70 has been designed in order to facilitate easy insertion of additional routines and removal of supplied routines. Each module is called from the main routine. In general, three modifications are required:

1. The insertion or deletion of a Fortran CALL statement which transfers to the routines to be inserted or deleted.

2. The addition or removal of the routines themselves.

3. Using the Dictionary of Variables, conflicts in variable names may be eliminated in the case of insertions, and Common blocks and variables no longer required may be removed in the case of deletions.

If these steps are followed, users having a working knowledge of Fortran should have little difficulty adding new capabilities to their versions of NASAP-70.

4.3.1 Removal of Supplied Routines

Under certain circumstances it may be desirable to remove some of the routines supplied with NASAP-70 (e.g., in order to run in less than 32K words). For proper operation NASAP-70 requires only one of the circuit description analysis routines and the Transfer Function routines. Thus, the following modules may be removed:

Either Subroutine NASAP or Subroutine READ.

Subroutine FINE.

Sensitivities and Worst Case option.

Plotting package.

Rootfinder.

Automatic Scaler.

For instance, if the Sensitivities and Worst Case option is to be omitted, all the references to Subroutines SENSC, WORSTC, MULT, SENSIT, WORST, and POLSEN should be deleted, as well as those routines themselves. Finally, Common blocks POLY and WORST1 and references to their variables should be removed.

4.3.2 Addition of User Routines

A user who is familiar with the operation of NASAP-70 may desire to add some of his own options to the supplied program, e.g., for further analysis of the Transfer Function. If any scanning of card images is to be performed, it is wise to become acquainted with the card scanning utilities ASCAN, BSCAN, CSCAN, SHIFT, NUMBER, etc. If bit manipulation is to be performed the utilities LOR, LOX, UNPAK, etc., are available. Finally, if operations are to be performed on the Transfer or Sensitivity Functions, the method of storage of these data in the 'VN' arrays should be examined (4.2.3 and 4.2.4).

If NASAP-70 is used with the Overlay structure, all user-written routines should be called by the main program only. This will eliminate troubles due to calling routines not in core. In general, workspace is available in the following Common blocks·

BITS	4095	Words
SPEED	2048	Words

This may not be true at all times, however, so the user should consult the Dictionary of Variables to ensure that he does not erase any useful data.

REFERENCES

1. IBM System/360, Fortran IV Language, Systems Reference Library Form No. C28-6515-7.
2. Potash, H. and L. P. McNamee, Application of Unilateral Graph Techniques to Analysis of Linear Circuits, Proc. 23rd Natl. ACM Conf., 1968, pp. 367-368.
3. McNamee, L. P. and H. Potash, A User's Guide and Programming Manual for NASAP, UCLA Department of Engineering Report No. 68-38, August 1968.
4. Happ, W. W., Flowgraph Techniques for Closed Systems, IEEE Transactions on Aerospace and Electronics Systems, Vol. AES -2, May 1966, pp. 252-264.
5. Knuth, D. C., The Art of Computer Programming, Vol. 1, Fundamental Algorithms, Addison-Wesley, 1968.
6. Russell, E. C., H. Okrent and L. P. McNamee, Instrumentation of a NASAP Subroutine, IEEE Transactions on Eduction, Vol. E-12, No. 4, December 1969, pp. 243-250.
7. Carpenter, R. M., "Computer-Oriented Sensitivity and Tolerance Techniques," Course Notes Automated Circuit Anal ysis, UCLA, April 3-7, 1967.
8. Vattuone and Dorf, Root Sensitivity as a Design Criterion, First Asilomar Conference, November 1967, pp. 287-290.
9. Haag, K., "Designers Manual for Computer-Aided Design of Communication Circuits," Illinois Institute of Technology, Final Report Contract NAS12-2064, December 1969.
10. Knuth, D. E., The Art of Computer Programming, Vol. 2, Seminumerical Algorithms, Addison-Wesley, 1969.
11. Muller, D. E., A Method for Solving Algebraic Equations Using an Automatic Computer, Mathematics of Computation (formerly Mathematical Tables and Other Aids to Computation), 1956, pp. 208-215.
12. Adams, A., A Stopping Criterion for Polynomial Root Finding, Communications of the ACM, Vol. 10, No. 10, October 1967, pp. 655-658.
13. IBM System/360 Operating System, Fortran IV (G and H) Programmer's Guide, Systems Reference Library Form No. C28-6817-1.
14. IBM System/360 Operating System, Linkage Editor and Loader, Systems Reference Library Form No. C28-6538-7.

CHAPTER 5
SYSTEM REQUIREMENTS

5.0 Computer Requirements

NASAP-70 requires a computer with no less than a 32K word memory to run, using the Overlay Structure in Section 5.3. For some computers having unusually large System Monitors the supplied program may be too large anyway. If this should be the case, some of the supplied options may have to be removed or restricted, e.g., the Sensitivity option could be reduced to handling fewer than 20 elements. This would decrease the size of Common block POLY. On computers with more than 32K words, the Overlay Structure may be neglected entirely. This would not only simplify the loading procedure but also decrease running times.

NASAP-70 needs no additional scratch units other than a card reader and line printer.

5.1 Running a NASAP Problem from the Source Deck

NASAP-70 is distributed as a Fortran source deck. The program must be compiled and loaded in order to run any problems on it. If no changes are to be made, this process may be performed directly. In the case of a machine with a smaller memory, the user may wish to use the Overlay Structure described in Section 5.3. When a user possesses a machine with a very limited memory, some modifications to the source deck may be necessary (e.g., Removal of some supplied routines, reduction of COMMON sizes). Guidelines for these changes are given in Section 4.3.2.

If a user possesses a machine which has a Fortran compiler which uses "backwarding addressing", (e.g., IBM 7094) he should ensure that all COMMON blocks are adjusted to the correct lengths as described in Section 5.5. Further, care should be taken to correct any problems with the Fortran AND function (Section 5.4).

Having made any necessary changes in the source deck, the program may be compiled and loaded. If the machine on which the program is to be run, is not IBM 360 compatible, it is strongly suggested that the sample problems be run and verified with the solutions given before any large-size user problems are attempted.

Here is the IBM/360 control card deck for running NASAP from the source deck [13,14]

```
// NASAP   JOB  (Installation dependent data).
// STEP1 EXEC  FORTHCLG, PARM.LKED='XREF, LIST, OVLY'
// FORT.SYSIN DD *
      Fortran Source Decks go here.

/*
//LKED.SYSIN DD *
    INSERT MAIN
    OVERLAY ONE
    INSERT ASCAN, BSCAN, CSCAN, ISORT, MSG, NUMBER, SHIFT, SORT
    OVERLAY TWO
    INSERT NASAP, SENSIT, WORST
    OVERLAY TWO
    INSERT READ, UNITS
    OVERLAY TWO
    INSERT WHAT, PLOT
    OVERLAY ONE
    INSERT FINE, SCALER, POLSEN
    OVERLAY ONE
    INSERT EQUAL, LOR, LOX, LSTOP, UNPAK
    OVERLAY THREE
    INSERT CALC, INBIT, GRAPH
    OVERLAY THREE
    INSERT LOOPS, CLEAR, REDUCE
    OVERLAY ONE
    INSERT MULT, SENSC, ANSWER, WORSTC.
    OVERLAY ONE
    INSERT INVERT
    OVERLAY FOUR
    INSERT INPUT, SAMPLE, FLIP
    OVERLAY FOUR
    INSERT ADJUST, TAYLOR, PRTPLT
    OVERLAY FOUR
    INSERT SCALE, ROUTH, CALCR
    OVERLAY ONE
```

```
  INSERT ROOTS
/*
//GO.SYSIN DD*
        User Problem Data Decks go here.
/*
//
```

5.2 NASAP-70 Input Data

The input data consists of three sections

1. Title Cards (Optional)
2. Circuit Description
3. Sensitivity and Worst Case Section (Optional)
4. Output Requests.

1. Up to 10 Title Cards are allowed. They are printed in the output listing exactly as they are read.

2. The Circuit Description section may be in one of three forms depending on the format of the first card

   a. NASAP
   b. TREE
   c. CIRCUIT (Operating Frequency)

a. For compatibility with other versions of NASAP, this heading card will cause the program to accept the circuit description in the form described in the previous NASAP User's Guide. This data format corresponds to the following conventions·

Each card contains one circuit element. The first letter describes the element and must be one of the following

- V Voltage Source
- I Current Source
- C Capacitor
- L Inductor
- R Resistor.

Each of these letters may be followed by up to 11 alphanumeric characters to bring the total number of characters in the element name to not more than 12.

The card format is as follows.

NAME ORIGIN TARGET VALUE UNITS (optional) DEPENDENCY (optional)

NAME refers to the previously described element name. ORIGIN and TARGET refer to the origin and target node numbers of the element. VALUE is the element value in any format. UNITS is optional and must be one of the following:

UF
PF
MH
UH
K
M

If UNITS is not specified, MKS units are assumed. DEPENDENCY is also optional and represents the name of some other element appearing in the circuit description, preceded by the letter V or I to denote dependency on voltage or current.

At least one blank must appear between each of the data fields on the card, otherwise the data is freefield. The last card in the circuit description must contain either the word OUTPUT or the word END.

This form of circuit description relieves the user of the need to organize the tree. Instead, the program automatically builds a tree which results in minimum computation time.

b. For a user-defined tree, a different input format is used. Following the TREE card are the circuit description cards. The last card in this section must be a card with the word END. The first 1 or 2 letters on each circuit description card must be one of the following:

E Independent Voltage Source

J Independent Current Source

CE Capacitor Branch

CJ Capacitor Link

LE Inductor Branch

LJ Inductor Link

RE Resistor Branch

RJ Resistor Link

DE Dependent Voltage Source

DJ Dependent Current Source

Each of these may be followed by up to 10 or 11 alphanumeric characters to bring the total to not more than 12.

The data card format is as follows

NAME (ORIGIN - TARGET) = VALUE UNITS (Optional)

or

NAME/DEPENDENCY (Optional) (ORIGIN - TARGET) = VALUE UNITS (Optional)

NAME refers to the element name described above. DEPENDENCY is one of the letters V or I to denote voltage or current dependence, followed by the name of the controlling element. ORIGIN and TARGET are the origin and target node numbers. VALUE is the element value. The optional UNITS field must be one of the following

MF

MMF

PF

MH

MMH

K

M

If the units are not specified, MKS units are assumed. All input cards are free-field. Blanks and/or commas may be used freely and are ignored by the program. All other punctuation must appear exactly as shown.

c. For a tree constructed for optimum accuracy at a specific frequency, the CIRCUIT card is used in conjunction with the operating frequency in cps enclosed in parentheses. The input data format is identical with that used after a TREE card except for the format of Capacitor, Resistor and Inductor names where it is no longer possible to differentiate between branches and links

C Capacitor

R Resistor

L Inductor.

If no operating frequency is specified, a default of 1 cps is supplied.

3. SENSITIVITY INPUT REQUESTS

SENSITIVITY FUNCTION·

To obtain the sensitivity of the transfer function to a particular circuit element, a sensitivity request card is required. These cards must immediately follow the 'END' or 'OUTPUT' cards terminating the circuit description. The format of the sensitivity request card is as follows·

SENSITIVITY = 'ELEMENT NAME,

or

SENS = 'ELEMENT NAME'

where 'ELEMENT NAME' is the name of a circuit element.

Sensitivity request for independent voltage and current sources are ignored. Also, a maximum of twenty valid sensitivity requests may be processed for each circuit.

The sensitivity function (in terms of S) is printed in the same format as the transfer function.

WORST CASE ANALYSIS:

A worst case analysis can be provided for circuits of twenty elements or less (not counting independent voltage or current sources). To obtain this analysis, a card must be provided immediately after the 'END' or 'OUTPUT' cards terminating the input description in the following format

WORST

To obtain a worst case analysis, the tolerance of each element must be known. These values may be provided by 'TOLERANCE' cards which have the following format

TOL = 'ELEMENT NAME' = 'VALUE'

where 'ELEMENT NAME' is the name of a circuit element, and 'VALUE' is the relative tolerance $\left(\frac{\Delta Q}{Q}\right)$ for this element. If no 'TOLERANCE' card is provided for an element, a default value of .1 (10%) is assumed for the tolerance.

As well as a function of S giving the square of the worst case tolerance, sensitivity functions for all the elements are also printed. Thus no 'SENSITIVITY' cards may be present when a 'WORST' card follows the circuit description.

4. The Output Requests section may contain
   a. Transfer Function requests
   b. Plot requests. (optional)
   c. Roots Requests. (optional).

Any number of Transfer Function requests may appear and any number of Plot requests may be included following a single Transfer Function request Only one Roots request may appear after each Transfer Function request. The last card in the Output Requests section must be a card containing either the word END or the word EXECUTE.

a. The Transfer Function requests have the following format

TYPE NAME 1 / TYPE NAME 2

where TYPE NAME is the name of an element which has appeared in the Circuit Description, preceded by the letter V or I to denote whether Voltage or Current is desired. TYPE NAME 2 must be an Independent Source.

b. The plot request is a single card of the following form:

PLOT (Option = Value / Option = Value....)

The following options are available

1.	TYPE = IMPULSE	for impulse response
	STEP	for a step function
	SINE	for a sine wave input
	EXPONENTIAL	for an exponential input
	PULSE	for a pulse or pulse train
	FREQUENCY	for a frequency response
	SENS	for a sensitivity function plot
	WORST	for a worst case function plot
	REAL	for a transfer function plot with σ as the independent variable

2. AMPLITUDE = a number

represents the magnitude of any of the waveforms.

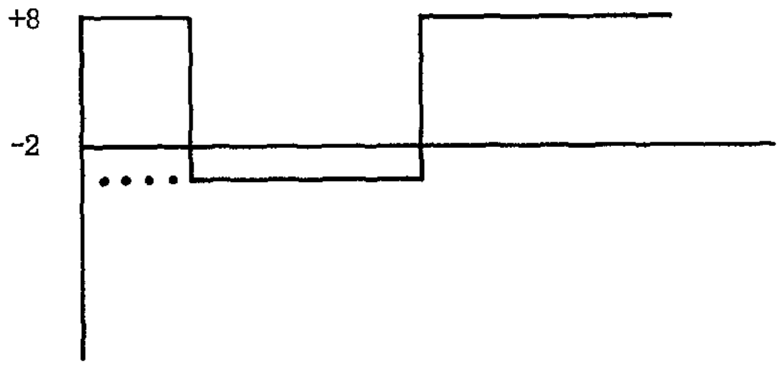
3. BIAS = a number

may be used only with PULSE to obtain a train of positive and negative pulses. It is combined with AMPLITUDE. e.g.,

AMPLITUDE = 10

BIAS = -2

gives



4. CONSTANT = a number

may be used only with EXPONENTIAL. It represents (a) in (Ke^{at}).

5. FREQUENCY = a number (in Hz/s.)

may be used only with SINE to specify the frequency of the sine wave.

6. DENSITY = a number

to specify the number of calculated points per plotted point.

7. TIME = a number (in Secs.)

should always appear with a time response. It specifies the duration of the plot.

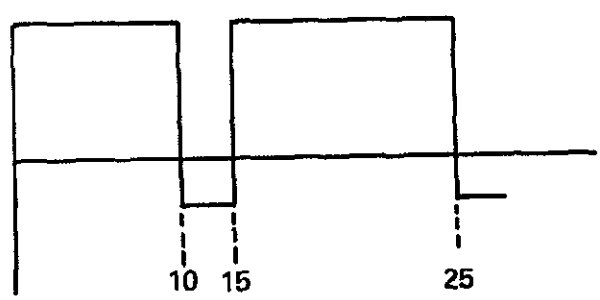
8. WIDTH = a number (in Secs.)

may be used only with PULSE to specify the duration of a pulse.

9. CYCLE = a number (in Secs.)

may be used only with PULSE to specify the time of 1 cycle.

e.g., WIDTH = 10
CYCLE = 15
gives:



10. STEP = a number (in Secs.)

specifies the time between individual calculated points. It may be used with any TYPE option.

11. FROM = a number, TO = a number

specifies the beginning and ending points of a frequency response (= $w/2\pi$), sensitivity, worst case or real plot. (= σ).

12. ELEMENT = an element name for which a sensitivity function has been requested may be used only with SENS to identify the sensitivity function to be plotted.

Only the first 2 letters of each word are used, therefore, AM may be used as an abbreviation for AMPLITUDE, IM for IMPULSE, etc.

Since continuation cards are not allowed, all the options for any one plot request must appear on one card. Not all the options are required since defaults are provided, e.g.:

PLOT(TIME = 1) produces an IMPULSE response of 100 steps and of duration 1 sec.

Defaults

TYPE = IMPULSE
TIME = 1 sec.
STEP = .01
AMPLITUDE = 1
DENSITY = 1
CONST = 1
FREQ = 1

The options may appear in any order on the Plot card. Inconsistent requests will be ignored, e.g., TYPE = IMPULSE, FREQ = 2 yields an Impulse response.

c. To obtain the roots of the numerator and denominator of the Transfer Function, a card with the word ROOTS must appear after the Transfer Function request.

The sensitivity of the poles of the transfer function may be obtained by adding the word 'POLES' to a 'ROOTS' card

ROOTS, POLES

The sensitivity of the Poles are produced for all elements for which a sensitivity request had been issued (or all elements if a 'WORST' card was included).

Several complete problems may run at one time. To terminate the execution of NASAP, a card containing the word STOP is used.

NASAP-70 can handle.

Up to 50 elements

Up to 50 circuit nodes

Up to 20 sensitivity requests

The circuit nodes should be numbered consecutively starting with 1 for best efficiency. This is not required, however. When more than 30 circuit elements are to be analyzed, the time required for the computation in Subroutine LOOPS begins to increase sharply. This effect is due to the presence of more than 2 words per block in Common Block BITS.

5.3 Overlay Organization

The recommended Overlay Structure is shown in Figure 5.1. This is not required for running NASAP but when included, it decreases the overall program size. However, the Overlay Structure is required to allow the program to fit into a computer with only a 32K word memory.

OVERLAY STRUCTURE FOR NASAP-70

```
                                                MAIN
 ---------------------------------------------------------------------------------------------------
         ASCAN               FINE               EQUAL        MULT            INVERT            ROOTS
         BSCAN               SCALER             LOR          SENSC   INPUT   ADJUST   SCALE
         CSCAN               POLSEN             LOX          ANSWER  SAMPLE  TAYLOR   ROUTH
         ISORT                                  LSTOR        WORSTC. FLIP    PRTPLT   CALCR
         MSG                                    UNPAK
         NUMBER                          CALC          LOOPS
         SHIFT                           INBIT         CLEAR
         SORT                            GRAPH         REDUCE
 NASAP   READ     WHAT
 SENSIT  UNITS    PLOT
 WORST
```

Figure 5 1

5.4 The Fortran AND and OR Functions

NASAP-70 requires the ability to perform a full word AND and OR on fixed point word variables in Fortran. This has been accomplished on the IBM 360 by means of the Fortran Logical .AND. and .OR. functions. The fixed point variables are given Logical type aliases and the AND or OR is performed on these logical type names. For example

```
LOGICAL A, B, C
EQUIVALENCE (A, I) (B, J), (C, K)
I = 12
J = 20
C = A.AND.B
```

would yield a value of 4 in K. On many other compilers a Logical type .AND. and .OR. does not operate on all the bits in the word. If NASAP-70 is compiled on such a system it will give incorrect results.

On systems which do not have the full word Logical .AND. and .OR., often another function is available for the same task. If no such facility is provided, the user must write his own AND and OR functions in Assembler Language. In either case statements of the type

```
C = A. AND.B
```

must be replaced by statements of the type

```
K = IAND (I, J)
```

This logical .AND. and .OR. is used in the following subprograms

Subroutine LOOPS

Function LOR

Function LOX

These are the only routines which are affected by the logical type AND. and .OR. but the user must ensure that these changes have been made before he attempts to run any NASAP-70 problems on a machine which is not IBM 360 compatible.

5.5 Compatibility Among Various Computers

Apart from the Fortran Logical type .AND. and .OR. functions described in the previous section, there are other facts about the IBM 360 Fortran compilers which could be a source of trouble on other computers.

The origins of Common Blocks on the IBM 360 are computed from the first variable in the block. On some compilers, however, the origin is computed from the last variable in the block, (e.g., IBM 7094). This allows occurrences of Common Blocks with the same name to have different lengths on the IBM 360. On compilers which compute the origin from the last variable (i.e., "backward addressing") all the occurrences of the same Common Block must be adjusted to the same length (with dummy variables if necessary) before compilation.

The test of the index in a DO loop is performed at the bottom of the DO. Thus in the following group of statements, the DO is executed once.

```
      N1 = 3
      N2 = 2
      DO 1 I = N1, N2
```

Further, in the case of the abnormal termination of a DO loop, the index variable retains its most recent value. NASAP has been programmed to avoid these problems with DO statements but they are mentioned here nevertheless to illustrate the way in which the IBM Fortran compilers handle special DO conditions.

Finally, a zero value for the index of a Fortran Computed GO TO is not treated as an error. Instead, execution proceeds with the statement following the GO TO. For example

```
      N = 0
      GO TO (1,2),N
      N = 2
1     N = 1-N
```

2 N = N + 7

would yield N = 6.

NASAP-70 is designed for IBM 360 compatible computers therefore some modification will be required if it is to be run on computers which are not IBM 360 compatible.

APPENDIX A

A. 1 PROGRAM LISTING

```
      BLOCK DATA
C        H.F.OKRENT           U.C.L.A.               1967
      INTEGER GAG
      COMMON/GAG/GAG(20),NSEN,SENS(50)
      COMMON/POLY/VN01(101,20),VN00(101,20),VN10(101,20),SMAX01(20),
     *SMIN01(20),SMAX00(20),SMIN00(20),SMAX10(20),SMIN10(20)
      COMMON/DATA/A(32)
      COMMON/SPEED/EMOD(2048)
      COMMON/BITS/F1(4450)
      COMMON/X/B(10)
      DATA GAG/2000,4000,8000,16000,32000,64000,128000,256000,512000,
     *1024000,2048000,4096000,8192000,16384000,32768000,65536000,
     *131072000,262144000,524288000,1048576000/
      DATA A,B/'C','I','R','L','E','J','D','/','(','V','F','S','T','O',
     1'P','N','.',' ','+','-',')','=','M','H','K',',','Y','A','X','U',
     2'B','W','0','1','2','3','4','5','6','7','8','9'/
      END
      INTEGER SMIN,SMAX,SMIN1,SMAX1,SMIN01,SMAX01,SMIN00,SMAX00,SMIN10,
     *SMAX10
      INTEGER TAG,FACTOR,SENS,GAG,SMAX2,SMIN2,S
      INTEGER GENER,TYPE,ORIGIN,TARGET,ERR,OUTPUT
      DIMENSION VN(101,2)
      DIMENSION SMIN(2),SMAX(2)
C
      COMMON/WORST1/NWORST,VW(101,2),TOL(50),SMAX2,SMIN2
      COMMON/GAG/GAG(20),NSEN,SENS(50)
      DIMENSION VNSEN(101,2,20),SMAX1(20),SMIN1(20)
      COMMON/TREE/ITREE
      COMMON/TAG/TAG
      COMMON/A/A(80)
      COMMON/DATA/C,AI,R,AL,E,AJ,D,AMOD(2),AV,F,AS,T,O,P,AN
     1,BMOD(11),AA,AX,WMOD(2),W
      COMMON/ERR/ERR,FACTOR
      COMMON/SPEED/V(1000),LS(1000),NLOOP
      COMMON/PATHS/LPATH(300,2),VPATH(300),S(300),NPATH
      COMMON/CIRCIT/CARD(50,12),TYPE(50),GENER(50),ORIGIN(50),TARGET(5
     10),INPUT,OUTPUT,NE,SMAX,SMIN
      COMMON/NTIMES/NTIMES
      COMMON/BITS/BITS(4450)
      DIMENSION DEP(50,13),MPATH(300,2)
      COMMON/POLY/POLY(6060),SMAX00(20),SMIN00(100)
      EQUIVALENCE (VNSEN(1,1,1),BITS(1)),(SMAX1(1),BITS(4041)),
     1(SMIN1(1),BITS(4061)),(DEP(1,1),BITS(3001)),(NNODES,BITS(3651)),
     2(INODE,BITS(3652)),(MPATH(1,1),BITS(3001))
      EQUIVALENCE (TYPE(1),VN(1,1))
    7 NE=0
      ITREE=0
      NNODES=0
      INODE=99
      NPATH=0
      ERR=0
```

```
      TAG=0
      NSEN=0
      NWORST=0
      DO 987 II=1,50
  987 SENS(II)=0
C          SKIP A PAGE
      WRITE(6,202)
  202 FORMAT('1****NASAP****'//'     NETWORK ANALYSIS AND SYSTEMS APPLICA
     1TION PROGRAM'//' THIS VERSION WAS DEVELOPED AT UCLA ENGR. DEPT.'//
     2/)
    1 READ(5,100)A
   11 NE=NE+1
      WRITE(6,200)A
  100 FORMAT(80A1)
  200 FORMAT(1X,80A1)
      CALL SHIFT(A)
      IF((A(1).NE.AN).OR.(A(2).NE.AA).OR.(A(3).NE.AS).OR.(A(4).NE.AA))
     1GO TO 2
      NE=0
      CALL NASAP
      GO TO 3
    2 IF((A(1).NE.T).OR.(A(2).NE.R).OR.(A(3).NE.E).OR.(A(4).NE.E))GO TO
     14
      NE=0
      CALL READ
      GO TO 3
    4 IF((A(1).NE.C).OR.(A(2).NE.AI).OR.(A(3).NE.R).OR.(A(4).NE.C))GO TO
     15
      NE=0
      ITREE=1
      CALL READ
      IF(ERR.EQ.0)CALL FINE
      GO TO 3
    5 IF((A(1).EQ.AS).AND.(A(2).EQ.T).AND.(A(3).EQ.O).AND.(A(4).EQ.P))ST
     1OP
C        PRINT   A    WARNING   AFTER    EVERY   10   HEADING    CARDS
      IF(NE-10)1,8,8
    8 WRITE(6,201)
  201 FORMAT(10X,'**** MORE THAN 10  TITLE  CARDS NOT ALLOWED--CHECK TO
     1SEE IF HEADING CARD WAS OMITTED.')
      NE=0
      GO TO 1
    3 IF(ERR.EQ.0)GO TO 6
    9 WRITE(6,204)
  204 FORMAT(//10X,'***** PROBLEM FLUSHED DUE TO EXCESSIVE ERRORS.')
      GO TO 7
    6 WRITE(6,203)
  203 FORMAT('1ELEMENT NUMBER   ELEMENT NAME   DEPENDENCY (IF ANY)   ORI
     1GIN NODE   TARGET NODE        VALUE              TAG        GENER'
     */1X,123('-')/)
      WRITE(6,211)(I,(CARD(I,J),J=1,12),(DEP(I,J),J=1,13),ORIGIN(I),TARG
     1ET(I),VPATH(I),TYPE(I),GENER(I),                    I=1,NE)
  211 FORMAT(1X,I10,8X,12A1,6X,13A1,5X,I7,I14,8X,1PE16.8, 9X,I1,10X,I1)
```

```
      CALL CALC                                                         UCLA1040
      IF(ERR.EQ.1)GØ TØ 9                                               UCLA1050
      CALL SCALER                                                       UCLA1060
      IF(FACTØR.NE.0)WRITE(6,409)FACTØR                                 UCLA1070
 409  FØRMAT(10X,'$$$$ AS A RESULT ØF AUTØMATIC SCALING,   THE   TRUE S UCLA1080
     1IS 10 TØ THE',I4,' PØWER TIMES THE PRØGRAM DEFINED S.')           UCLA1090
      IF (TAG.EQ.1)GØ TØ 301                                            UCLA1100
      READ(5,100)A                                                      UCLA1110
  301 CALL SHIFT(A)                                                     UCLA1120
      IF((A(1).NE.AS).ØR.(A(2).NE.E).ØR.(A(3).NE.AN).ØR.(A(4).NE.AS))   UCLA1130
     *GØ TØ 390                                                         UCLA1140
      CALL SENSIT                                                       UCLA1150
      GØ TØ 300                                                         UCLA1160
  390 IF((A(1).EQ.W).AND.(A(2).EQ.Ø).AND.(A(3).EQ.R).AND.(A(4).EQ.AS))  UCLA1170
     *CALL WØRST                                                        UCLA1180
  300 CALL GRAPH                                                        UCLA1190
      WRITE(6,250)                                                      UCLA1200
 250  FØRMAT('1FLØWGRAPH'/' FRØM      ',' TØ',8X,'S',9X,'VALUE')        UCLA1210
      DØ 521 I=1,NPATH                                                  UCLA1220
      CALL REDUCE(S(I),II,K)                                            UCLA1230
  521 WRITE(6,251)LPATH(I,1),LPATH(I,2),II,VPATH(I)                     UCLA1240
 251  FØRMAT(1X,I3,2I9,1PE20.8)                                         UCLA1250
      NPATH=NPATH+1                                                     UCLA1260
      VPATH(NPATH)=1                                                    UCLA1270
      S(NPATH)=1000                                                     UCLA1280
      NTIMES=(((2*NE)-1)/30)+1                                          UCLA1290
      INP=0                                                             UCLA1300
      ERR=0                                                             UCLA1310
      TAG=1                                                             UCLA1315
      GØ TØ 10                                                          UCLA1320
 17   INP=0                                                             UCLA1330
      ERR=0                                                             UCLA1340
 14   READ(5,100)A                                                      UCLA1350
 10   WRITE(6,205)A                                                     UCLA1360
 205  FØRMAT('1',80A1)                                                  UCLA1370
 19   IF(TAG.EQ.0)CALL SHIFT(A)                                         UCLA1380
      TAG=0                                                             UCLA1385
      IF((A(1).EQ.E).AND.(A(2).EQ.AN).AND.(A(3).EQ.D))GØ TØ 7           UCLA1390
      IF((A(1).EQ.E).AND.(A(2).EQ.AX))GØ TØ 7                           UCLA1400
      IF((A(1).EQ.C).AND.(A(2).EQ.AI).AND.(A(3).EQ.R).AND.(A(4).EQ.C))  UCLA1410
     1GØ TØ 12                                                          UCLA1420
      IF((A(1).EQ.T).AND.(A(2).EQ. R).AND.(A(3).EQ.E).AND.(A(4).EQ.E))  UCLA1430
     1GØ TØ 12                                                          UCLA1440
      IF((A(1).EQ.AN).AND.(A(2).EQ.AA).AND.(A(3).EQ.AS).AND.(A(4).EQ.AA)UCLA1450
     1)GØ TØ 12                                                         UCLA1460
      IF((A(1).EQ.AS).AND.(A(2).EQ.T).AND.(A(3).EQ.Ø).AND.(A(4).EQ.P))  UCLA1470
     1STØP                                                              UCLA1480
      IF(INP.EQ.0)GØ TØ 13                                              UCLA1490
      IF((A(1).EQ.AV).ØR.(A(1).EQ.AI))GØ TØ 16                          UCLA1495
      IF((A(1).NE.P).ØR.(A(2).NE.AL).ØR.(A(3).NE.Ø).ØR.(A(4).NE.T).ØR.(EUCLA1500
     1RR.NE.0))GØ TØ 20                                                 UCLA1510
      CALL PLØT                                                         UCLA1520
      IF(ERR.EQ.0)CALL INVERT                                           UCLA1530
```

```
      ERR=0                                                             UCLA1540
 20   IF((A(1).EQ.R).AND.(A(2).EQ.O).AND.(A(3).EQ.O).AND.(A(4).EQ.T).ANDUCLA1550
     1.(ERR.EQ.0))CALL ROOTS                                            UCLA1560
      CALL SHIFT(A)                                                     UCLA1570
      IF((A(1).EQ.P).AND.(A(2).EQ.O).AND.(A(3).EQ.AL).AND.(A(4).EQ.E)   UCLA1580
     *.AND.(ERR.EQ.0))CALL POLSEN                                       UCLA1590
      IF((A(1).NE.W).OR.(A(2).NE.O).OR.(A(3).NE.R).OR.(A(4).NE.AS))     UCLA1610
     *GO TO 610                                                         UCLA1620
      WRITE(6,302)                                                      UCLA1630
  302 FORMAT(/10X,'**** SENSITIVITY OR WORST CASE REQUESTS MUST IMMEDIATUCLA1640
     *ELY FOLLOW THE CIRCUIT DESCRIPTION---REQUEST IGNORED.')           UCLA1650
      GO TO 14                                                          UCLA1660
  610 IF((A(1).NE.AS).OR.(A(2).NE.E).OR.(A(3).NE.AN).OR.(A(4).NE.AS))   UCLA1670
     *GO TO 611                                                         UCLA1680
      WRITE(6,302)                                                      UCLA1690
      GO TO 14                                                          UCLA1700
  611 IF((A(1).NE.T).OR.(A(2).NE.O).OR.(A(3).NE.AL)) GO TO 14           UCLA1710
      WRITE(6,312)                                                      UCLA1720
  312 FORMAT(/10X,'**** TOLERANCE CARDS MUST IMMEDIATELY FOLLOW THE WORSUCLA1730
     *T CASE CARD--CARD IGNORED.')                                      UCLA1740
      GO TO 14                                                          UCLA1750
 12   WRITE(6,210)                                                      UCLA1760
 210  FORMAT(/10X,'**** MISSING ''END OF PROBLEM'' CARD SIMULATED BY PROUCLA1770
     1GRAM.')                                                           UCLA1780
      WRITE(6,202)                                                      UCLA1790
      GO TO 11                                                          UCLA1800
 13   IF((A(1).NE.P).OR.(A(2).NE.AL).OR.(A(3).NE.O).OR.(A(4).NE.T))GO TOUCLA1810
     115                                                                UCLA1820
      WRITE(6,206)                                                      UCLA1830
  206 FORMAT(/10X,'**** PLOT, ROOT AND POLE SENSITIVITY REQUESTS MUST FOUCLA1840
     *LLOW THEIR CORRESPONDING OUTPUT REQUESTS.'/15X,'REQUEST IGNORED.')UCLA1850
      GO TO 14                                                          UCLA1860
 15   IF((A(1).NE.R).OR.(A(2).NE.O).OR.(A(3).NE.O).OR.(A(4).NE.T))GO TO UCLA1870
     *303                                                               UCLA1880
      WRITE(6,206)                                                      UCLA1890
      GO TO 14                                                          UCLA1900
  303 IF((A(1).NE.P).OR.(A(2).NE.O).OR.(A(3).NE.AL).OR.(A(4).NE.E))     UCLA1910
     *GO TO 393                                                         UCLA1920
      WRITE(6,206)                                                      UCLA1930
      GO TO 14                                                          UCLA1940
  393 IF((A(1).NE.AS).OR.(A(2).NE.E).OR.(A(3).NE.AN).OR.(A(4).NE.AS))   UCLA1950
     *GO TO 304                                                         UCLA1960
      WRITE(6,302)                                                      UCLA1970
      GO TO 14                                                          UCLA1980
  304 IF((A(1).NE.T).OR.(A(2).NE.O).OR.(A(3).NE.AL)) GO TO 315          UCLA1990
      WRITE(6,312)                                                      UCLA2000
      GO TO 14                                                          UCLA2010
  315 IF((A(1).NE.W).OR.(A(2).NE.O).OR.(A(3).NE.R).OR.(A(4).NE.AS))     UCLA2020
     *GO TO 16                                                          UCLA2030
      WRITE(6,302)                                                      UCLA2040
      GO TO 14                                                          UCLA2050
 16   INP=1                                                             UCLA2060
      WRITE(6,207)NTIMES                                                UCLA2070
```

```
 207 FORMAT(//1X,'NTIMES=',I4)
     CALL WHAT
     IF(ERR.EQ.1)GO TO 17
     DO18I=1,NPATH
     CALL INBIT(I,0,-NTIMES)
     CALL INBIT(I,MPATH(I,1),-NTIMES)
 18  CALL INBIT(I,MPATH(I,2),-NTIMES)
     CALL LOOPS
     DO99I=1,4040
 99  BITS(I)=0.0
     DO88I=1,20
     SMAX1(I)=0
 88  SMIN1(I)=0
     NSMIN=MIN0(SMIN(1),SMIN(2))
     NSMAX=MAX0(SMAX(1),SMAX(2))
     CALL ANSWER(VN,NSMAX,NSMIN,0)
     IF( SEN.EQ.0)GO TO 14
     CALL SENSC
     DO21I=1,4040
 21  POLY(I)=BITS(I)
     DO22I=1,20
     SMAXOO(I)=SMAX1(I)
 22  SMINOO(I)=SMIN1(I)
     IF(NWORST.EQ.0)GO TO 14
     CALL WORSTC
     CALL ANSWER(VW,SMAX2,SMIN2,-1)
     GO TO 14
     END
     SUBROUTINE ASCAN(B,I,J)
C        H.F.OKRENT           U.C.L.A.               1967
     REAL LEFT,MINUS
     COMMON/DATA/AMOD(7),SLASH,LEFT,BMOD(8),BLANK,PLUS,MINUS,RIGHT,EQU
     COMMON/A/A(80)
     DO1K=I,80
     J=K-1
     B=A(K)
     IF(A(K).EQ.LEFT)RETURN
     IF(A(K).EQ.SLASH)RETURN
     IF(A(K).EQ.MINUS)RETURN
     IF(A(K).EQ.RIGHT)RETURN
     IF(A(K).EQ.EQU)RETURN
 1   IF(A(K).EQ.BLANK)RETURN
     RETURN
     END
     SUBROUTINE BSCAN(B,I,J)
     COMMON/A/A(80)
C        H.F.OKRENT           U.C.L.A.               1967
     COMMON/DATA/C,AI,AMOD(7),AV,FMOD(3),B,P,BMOD(2),BLANK,CMOD(4),
    1AN,H AK,DMOD(4),U
     DO1K=I,80
     J=K-1
     B=A(K)
     IF(A(K).EQ.U)RETURN
```

A-5

```
      IF(A(K).EQ.BLANK)RETURN                                           UCLA2610
      IF(A(K).EQ.AM)RETURN                                              UCLA2620
      IF(A(K).EQ.AK)RETURN                                              UCLA2630
      IF(A(K).EQ.P)RETURN                                               UCLA2640
      IF(A(K).EQ.Ø)RETURN                                               UCLA2650
      IF(A(K).EQ.AV)RETURN                                              UCLA2660
    1 IF(A(K).EQ.AI)RETURN                                              UCLA2670
      RETURN                                                            UCLA2680
      END                                                               UCLA2690
      SUBRØUTINE CSCAN(B,I,J)                                           UCLA2700
C        H.F.ØKRENT           U.C.L.A.                  1967            UCLA2710
      CØMMØN/A/A(80)                                                    UCLA2720
      REAL LEFT,MINUS                                                   UCLA2730
      CØMMØN/DATA/C,AI,AMØD(2),E,BMØD(8),Ø,CMØD(2),PNT,BLANK,PLUS,MINUS UCLA2740
      DØ1K=I,80                                                         UCLA2750
      J=K-1                                                             UCLA2760
      B=A(K)                                                            UCLA2770
      IF(A(K).EQ.PNT)RETURN                                             UCLA2780
      IF(A(K).EQ.E)RETURN                                               UCLA2790
      IF(A(K).EQ.PLUS)RETURN                                            UCLA2800
      IF(A(K).EQ.MINUS)RETURN                                           UCLA2810
    1 IF(A(K).EQ.BLANK)RETURN                                           UCLA2820
      RETURN                                                            UCLA2830
      END                                                               UCLA2840
      FUNCTIØN ISØRT(A)                                                 UCLA2850
C        H.F.ØKRENT           U.C.L.A.                  1967            UCLA2860
      INTEGER FLAG                                                      UCLA2870
      CØMMØN/FLAG/FLAG                                                  UCLA2880
      CØMMØN/X/B(10)                                                    UCLA2890
      FLAG=0                                                            UCLA2900
      DØ1I=1,10                                                         UCLA2910
      ISØRT=I-1                                                         UCLA2920
    1 IF(A.EQ.B(I))RETURN                                               UCLA2930
      ISØRT=0                                                           UCLA2940
      FLAG=1                                                            UCLA2950
      RETURN                                                            UCLA2960
      END                                                               UCLA2970
      SUBRØUTINE MSG(A,I)                                               UCLA2980
C        H.F.ØKRENT           U.C.L.A.                  1967            UCLA2990
      WRITE(6,200)I,A                                                   UCLA3000
  200 FØRMAT(10X,'**** CHARACTER',I3,' WAS EXPECTED TØ BE A ',A2,'. CARDUCLA3010
     1 IGNØRED.')                                                       UCLA3020
      RETURN                                                            UCLA3030
      END                                                               UCLA3040
      SUBRØUTINE NUMBER(N1,N2)                                          UCLA3050
C        H.F.ØKRENT           U.C.L.A.                  1967            UCLA3060
      CØMMØN/A/A(80)                                                    UCLA3070
      CØMMØN/DATA/AMØD(4),E,BMØD(11),PNT,BLANK,PLUS,MINUS,RIGHT,EQU     UCLA3080
      REAL MINUS                                                        UCLA3090
      INTEGER TAG,FLAG,EXPØ,PØW                                         UCLA3100
      CØMMØN/FLAG/FLAG                                                  UCLA3110
      CØMMØN/CIRCIT/CMØD(802),NE                                        UCLA3120
      CØMMØN/PATHS/LPATH(300,2),VPATH(300),S(300),NPATH                 UCLA3130
```

```
      K=N1                                                              UCLA3140
      TAG=0                                                             UCLA3150
      VPATH(NE)=0.                                                      UCLA3160
      POW=0                                                             UCLA3170
      IF(K.LE.N2)GO TO 1                                                UCLA3180
    4 IF(VPATH(NE).LT.1.E-30)VPATH(NE)=1.0                              UCLA3190
      IF(TAG.EQ.1)VPATH(NE)=-VPATH(NE)                                  UCLA3200
      WRITE(6,200)K,VPATH(NE)                                           UCLA3210
  200 FORMAT(10X,        '**** VALUE CANNOT BE COMPLETELY DETERMINED DUEUCLA3220
     1 TO AN ERROR IN CHARACTER',I3/10X,'**** VALUE ASSUMED TO BE',E16.8UCLA3230
     1)                                                                 UCLA3240
      RETURN                                                            UCLA3250
    1 I=K                                                               UCLA3260
C        CHECK      FOR     +    OR    -     SIGNS                      UCLA3270
      IF(A(I).EQ.MINUS)TAG=1                                            UCLA3280
      IF((A(I).EQ.MINUS).OR.(A(I).EQ.PLUS))I=I+1                        UCLA3290
C        SCAN     FOR     +,    -,    E,    .,     BLANK                UCLA3300
      CALL CSCAN(B,I,J)                                                 UCLA3310
      IF(J.GT.N2)J=N2                                                   UCLA3320
      IF(J.LT.I)GO TO 2                                                 UCLA3330
C        INSERT    NUMBERS   GREATER    THAN      1.0                   UCLA3340
      DO3K=I,J                                                          UCLA3350
      VPATH(NE)=VPATH(NE)+(SORT(A(K))*(10.**(J-K)))                     UCLA3360
      IF(FLAG.EQ.1)GO TO 4                                              UCLA3370
    3 CONTINUE                                                          UCLA3380
C        CHECK    FOR   THE    END    OF     THE      NUMBER    FIELD   UCLA3390
    2 IF(J.GE.N2)GO TO 5                                                UCLA3400
C        HANDLE     NUMBERS      LESS    THAN    1.0                    UCLA3410
      IF(B.NE.PNT)GO TO 6                                               UCLA3420
      I=J+2                                                             UCLA3430
      CALL CSCAN(B,I,J)                                                 UCLA3440
      IF(J.GT.N2)J=N2                                                   UCLA3450
      IF(J.LT.I)GO TO 8                                                 UCLA3460
      DO 7 K=I,J                                                        UCLA3470
      VPATH(NE)=VPATH(NE)+(SORT(A(K))*(10.**(I-K-1)))                   UCLA3480
      IF(FLAG.EQ.1)GO TO 4                                              UCLA3490
    7 CONTINUE                                                          UCLA3500
    8 IF(J.GE.N2)GO TO 5                                                UCLA3510
    6 IF(B.EQ.E)J=J+1                                                   UCLA3520
C        EVALUATE     THE     EXPONENT                                  UCLA3530
      EXPO=0                                                            UCLA3540
      IF(A(J+1).EQ.MINUS)EXPO=1                                         UCLA3550
      IF((A(J+1).NE.PLUS).AND.(A(J+1).NE.MINUS))J=J-1                   UCLA3560
      I=J+2                                                             UCLA3570
      CALL BSCAN(B,I,K)                                                 UCLA3580
      IF(K.GT.N2)K=N2                                                   UCLA3590
      IF(K-I.LE.1)GO TO 9                                               UCLA3600
      WRITE(6,201)                                                      UCLA3610
  201 FORMAT(10X,'**** EXPONENTS WITH MORE THAN 2 DIGITS ARE NOT ALLOWEDUCLA3620
     1. ONLY THE LAST 2 WILL BE USED.')                                 UCLA3630
      I=K-1                                                             UCLA3640
    9 POW=ISORT(A(K))                                                   UCLA3650
      IF(FLAG.EQ.1)GO TO 4                                              UCLA3660
```

```
      IF(K.EC.1)GO TO 10                                                UCLA3670
      K=I                                                               UCLA3680
      PBW=PBW+(10*ISORT(A(K)))                                          UCLA3690
      IF(FLAG.EG.1)GO TO 4                                              UCLA3700
 10   IF(TAPP.EQ.1)PBW=-PBW                                             UCLA3710
      VPATH(NE)=VPATH(NE)*(10.**PBW)                                    UCLA3720
 5    IF(TAG.EQ.1)VPATH(NE)=-VPATH(NE)                                  UCLA3730
      RETURN                                                            UCLA3740
      END                                                               UCLA3750
      SUBROUTINE SHIFT(A)                                               UCLA3760
C       H.F.BKRENT           U.C.L.A.              1967                 UCLA3770
      DIMENSION A(80)                                                   UCLA3780
      COMMON/DATA/AMOD(17),BLANK,BMOD(7),COMMA                          UCLA3790
      K=0                                                               UCLA3800
      DO1I=1,80                                                         UCLA3810
      J=I-K                                                             UCLA3820
      A(J)=A(I)                                                         UCLA3830
      IF(A(I).EQ.BLANK)K=K+1                                            UCLA3840
 1    IF(A(I).EQ.COMMA)K=K+1                                            UCLA3850
      IF(K.EQ.0)K=1                                                     UCLA3860
      A(81-K)=BLANK                                                     UCLA3870
      RETURN                                                            UCLA3880
      END                                                               UCLA3890
      FUNCTION SORT(A)                                                  UCLA3900
C       H.F.BKRENT           U.C.L.A.              1967                 UCLA3910
      SORT=ISORT(A)                                                     UCLA3920
      RETURN                                                            UCLA3930
      END                                                               UCLA3940
      SUBROUTINE NASAP                                                  UCLA3950
C       H.F.BKRENT            U.C.L.A.                 1969             UCLA3960
      INTEGER TYPE,GENER                                                UCLA3970
      INTEGER FLAG,ERR,ORIGIN,TARGET                                    UCLA3980
      COMMON/PATHS/LPATH(300,2),VPATH(300),S(300),FREQ                  UCLA3990
      COMMON/A/A(80)                                                    UCLA4000
      COMMON/CIRCIT/CARD(50,12),TYPE(50),GENER(50),ORIGIN(50),TARGET(50)UCLA4010
     1,INPUT,OUTPUT,NE,SMAX,SMIN                                        UCLA4020
      COMMON/DATA/C,AI,AMOD(2),E,BMOD(4),AV,FMOD(3),O,P,CMOD(2),        UCLA4030
     1BLANK,                                                            UCLA4040
     1DMOD(4),AM,H,AK,EMOD(4),U                                         UCLA4050
      COMMON/FLAG/FLAG                                                  UCLA4060
      COMMON/ERR/ERR                                                    UCLA4070
      COMMON/BITS/N(3000),DEP(50,13),NNODES,INODE                       UCLA4080
      NE=0                                                              UCLA4090
 18   READ(5,100)A                                                      UCLA4100
      WRITE(6,200)A                                                     UCLA4110
 100  FORMAT(80A1)                                                      UCLA4120
 200  FORMAT(1X,80A1)                                                   UCLA4130
      I=1                                                               UCLA4140
 2    CALL ASCAN(B,I,J)                                                 UCLA4150
      IF(J.NE.I-1)GO TO 1                                               UCLA4160
      I=J+2                                                             UCLA4170
      IF(I.GT.72)GO TO 6                                                UCLA4180
      GO TO 2                                                           UCLA4190
```

```
1     IF(A(I).EQ.E)GØ TØ 17                                          UCLA4200
      IF(A(I).EQ.S)GØ TØ 17                                          UCLA4210
      NE=NE+1                                                        UCLA4220
      DØ3K=1,12                                                      UCLA4230
      L=I+K-1                                                        UCLA4240
      CARD(NE,K)=A(L)                                                UCLA4250
      IF(L.GT.J)CARD(NE,K)=BLANK                                     UCLA4260
3     CØNTINUE                                                       UCLA4270
      I=J+2                                                          UCLA4280
4     CALL ASCAN(B,I,J)                                              UCLA4290
      IF(J.NE.I-1)GØ TØ 5                                            UCLA4300
      I=J+2                                                          UCLA4310
      IF(I.GT.72)GØ TØ 6                                             UCLA4320
      GØ TØ 4                                                        UCLA4330
5     ØRIGIN(NE)=ISØRT(A(J))                                         UCLA4340
      IF(FLAG.EQ.1)GØ TØ 6                                           UCLA4350
      IF(J.NE.I)ØRIGIN(NE)=ØRIGIN(NE)+(ISØRT(A(I))*10)               UCLA4360
      IF(FLAG.EQ.1)GØ TØ 6                                           UCLA4370
      I=J+2                                                          UCLA4380
8     CALL ASCAN(B,I,J)                                              UCLA4390
      IF(J.NE.I-1)GØ TØ 7                                            UCLA4400
      I=J+2                                                          UCLA4410
      IF(I.GT.72)GØ TØ 6                                             UCLA4420
      GØ TØ 8                                                        UCLA4430
7     TARGET(NE)=ISØRT(A(J))                                         UCLA4440
      IF(FLAG.EQ.1)GØ TØ 6                                           UCLA4450
      IF(J.NE.I)TARGET(NE)=TARGET(NE)+(ISØRT(A(I))*10)               UCLA4460
      IF(FLAG.EQ.1)GØ TØ 6                                           UCLA4470
      I=J+2                                                          UCLA4480
10    CALL BSCAN(B,I,J)                                              UCLA4490
      IF(J.NE.I-1)GØ TØ 9                                            UCLA4500
      IF(I.GT.72)GØ TØ 6                                             UCLA4510
      I=J+2                                                          UCLA4520
      GØ TØ 10                                                       UCLA4530
9     K=72                                                           UCLA4540
      CALL NUMBER(I,J)                                               UCLA4550
14    IF(B.EQ.BLANK)GØ TØ 11                                         UCLA4560
      IF(B.EQ.U)VPATH(NE)=VPATH(NE)*1.E-6                            UCLA4570
      IF(B.EQ.P)VPATH(NE)=VPATH(NE)*1.E-12                           UCLA4580
      IF(B.EQ.AK)VPATH(NE)=VPATH(NE)*1.E3                            UCLA4590
      IF((B.EQ.AM).AND.(A(J+2).NE.H))VPATH(NE)=VPATH(NE)*1.E6        UCLA4600
      IF((B.EQ.AM).AND.(A(J+2).EQ.H))VPATH(NE)=VPATH(NE)*1.E-3       UCLA4610
11    I=J+2                                                          UCLA4620
      IF((B.EQ.AV).ØR.(B.EQ.AI))GØ TØ 16                             UCLA4630
      IF(I.GT.72)GØ TØ 12                                            UCLA4640
13    CALL BSCAN(B,I,J)                                              UCLA4650
      GØ TØ 14                                                       UCLA4660
16    CALL ASCAN(B,I,K)                                              UCLA4670
      I=K+2                                                          UCLA4680
      IF(B.EQ.BLANK)GØ TØ 12                                         UCLA4690
      IF(I.GT.72)GØ TØ 12                                            UCLA4700
      GØ TØ 16                                                       UCLA4710
12    DØ15I=1,13                                                     UCLA4720
```

```
      L=I+J
      DEP(NE,I)=A(L)
      IF(L.GT.K)DEP(NE,I)=BLANK
   15 CØNTINUE
      GENER(NE)=0
      TYPE(NE)=0
      IF((CARD(NE,1).EQ. V).ØR.(CARD(NE,1).EQ.AI))GENER(NE)=1
      IF(CARD(NE,1).EQ.A )TYPE(NE)=1
      NNØDES=MAX0(NNØDES,TARGET(NE),ØRIGIN(NE))
      INØDE=MIN0( INØDE,TARGET(NE),ØRIGIN(NE))
      GØ TØ 18
   17 DØ32I=INØDE,NNØDES
   32 N(I)=0
      DØ19I=1,NE
      J=ØRIGIN(I)
      N(J)=N(J)+1
      J=TARGET(I)
   19 N(J)=N(J)+1
      J=0
      DØ20I=INØDE,NNØDES
   20 J=MAX0(J,N(I))
   28 DØ21I=INØDE,NNØDES
      IF(J.EQ.N(I))GØ TØ 22
   21 CØNTINUE
   22 N(I)=0
      M=0
      DØ23J=1,NE
      IF (GENER(J)*TYPE(J).EQ.0)GØ TØ 23
      K=ØRIGIN(J)
      L=TARGET(J)
      IF(N(K).GT.0)GØ TØ 36
      IF(N(L).GT.0)M=1
      N(L)=-IABS(N(L))
   36 IF(N(L).GT.0)GØ TØ 23
      IF(N(K).GT.0)M=1
      N(K)=-IABS(N(K))
   23 CØNTINUE
      IF(M.EQ.1)GØ TØ 22
      DØ26J=1,NE
      IF(GENER(J)+TYPE(J).NE.0)GØ TØ 26
      K=ØRIGIN(J)
      L=TARGET(J)
      IF(K.NE.I)GØ TØ 27
      IF(N(L).LE.0)GØ TØ 27
      N(L)=-N(L)
      GØ TØ 34
   27 IF(L.NE.I)GØ TØ 26
      IF(N(K).LE.0)GØ TØ 26
      N(K)=-N(K)
   34 TYPE(J)=1
   37 M=0
      DØ31J1=1,NE
      IF(TYPE(J1)+GENER(J1).NE.2)GØ TØ 31
```

```
      K=ORIGIN(J1)                                                      UCLA5260
      L=TARGET(J1)                                                      UCLA5270
      IF(N(K)*N(L).GT.0)GO TO 31                                        UCLA5280
      IF(N(K).GE.0)GO TO 33                                             UCLA5290
      IF(N(L).GT.0)M=1                                                  UCLA5300
      N(L)=-IABS(N(L))                                                  UCLA5310
 33   IF(N(L).GE.0)GO TO 31                                             UCLA5320
      IF(N(K).GT.0)M=1                                                  UCLA5330
      N(K)=-IABS(N(K))                                                  UCLA5340
 31   CONTINUE                                                          UCLA5350
 26   CONTINUE                                                          UCLA5360
 35   J=0                                                               UCLA5370
      DO29I=INODE,NNODES                                                UCLA5380
 29   J=MINO(J,N(I))                                                    UCLA5390
      IF(J.EQ.0)GO TO 30                                                UCLA5400
      IF(J.LT.-1000)GO TO 28                                            UCLA5410
 24   DO25I=INODE,NNODES                                                UCLA5420
      IF(N(I).LT.0)N(I)=N(I)-1000                                       UCLA5430
 25   CONTINUE                                                          UCLA5440
      GO TO 35                                                          UCLA5450
 6    WRITE(6,201)J                                                     UCLA5460
 201  FORMAT(' INPUT CODING ERROR NEAR COLUMN ',I3)                     UCLA5470
      ERR=1                                                             UCLA5480
 30   RETURN                                                            UCLA5490
      END                                                               UCLA5500
      SUBROUTINE SENSIT                                                 UCLA5510
C                                                                       UCLA5520
C        S. GRANDI   U.C.L.A.    1969                                   UCLA5530
C        PROCESSES SENSITIVITY REQUESTS                                 UCLA5540
C                                                                       UCLA5550
      INTEGER TAG,SENS                                                  UCLA5560
      DIMENSION SMIN(2),SMAX(2)                                         UCLA5570
      COMMON/A/A(80)                                                    UCLA5580
      COMMON/CIRCIT/CARD(50,12),TYPE(50),GENER(50),ORIGIN(50),          UCLA5590
     *TARGET(50),INPUT     ,OUTPUT,NE,SMAX,SMIN                         UCLA5600
      COMMON/DATA/AMOD(4),E,BMOD(6),AS,CMOD(3),AN,DMOD,BLANK,EMOD(3),EQUUCLA5610
      COMMON/JAG/TAG(20),NSEN,SENS(50)                                  UCLA5620
      COMMON/BITS/LBITS(3000),DEP(50,13),NNODES,INODE                   UCLA5630
      DATA LIMIT/20/                                                    UCLA5640
C        INITIALIZE                                                     UCLA5650
      NE1=0                                                             UCLA5660
      WRITE (6,106)A                                                    UCLA5670
  106 FORMAT('1',80A1)                                                  UCLA5680
      GO TO 1                                                           UCLA5690
C        IF NEXT CARD IS NOT SENSITIVITY CARD, RETURN                   UCLA5700
    2 READ(5,100)A                                                      UCLA5710
  100 FORMAT(80A1)                                                      UCLA5720
      CALL SHIFT(A)                                                     UCLA5730
      IF((A(1).NE.AS).OR.(A(2).NE.E).OR.(A(3).NE.AN).OR.(A(4).NE.AS))   UCLA5740
     *RETURN                                                            UCLA5750
    1 WRITE(6,200)A                                                     UCLA5760
  200 FORMAT(' ',80A1)                                                  UCLA5770
C        SCAN FOR EQUALS SIGN AFTER SENS                                UCLA5780
```

```
    1 CALL ASCAN(B,1,J)
      IF (B.EQ.EQU)GØ TØ 3
      WRITE(6,102)
  102 FØRMAT (/10X,'**** EQUALS SIGN NØT FØUND--REQUEST IGNØRED.')
      GØ TØ 2
C        FIND CIRCUIT ELEMENT
    3 K=J+2
      L=K+11
      DØ 4 I=1,NE
      DØ 5 I1=K,L
      I2=I1-J-1
      IF (A(I1).NE.CARD(I,I2))GØ TØ 4
    5 CØNTINUE
C        CHECK FØR INVALID ELEMENT (INDEPENDENT SØURCE)
      IF(GENER(I).EQ.0)GØ TØ 9
      IF(DEP(I,1).NE.BLANK)GØ TØ 9
      WRITE(6,105)
  105 FØRMAT(/10X,'**** INVALID ELEMENT TYPE--REQUEST IGNØRED.')
      GØ TØ 2
C        CHECK FØR TØØ MANY REQUESTS
    9 NE1=NE1+1
      IF(NE1.LE.LIMIT)GØ TØ 10
      WRITE(6,101)
  101 FØRMAT (/10X,'**** TØØ MANY SENSITIVITY REQUESTS--REQUEST IGNØRED.
     *')
      GØ TØ 2
C        CHECK FØR DUPLICATE REQUESTS
   10 IF(SENS(I).EQ.0)GØ TØ 11
      WRITE(6,104)
  104 FØRMAT(/10X,'**** DUPLICATE REQUEST--REQUEST IGNØRED.')
      GØ TØ 2
C        INSERT SENSITIVITY TAGS
   11 NSEN=NSEN+1
      SENS(I)=TAG(NSEN)
      GØ TØ 2
    4 CØNTINUE
C        ELEMENT NØT FØUND IN CIRCUIT
      WRITE(6,103)
  103 FØRMAT (/10X,'**** SENSITIVITY REQUEST FØR UNKNØWN ELEMENT--REQUES
     *T IGNØRED.')
      GØ TØ 2
      END
      SUBRØUTINE WØRST
C
C        S. GRANDI    U.C.L.A.    1969
C        PRØCESSES WØRST CASE REQUESTS
C
      INTEGER TAG,SENS
      DIMENSION SMIN(2),SMAX(2)
      CØMMØN/A/A(80)
      CØMMØN/CIRCIT/CARD(50,12),TYPE(50),GENER(50),ØRIGIN(50),
     *TARGET(50),INPUT,ØUTPUT,NE
      CØMMØN/GAG/TAG(20),NSEN,SENS(50)
```

```
      COMMON/BITS/LBITS(3000),DEP(50,13)                                UCLA6320
      COMMON/WORST1/NWORST,VW(101,2),TOL(50)                            UCLA6330
      COMMON/DATA/AMOD(2),R,AL,E,BMOD(6),S,T,O,CMOD,AN,FMOD,BLANK,      UCLA6340
     *DMOD(3),EQU,EMOD(9),W                                             UCLA6350
      COMMON/PATHS/LPATH(600),VPATH(300)                                UCLA6360
      DATA LIMIT/20/                                                    UCLA6370
C         INITIALIZE                                                    UCLA6380
      NFLAG=0                                                           UCLA6390
      WRITE(6,100)A                                                     UCLA6400
  100 FORMAT('1',80A1)                                                  UCLA6410
C         INSERT SENSITIVITY TAGS FOR ALL VALID ELEMENTS                UCLA6420
      DO 1 I=1,NE                                                       UCLA6430
      IF(GENER(I).EQ.0)GO TO 2                                          UCLA6440
      IF(DEP(I,1).NE.BLANK)GO TO 2                                      UCLA6450
      GO TO 1                                                           UCLA6460
    2 NSEN=NSEN+1                                                       UCLA6470
      IF(NSEN.GT.LIMIT)GO TO 3                                          UCLA6480
      SENS(I)=TAG(NSEN)                                                 UCLA6490
    1 CONTINUE                                                          UCLA6500
      NWORST=1                                                          UCLA6510
      GO TO 11                                                          UCLA6520
C         TOO MANY ELEMENTS                                             UCLA6530
    3 NSEN=NSEN-1                                                       UCLA6540
      WRITE(6,101)                                                      UCLA6550
  101 FORMAT(/10X,'**** TOO MANY ELEMENTS FOR A WORST CASE ANALYSIS--REQUCLA6560
     *UEST IGNORED')                                                    UCLA6570
      NFLAG=1                                                           UCLA6580
      GO TO 10                                                          UCLA6590
C         SET ALL TOLERANCES TO .1 DEFAULT                              UCLA6600
   11 DO 4 I=1,NE                                                       UCLA6610
    4 TOL(I)=.1                                                         UCLA6620
C         READ TOLERANCE CARDS                                          UCLA6630
   10 READ(5,103)A                                                      UCLA6640
  103 FORMAT(80A1)                                                      UCLA6650
      CALL SHIFT(A)                                                     UCLA6660
C         IF CARD NOT TOLERANCE, RETURN                                 UCLA6670
      IF((A(1).NE.T).OR.(A(2).NE.O).OR.(A(3).NE.AL))GO TO 12            UCLA6680
      WRITE(6,113)A                                                     UCLA6690
  113 FORMAT(' ',80A1)                                                  UCLA6700
      IF(NFLAG.EQ.1)GO TO 10                                            UCLA6710
C         CHECK FOR EQUALS SIGN                                         UCLA6720
      CALL ASCAN(B,1,J)                                                 UCLA6730
      IF(B.EQ.EQU)GO TO 5                                               UCLA6740
C         BAD CARD                                                      UCLA6750
    7 WRITE(6,102)                                                      UCLA6760
  102 FORMAT(/10X,'**** ERROR IN TOLERANCE CARD--IGNORED')              UCLA6770
      GO TO 10                                                          UCLA6780
    5 K=J+2                                                             UCLA6790
C         OBTAIN CIRCUIT ELEMENT                                        UCLA6800
      CALL ASCAN(B,K,J)                                                 UCLA6810
      IF(J.EQ.K-1)GO TO 7                                               UCLA6820
      IF(B.NE.EQU)GO TO 7                                               UCLA6830
    6 L=J-K+1                                                           UCLA6840
```

```
      IF(L.GT.12)L=12                                                   UCLA6850
      NE1=NE                                                            UCLA6860
      DØ 8 I=1,NE1                                                      UCLA6870
      DØ 9 II=1,L                                                       UCLA6880
      III=II+K-1                                                        UCLA6890
      IF(CARD(I,II).NE.A(III))GØ TØ 8                                   UCLA6900
    9 CØNTINUE                                                          UCLA6910
C        ØBTAIN TØLERANCE VALUE                                         UCLA6920
      NE=NE+1                                                           UCLA6930
      M=J+2                                                             UCLA6940
      CALL ASCAN(B,M,N)                                                 UCLA6950
      CALL NUMBER(M,N)                                                  UCLA6960
      TØL(I)=VPATH(NE)                                                  UCLA6970
      NE=NE-1                                                           UCLA6980
      GØ TØ 10                                                          UCLA6990
    8 CØNTINUE                                                          UCLA7000
      GØ TØ 7                                                           UCLA7010
C        WRITE TØLERANCE TABLE                                          UCLA7020
   12 IF(NFLAG.EQ.1)RETURN                                              UCLA7030
      WRITE(6,120)                                                      UCLA7040
  120 FØRMAT(//' ELEMENT NAME',10X,'TØLERANCE'/)                        UCLA7050
      DØ 13 I=1,NE                                                      UCLA7060
      IF(SENS(I).EQ.0)GØ TØ 13                                          UCLA7070
      WRITE(6,121)(CARD(I,J),J=1,12),TØL(I)                             UCLA7080
  121 FØRMAT(1X,12A1,11X,F7.4)                                          UCLA7090
   13 CØNTINUE                                                          UCLA7100
      RETURN                                                            UCLA7110
      END                                                               UCLA7120
      SUBRØUTINE READ                                                   UCLA7130
C        H.F.ØKPENT          U.C.L.A.               1967                UCLA7140
C          ANALYSES    FREE    FIELD     CIRCUIT    DESCRIPTIØN    CARDS UCLA7150
      CØMMØN/TREE/TREE                                                  UCLA7160
      INTEGER TREE                                                      UCLA7170
      INTEGER TYPE,GENER                                                UCLA7180
      INTEGER ØRIGIN,TARGET                                             UCLA7190
      INTEGER TAG,EPR,FLAG,ØUTPUT                                       UCLA7200
      REAL LEFT,MINUS                                                   UCLA7210
      CØMMØN/DATA/C,AI,R,AL,E,AJ,D,SLASH,LEFT,AV,F,AS,T,Ø,P,AN,PNT,BLANKUCLA7220
     1,PLUS,MINUS,RIGHT,EQU,AM,H,AK,CØMMA,Y,AA                          UCLA7230
      CØMMØN/A/A(80)                                                    UCLA7240
      CØMMØN/TAG/TAG                                                    UCLA7250
      CØMMØN/ERR/ERR                                                    UCLA7260
      CØMMØN/X/AC(10)                                                   UCLA7270
      CØMMØN/FLAG/FLAG                                                  UCLA7280
      CØMMØN/BITS/LØØP(3000),DEP(50,13),NNØDES,INØDE                    UCLA7290
      CØMMØN/CIRCIT/CARD(50,12),TYPE(50),GENER(50),ØRIGIN(50),TARGET(5  UCLA7300
     10),INPUT,ØUTPUT,NE,SMAX,SMIN                                      UCLA7310
      CØMMØN/PATHS/LPATH(300,2),VPATH(300),S(300),FREQ                  UCLA7320
      NE=0                                                              UCLA7330
      NP=0                                                              UCLA7340
      ERR=0                                                             UCLA7350
      INPUT=0                                                           UCLA7360
      ØUTPUT=0                                                          UCLA7370
```

```
      TAG=0                                                             UCLA7380
      IF(TREE.EQ.0)GO TO 2                                              UCLA7390
 48   TREE=1                                                            UCLA7400
C        EXTRACT          CIRCUIT          FREQUENCY                    UCLA7410
      FREQ=1.0                                                          UCLA7420
      CALL ASCAN(B,1,I)                                                 UCLA7430
      IF(B.EQ.BLANK)GO TO 2                                             UCLA7440
      IF(B.EQ.LEFT)GO TO 49                                             UCLA7450
      CALL MSG(LEFT,I+1)                                                UCLA7460
      WRITE(6,224)FREQ                                                  UCLA7470
 224  FORMAT(10X,'**** FREQUENCY ASSUMED TO BE ',E16.8)                 UCLA7480
      GO TO 2                                                           UCLA7490
 49   I=I+2                                                             UCLA7500
      CALL ASCAN(B,I,J)                                                 UCLA7510
 51   IF((B.NE.EQU).AND.(B.NE.MINUS))GO TO 52                           UCLA7520
      CALL ASCAN(B,J+2,J)                                               UCLA7530
      GO TO 51                                                          UCLA7540
 52   IF(B.NE.RIGHT)CALL MSG(RIGHT,J+1)                                 UCLA7550
      NE=1                                                              UCLA7560
      CALL NUMBER(I,J)                                                  UCLA7570
      FREQ=VPATH(NE)                                                    UCLA7580
 2    NE=0                                                              UCLA7590
C                                                                       UCLA7600
C        BEGIN      CIRCUIT    DATA    ANALYSIS                         UCLA7610
 11   READ(5,100)A                                                      UCLA7620
      WRITE(6,200)A                                                     UCLA7630
 100  FORMAT(80A1)                                                      UCLA7640
 200  FORMAT(1X,80A1)                                                   UCLA7650
      CALL SHIFT(A)                                                     UCLA7660
      IF(A(1).NE.BLANK)GO TO 44                                         UCLA7670
      WRITE(6,218)                                                      UCLA7680
 218  FORMAT(10X,'**** BLANK CARD IS IGNORED.')                         UCLA7690
      GO TO 11                                                          UCLA7700
C        IF  'END OF DATA'    IS     ENCOUNTERED,   END   ANALYSIS      UCLA7710
 44   IF((A(1).EQ.E).AND.(A(2).EQ.AN).AND.(A(3).EQ.D))RETURN            UCLA7720
      IF((A(1).NE.P).OR.(A(2).NE.AL))GO TO 50                           UCLA7730
      WRITE(6,217)                                                      UCLA7740
 217  FORMAT(10X,'**** PLOT CARD IGNORED.')                             UCLA7750
      GO TO 11                                                          UCLA7760
 50   IF(NE.LE.49)GO TO 23                                              UCLA7770
      WRITE(6,221)                                                      UCLA7780
 221  FORMAT(10X,'**** MORE THAN 50 ELEMENTS NOT ALLOWED. ONLY THE FIRSTUCLA7790
     1 50 WILL BE USED.')                                               UCLA7800
 23   DEP(NE+1,1)=BLANK                                                 UCLA7810
      TYPE(NE+1)=0                                                      UCLA7820
      GENE(NE+1)=0                                                      UCLA7830
      DO45I=1,12                                                        UCLA7840
      DEP(NE+1,I+1)=BLANK                                               UCLA7850
 45   CAPS(NE+1,I)=BLANK                                                UCLA7860
C        CHECK   FIRST  LETTER  FOR  ELMENT  TYPE                       UCLA7870
      IF((A(1).EQ.E).OR.(A(1).EQ.AJ))GO TO 4                            UCLA7880
      IF((A(1).EQ.R).OR.(A(1).EQ.AL).OR.(A(1).EQ.C).OR.(A(1).EQ.D))GOTO5UCLA7890
      IF((A(1).EQ.AV).OR.(A(1).EQ.AI))GO TO 6                           UCLA7900
```

```
      IF(A(1).EQ.P)G0 T0 53                                             UCLA7910
      WRITE(6,220)                                                      UCLA7920
  220 F0RMAT(10X,'**** ELEMENT TYPE (FIRST CHARACTER) CANN0T BE IDENTIFIUCLA7930
     1ED. DEFAULT VALUE IS R.')                                         UCLA7940
      A(1)=R                                                            UCLA7950
      G0 T0 5                                                           UCLA7960
    6 WRITE(6,203)                                                      UCLA7970
  203 F0RMAT(10X,'**** ELEMENT TYPE CANN0T BE IDENTIFIED. ''END 0F DATA'UCLA7980
     1' CARD ASSUMED T0 BE MISSING.'/14X,'AB0VE CARD BEC0MES FIRST 0UTPUUCLA7990
     2T REQUEST.')                                                      UCLA8000
      TAG=1                                                             UCLA8010
      RETURN                                                            UCLA8020
   53 WRITE(6,223)                                                      UCLA8030
  223 F0RMAT(10X,'**** PL0T REQUESTS MUST F0LL0W THE C0RRESP0NDING 0UTPUUCLA8040
     1T REQUEST.'/10X,'     PL0T REQUEST IGN0RED AND DEFAULT PL0T ASSUMEUCLA8050
     2D.')                                                              UCLA8060
      G0 T0 11                                                          UCLA8070
C                                                                       UCLA8080
    4 IF(A(1).EQ.E)TYPE(NE+1)=1                                         UCLA8090
      GENER(NE+1)=1                                                     UCLA8100
      G0 T0 7                                                           UCLA8110
    5 IF(A(1).EQ.D)GENER(NE+1)=1                                        UCLA8120
      IF(A(2).EQ.E)G0 T0 8                                              UCLA8130
      IF(TREF.EQ.1)G0 T0 7                                              UCLA8140
      IF(A(2).EQ.AJ)G0 T0 7                                             UCLA8150
      WRITE(6,204)                                                      UCLA8160
  204 F0RMAT(10X,'**** ELEMENT TYPE CANN0T BE IDENTIFIED AS BRANCH 0R LIUCLA8170
     1NK. DEFAULT VALUE IS J(LINK).')                                   UCLA8180
      A(2)=AJ                                                           UCLA8190
      G0 T0 7                                                           UCLA8200
    8 TYPE(NE+1)=1                                                      UCLA8210
    7 CALL ASCAN(B,1,I)                                                 UCLA8220
      IF(I.GT.12)WRITE(6,205)(A(N),N=1,12)                              UCLA8230
  205 F0RMAT(10X,'**** ELEMENT NAME WITH M0RE THAN 12 CHARACTERS HAS BEEUCLA8240
     1N TRUNCATED T0 ',12A1)                                            UCLA8250
      IF(A(1).EQ.D)G0 T0 9                                              UCLA8260
      IF(B.EQ.LEFT)G0 T0 10                                             UCLA8270
      CALL MSG(LEFT,I+1)                                                UCLA8280
      G0 T0 11                                                          UCLA8290
    9 DEP(NE+1,1)=LEFT                                                  UCLA8300
      IF(B.EQ.SLASH)G0 T0 10                                            UCLA8310
      IF(B.EQ.LEFT)G0 T0 12                                             UCLA8320
      CALL MSG(SLASH,I+1)                                               UCLA8330
      G0 T0 11                                                          UCLA8340
   12 WRITE(6,206)                                                      UCLA8350
  206 F0RMAT(10X,'**** ELEMENT HAS ILLEGAL DEPENDENCY. ASSUMED T0 BE SELUCLA8360
     1F DEPENDENT.')                                                    UCLA8370
      DEP(NE+1,1)=PNT                                                   UCLA8380
C        INSERT   NAME   IN   LIST   AND   PAD   WITH   BLANKS          UCLA8390
   10 IF(I.GT.12)I=12                                                   UCLA8400
      D013J=1,I                                                         UCLA8410
   13 CARD(NE+1,J)=A(J)                                                 UCLA8420
C        IS   NAME     UNIQUE                                           UCLA8430
```

```
 14   IF(NE.EQ.0)GØ TØ 16                                                 UCLA8440
      DØ17J=1,NE                                                          UCLA8450
      DØ18K=1,12                                                          UCLA8460
      IF(CARD(J,K).NE.CARD(NE+1,K))GØ TØ 17                               UCLA8470
 18   CØNTINUE                                                            UCLA8480
      NR=NR+1                                                             UCLA8490
      IF(NR.GT.10)ERR=1                                                   UCLA8500
      DØ19K=3,11                                                          UCLA8510
 19   CARD(NE+1,K)=CØMMA                                                  UCLA8520
      CARD(NE+1,12)=AC(NR)                                                UCLA8530
      WRITE(6,207)(CARD(NE+1,K),K=1,12)                                   UCLA8540
 207  FØRMAT(10X,'**** DUPLICATE NAME ENCØUNTERED. PRØGRAM ASSIGNS THE NUCLA8550
     1AME ',12A1/10X,'**** WARNING--RESULTS ARE UNRELIABLE.')             UCLA8560
      GØ TØ 16                                                            UCLA8570
 17   CØNTINUE                                                            UCLA8580
 16   IF(A(1).NE.D)GØ TØ 20                                               UCLA8590
      IF(DEP(NE+1,1).EQ.PNT)GØ TØ 20                                      UCLA8600
C        INSERT   DEPENDENCY   NAME                                       UCLA8610
      I=I+2                                                               UCLA8620
      CALL ASCAN(B,I,J)                                                   UCLA8630
      M=1                                                                 UCLA8640
      IF(B.EQ.LEFT)GØ TØ 21                                               UCLA8650
      CALL MSG(LEFT,J+1)                                                  UCLA8660
      GØ TØ 11                                                            UCLA8670
 21   IF((A(I).EQ.AV).ØR.(A(I).EQ.AI))GØ TØ 15                            UCLA8680
      IF(A(2).EQ.E)DEP(NE+1,1)=AV                                         UCLA8690
      IF(A(2).EQ.AJ)DEP(NE+1,1)=AI                                        UCLA8700
      WRITE(6,219)DEP(NE+1,1)                                             UCLA8710
 219  FØRMAT(10X,'**** DEPENDENCY TYPE (FIRST LETTER ØF DEPENDENCY NAME)UCLA8720
     1 IS ILLEGAL. PRØGRAM ASSIGNS ',A1)                                  UCLA8730
      M=2                                                                 UCLA8740
 15   IF(J-I+M.GT.13)J=13+I-M                                             UCLA8750
      DØ22K=I,J                                                           UCLA8760
      N=K-I+M                                                             UCLA8770
 22   DEP(NE+1,N)=A(K)                                                    UCLA8780
 24   I=J                                                                 UCLA8790
 20   I=I+2                                                               UCLA8800
      CALL ASCAN(B,I,J)                                                   UCLA8810
      IF(J.LT.I)J=I                                                       UCLA8820
C        EXTRACT   ØRIGIN   NØDE                                          UCLA8830
      IF(J-I.LE.1)GØ TØ 26                                                UCLA8840
 27   WRITE(6,208)                                                        UCLA8850
 208  FØRMAT(10X,'**** MØRE THAN 99 NØDES NØT ALLØWED. CARD IGNØRED')     UCLA8860
      GØ TØ 11                                                            UCLA8870
 26   ØRIGIN(NE+1)=ISØRT(A(J))                                            UCLA8880
      IF(FLAG.EQ.1)GØ TØ 29                                               UCLA8890
      IF(J.NE.I)ØRIGIN(NE+1)=ØRIGIN(NE+1)+(10*ISØRT(A(I)))                UCLA8900
      IF(FLAG.EQ.0)GØ TØ 28                                               UCLA8910
 29   WRITE(6,209)                                                        UCLA8920
 209  FØRMAT(10X,'**** ILLEGAL CHARACTER IN ØRIGIN NUMBER. CARD IGNØRED'UCLA8930
     1)                                                                   UCLA8940
      GØ TØ 11                                                            UCLA8950
 28   IF(ØRIGIN(NE+1).NE.0)GØ TØ 46                                       UCLA8960
```

```
      WRITE(6,222)
222   FØRMAT(10X,'**** NØDE 0 NØT ALLØWED. 99 SUBSTITUTED.'/10X,'**** WA
     1RNING--RESULTS ARE UNRELIABLE.')
      ØRIGIN(NE+1)=99
46    IF(B.EQ.MINUS)GØ TØ 25
      CALL MSG(MINUS,J+1)
      GØ TØ 11
C        E TRACT     TARGET    NØDE
25    I=J+2
      CALL ASCAN(B,I,J)
      IF(J.LT.I)J=I
      IF(J-I.GT.1)GØ TØ 27
      TARGET(NE+1)=ISØRT(A(J))
      IF(FLAG.EQ.1)GØ TØ 30
      IF(J.NE.I)TARGET(NE+1)=TARGET(NE+1)+(10*ISØRT(A(I)))
      IF(FLAG.EQ.0)GØ TØ 31
30    WRITE(6,211)
211   FØRMAT(10X,'**** ILLEGAL CHARACTER IN TARGET NUMBER. CARD IGNØRED'
     1)
      GØ TØ 11
31    IF(TARGET(NE+1).NE.0)GØ TØ 47
      WRITE(6,222)
      TARGET(NE+1)=99
47    IF(B.NE.RIGHT)WRITE(6,210)
210   FØRMAT(10X,'**** MISSING RIGHT PARENTHESIS INSERTED BY PRØGRAM.')
      IF((B.NE.EQU).AND.(A(J+2).NE.EQU))WRITE(6,212)
212   FØRMAT(10X,'**** ''='' FØLLØWING RIGHT PARENTHESIS IS MISSING BUT
     1HAS BEEN INSERTED BY PRØGRAM.')
      I=J+2
      IF(B.EQ.BLANK)I=J+1
      IF(A(J+2).EQ.EQU)I=J+3
C        CARD  IS  NØW  SUFFICIENTLY  IN  ØRDER  FØR  CØNSIDERATIØN
      NE=NE+1
      NNØDES=MAX0(ØRIGIN(NE),TARGET(NE),NNØDES)
      INØDE=MIN0(ØRIGIN(NE),TARGET(NE),INØDE)
C        E TRACT     NUMERICAL    VALUE
      CALL BSCAN(B,I,J)
      CALL NUMBER(I,J)
      K=0
      IF((B.EQ.BLANK).ØR.(B.EQ.AI).ØR.(B.EQ.Ø))GØ TØ 33
      IF(B.NE.P)GØ TØ 34
      K=-12
      CALL UNITS('PF')
      GØ TØ 33
34    IF(B.NE.AK)GØ TØ 35
      K=3
      CALL UNITS('K')
      GØ TØ 33
35    B=A(J+2)
      IF(B.NE.F)GØ TØ 36
      K=-6
      CALL UNITS('MF')
      GØ TØ 33
```

```
36    IF((B.NE.BLANK).AND.(B.NE.AI).AND.(B.NE.0))G0 T0 32                UCLA9500
      K=6                                                                 UCLA9510
      CALL UNITS('M')                                                     UCLA9520
      G0 T0 33                                                            UCLA9530
32    IF(B.NE.H)G0 T0 37                                                  UCLA9540
      K=-3                                                                UCLA9550
      CALL UNITS('MH')                                                    UCLA9560
      G0 T0 33                                                            UCLA9570
37    IF(B.NE.AM)G0 T0 33                                                 UCLA9580
      B=A(J+3)                                                            UCLA9590
      IF(B.NE.F)G0 T0 38                                                  UCLA9600
      K=-12                                                               UCLA9610
      CALL UNITS('MMF')                                                   UCLA9620
      G0 T0 33                                                            UCLA9630
38    IF(B.NE.H)G0 T0 33                                                  UCLA9640
      K=-6                                                                UCLA9650
      CALL UNITS('MMH')                                                   UCLA9660
33    VPATH(NE)=VPATH(NE)*(10.**K)                                        UCLA9670
      G0 T0 11                                                            UCLA9680
      END                                                                 UCLA9690
      SUBR0UTINE UNITS(A)                                                 UCLA9700
C        H.F.0KRENT          U.C.L.A.             1967                    UCLA9710
      WRITE(6,200)A                                                       UCLA9720
200   F0RMAT(10X,'$$$$ UNITS ARE ',A4)                                    UCLA9730
      RETURN                                                              UCLA9740
      END                                                                 UCLA9750
      SUBR0UTINE WHAT                                                     UCLA9760
C        H.F.0KRENT          U.C.L.A.             1967                    UCLA9770
C        DEC0DES       0UTPUT      REQUESTS                               UCLA9780
      INTEGER S                                                           UCLA9790
      INTEGER TYPE                                                        UCLA9800
      INTEGER TAG,ERR                                                     UCLA9810
      C0MM0N/TAG/TAG                                                      UCLA9820
      C0MM0N/ERR/ERR                                                      UCLA9830
      C0MM0N/A/A(80)                                                      UCLA9840
      C0MM0N/PATHS/JSAVE(1201)                                            UCLA9850
      C0MM0N/CIRCIT/CARD(50,12),VN(202),NE                                UCLA9860
      REAL LEFT                                                           UCLA9870
      C0MM0N/DATA/C,AI,R,AL,E,AJ,D,SLASH,LEFT,AV,AM0D(7),BLANK            UCLA9880
      C0MM0N/NTIMES/NTIMES                                                UCLA9890
      C0MM0N/BITS/LBITS(3000),LPATH(300,2),VPATH(300),S(300),NPATH,       UCLA9900
     1SM0D(99),TYPE(30)                                                   UCLA9910
      DIMENSI0N ISAVE(1201)                                               UCLA9920
      EQUIVALENCE (LPATH(1,1),ISAVE(1))                                   UCLA9930
      D018I=1,1201                                                        UCLA9940
18    ISAVE(I)=JSAVE(I)                                                   UCLA9950
C        SET       STARTING       P0INTER                                 UCLA9960
      I=2                                                                 UCLA9970
      B=A(1)                                                              UCLA9980
      IF((B.EQ.R).0R.(B.EQ.C).0R.(B.EQ.AL).0R.(B.EQ.E).0R.(B.EQ.AJ).0R.(UCLA9990
     1B.EQ.D))I=1                                                         UCL10000
C        L00K          F0R       THE         SLASH                        UCL10010
      CALL ASCAN(B,I,J)                                                   UCL10020
```

```
      IF((B.EQ.SLASH).AND.(J.GE.I))GØ TØ 1
      WRITE(6,200)
200   FØRMAT(10X,'**** ILLEGAL NAME IN ØUTPUT FIELD, CARD IGNØRED.')
3     ERR=1
      RETURN
1     IF(J-I.GT.11)J=I+11
      DØ2K=1,NE
      N=K
      DØ4L=I,J
      M=L-I+1
      IF(A(L).NE.CARD(K,M))GØ TØ 2
4     CØNTINUE
      GØ TØ 5
2     CØNTINUE
13    WRITE(6,201)(A(K),K=I,J)
201   FØRMAT(10X,'**** ',12A1)
      WRITE(6,202)
202   FØRMAT(15X,'IS AN UNDEFINED ELEMENT NAME. CARD IGNØRED.')
      GØ TØ 3
5     IF(I.EQ.2)GØ TØ 6
8     LPATH(NPATH,1)=N+(NE*(TYPE(N)-1))
      B=AV
      IF(TYPE(N).EQ.1)B=AI
      WRITE(6,203)B
203   FØRMAT(10X,'**** ØUTPUT ELEMENT HAS ILLEGAL DEPENDENCY TYPE. PRØGR
     1AM ASSIGNS ',A1,' .')
      GØ TØ 7
6     IF(A(1).EQ.AV)LPATH(NPATH,1)=NE+N
      IF(A(1).EQ.AI)LPATH(NPATH,1)=N
      IF(LPATH(NPATH,1).EQ.0)GØ TØ 8
7     I=J+3
      B=A(I-1)
      IF((B.EQ.R).ØR.(B.EQ.C).ØR.(B.EQ.AL).ØR.(B.EQ.E).ØR.(B.EQ.AJ).ØR.(
     1B.EQ.D))I=J+2
      N1=J
      CALL ASCAN(B,I,J)
      IF((B.EQ.BLANK).AND.(J.GE.I))GØ TØ 9
      WRITE(6,204)
204   FØRMAT(10X,'**** ILLEGAL NAME IN INPUT FIELD. CARD IGNØRED.')
      GØ TØ 3
9     IF(J-I.GT.11)J=I+11
      DØ10K=1,NE
      N=K
      DØ11L=I,J
      M=L-I+1
      IF(A(L).NE.CARD(K,M))GØ TØ 10
11    CØNTINUE
      GØ TØ 12
10    CØNTINUE
      GØ TØ 13
12    IF(I.EQ.N1+3)GØ TØ 14
16    LPATH(NPATH,2)=N+(NE*TYPE(N))
      B=AV
```

```
      IF(TYPE(N).EQ.0)B=AI
      WRITE(6,205)B
  205 FØRMAT(10X,'**** INPUT ELEMENT HAS ILLEGAL DEPENDENCY TYPE. PRØGRA
     1M ASSIGNS ',A1,' .')
      GØ TØ 15
   14 IF(A(I-1).EQ.AV)LPATH(NPATH,2)=N+NE
      IF(A(I-1).EQ.AI)LPATH(NPATH,2)=N
      IF(LPATH(NPATH,2).EQ.0)GØ TØ 16
   15 RETURN
      END
      SUBRØUTINE PLØT
C        H.F.ØKRENT            U.C.L.A.              1967
      INTEGER ERR
      INTEGER SMAX,SMIN,GAG,SMAX1,SMIN1,SENS,SMAX2,SMIN2,BMØD(8),S
      REAL LEFT,MINUS
      DIMENSIØN SMIN(2),SMAX(2)
      CØMMØN/CIRCIT/CARD(50,12),VN(101,2),NE,SMAX,SMIN
      CØMMØN/GAG/GAG(25),KSEN,SENS(50)
      CØMMØN/WØRST1/NWØRST,VW(101,2),TØL(50),SMAX2,SMIN2
      CØMMØN/DATA/AC,AI,R,AL,E,AJ,D,SLASH,LEFT,AV,F,AS,T,Ø,P,CMØD(2),
     1BLANK,PLUS,MINUS,RIGHT,EQU,AM,H,AK,CØMMA,Y,AA,X,U,AB,W
      CØMMØN/SPEED/F2(2048)
      CØMMØN/PATHS/LPATH(600),VPATH(300),S(300),NPATH
      CØMMØN/A/A(80)
      CØMMØN/PØLY/VNSEN(101,2,20),VNØØ(2020),SMAX1(20),SMIN1(20)
      CØMMØN/BITS/F1(4096),
     9B1(50),C(50),NXFR,A1(50),NXTI,AMØD(8),
     1NUMFR,NUMTI,NF(11),NR(11),MØR,MØRNUM,NTAY,ISN,SCMAG,SCFREQ,SIGMA,
     2T1,T2,T3,DMØD(8)
      CØMMØN/ERR/ERR
      EQUIVALENCE (AMØD(1),BMØD(1))
      I1=0
      I2=0
      NE1=NE
      NE=NPATH+1
      DØ1I=1,8
    1 BMØD(I)=-1
      I=1
    6 CALL ASCAN(B,I,I)
    4 IF(B.NE.LEFT)CALL MSG(LEFT,I+1)
      IF(B.EQ.BLANK)GØ TØ 2
      IF((B.EQ.SLASH).ØR.(B.EQ.LEFT))GØ TØ 3
      CALL ASCAN(B,I+2,I)
      GØ TØ 4
    3 CALL ASCAN(B,I+2,J)
      IF(B.NE.EQU)CALL MSG(EQU,J+1)
      IF((B.EQ.BLANK).ØR.(B.EQ.RIGHT))GØ TØ 2
      IF(B.NE.SLASH)GØ TØ 12
      I=J+2
      GØ TØ 3
   12 K=J+2
    5 CALL ASCAN(B,K,L)
      IF(B.NE.MINUS)GØ TØ 7
```

```
      K=L+2
      GØ TØ 5
7     IF((A(I+2).NE.T).ØR.(A(I+3).NE.Y))GØ TØ 8
      IF((A(J+2).EQ.AI).AND.(A(J+3).EQ.AM))BMØD(3)=0
      IF((A(J+2).EQ.AS).AND.(A(J+3).EQ.T ))BMØD(3)=1
      IF((A(J+2).EQ.E).AND.A(J+3).EQ.X) BMØD(3)=2
      IF((A(J+2).EQ.AS).AND.(A(J+3).EQ.AI))BMØD(3)=3
      IF((A(J+2).EQ.P) AND.A(J+3).EQ.U) BMØD(3)=4
      IF((A(J+2).EQ.F).AND.A(J+3).EQ.R) BMØD(3)=6
      IF((A(J+2).EQ.AS).AND.(A(J+3).EQ.E))BMØD(3)=7
      IF((A(J+2).EQ.W ).AND.(A(J+3).EQ.Ø))BMØD(3)=8
      IF((A(J+2).EQ.R ).AND.(A(J+3).EQ.E))BMØD(3)=9
      GØ TØ 9
    8 IF((A(I+2).EQ.E).AND.(A(I+3).EQ.AL))GØ TØ 20
      CALL NUMBER(J+2,L)
      IF((A(I+2).EQ.AS).AND.(A(I+3).EQ. T))AMØD(1)=VPATH(NE)
      IF((A(I+2).EQ.T).AND.(A(I+3).EQ.AI))AMØD(2)=VPATH(NE)
      IF((A(I+2).EQ.AA).AND.(A(I+3).EQ.AM))AMØD(4)=VPATH(NE)
      IF((A(I+2).EQ.AB).AND.(A(I+3).EQ.AI))AMØD(5)=VPATH(NE)
      IF((A(I+2).EQ.AC).AND.(A(I+3).EQ.Ø))AMØD(6)=VPATH(NE)
      IF((A(I+2).EQ.AC).AND.(A(I+3).EQ.Y))AMØD(6)=VPATH(NE)
      IF((A(I+2).EQ. F).AND.(A(I+3).EQ. R))AMØD(6)=VPATH(NE)
      IF((A(I+2).EQ.T).AND.(A(I+3).EQ.Ø))AMØD(7)=VPATH(NE)
      IF((A(I+2).EQ.W).AND.(A(I+3).EQ.AI))AMØD(7)=VPATH(NE)
      IF((A(I+2).EQ. D).AND.(A(I+3).EQ. E))BMØD(8)=VPATH(NE)+0.5
9     I=L
      IF((B.EQ.BLANK).ØR.(B.EQ.RIGHT))GØ TØ 2
      GØ TØ 3
2     IF(BMØD(3).LT.6)GØ TØ 14
      IF(BMØD(6).EQ.-1)AMØD(6)=1.0E0
      IF(BMØD(7).EQ.-1)AMØD(7)=1.E6
      IF(BMØD(1).EQ.-1)AMØD(1)=(AMØD(7)/AMØD(6))**1.E-2
      GØ TØ 15
14    IF((BMØD(1).NE.-1).ØR.(BMØD(2).NE.-1))GØ TØ 10
      AMØD(1)=0.01
      AMØD(2)=1.0
10    IF(BMØD(1).EQ.-1)AMØD(1)=AMØD(2)/100.
      IF(BMØD(2).EQ.-1)AMØD(2)=100.*AMØD(1)
      IF(BMØD(3).EQ.-1)BMØD(3)=0
      IF(BMØD(4).EQ.-1)AMØD(4)=1.0
      IF(BMØD(5).NE.-1)AMØD(4)=AMØD(4)+AMØD(5)
      IF(BMØD(6).EQ.-1)AMØD(6)=1.0
      IF(BMØD(7).NE.-1)BMØD(3)=5
15    IF(BMØD(8).EQ.-1)BMØD(8)=1
      IF(BMØD(3).EQ.7)GØ TØ 21
      IF(BMØD(3).EQ.8)GØ TØ 22
      NSMIN=MIN0(SMIN(1),SMIN(2))
      NSMAX=MAX0(SMAX(1),SMAX(2))
      I=NSMAX-NSMIN+1
      DØ13J=1,I
      K=50+NSMIN+J
      A1(J)=-VN(K,2)
13    B1(J)=VN(K,1)
```

```
      J=I+1
      DO11I=J,50
      A1(I)=0.0
11    B1(I)=0.0
   25 NE=NE1
      RETURN
   20 M=L-J-1
      IF(M.GT.12)M=12
      DO 31 M2=1,NE1
      DO 30 M1=1,M
      IF(CARD(M2,M1).NE.A(J+1+M1))GO TO 31
   30 CONTINUE
      I1=M2
      GO TO 35
   31 CONTINUE
   99 WRITE(6,100)
  100 FORMAT(/10X,'**** INVALID REQUEST FOR SENSITIVITY OR WORST CASE PL
     *OT--IGNORED.')
      ERR=1
      GO TO 25
   35 IF(SENS(I1).NE.0)GO TO 36
      GO TO 99
   36 DO 37 N=1,NSEN
      IF(SENS(I1).NE.GAG(N))GO TO 37
      I2=N
      GO TO 9
   37 CONTINUE
   21 IF(I2.EQ.0)GO TO 99
      I=SMAX1(I2)-SMIN1(I2)+1
      DO 50 J=1,I
      K=50+SMIN1(I2)+J
      A1(J)=VNSEN(K,2,I2)
   50 B1(J)=VNSEN(K,1,I2)
      J=I+1
      DO 51 I=J,50
      A1(I)=0.0
   51 B1(I)=0.0
      WRITE(6,101)(CARD(I1,I),I=1,12)
  101 FORMAT('1','SENSITIVITY WITH RESPECT TO ',12A1)
      GO TO 25
   22 IF(NWORST.EQ.0)GO TO 99
      I=SMAX2-SMIN2+1
      DO 40 J=1,I
      K=50+SMIN2+J
      A1(J)=VW(K,2)
   40 B1(J)=VW(K,1)
      J=I+1
      DO 41 I=J,50
      A1(I)=0.0
   41 B1(I)=0.0
      WRITE(6,102)
  102 FORMAT('1',10X,'WORST CASE TOLERANCE')
      GO TO 25
```

```
      END                                                               UCL12140
      SUBROUTINE  FINE                                                  UCL12150
C        DAVID  PALETZ         U.C.L.A.                  1967           UCL12160
      INTEGER ORIGIN,TARGET,OORIGN,OTARGT,ODUM,OGENER                   UCL12170
      INTEGER TAG,ERR,GENER                                             UCL12180
      INTEGER BLANK                                                     UCL12190
      COMMON/ERR/ERR                                                    UCL12200
      COMMON/CIRCIT/CARD(50,12),TAG(50), GENER(50),ORIGIN(50),TARGET(50 UCL12210
     1),INPUT,OUTPUT,NE                                                 UCL12220
      COMMON/PATHS/OCARD(50,12),VPATH(300),Z(100),OZ(100),OORIGN(50),   UCL12230
     1OTARGT(50),W                                                      UCL12240
      COMMON/BITS/LOOP(3000),DEP(50,13),NNODES,INODE,LINE(132),INDIC(50)UCL12250
     1,ODUM(50),NUMB(50),NINDIC(100),LJUNK(18),OGENER(50),JUNK(48),OVPATUCL12260
     1H(300)                                                            UCL12270
      COMMON/DATA/C,AI,R,AL,E,AJ,D,SLASH,LEFT,AV,F,AS,T,9,P,AN,PNT,BLANKUCL12280
     1,PLUS,MINUS,RIGHT,EQU,AM,H,AK,COMMA                               UCL12290
      COMMON/X/B(10)                                                    UCL12300
      WRITE(6,99)W                                                      UCL12310
   99 FORMAT(//2X,'W=',E12.4)                                           UCL12320
C       CALCULATING  THE  RESISTANCE,  IM  OHMS,  OF THE ELEMENTS.      UCL12330
C     9.9*10**30  STAYS  FOR  INFINITY.                                 UCL12340
      J=1                                                               UCL12350
      DO 1  I=1,NE                                                      UCL12360
      IF(CARD(I,J).EQ.E)Z(I)=0.                                         UCL12370
      IF(CARD(I,J).EQ.AJ)Z(I)=9.9E30                                    UCL12380
      IF(CARD(I,J).EQ.C)Z(I)=1./(VPATH(I)*W)                            UCL12390
      IF(CARD(I,J).EQ.AL)Z(I)=VPATH(I)*W                                UCL12400
      IF(CARD(I,J).EQ.R)Z(I)=VPATH(I)                                   UCL12410
      IF(CARD(I,J).EQ.D)GO TO 2                                         UCL12420
      GO TO 1                                                           UCL12430
    2 J=2                                                               UCL12440
      IF(CARD(I,J).EQ.E)Z(I)=0.                                         UCL12450
      IF(CARD(I,J).EQ.AJ)Z(I)=9.9E30                                    UCL12460
      J=1                                                               UCL12470
    1 CONTINUE                                                          UCL12480
C  ARRANGING  THE  ELEMENTS  IN  A  SEQUENCE  OF  INCREASING  RESISTANCEUCL12490
C   VALUE.                                                              UCL12500
      DO 4   I=1,NE                                                     UCL12510
      DO 4   J=1,NE                                                     UCL12520
      IF(Z(I)-Z(J))3,3,4                                                UCL12530
    3 DUMMY=Z(J)                                                        UCL12540
      Z(J)=Z(I)                                                         UCL12550
      Z(I)=DUMMY                                                        UCL12560
      OZ(J)=Z(J)                                                        UCL12570
      DUM=ORIGIN(J)                                                     UCL12580
      ORIGIN(J)=ORIGIN(I)                                               UCL12590
      ORIGIN(I)=DUM                                                     UCL12600
      OORIGN(J)=ORIGIN(J)                                               UCL12610
      DOM=TARGET(J)                                                     UCL12620
      TARGET(J)=TARGET(I)                                               UCL12630
      TARGET(I)=DOM                                                     UCL12640
      OTARGT(J)=TARGET(J)                                               UCL12650
      KDUMMY=GENER(J)                                                   UCL12660
```

```
      GENER(J)=GENER(I)                                                 UCL12670
      GENER(I)=KDUMMY                                                   UCL12680
      ØGENER(J)=GENER(J)                                                UCL12690
      ADUMMY=VPATH(J)                                                   UCL12700
      VPATH(J)=VPATH(I)                                                 UCL12710
      VPATH(I)=ADUMMY                                                   UCL12720
      ØVPATH(J)=VPATH(J)                                                UCL12730
      DRUM=DEP(J,13)                                                    UCL12740
      DEP(J,13)=DEP(I,13)                                               UCL12750
      DEP(I,13)=DRUM                                                    UCL12760
      DØ 4  N=1,12                                                      UCL12770
      DRUM=DEP(J,N)                                                     UCL12780
      DEP(J,N)=DEP(I,N)                                                 UCL12790
      DEP(I,N)=DRUM                                                     UCL12800
      DEEM=CARD(J,N)                                                    UCL12810
      CARD(J,N)=CARD(I,N)                                               UCL12820
      CARD(I,N)=DEEM                                                    UCL12830
      ØCARD(J,N)=CARD(J,N)                                              UCL12840
    4 CØNTINUE                                                          UCL12850
      DØ 70  I=1,NE                                                     UCL12860
   70 VPATH(I)=ØVPATH(I)                                                UCL12870
C  DETERMINING  HØW  MANY  NØDES  THERE  ARE  IN  THE  CIRCUIT.         UCL12880
      J=1                                                               UCL12890
      DØ 40   I=2,50,2                                                  UCL12900
      N=I-1                                                             UCL12910
      NINDIC(N)=ØØRIGN(J)                                               UCL12920
      NINDIC(I)=ØTARGT(J)                                               UCL12930
      J=J+1                                                             UCL12940
      IF(J.GT.NE)GØ TØ 47                                               UCL12950
   40 CØNTINUE                                                          UCL12960
   47 NUMIND=1                                                          UCL12970
      NUMB(1)=NINDIC(1)                                                 UCL12980
      NUMB(2)=NINDIC(2)                                                 UCL12990
        J=2                                                             UCL13000
   42 DØ 41   I=3,NUMIND                                                UCL13010
      NTAG=1                                                            UCL13020
      DØ 44   N=1,J                                                     UCL13030
      IF(NINDIC(I).EQ.NUMB(N))NTAG=0                                    UCL13040
   44 CØNTINUE                                                          UCL13050
      IF(NTAG.EQ.1)GØ TØ 45                                             UCL13060
   41 CØNTINUE                                                          UCL13070
      GØ TØ 46                                                          UCL13080
   45 J=J+1                                                             UCL13090
      NUMB(J)=NINDIC(I)                                                 UCL13100
      GØ TØ 42                                                          UCL13110
   46 NUMNØD=J                                                          UCL13120
C  DETERMINING  THE  ELEMENTS  THAT  FØRM  TREE.  I.E.  HAVING  TAG  1  UCL13130
      DØ 9   I=1,50                                                     UCL13140
      INDIC(I)=0                                                        UCL13150
    9 TAG(I)=0                                                          UCL13160
      INDIC(1)=ØØRIGN(1)                                                UCL13170
      INDIC(2)=ØTARGT(1)                                                UCL13180
      TAG(1)=1                                                          UCL13190
```

```
   13 L=3
   15 I=2
   16 J=1
   17 IF(ØØRIGN(I).EQ.INDIC(J))GØ TØ 10
      IF(ØTARGT(I).EQ.INDIC(J))GØ TØ 11
      J=J+1
      IF(J.GT.50)GØ TØ 20
      GØ TØ 17
   20 I=I+1
      GØ TØ 16
   10 ØDUM(I)=ØTARGT(I)
      GØ TØ 18
   11 ØDUM(I)=ØØRIGN(I)
   18 IF(TAG(I).EQ.1)GØ TØ 34
      IF(INDIC(L))33,32,33
   32 IF(L.GT.NUMNØD)GØ TØ 12
      TAG(I)=1
      INDIC(L)=ØDUM(I)
      GØ TØ 13
   33 DØ 31  NN=1,50
      IF(INDIC(NN).EQ.0)GØ TØ 30
   31 CØNTINUE
      GØ TØ 34
   30 L=NN
      GØ TØ 32
   34 I=I+1
      GØ TØ 16
   12 WRITE(6,100)
  100 FØRMAT(//' THE TREE  IS  FØRMED  BY  THE  ELEMENTS  THAT  HAVE  TA
     3G  1'//)
      WRITE(6,101)
  101 FØRMAT(' ELEMENT NUMBER',5X,'ELEMENT  NAME',5X,'ØRIGIN  NØDE',5X,'
     1TARGET  NØDE'9X,'VALUE(ØHMS)',13X,'TAG',12X,'GENER')
      DØ 7  I=1,132
    7 LINE(I)=BLANK
      DØ 8  I=1,120
    8 LINE(I)=MINUS
      WRITE(6,104)(LINE(I),I=1,120)
  104 FØRMAT(120A1)
      DØ 5  I=1,NE
      WRITE(6,102)I,(ØCARD(I,J),J=1,12),ØØRIGN(I),ØTARGT(I),ØZ(I),TAG(I)
     1,ØGENER(I)
  102 FØRMAT(  7X,I2,17X,12A1,6X,I2,15X,I2,10X,E15.8,13X,I1,15X,I1)
    5 CØNTINUE
      RETURN
      END
      SUBRØUTINE SCALER
C       H.F.ØKRENT           UCLA                    1969
      INTEGER FACTØR
      CØMMØN/ERR/ERR,FACTØR
      CØMMØN/CIRCIT/CARD(50,12),BMØD(202),NE
      CØMMØN/PATHS/LPATH(600),VPATH(300),S(300),NPATH
      CØMMØN/DATA/AC,AMØD(2),AL
```

```
      FACTØR=0
    8 D=0.
      DL=0.
      DF=0.
      DØ1I=1,NE
      IF((CARD(I,1).NE.AC).AND.(CARD(I,1).NE.AL))GØ TØ 1
      IF(VPATH(I).EQ.0.)GØ TØ 1
      A=ALØG10(ABS(VPATH(I)))
      D=D+ABS(A+FLØAT(FACTØR))
      DL=DL+ABS(A+FLØAT(FACTØR-1))
      DR=DR+ABS(A+FLØAT(FACTØR+1))
    1 CØNTINUE
      IF((D.LT.60.).AND.(FACTØR.EQ.0))RETURN
      IF(DL.LT.D)GØ TØ 3
      IF(DR.LT.D)GØ TØ 4
      IF(FACTØR.EQ.0)GØ TØ 6
      DØ5I=1,NE
      IF((CARD(I,1).NE.AC).AND.(CARD(I,1).NE.AL))GØ TØ 5
      VPATH(I)=VPATH(I)*(10.**FACTØR)
    5 CØNTINUE
    7 RETURN
    6 WRITE(6,200)
  200 FØRMAT(10X,'**** THE RANGE ØF ELEMENT VALUES IS TØØ LARGE TØ BE SC
     1ALED.'/10X,'**** ERRØNEØUS RESULTS DUE TØ ØVERFLØW MAY ØCCUR.')
      GØ TØ 7
    3 D=DL
      FACTØR=FACTØR-1
      GØ TØ 8
    4 D=DR
      FACTØR=FACTØR+1
      GØ TØ 8
      END
      SUBRØUTINE PØLSEN
C
C        S. GRANDI    U.C.L.A.     1969
C        CØMPUTE THE SENSITIVITES ØF THE PØLES ØF THE TRANSFER FUNCTIØN
C
      INTEGER SENS,GAG
      INTEGER SMIN,SMAX,SMIN1,SMAX1,SMIN01,SMAX01,SMIN00,SMAX00,SMIN10,
     *SMAX10
      CØMPLEX PØLE
      CØMMØN/PØLY/VN01(101,20),VN00(101,20),VN10(101,20),SMAX01(20),
     *SMIN01(20),SMAX00(20),SMIN00(20),SMAX10(20),SMIN10(20)
      CØMMØN/SPEED/PØLE(50),NPØLES
      CØMMØN/CIRCIT/CARD(50,12),VN(101,2),NE,SMAX,SMIN
      CØMMØN/GAG/GAG(20),NSEN,SENS(50)
      DIMENSIØN SMIN(2),SMAX(2)
      DIMENSIØN VDERIV(101),A(50),B(50),C(50),D(50)
C        RETURN IF RØØTS ØR SENSITIVITY FCN NØT PREVIØUSLY FØUND
      IF(NPØLES.NE.0.AND.NSEN.NE.0)GØ TØ 1
      WRITE(6,100)
  100 FØRMAT(/10X,'**** RØØTS AND/ØR SENSITIVITY REQUEST NØT PREVIØUSLY
     *MADE--PØLE SENSITIVITY REQUEST IGNØRED.')
```

```
      RETURN                                                            UCL14260
C        TAKE DERIV. ØF DENØMINATØR ØF TRANSFER FUNCTIØN                UCL14270
    1 I1=SMIN(1)+51                                                     UCL14280
      I2=SMAX(1)+51                                                     UCL14290
      DØ 5 I=I1,I2                                                      UCL14300
    5 VDERIV(I-1)=(I-51)*VN(I,1)                                        UCL14310
C        EVALUATE IT AT THE PØLES                                       UCL14320
      DØ 7 K=1,50                                                       UCL14330
      C(K)=0.0                                                          UCL14340
    7 D(K)=0.0                                                          UCL14350
      DØ 6 L=1,NPØLES                                                   UCL14360
      A(L)=REAL(PØLE(L))                                                UCL14370
      B(L)=AIMAG(PØLE(L))                                               UCL14380
      DØ 6 K=I1,I2                                                      UCL14390
      IF(A(L).EQ.0)GØ TØ 60                                             UCL14400
      C(L)=VDERIV(K-1)*A(L)**(K-52)+C(L)                                UCL14410
   60 IF(B(L).EQ.0)GØ TØ 6                                              UCL14420
      D(L)=VDERIV(K-1)*B(L)**(K-52)+D(L)                                UCL14430
    6 CØNTINUE                                                          UCL14440
C        FIND SENSITIVITY FCNS FØR ALL SPECIFIED ELEMENTS               UCL14450
      DØ 2 I=1,NE                                                       UCL14460
      IF(SENS(I).EQ.0)GØ TØ 2                                           UCL14470
      DØ 3 J=1,NSEN                                                     UCL14480
      IF(SENS(I).EQ.3AG(J))GØ TØ 4                                      UCL14490
    3 CØNTINUE                                                          UCL14500
    4 WRITE(6,101)(CARD(I,K),K=1,12)                                    UCL14510
  101 FØRMAT('1SENSITIVITY ØF PØLES TØ ',12A1///)                       UCL14520
      WRITE(6,102)                                                      UCL14530
  102 FØRMAT(15X,'PØLES',24X,'SENSITIVITY'//9X,'REAL',10X,'IMAG',14X,'REUCL14540
     *AL',10X,'IMAG'/)                                                  UCL14550
C        EVALUATE NUMERATØR ØF RØØT SENSITIVITY FCN AT THE PØLES        UCL14560
      DØ 2 L=1,NPØLES                                                   UCL14570
      X=0                                                               UCL14580
      Y=0                                                               UCL14590
      I1=SMIN10(J)+51                                                   UCL14600
      I2=SMAX10(J)+51                                                   UCL14610
      DØ 11 K=I1,I2                                                     UCL14620
      IF(A(L).EQ.0)GØ TØ 110                                            UCL14630
      X=X+VN10   (K,J)*A(L)**(K-51)                                     UCL14640
  110 IF(B(L).EQ.0)GØ TØ 11                                             UCL14650
      Y=Y+VN10   (K,J)*B(L)**(K-51)                                     UCL14660
   11 CØNTINUE                                                          UCL14670
C        ØBTAIN ANSWERS (PREVENTING ZERØDIVIDE)                         UCL14680
      IF(C(L).NE.0)GØ TØ 12                                             UCL14690
      P=0                                                               UCL14700
      GØ TØ 23                                                          UCL14710
   12 P=X/C(L)                                                          UCL14720
   23 IF(D(L).NE.0)GØ TØ 13                                             UCL14730
      Q=0                                                               UCL14740
      GØ TØ 14                                                          UCL14750
   13 Q=Y/D(L)                                                          UCL14760
   14 WRITE(6,103)A(L),B(L),P,Q                                         UCL14770
  103 FØRMAT(6X,1PE11.4,3X,1PE11.4,6X,1PE11.4,3X,1PE11.4)               UCL14780
```

```
    2 CØNTINUE
      RETURN
      END
      SUBRØUTINE EQUAL(I,J,N,NTIMES)
C        H.F.ØKRENT          U.C.L.A.                1967
      CØMMØN/BITS/K(3000)
      IF(NTIMES.GT.0)GØ TØ 2
      IF(I.LT.0)I1=(IABS(I)-1)*IABS(NTIMES)
      IF(J.LT.0)J1=(IABS(J)-1)*IABS(NTIMES)
      IF(I.GT.0)I1=3000+(NTIMES*I)
      IF(J.GT.0)J1=3000+(NTIMES*J)
      GØ TØ 3
    2 I2=-1
      J2=-1
      IF(I.LT.0)I2=99
      IF(J.LT.0)J2=99
      I1=NTIMES*(IABS(I)+I2)
      J1=NTIMES*(IABS(J)+J2)
    3 N1=IABS(NTIMES)
      DØ1M=1,N1
      M1=I1+M
      M2=J1+M
    1 L=N(K(M1),K(M2))
      RETURN
      END
      FUNCTIØN LØR(I,J)
C        H.F.ØKRENT          U.C.L.A.                1967
      LØGICAL K1,L1
      EQUIVALENCE (K1,K),(L1,L)
      LØR=0
      K=I
      L=J
      K1=K1.ØR.L1
      I=K
      RETURN
      END
      FUNCTIØN LØX(I,J)
C        H.F.ØKRENT          U.C.L.A.                1967
      LØGICAL K1,L1,M1
      EQUIVALENCE (K1,K),(L1,L),(M1,M)
      LØX=0
      K=I
      L=J
      M1=K1.AND.L1
      J=J-M
      I=I-M
      RETURN
      END
      FUNCTIØN LSTØR(I,J)
C        H.F.ØKRENT          U.C.L.A.                1967
      LSTØR=0
      I=J
      RETURN
```

```
      END
      SUBROUTINE UNP/ (I J,NN)
C        H.F.OKRENT             U.C.L.A.                1967
      COMMON/BITS/L([illegible])
      COMMON/NTIMES/NTIMES
      K=-1
      IF(I.LT.0)K=99
      I1=NTIMES*(IABS(I)+K)
      DO3M=1,NTIMES
      N=I1+M
      IF(L(N).NE.0)GO TO 2
    3 CONTINUE
    4 J=0
      RETURN
    2 N=1
      M=1
      N1=1
    6 IF(N.GT.NN)GO TO 4
      IF(N1.LE.30)GO TO 5
      M=M+1
      N1=N1-30
    5 M1=I1+M
      J=2**N1
      K=LOX(L(M1),J)
      IF(J.EQ.0)GO TO 1
      N=N+1
      N1=N1+1
      GO TO 6
    1 J=N1+(30*(M-1))
      RETURN
      END
      SUBROUTINE CALC
C        H.F.OKRENT             U.C.L.A.                1967
C        EVALUATES  CURRENT  EQUATIONS   BY   BIT   MANIPULATION
      INTEGER TARGET,ORIGIN,TYPE,BRANCH
      INTEGER ERR
      COMMON/NTIMES/NTIMES
      EXTERNAL LOX,LOR
      COMMON/ERR/ERR
      COMMON/CIRCIT/CARD(50,12),TYPE(50),GENER(50),ORIGIN(50),TARGET(5
     10),INPUT,OUTPUT,NE
      COMMON/BITS/LINKS(500),BRANCH(200),NQ(100),NEL(100),NBR(100),NS(10
     10),AMOD(1900),DEP(50,13),NNODES,INODE
      WRITE(6,199)
  199 FORMAT('1')
      NTIMES=((NE-1)/30)+1
C        INITIALISE
      NQ(INODE)=1
      NS(INODE)=1
      NB=0
      NN=0
      DO30I=INODE,NNODES
      NEL(I)=0
```

```
      NBR(I)=0
      CALL INBIT(I,0,NTIMES)
   30 CALL INBIT(-I,0,NTIMES)
C        FORM     EQUATIONS
      DO1I=1,NE
      J=ORIGIN(I)
      K=TARGET(I)
      NEL(J)=NEL(J)+1
      NEL(K)=NEL(K)+1
      IF(TYPE(I).EQ.1)GO TO 29
      CALL INBIT(J,I,NTIMES)
      CALL INBIT(-K,I,NTIMES)
      GO TO 1
   29 NBR(J)=NBR(J)+1
      NBR(K)=NBR(K)+1
      NB=NB+1
    1 CONTINUE
  300 FORMAT(1X,10I10)
C        CHECK    TREE     LEGALITY    AND    ENTER     BRANCHES
      DO2I=INODE,NNODES
      NS(I+1)=NS(I)+NBR(I)
      NQ(I+1)=NS(I+1)
      IF(NEL(I).NE.0)GO TO 3
      WRITE(6,202)I
  202 FORMAT(10X,'**** NODE',I3,' IS ISOLATED DUE TO NON-CONSECUTIVE NUM
     1BERING. WARNING--PROGRAM INEFFICIENCY WILL RESULT.')
      GO TO 2
    3 NN=NN+1
      IF(NEL(I).NE.1)GO TO 4
      ERR=1
      WRITE(6,203)I
  203 FORMAT(10X,'**** NODE',I3,' IS CONNECTED TO ONLY ONE ELEMENT. SOLU
     1TION IS IMPOSSIBLE.')
    4 IF(NBR(I).NE.0)GO TO 2
      ERR=1
      WRITE(6,204)I
  204 FORMAT(10X,'**** NODE',I3,' HAS NO BRANCH. THE TREE IS ILLEGAL. SO
     1LUTION IS IMPOSSIBLE.')
    2 CONTINUE
      IF(NB.EQ.NN-1)GO TO 5
      ERR=1
      IF(NB.GT.NN-1)WRITE(6,205)
  205 FORMAT(10X,'**** THE TREE IS OVER-DEFINED (THERE ARE TOO MANY BRAN
     1CHES). SOLUTION IS IMPOSSIBLE.')
      IF(NB.LT.NN-1)WRITE(6,206)
  206 FORMAT(10X,'**** THE TREE IS UNDER-DEFINED (THERE ARE TOO FEW BRAN
     1CHES). SOLUTION IS IMPOSSIBLE.')
    5 IF(ERR.EQ.1)RETURN
      DO7I=1,NE
      IF(TYPE(I).EQ.0)GO TO 7
      J=ORIGIN(I)
      K=TARGET(I)
      L=NS(J)
```

```
      BRANCH(L)=-I
      L=NS(K)
      BRANCH(L)=I
      NS(J)=NS(J)+1
      NS(K)=NS(K)+1
 7    CONTINUE
      JNODE=INODE+1
 25   K=0
      J=0
C        SEARCH FOR THE FIRST CUT-SET AND EQUATION
      DO22I=JNODE,NNODES
      IF((K.EQ.0).AND.(NBR(I).EQ.1))J=I
      IF((K.EQ.0).AND.(NBR(I).GT.1))K=I
      IF((K.NE.0).AND.(J.NE.0))GO TO 24
 22   CONTINUE
C        IF  NONE  IS  FOUND,  SOLUTION  IS  COMPLETE
      IF(K.EQ.0)GO TO 23
      ERR=1
      WRITE(6,207)
 207  FORMAT(10X,'**** SOLUTION IS IMPOSSIBLE DUE TO A LOOP CONCERNING T
     1HE FOLLOWING NODES-')
      DO6I=JNODE,NNODES
 6    IF(NBR(I).NE.0)WRITE(6,208)I
 208  FORMAT(85X,I3)
      RETURN
 24   N3=NQ(J)
      DO26I=K,NNODES
      IF(NBR(I).LE.1)GO TO 26
      L=NBR(I)
      N5=NQ(I)+L-1
      DO8M=1,L
      N4=NQ(I)+M-1
      IF(IABS(BRANCH(N3)).NE.IABS(BRANCH(N4)))GO TO 8
      N1=1
      N2=-1
      IF((BRANCH(N3)*BRANCH(N4)).LT.0)GO TO 13
      N1=-1
      N2=1
 13   CALL EQUAL(I,J*N1,LOR,NTIMES)
      CALL EQUAL(-I,J*N2,LOR,NTIMES)
      CALL EQUAL(I,-I,LOX,NTIMES)
      BRANCH(N4)=BRANCH(N5)
      NBR(I)=NBR(I)-1
      GO TO 26
 8    CONTINUE
 26   CONTINUE
      NBR(J)=0
      GO TO 25
 23   DO10I=JNODE,NNODES
      IF(NQ(I).EQ.NQ(I+1))GO TO 10
      J=NQ(I)
      K=NTIMES*(I-1)
      M=NTIMES*(I+99)
```

```
      WRITE(6,201)I,BRANCH(J),(LINKS(K+L),LINKS(M+L),L=1,NTIMES)
  201 FORMAT(1X,8I10)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE INBIT(I,J,NTIMES)
C        H.F.OKRENT          U.C.L.A.             1967
      COMMON/BITS/K(3000)
      N=-1
      L=J
      IF(I.LT.0)N=99
      M=(NTIMES*(IABS(I)+N))+1
      IF(NTIMES.LT.0)M=3001-(IABS(NTIMES)*I)
      IF(J.EQ.0)GO TO 3
    2 IF(L.LE.30)GO TO 1
      L=L-30
      M=M+1
      GO TO 2
    1 K(M)=K(M)+(2**L)
    5 RETURN
    3 M=M-1
      N=IABS(NTIMES)
      DO4L=1,N
    4 K(M+L)=0
      GO TO 5
      END
      SUBROUTINE GRAPH
C        H.F.OKRENT          U.C.L.A.             1967
C        COMPILES     FLOWGRAPH     DATA
      COMMON/DATA/C,AI,R,AL,E,AJ,D,AMOD(2),AV,BMOD(6),PNT,BLANK
      INTEGER SENS
      INTEGER BRANCH,ADDR
      INTEGER GENER,S
      INTEGER TYPE,TYP
      DIMENSION BRANCH(200),ADDR(100)
      COMMON/GAG/GAG(20),NSEN,SENS(50)
      COMMON/BITS/LBITS(3000),DEP(50,13),NNODES,INODE,XMOD(648),TYP(50)
      COMMON/CIRCIT/CARD(50,12),TYPE(50),GENER(50),ORIGIN(50),TARGET(5
     10),INPUT,OUTPUT,NE
      COMMON/PATHS/LPATH(300,2),VPATH(300),S(300),NPATH
      EQUIVALENCE (LBITS(501),BRANCH(1)),(LBITS(701),ADDR(1))
      NPATH=0
      DO 100 I=1,NE
      TYP(I)=TYPE(I)
      S(NPATH+1)=0
      VPATH(NPATH+1)=VPATH(I)
      IF(DEP(I,1).NE.BLANK)GO TO 11
      IF(GENER(I).EQ.1)GO TO 100
C        INSERT     PATHS     DUE     TO     IMPEDANCES
      NPATH=NPATH+1
      LPATH(NPATH,1)=I
      LPATH(NPATH,2)=I+NE
      IF(TYPE(I).EQ.1)GO TO 12
```

```
      LPATH(NPATH,1)=I+NE                                                UCL17440
      LPATH(NPATH,2)=I                                                   UCL17450
 12   IF(CARD(I,1).NE.R)GO TO 13                                         UCL17460
      IF(TYPE(I).EQ.1)GO TO 10                                           UCL17470
      GO TO 14                                                           UCL17480
 13   S(NPATH)=1                                                         UCL17490
      IF(CARD(I,1).NE.AL)GO TO 15                                        UCL17500
      IF(TYPE(I).EQ.1)GO TO 10                                           UCL17510
      GO TO 16                                                           UCL17520
 15   IF(TYPE(I).EQ.0)GO TO 10                                           UCL17530
 16   S(NPATH)=-S(NPATH)                                                 UCL17540
 14   VPATH(NPATH)=1./VPATH(NPATH)                                       UCL17550
      GO TO 10                                                           UCL17560
C        INSERT     PATHS     DUE      TO     DEPENDENT     ELEMENTS     UCL17570
 11   IF(DEP(I,1).EQ.PNT)GO TO 10                                        UCL17580
      DO17J=1,NE                                                         UCL17590
      L=J                                                                UCL17600
      DO18K=1,12                                                         UCL17610
      IF(CARD(J,K).NE.DEP(I,K+1))GO TO 17                                UCL17620
 18   CONTINUE                                                           UCL17630
      GO TO 19                                                           UCL17640
 17   CONTINUE                                                           UCL17650
      WRITE(6,200)(CARD(I,J),J=1,12)                                     UCL17660
 200  FORMAT(10X,'**** ',12A1,' DEPENDS ON NON-EXISTENT ELEMENT, DEPENDEUCL17670
     1NCY IS IGNORED.')                                                  UCL17680
      GO TO 10                                                           UCL17690
 19   NPATH=NPATH+1                                                      UCL17700
      LPATH(NPATH,1)=L                                                   UCL17710
      IF(DEP(I,1).EQ.AV)LPATH(NPATH,1)=L+NE                              UCL17720
      LPATH(NPATH,2)=I                                                   UCL17730
      IF(TYPE(I).EQ.1)LPATH(NPATH,2)=I+NE                                UCL17740
   10 IF (SENS(I).EQ.0)GO TO 100                                         UCL17750
      S(NPATH)=S(NPATH)+SENS(I)                                          UCL17760
  100 CONTINUE                                                           UCL17770
C        INSERT     CURRENT     AND     VOLTAGE     RELATIONSHIPS        UCL17780
      JNODE=INODE+1                                                      UCL17790
      DO1I=JNODE,NNODES                                                  UCL17800
      M=0                                                                UCL17810
      X=1.                                                               UCL17820
      N=1                                                                UCL17830
      IF(ADDR(I).EQ.ADDR(I+1))GO TO 1                                    UCL17840
      L=ADDR(I)                                                          UCL17850
      IF(BRANCH(L).LT.0)N=-1                                             UCL17860
      K=IABS(BRANCH(L))                                                  UCL17870
 6    M=M+1                                                              UCL17880
      IF(M.GE.3)GO TO 1                                                  UCL17890
 4    CALL UNPAK(I*N,J,NE)                                               UCL17900
      IF(J)3,2,3                                                         UCL17910
 3    NPATH=NPATH+2                                                      UCL17920
      LPATH(NPATH-1,2)=K                                                 UCL17930
      LPATH(NPATH,1)=K+NE                                                UCL17940
      LPATH(NPATH-1,1)=J                                                 UCL17950
      VPATH(NPATH-1)=X                                                   UCL17960
```

```
      LPATH(NPATH,2)=J+NE
      VPATH(NPATH)=X
      IF(GENER(K).EQ.GENER(J))VPATH(NPATH)=-X
 5    S(NPATH-1)=0
      S(NPATH)=0
      GØ TØ 4
 2    N=-N
      X=-1.
      GØ TØ 6
 1    CONTINUE
      RETURN
      END
      SUBRØUTINE LØØPS
C        H.F.ØKRENT          U.C.L.A.             1967
C        ØBTAINS    THE   SUM   ØF   EACH   ØRDER   ØF   LØØPS
      LØGICAL   LØGIC(3000),LØGIC1
      INTEGER SMIN,SMAX,SMIN1,SMAX1,SMIN01,SMAX01,SMIN00,SMAX00,SMIN10,
     *SMAX10
      INTEGER GAG,CØUNT(100),FLAG,S,SENS,ØRDER
C
      EXTERNAL LØR,LSTØR
      CØMMØN/GAG/GAG(20),NSEN,SENS(50)
      CØMMØN/PØLY/VN01(101,20),VN00(101,20),VN10(101,20),SMAX01(20),
     *SMIN01(20),SMAX00(20),SMIN00(20),SMAX10(20),SMIN10(20)
      CØMMØN/CIRCIT/CARD(50,12),VN(101,2),NE,SMAX,SMIN
      CØMMØN/SPEED/V(1000),LS(1000),NLØØP
      CØMMØN/NTIMES/NTIMES
      CØMMØN/BITS/LØØP(3000),LPATH(300,2),VPATH(300),S(300),NPATH,ØRDER(
     1100),NØDE(100)
      CØMMØN/FLAG/FLAG
      DIMENSIØN SMIN(2),SMAX(2)
      DIMENSIØN VM01(2020),VM00(2020),VM10(2020)
      DIMENSIØN VM(202)
      EQUIVALENCE (VN01(1,1),VM01(1)),
     *(VN00(1,1),VM00(1)),(VN10(1,1),VM10(1))
      EQUIVALENCE (LØØP(1),LØGIC(1)),(LØGIC1,M1)
      EQUIVALENCE (ØRDER(1),CØUNT(1)),(VN(1,1),VM(1))
 500  WRITE(6,300)LPATH(NPATH,1),LPATH(NPATH,2)
  300 FØRMAT(/1X,'THE UNKNØWN TRANSMITTANCE GØES FRØM NØDE',I3,' TØ',I3)
C        INITIALISE
      NN=2*NE
      DØ2I=1,NN
      NØDE(I)=0
 2    CØUNT(I)=0
      DØ10I=1,202
 10   VM(I)=0.0
      DØ 99 I=1,2020
      VM01(I)=0.0
      VM00(I)=0.0
   99 VM10(I)=0.0
      DØ 85 I=1,20
      SMAX 1(I)=0
      SMIN01(I)=0
```

```
      SMAX00(I)=0
      SMIN00(I)=0
      SMAX10(I)=0
   88 SMIN10(I)=0
      M1=0
      M2=0
      M3=0
      NLOOP=0
      DO 98 I=1,20
   98 VN00(51,I)=-1.
      VN(51,1)=1.
      SMAX(1)=0
      SMAX(2)=0
      SMIN(1)=0
      SMIN(2)=0
C        FIND   HOW    MANY     TRANSMITTANCES    LEAVE    NODE    J
      DO1I=1,NPATH
      K=LPATH(I,2)
      NODE(K)=NODE(K)+1
      J=LPATH(I,1)
 1    COUNT(J)=COUNT(J)+1
C        OBTAIN   THE    FIRST    NODE    WITH    ONE    LEAVING
C        TRANSMITTANCE
      DO3I=1,NPATH
      J=LPATH(I,1)
      IF(COUNT(J)-1)3,4,3
C        FORM     THE     UNIQUE    PATH
 4    DO5J=1,NPATH
      IF(LPATH(J,2).NE.LPATH(I,1))GO TO 5
      CALL EQUAL(J,I,LOR,-NTIMES)
      VPATH(J)=VPATH(J)*VPATH(I)
      S(J)=S(J)+S(I)
      LPATH(J,2)=LPATH(I,2)
 5    CONTINUE
      LPATH(I,1)=0
 3    CONTINUE
C        STORE     ANY     LOOPS     THAT    MAY     HAVE    BEEN
C        FORMED
      DO6I=1,NPATH
      IF(LPATH(I,1).NE.LPATH(I,2))GO TO 6
      NLOOP=NLOOP+1
      CALL EQUAL(-NLOOP,I,LSTOR,-NTIMES)
      V(NLOOP)=VPATH(I)
      LS(NLOOP)=S(I)
      JJ=NTIMES*(NLOOP-1)
      WRITE(6,999)(LOOP(II+JJ),II=1,NTIMES),LS(NLOOP)
 999  FORMAT(1X,2I20/1X,I20)
      LPATH(I,1)=0
 6    CONTINUE
      DO11I=1,NPATH
      J=LPATH(I,1)
      K=LPATH(I,2)
      IF(J.EQ.0)GO TO 13
```

```
      IF(NØDE(J).EQ.0)LPATH(I,1)=0
   13 IF(K.EQ.0)GØ TØ 11
      IF(CØUNT(K).EQ.0)LPATH(I,1)=0
   11 CØNTINUE
      CALL CLEAR
      IF(NPATH.EQ.0)GØ TØ 27
C        BEGIN   MATCHING    UNIQUE    PATHS
   19 I=1
      ØRDER(1)=1
   12 ØRDER(I+1)=ØRDER(1)
      IF(ØRDER(1).GE.NPATH)GØ TØ 27
    9 ØRDER(I+1)=ØRDER(I+1)+1
      IF(ØRDER(I+1).LE.NPATH)GØ TØ 8
      I=I-1
      GØ TØ 9
    8 IF(I.LE.0)GØ TØ 7
      N=ØRDER(I)
      M=ØRDER(I+1)
      L=NPATH+I
      IF(I.GT.1)N=L-1
      IF(LPATH(N,1).NE.LPATH(M,2))GØ TØ 9
      I1=NTIMES*NLØØP
      J1=3000-(NTIMES*M)
      K1=3000-(NTIMES*N)
      DØ 702MM=1,NTIMES
  702 LØGIC(I1+MM)=LØGIC(J1+MM).AND.LØGIC(K1+MM)
      CALL UNPAK(NLØØP+1,N1,NN)
      CALL UNPAK(NLØØP+1,N2,NN)
      CALL UNPAK(NLØØP+1,N3,NN)
      IF(LPATH(N,2).NE.LPATH(M,1))GØ TØ 32
      IF(N3.NE.0)GØ TØ 9
      IF((LPATH(N,1).EQ.N1).AND.(LPATH(N,2).EQ.N2))GØ TØ 33
      IF((LPATH(N,1).EQ.N2).AND.(LPATH(N,2).EQ.N1))GØ TØ 33
      GØ TØ 9
   33 NLØØP=NLØØP+1
      CALL EQUAL(-NLØØP,N,LSTØR,-NTIMES)
      CALL EQUAL(-NLØØP,M,LØR,-NTIMES)
      V(NLØØP)=VPATH(N)*VPATH(M)
      LS(NLØØP)=S(N)+S(M)
      JJ=NTIMES*(NLØØP-1)
      WRITE(6,999)(LØØP(II+JJ),II=1,NTIMES),LS(NLØØP)
      GØ TØ 9
   32 IF((N3.NE.0).ØR.(N2.NE.0))GØ TØ 9
      IF(N1.NE.LPATH(N,1))GØ TØ 9
      CALL EQUAL(L,N,LSTØR,-NTIMES)
      CALL EQUAL(L,M,LØR,-NTIMES)
      VPATH(L)=VPATH(N)*VPATH(M)
      S(L)=S(N)+S(M)
      LPATH(L,1)=LPATH(M,1)
      LPATH(L,2)=LPATH(N,2)
    7 I=I+1
      GØ TØ 12
C        ENTER    FIRST    ØRDER    LØØP    VALUES
```

```
27    DØ25I=1,NLØØP                                                    UCL19560
      IF(NSEN.EQ.0)GØ TØ 280                                           UCL19570
      J=LS(I)                                                          UCL19580
      K8=21-NSEN                                                       UCL19590
      DØ 979 K9=K8,20                                                  UCL19600
      LØC=21-K9                                                        UCL19610
      IF(J.GT.GAG(LØC)  -500)GØ TØ 998                                 UCL19620
      CALL REDUCE(J,K3,K2)                                             UCL19630
      IF(K2.EQ.2)GØ TØ 800                                             UCL19640
      VN00(K3+51,LØC)=VN00(K3+51,LØC)+V(I)                             UCL19650
      SMAX00(LØC)=MAX0(SMAX00(LØC),K3)                                 UCL19660
      SMIN00(LØC)=MIN0(SMIN00(LØC),K3)                                 UCL19670
      GØ TØ 979                                                        UCL19680
  800 VN01(K3+51,LØC)=VN01(K3+51,LØC)+V(I)                             UCL19690
      SMAX01(LØC)=MAX0(SMAX01(LØC),K3)                                 UCL19700
      SMIN01(LØC)=MIN0(SMIN01(LØC),K3)                                 UCL19710
      GØ TØ 979                                                        UCL19720
  998 J=J-GAG(LØC)                                                     UCL19730
      CALL REDUCE(J,K3,K2)                                             UCL19740
      IF(K2.EQ.2)GØ TØ 979                                             UCL19750
      VN10(K3+51,LØC)=VN10(K3+51,LØC)+V(I)                             UCL19760
      SMAX10(LØC)=MAX0(SMAX10(LØC),K3)                                 UCL19770
      SMIN10(LØC)=MIN0(SMIN10(LØC),K3)                                 UCL19780
  979 CØNTINUE                                                         UCL19790
  280 J=LS(I)                                                          UCL19800
      CALL REDUCE(J,K9,K)                                              UCL19810
      SMAX(K)=MAX0(SMAX(K),K9)                                         UCL19820
      SMIN(K)=MIN0(SMIN(K),K9)                                         UCL19830
      VN(K9+51,K)=VN(K9+51,K)+V(I)                                     UCL19840
   25 CØNTINUE                                                         UCL19850
C        ØBTAIN     HIGHER    ØRDER     LØØP      VALUES               UCL19860
C         SET    LØØP      ØRDER      PØINTER                          UCL19870
14    I=1                                                              UCL19880
      WRITE(6,990)NLØØP                                                UCL19890
990   FØRMAT(' NØ. ØF FIRST ØRDER LØØPS=',I3)                          UCL19900
      ØRDER(1)=1                                                       UCL19910
17    ØRDER(I+1)=ØRDER(I)                                              UCL19920
C        CHECK    FØR     MØRE      FIRST     ØRDER     LØØPS          UCL19930
      IF(I.EQ.1)WRITE(6,999)ØRDER(1)                                   UCL19940
      IF(ØRDER(1).GE.NLØØP)RETURN                                      UCL19950
16    ØRDER(I+1)=ØRDER(I+1)+1                                          UCL19960
      IF(ØRDER(I+1).LE.NLØØP)GØ TØ 18                                  UCL19970
      I=I-1                                                            UCL19980
      GØ TØ 16                                                         UCL19990
18    IF(I.LE.0)GØ TØ 24                                               UCL20000
      N=ØRDER(I)                                                       UCL20010
      M=ØRDER(I+1)                                                     UCL20020
      L=NLØØP+I                                                        UCL20030
      IF(I.GT.1)N=L-1                                                  UCL20040
      I1=NTIMES*(L-1)                                                  UCL20050
      J1=NTIMES*(M-1)                                                  UCL20060
      K1=NTIMES*(N-1)                                                  UCL20070
      DØ70 MM=1,NTIMES                                                 UCL20080
```

```
      LØGIC1=LØGIC(J1+MM).AND.LØGIC(K1+MM)
      IF(M1.NE.0)GØ TØ 14
C        E TER    THE    LØØP    ØF    ØRDER    I+1
 701  LØGIC(I1+MM)=LØGIC(J1+MM).ØR.LØGIC(K1+MM)
      V(L)=V(N)*V(M)
      LS(L)=LS(N)+LS(M)
      IF(NSEN.EQ.0)GØ TØ 281
      J=LS(L)
      K8=21-NSEN
      DØ 799 K9=K8,20
      LØC=21-K9
      IF(J.GT.GAS(LØC)  -500)GØ TØ 798
      CALL REDUCE(J,K3,K2)
      IF(K2.EQ.2)GØ TØ 850
      VN00(K3+51,LØC)=VN00(K3+51,LØC)-(V(L)*((-1)**(I+1)))
      SMAX00(LØC)=MAX0(SMAX00(LØC),K3)
      SMIN00(LØC)=MIN0(SMIN00(LØC),K3)
      GØ TØ 799
  850 VN01(K3+51,LØC)=VN01(K3+51,LØC)+(V(L)*((-1)**(I+1)))
      SMAX01(LØC)=MAX0(SMAX01(LØC),K3)
      SMIN01(LØC)=MIN0(SMIN01(LØC),K3)
      GØ TØ 799
  798 J=J-GAS(LØC)
      CALL REDUCE(J,K3,K2)
      IF(K2.EQ.2)GØ TØ 799
      VN10(K3+51,LØC)=VN10(K3+51,LØC)-(V(L)*((-1)**(I+1)))
      SMAX10(LØC)=MAX0(SMAX10(LØC),K3)
      SMIN10(LØC)=MIN0(SMIN10(LØC),K3)
  799 CØNTINUE
  281 J=LS(L)
      CALL REDUCE(J,K9,K)
      SMAX(K)=MAX0(SMAX(K),K9)
      SMIN(K)=MIN0(SMIN(K),K9)
      VN(K9+51,K)=VN(K9+51,K)+ (V(L)*((-1)**(I+1)))
 24   I=I+1
      GØ TØ 17
      END
      SUBRØUTINE CLEAR
C        H.F.ØKRENT          U.C.L.A.            1967
      INTEGERS
      CØMMØN/BITS/LØØP(3000),LPATH(300,2),VPATH(300),S(300),NPATH
      EXTERNAL LSTØR
      CØMMØN/NTIMES/NTIMES
      N=0
      DØ1I=1,NPATH
      IF(LPATH(I,1).EQ.0)GØ TØ 1
      N=N+1
      LPATH(N,1)=LPATH(I,1)
      LPATH(N,2)=LPATH(I,2)
      CALL EQUAL(N,I,LSTØR,-NTIMES)
      VPATH(N)=VPATH(I)
      S(N)=S(I)
 1    CØNTINUE
```

```
      NPATH=N
      RETURN
      END
      SUBROUTINE REDUCE(I,J,K)
C     S. GRANDI   U.C.L.A.   1969
      INTEGER TAG
      COMMON/GAG/TAG(20),NSEN,SENS(50)
      K=1
      J=I
      IF(NSEN.EQ.0)GO TO 2
      M=21-NSEN
      DO 1 L=M,20
      IF(J.GT.TAG(21-L)-500)J=J-TAG(21-L)
    1 CONTINUE
    2 IF(J.LT.500)RETURN
      J=J-1000
      K=2
      RETURN
      END
      SUBROUTINE MULT(A,B,C,NMINA,NMAXA,NMINB,NMAXB)
C        S. GRANDI   U.C.L.A.   1969
C        MULTIPLIES TWO POLYNOMIALS
      DIMENSION A(101),B(101),C(101)
C        ZERO ANSWER
      DO 1 I=1,101
    1 C(I)=0.0
C        LIMITS OF MULTIPLICATION
      IMINA=NMINA+51
      IMAXA=NMAXA+51
      IMINB=NMINB+51
      IMAXB=NMAXB+51
C        MULTIPLY
      DO 2 I=IMINA,IMAXA
      DO 2 J=IMINB,IMAXB
    2 C(I+J-51)=C(I+J-51)+A(I)*B(J)
      RETURN
      END
      SUBROUTINE SENSC
C        S. GRANDI  U.C.L.A.   1969
C        FORMS THE SENSITIVITY FUNCTIONS
      INTEGER GAG,SENS
      INTEGER SMIN,SMAX,SMIN1,SMAX1,SMIN01,SMAX01,SMIN00,SMAX00,SMIN10,
     *SMAX10
      DIMENSION SMIN(2),SMAX(2)
      DIMENSION G(101),B(101)
      COMMON/GAG/GAG(20),NSEN,SENS(50)
      COMMON/BITS/VNSEN(101,2,20),SMAX1(20),SMIN1(20)
      COMMON/POLY/VN01(101,20),VN00(101,20),VN10(101,20),SMAX01(20),
     *SMIN01(20),SMAX00(20),SMIN00(20),SMAX10(20),SMIN10(20)
      COMMON/CIRCIT/CARD(50,12),VN(101,2),NE,SMAX,SMIN
      DO 305 I1=1,NE
      IF(SENS(I1).EQ.0)GO TO 305
      DO 306  II=1,NSEN
```

```
      IF(SENS(I1).EQ.GAG(II ))GO TO 307                                 UCL21150
  306 CONTINUE                                                          UCL21160
C        CHECK FOR ZERO VNO1                                            UCL21170
  307 DO 701 I=1,101                                                    UCL21180
      IF(VNO1(I,II).NE.0.0)GO TO 710                                    UCL21190
  701 CONTINUE                                                          UCL21200
      GO TO 720                                                         UCL21210
C        NON ZERO VNO1                                                  UCL21220
  710 CALL MULT(VN(1,1),VNO1(1,II),G,SMIN(1),SMAX(1),SMINO1(II),        UCL21230
     *SMAXO1(II))                                                       UCL21240
      CALL MULT(VN(1,2),VNOO(1,II),B,SMIN(2),SMAX(2),SMINOO(II),        UCL21250
     *SMAXOO(II))                                                       UCL21260
      DO 702 I=1,101                                                    UCL21270
  702 VNSEN(I,2,II)=G(I)+B(I)                                           UCL21280
      CALL MULT(VN(1,1),VN(1,2),VNSEN(1,1,II),SMIN(1),SMAX(1),SMIN(2),  UCL21290
     *SMAX(2))                                                          UCL21300
      SMIN1(II) =MINO(SMIN(1)+SMIN(2),SMIN(1)+SMINO1(II),SMIN(2)+       UCL21310
     *SMINOO(II))                                                       UCL21320
      SMAX1(II)=MAXO(SMAX(1)+SMAX(2),SMAX(1)+SMAXO1(II),SMAX(2)+        UCL21330
     *SMAXOO(II))                                                       UCL21340
      GO TO 800                                                         UCL21350
C        ZERO VNO1                                                      UCL21360
  720 DO 703 I=1,101                                                    UCL21370
      VNSEN(I,2,II)=VNOO(I,II)                                          UCL21380
  703 VNSEN(I,1,II)=VN(I,1)                                             UCL21390
      SMIN1(II)=MINO(SMINOO(II),SMIN(1))                                UCL21400
      SMAX1(II)=MAXO(SMAXOO(II),SMAX(1))                                UCL21410
  800 CALL ANSWER(VNSEN(1,1,II),SMAX1(II),SMIN1(II),I1)                 UCL21420
  305 CONTINUE                                                          UCL21430
      RETURN                                                            UCL21440
      END                                                               UCL21450
      SUBROUTINE ANSWER(VN,SMAX,SMIN,ITAG)                              UCL21460
C        DAVID  PALETZ       U.C.L.A.              1967                 UCL21470
      DIMENSION VN(101,2)                                               UCL21480
      COMMON/CIPCIT/CARD(50,12)                                         UCL21490
      COMMON/SPEED/NEXPS(100,2),AS(100,2),ASIGN(100,2),LINE(132),NUM(50,UCL21500
     13),JUNK(119)                                                      UCL21510
      INTEGER BLANK,SMAX,SMIN,ASIGN,PLUS,MINUS,AS,LINE                  UCL21520
      COMMON/DATA/AMOD(11),NS,BMOD(5),BLANK,PLUS,MINUS                  UCL21530
      COMMON/X/IDEC(10)                                                 UCL21540
      COMMON/BITS/F1(4096),VM(101,2)                                    UCL21550
      DO1I=1,100                                                        UCL21560
      AS(I,1)=NS                                                        UCL21570
    1 AS(I,2)=NS                                                        UCL21580
C                                                                       UCL21590
      L=51+SMIN                                                         UCL21600
      K=SMAX-SMIN+1                                                     UCL21610
C     SKIP  A  PAGE  AND  WRITE  HEADING                                UCL21620
      IF(ITAG.EQ.0)WRITE(6,201)                                         UCL21630
  201 FORMAT('1','TRANSFER  FUNCTION'//)                                UCL21640
      IF(ITAG.EQ.-1)WRITE(6,220)                                        UCL21650
  220 FORMAT('1','SQUARE OF WORST CASE TOLERANCE'//)                    UCL21660
      IF(ITAG.GT.0)WRITE(6,210)(CARD(ITAG,J),J=1,12)                    UCL21670
```

```
  210 FORMAT('1','SENSITIVITY TO ',12A1//)
C       SETTING THE NUMERATOR IF J=2 AND THE DENOMINATOR WHEN J=1
        J=2
C          FAN IS THE NUMERATOR FACTOR, FAD IS THE DENOMINATOR FACTOR
 11     KK=5I+SMAX
 3      IF(VN(KK,J).NE.0.)GO TO 7
        KK=KK-1
        GO TO 3
 7      IF(J.EQ.2)FAN=VN(KK,J)
        IF(J.EQ.1)FAD=VN(KK,J)
        DO2I=1,K
        ASIGN(I,J)=PLUS
        NEXPS(I,J)=I-1
        VM(I,J)=VN(L,J)
        IF(J.EQ.2.AND.VM(I,J).NE.0)VM(I,J)=-VM(I,J)/FAN
        IF(J.EQ.2.AND.ITAG.NE.0.AND.VM(I,J).NE.0)VM(I,J)=-VM(I,J)
        IF(J.EQ.1)VM(I,J)=VM(I,J)/FAD
        IF(VM(I,J))4,6,5
 4      VM(I,J)=-VM(I,J)
        ASIGN(I,J)=MINUS
        GO TO 5
    6 NEXPS(I,J)=0
        AS(I,J)=BLANK
    5 L=L+1
 2      CONTINUE
        IF(ASIGN(1,J).EQ.PLUS)ASIGN(1,J)=BLANK
        AS(1,J)=BLANK
        DO  40  N=1,K
        IF(NEXPS(N,J) .GE. 10) GO TO 41
        LEXPS=NEXPS(N,J)+1
        NUM(N,3)=IDEC(LEXPS)
        IF(LEXPS .EQ. 1 .OR. LEXPS .EQ. 2) NUM(N,3)=BLANK
        NUM(N,2)=BLANK
        NUM(N,1)=BLANK
        GO TO 40
   41 IF(NEXPS(N,J) .GE. 100) GO TO 42
        IARG=NEXPS(N,J)/10
        LEXPS=NEXPS(N,J)-10*IARG
        NUM(N,3)=IDEC(LEXPS+1)
        NUM(N,2)=IDEC(IARG+1)
        NUM(N,1)=BLANK
        GO TO 40
   42 IARG=NEXPS(N,J)/100
        LEXPS=NEXPS(N,J)-100*IARG
        LEXPS1=LEXPS/10
        LEXPS2=LEXPS-10*LEXPS1
        NUM(N,3)=IDEC(LEXPS2+1)
        NUM(N,2)=IDEC(LEXPS1+1)
        NUM(N,1)=IDEC(IARG+1)
   40 CONTINUE
        NBEG=1
    8 NEND=NBEG+7
        IF(NEND.GE.K)GO TO 9
```

```
      WRITE(6,202)((NUM(M,I),I=1,3),M=NBEG,NEND)                        UCL22210
  202 FORMAT(1H0,8(11X,3A1))                                            UCL22220
      WRITE(6,204)((ASIGN(I,J),VM(I,J),AS(I,J)),I=NBEG,NEND)            UCL22230
  204 FORMAT(1H ,8(A1,1PE10.3,A1,2X))                                   UCL22240
      NBEG=NBEG+8                                                       UCL22250
      GO TO 8                                                           UCL22260
    9 NEND=K                                                            UCL22270
      WRITE(6,202)((NUM(M,I),I=1,3),M=NBEG,NEND)                        UCL22280
      WRITE(6,204)((ASIGN(I,J),VM(I,J),AS(I,J)),I=NBEG,NEND)            UCL22290
      IF(J.EQ.1)GO TO 10                                                UCL22300
C     WRITING  THE  DIVISION  LINE                                      UCL22310
      IF(K.GE.8)GO TO 13                                                UCL22320
      LEND=K                                                            UCL22330
      GO TO 14                                                          UCL22340
   13 LEND=8                                                            UCL22350
   14 LTOTAL=LEND*14                                                    UCL22360
      DO 15 I=1,132                                                     UCL22370
   15 LINE(I)=BLANK                                                     UCL22380
      DO 16 I=1,LTOTAL                                                  UCL22390
   16 LINE(I)=MINUS                                                     UCL22400
      WRITE(6,203)(LINE(I),I=1,LTOTAL)                                  UCL22410
  203 FORMAT(/132A1/)                                                   UCL22420
      L=51+SMIN                                                         UCL22430
      K=SMAX-SMIN+1                                                     UCL22440
      J=1                                                               UCL22450
      GO TO 11                                                          UCL22460
C        FACT IS THE FUNCTION FACTOR                                    UCL22470
 10   FACT=FAN/FAD                                                      UCL22480
      WRITE(6,205)FACT                                                  UCL22490
  205 FORMAT(//' THE FUNCTION FACTOR =',3X,1PE10.3///)                  UCL22500
      WRITE(6,209)VN                                                    UCL22510
  209 FORMAT(1X,1P10E13.5)                                              UCL22520
      RETURN                                                            UCL22530
      END                                                               UCL22540
      SUBROUTINE WORSTC                                                 UCL22550
C                                                                       UCL22560
C        S. GRANDI    U.C.L.A.    1969                                  UCL22570
C        CALCULATES WORST CASE TOLERANCE                                UCL22580
C                                                                       UCL22590
      INTEGER SENS,SMAX2,SMIN2,SMAX1,SMIN1,SMIN,SMAX                    UCL22600
      COMMON/WORST1/VW RST,VW(101,2),TOL(50),SMAX2,SMIN2                UCL22610
      COMMON/GAG/GAG(20),NSEN,SENS(50)                                  UCL22620
      COMMON/BITS/VNSEN(101,2,20),SMAX1(20),SMIN1(20)                   UCL22630
      COMMON/CIRCIT/CARD(50,12),VN(101,2),NE,SMAX,SMIN                  UCL22640
      DIMENSION VWORKN(101),VW1(101),VW2(101),SMIN(2),SMAX(2)           UCL22650
C        INITIALIZE                                                     UCL22660
      I1=SMIN(1)+51                                                     UCL22670
      I2=SMAX(1)+51                                                     UCL22680
      NDEM=0                                                            UCL22690
      MAX1=0                                                            UCL22700
      MIN1=0                                                            UCL22710
      MAX2=0                                                            UCL22720
      MIN2=0                                                            UCL22730
```

```
      DØ 1 I=1,101
      VW1(I)=0.0
      VW2(I)=0.0
      VW(I,1)=0.0
    1 VW(I,2)=0.0
C        ØBTAIN SENSITIVITY SENSITIVITY FCNS FØR ALL VALID ELEMENTS
      NSEN1=0
      DØ 10 K=1,NE
      IF(SENS(K).EQ.0)GØ TØ 10
C        SQUARE THE TØLERANCE
      TØL1=TØL(K)*TØL(K)
      NSEN1=NSEN1+1
C        SQUARE SENSITIVITY FUNCTIØN
      CALL MULT(VNSEN(1,2,NSEN1),VNSEN(1,2,NSEN1),VWØRKN,
     *SMIN1(NSEN1),SMAX1(NSEN1),SMIN1(NSEN1),SMAX1(NSEN1))
      NNMIN=SMIN1(NSEN1)+SMIN1(NSEN1)
      NNMAX=SMAX1(NSEN1)+SMAX1(NSEN1)
      L=51+NNMIN
      M=51+NNMAX
C        FIND TYPE ØF DENØM
      J1=MIN0(SMIN1(NSEN1)+51,I1)
      J2=MAX0(SMAX1(NSEN1)+51,I1)
      DØ 20 J=J1,J2
      IF(V (J,1).NE.VNSEN(J,1,NSEN1))GØ TØ 21
   20 CØNTINUE
C        DENØM SAME AS TRANSFER FUNCTIØN
      DØ 11 J=L,M
   11 VW1(J)=VW1(J)+TØL1*VWØRKN(J)
      MAX1=MAX0(MAX1,NNMAX)
      MIN1=MIN0(MIN1,NNMIN)
      GØ TØ 10
C        DENØM NØT SAME AS TRANSFER FCN
   21 IF(NDEM.NE.1)NDEM=1
      DØ 12 J=L,M
   12 VW2(J)=VW2(J)+TØL1*VWØRKN(J)
      MAX2=MAX0(MAX2,NNMAX)
      MIN2=MIN0(MIN2,NNMIN)
   10 CØNTINUE
C        GET FUNCTIØN
      IF(NDEM.EQ.0)GØ TØ 30
C        BØTH TYPES PRESENT
      CALL MULT(VN(1,2),VN(1,2),VWØRKN,SMIN(2),SMAX(2),SMIN(2),SMAX(2))
      N1=SMIN(2)+SMIN(2)
      N2=SMAX(2)+SMAX(2)
      CALL MULT(VW1,VWØRKN,VW(1,2),MIN1,MAX1,N1,N2)
      DØ 31 I=1,101
   31 VW(I,2)=VW(I,2)+VW2(I)
      CALL MULT(VN(1,1),VN(1,2),VWØRKN ,SMIN(1),SMAX(1),SMIN(2),SMAX(2))
      SMIN2=SMIN(1)+SMIN(2)
      SMAX2=SMAX(1)+SMAX(2)
      CALL MULT(VWØRKN,VWØRKN,VW(1,1),SMIN2,SMAX2,SMIN2,SMAX2)
      SMIN2=SMIN2+SMIN2
      SMAX2=SMAX2+SMAX2
```

```
      GO TO 100
C        FIRST TYPE ONLY
   30 DO 40 I=1,101
   40 VW(I,2)=VW1(I)
      CALL MULT(VN(1,1),VN(1,1),VW(1,1),SMIN(1),SMAX(1),SMIN(1),SMAX(1))
      SMIN2=SMIN(1)+SMIN(1)
      SMAX2=SMAX(1)+SMAX(1)
C        FIND MIN AND MAX POWERS OF VW
  100 DO 50 I=1,101
      IF(VW(I,2).NE.0.0)GO TO 51
   50 CONTINUE
      GO TO 99
   51 L=I-51
      SMIN2=MINO(SMIN2,L)
      DO 52 J=1,101
      I=102-J
      IF(VW(I,2).NE.0.0)GO TO 53
   52 CONTINUE
   53 L=I-51
      SMAX2=MAXO(SMAX2,L)
   99 RETURN
      END
      SUBROUTINE INVERT
C    A. R. TYRRILL,  UCLA,  6/21/68
      COMPLEX Z,CA(50),CB(50),G1,G2
      DIMENSION FCTN(512),TEE(512),PH(512)
      COMMON/SPEED/F2(2048)
      COMMON/BITS/F1(4096),B(50),C(50),NXFR,A(50),NXTI,STEP,TOTEE,
     ANTYPE,AMP1,AMP2,FREQ1,FREQ2,NPLOT,
     1NUMFR,NUMTI,NF(11),NR(11),MOR,MORNUM,NTAY,ISN,SCMAG,SCFREQ,SIGMA,
     2T1,T2,T3,DMOD(6),ABSC,ORD
      EQUIVALENCE (F2(1),FCTN(1)),(F2(513),TEE(1)),(F2(1025),CA(1)),
     1(F2(1125),CB(1)),(F2(1225),PH(1))
      DATA FREQ,AMAG,PHA,TIM/'FREQ','MAG ','PHA ','TIME'/
  112 FORMAT (///// 10X, '**** NOT RATIONAL, CANNOT INVERT.')
  115 FORMAT (///// 10X, '**** TOO SIMPLE.  DO BY HAND.')
  120 FORMAT (//// 5X, 'THE TRANSFORM OF THE RESPONSE IS ---' // 20X,
     2'EXP. OF S', 6X, 'NUMER. COEFFS.', 6X, 'DENOM. COEFFS.' / 20X,
     3'---------', 6X, '--------------', 6X, '--------------')
  121 FORMAT (23X, I2, 3X, 1P2E20.5)
   12 FORMAT('0',22X,'+ ',1P2E20.5)
   14 FORMAT(//10X,'**** THE PLOT IS OF THE NON-CONSTANT REMAINDER TERM
     *ONLY')
      NFLAG=0
      IF (STEP .LE. 0.0)  STEP = 1.0
      SCFREQ = STEP
      NF(1) = 1
      DO 116  I = 2, 11
  116 NF(I) = 2*NF(I-1)
      INTS = TOTEE/STEP
      IF (INTS .LT. 10)  INTS = 10
      DO 117  I = 1, 10
      IF (NF(I) - INTS)  117, 118, 118
```

```
117 CØNTINUE                                                            UCL23800
    I = 10                                                              UCL23810
118 NXTI = I - 1                                                        UCL23820
    NXFR = I + 1                                                        UCL23830
    DØ 119  I = 1, NXFR                                                 UCL23840
119 NR(I) = NF(NXFR + 1 - I)                                            UCL23850
    NUMFR = 2*NR(1)                                                     UCL23860
    NUMTI = NR(2)                                                       UCL23870
    IF(NTYPE.LT.6)CALL INPUT                                            UCL23880
    DØ 122  I = 1, 50                                                   UCL23890
    IF (B(51-I))  123, 122, 123                                         UCL23900
122 CØNTINUE                                                            UCL23910
123 MØR = 51 - I                                                        UCL23920
    DØ 124  I = 1, 50                                                   UCL23930
    IF (A(51-I)) 125, 124, 125                                          UCL23940
124 CØNTINUE                                                            UCL23950
125 MØRNUM = 51 - I                                                     UCL23960
    MRMAX = MAX0(MØR, MØRNUM)                                           UCL23970
    IF (MØR-MØRNUM)110,10,111                                           UCL23980
110 IF(NTYPE.GT.5)GØ TØ 111                                             UCL23990
    WRITE(6,112)                                                        UCL24000
    RETURN                                                              UCL24010
 10 IF(NTYPE.GT.5)GØ TØ 111                                             UCL24020
    CØNST=A(MØR)/B(MØR)                                                 UCL24030
    MØRNUM=MØRNUM-1                                                     UCL24040
    DØ 11 I=1,MØRNUM                                                    UCL24050
 11 A(I)=A(I)-CØNST*B(I)                                                UCL24060
    A(MØR)=0.0                                                          UCL24070
    NFLAG=1                                                             UCL24080
111 WRITE(6,120)                                                        UCL24090
    DØ 126  I = 1, MRMAX                                                UCL24100
    J = I - 1                                                           UCL24110
126 WRITE (6, 121) J, A(I), B(I)                                        UCL24120
    IF(NFLAG.NE.1) GØ TØ 13                                             UCL24130
    WRITE(6,12)CØNST                                                    UCL24140
    WRITE(6,14)                                                         UCL24150
 13 IF(NTYPE.GT.5)GØ TØ 1                                               UCL24160
    IF (MØR - 2)  113, 113, 127                                         UCL24170
113 WRITE (6, 115)                                                      UCL24180
    RETURN                                                              UCL24190
127 ABSC=TIM                                                            UCL24200
    ØRD=AMAG                                                            UCL24210
    CALL SCALE                                                          UCL24220
    CALL RØUTH                                                          UCL24230
    CALL SAMPLE                                                         UCL24240
    CALL FLIP                                                           UCL24250
    CALL ADJUST                                                         UCL24260
    RETURN                                                              UCL24270
C      H.F.ØKRENT           U.C.L.A.                    1969            UCL24280
1   IF(NTYPE.NE.9)GØ TØ 6                                               UCL24285
    NUMTI=(FREQ2-FREQ1)/STEP                                            UCL24286
    Z=CMPLX(FREQ1,0.E0)                                                 UCL24287
    GØ TØ 7                                                             UCL24288
```

```
6     NUMTI=ALØG10(ABS(FREQ2/FREQ1))/ALØG10(ABS(STEP))                  UCL24290
      Z=CMPLX(0.E0,6.2832*FREQ1)                                          UCL24295
7     IF(NUMTI.GT.512)NUMTI=512                                           UCL24300
      IF(NUMTI.LE.0)NUMTI=100                                             UCL24310
      DØ2J=1,MRMAX                                                        UCL24330
      K=MRMAX-J+1                                                         UCL24340
      CA(K)=CMPLX(A(J),0.E0)                                              UCL24350
2     CB(K)=CMPLX(B(J),0.E0)                                              UCL24360
      DØ4I=1,NUMTI                                                        UCL24370
      G1=(0.E0,0.E0)                                                      UCL24380
      G2=(0.E0,0.E0)                                                      UCL24390
      K=MRMAX-1                                                           UCL24400
      DØ3J=1,K                                                            UCL24410
      G1=(G1+CA(J))*Z                                                     UCL24420
3     G2=(G2+CB(J))*Z                                                     UCL24430
      G1=G1+CA(MRMAX)                                                     UCL24440
      G2=G2+CB(MRMAX)                                                     UCL24450
      G1=G1/G2                                                            UCL24460
      IF(NTYPE.NE.9)GØ TØ 8                                               UCL24461
      TEE(I)=REAL(Z)                                                      UCL24462
      FCTN(I)=REAL(C1)                                                    UCL24463
      Z=CMPLY(FREQ1+(STEP*I),0.E0)                                        UCL24464
      GØ TØ 4                                                             UCL24465
8     TEE(I)=AIMAG(Z)/6.2832                                              UCL24470
      FCTN(I)=SQRT((REAL(G1)**2)+(AIMAG(G1)**2))                          UCL24480
      IF(NTYPE.EQ.8)FCTN(I)=SQRT(FCTN(I))                                 UCL24490
      PH(I)=ATAN2(AIMAG(G1),REAL(G1))*57.2958                             UCL24500
      Z=CMPLX(0.E0,6.2832*FREQ1*(STEP**I))                                UCL24510
4     CØNTINUE                                                            UCL24515
      ABSC=FREQ                                                           UCL24520
      ØRD=AMAG                                                            UCL24530
      CALL PRTPLT                                                         UCL24540
      IF(NTYPE.GE.8)RETURN                                                UCL24550
      ØRD=PHA                                                             UCL24560
      DØ5I=1,NUMTI                                                        UCL24570
5     FCTN(I)=PH(I)                                                       UCL24580
      CALL PRTPLT                                                         UCL24590
      RETURN                                                              UCL24600
      END                                                                 UCL24610
      SUBRØUTINE INPUT                                                    UCL24620
C     A. R. TYRRILL,  UCLA,   6/21/68                                     UCL24630
      CØMMØN/BITS/F1(4096),B(50),C(50),NXFR,A(50),NXTI,STEP,TØTEE,        UCL24640
     ANTYPE,AMP1,AMP2,FREQ1,FREQ2,NPLØT,                                  UCL24650
     1NUMFR,NUMTI,NF(11),NR(11),MØR,MØRNUM,NTAY,ISN,SCMAG,SCFREQ,SIGMA,   UCL24660
     2T1,T2,T3,DMØD(6),ABSC,ØRD                                           UCL24670
  310 FØRMAT (1H1, 10X, 'SPECIFIED INPUT IS ---')                         UCL24680
  316 FØRMAT (20X, 'IMPULSE, STRENGTH = ', 1PE12.5)                       UCL24690
  317 FØRMAT (20X,'STEP, MAGNITUDE = ', 1PE12.5)                          UCL24700
  318 FØRMAT (20X, 'EXPØNENTIAL, MAGNITUDE = ', 1PE12.5, 5X, 'CØEFF. = 'UCL24710
     2, 1PE12.5)                                                          UCL24720
  319 FØRMAT (20X, 'SINE, MAGNITUDE = ', 1PE12.5, 5X, 'FREQUENCY = ',     UCL24730
     2 1PE12.5, ' HZ.')                                                   UCL24740
  321 FØRMAT (20X, 'FINITE PULSE, MAGNITUDE = ', 1PE12.5, 5X, 'DURATIØN UCL24750
```

```
      2= ', 1PE12.5, ' SEC.')
  322 FØRMAT (20X, 'PULSE TRAIN, WAVELENGTH = ', 1PE12.5, ' SEC.', 5X,
     2'PULSE DURATIØN = ', 1PE12.5, ' SEC.'/33X, 'TØTAL MAG. = ', 1PE12.
     35, 1 X, 'BACKGRØUND = ', 1PE12.5)
      WRITE(6, 310)
      GØ TØ (311, 312, 313, 314, 315), NTYPE
      WRITE (6, 316) AMP1
      GØ TØ 320
  311 WRITE (6, 317) AMP1
      GØ TØ 320
  312 WRITE (6, 318) AMP1, FREQ1
      GØ TØ 320
  313 WRITE (6, 319) AMP1, FREQ1
      GØ TØ 320
  314 WRITE (6, 321) AMP1, FREQ1
      GØ TØ 320
  315 WRITE (6, 322) FREQ1, FREQ2, AMP1, AMP2
  320 ISN = 1
      NTAY = 1
      SCMAG = AMP1
      GØ TØ (305, 301, 303, 305, 305), NTYPE
  300 RETURN
  301 DØ 302  I = 1, 48
  302 B(50-I) = B(49-I) - B(50-I)*FREQ1
      B(1) = -B(1)*FREQ1
      RETURN
  303 IF (FREQ1 .LE. 0.0)  FREQ1= 1.0
      ØMEGA = FREQ1*6.2831853
      ØMGSQ = ØMEGA**2
      SCMAG = SCMAG*ØMEGA
      DØ 304  I = 1, 48
  304 B(51-I) = B(49-I) + B(51-I)*ØMGSQ
      B(1) = B(1)*ØMGSQ
      B(2) = B(2)*ØMGSQ
      RETURN
  305 DØ 306  I = 1, 48
  306 B(50-I) = B(49-I)
      B(1) = 0.0
      IF (FREQ1 .LE. 0.0)  FREQ1= 1.0
      T1 = FREQ1
      IF (NTYPE-4) 308, 307, 309
  307 ISN = 2
      IF (T1 .LE. 5*STEP)  NTAY = 0
  308 RETURN
  309 ISN = 3
      T3 = AMP2/AMP1
      IF (FREQ2 .LE. 0.0 .ØR. FREQ2 .GE. FREQ1)  FREQ2 = 0.5*FREQ1
      T2 = FREQ2
      IF (T2 .LE. 5*STEP)  NTAY = 0
      RETURN
      END
      SUBRØUTINE SAMPLE
C     A. R. TYRRILL, UCLA, 6/21/68
```

```
      CØMPLEX*8 RSPNS(2048), EXCMP(1024), SPØW(50), S, SUMNM, SUMDN          UCL25290
      CØMPLEX*8 ZØWIE, ZZ, Z1                                                UCL25300
      DØUBLE PRECISIØN X, Y                                                  UCL25310
      CØMMØN/BITS/RSPNS    ,B(50),C(50),NXFR,A(50),NXTI,STEP,TØTEE,          UCL25320
     ANTYPE,AMP1,AMP2,FREQ1,FREQ2,NPLØT,                                     UCL25330
     1NUMFR,NUMTI,NF(11),NR(11),MØR,MØRNUM,NTAY,ISN,SCMAG,SCFREQ,SIGMA,      UCL25340
     2T1,T2,T3,DMØD(6),ABSC,ØRD                                              UCL25350
      CØMMØN/SPEED/EXCMP                                                     UCL25360
      X = 6.2831853071795860+00/FLØAT(NUMFR)                                 UCL25370
      Y = 0.00+00                                                            UCL25380
      SPØW(1) = (1.0,0.0)                                                    UCL25390
      DØ 130  I = 1, NUMFR                                                   UCL25400
      S = CMPLX(SIGMA, SNGL(Y))                                              UCL25410
      IF (I - NR(1))  134, 134, 133                                          UCL25420
  134 ZØWIE = CMPLX(0.0, SNGL(Y))                                            UCL25430
      EXCMP(I) = CEXP(ZØWIE)                                                 UCL25440
  133 Y = X + Y                                                              UCL25450
      SUMNM=(0.0,0.0)                                                        UCL25460
      SUMDN = (0.0,0.0)                                                      UCL25470
      DØ 131  J = 2, MØR                                                     UCL25480
  131 SPØW(J) = SPØW(J-1)*S                                                  UCL25490
      DØ 132  J = 1, MØR                                                     UCL25500
      SUMNM = A(J)*SPØW(J) + SUMNM                                           UCL25510
  132 SUMDN = B(J)*SPØW(J) + SUMDN                                           UCL25520
      IF (ISN-2)  135, 136, 137                                              UCL25530
  135 ZZ = (1.0, 0.0)                                                        UCL25540
      GØ TØ 130                                                              UCL25550
  136 ZZ = 1.0 - CEXP(-T1*S/SCFREQ)                                          UCL25560
      GØ TØ 130                                                              UCL25570
  137 Z1 = CEXP(-T1*S/SCFREQ)                                                UCL25580
      ZZ = (1.0 + (T3-1.0)*CEXP(-T2*S/SCFREQ) - T3*Z1)/(1.0 - Z1)            UCL25590
  130 RSPNS(I) = SUMNM*ZZ/SUMDN                                              UCL25600
      RSPNS(1) = RSPNS(1)/2.0                                                UCL25610
      RETURN                                                                 UCL25620
      END                                                                    UCL25630
      SUBRØUTINE FLIP                                                        UCL25640
C     A. R. TYPRILL,  UCLA,  6/21/68                                         UCL25650
      CØMPLEX RSPNS(2048), EXCMP(1024), Z                                    UCL25660
      DØUBLE PRECISIØN X, Y                                                  UCL25670
      CØMMØN/BITS/RSPNS    ,B(50),C(50),NXFR,A(50),NXTI,STEP,TØTEE,          UCL25680
     ANTYPE,AMP1,AMP2,FREQ1,FREQ2,NPLØT,                                     UCL25690
     1NUMFR,NUMTI,NF(11),NR(11),MØR,MØRNUM,NTAY,ISN,SCMAG,SCFREQ,SIGMA,      UCL25700
     2T1,T2,T3,DMØD(6),ABSC,ØRD                                              UCL25710
      CØMMØN/SPEED/EXCMP                                                     UCL25720
  136 FØRMAT (//// 20X, 'SIGMA =', 1PE12.5)                                  UCL25730
      SIGACT = SIGMA/SCFREQ                                                  UCL25740
      WRITE (6, 136) SIGACT                                                  UCL25750
  102 NXH =  NXFR/2                                                          UCL25760
      DØ 106  L = 1, NXH                                                     UCL25770
      NFØ = NF(L)                                                            UCL25780
      NDIF = NR(L) - NFØ                                                     UCL25790
      N2FØ = NF(L + 1)                                                       UCL25800
      DØ 106  K = 1, NFØ                                                     UCL25810
```

```
      JUMP = (K-1)*2*NR(L)
  105 DO 106  J = NFO, NDIF, N2FO
      JI = J + JUMP
      JF = JI + NDIF
      DO 106  I = 1, NFO
      JJI = JI + I
      JJF = JF + I
      Z = RSPNS(JJI)
      RSPNS(JJI) = RSPNS(JJF)
  106 RSPNS(JJF) = Z
      DO 110  L = 1, NXTI
      NFO = NF(L)
      NRE = NR(L)
      DO 110  K = 1, NRE
      JI = (K-1)*NF(L+1)
      JF = JI + NFO
      DO 110  J = 1, NFO
      JJI = JI + J
      JJF = JF + J
      Z = RSPNS(JJF)*EXCMP((J-1)*NRE + 1)
      RSPNS(JJF) = RSPNS(JJI) - Z
  110 RSPNS(JJI) = RSPNS(JJI) + Z
      DO 108  K = 1, 2
      NFO = NF(3-K)
      DO 108  J = 1, NFO
      JI = (J-1)*NR(1)
      JF = (J-1)*NR(2)
      DO 108  I = 1, NUMTI
      JJI = JI + I
  108 RSPNS(JF+I) = RSPNS(JJI) + RSPNS(JJI+NUMTI)*EXCMP((I-1)*NFO + 1)
      RETURN
      END
      SUBROUTINE ADJUST
C     A. R. TYRRILL, UCLA, 6/21/68
      COMPLEX RSPNS(2048)
      DIMENSION FNP(4)
      COMMON/BITS/RSPNS,FACTL(50),TERM(50),NXFR,POW(50),NXTI,STEP,
     BTOTEE,NTYPE,AMP1,AMP2,FREQ1,FREQ2,NPLOT,
     1NUMFR,NUMTI,NF(11),NR(11),MOR,MORNUM,NTAY,ISN,SCMAG,SCFREQ,SIGMA,
     2T1,T2,T3,DMOD(6),ABSC,ORD
      COMMON/SPEED/FCTN(512),TEE(512)
      DLW = 2.0*SCMAG/(SCFREQ*FLOAT(NUMFR))
      DO 150  I = 1, NUMTI
      TEE(I) = (FLOAT(I-1))*SCFREQ
  150 FCTN(I) = DLW*(REAL(RSPNS(I)))
      IF (SIGMA) 151, 153, 151
  151 SGQ = SIGMA/SCFREQ
      DO 152  I = 2, NUMTI
  152 FCTN(I) = FCTN(I)*EXP(SGQ*TEE(I))
  153 CALL TAYLOR
      DLW = SCMAG/SCFREQ
      FCTN(1) = TERM(2)*DLW
      IF (NTAY) 156, 156, 159
```

```
  159 FACTL(2) = 1.0
      DØ 154  I = 3, 50
  154 FACTL(I) = FACTL(I-1)*(I-2)*0.25
      DØ 155  I = 2, 50
  155 TERM(I) = TERM(I)/FACTL(I)
      DØ 157  I = 1, 4
      FNP(I) = 0.0
      PØW(2) = 1.0
      DØ 158  J = 2, 50
      FNP(I) = FNP(I) + TERM(J)*PØW(J)
      IF (FNP(I) .GT. 1.0E+03)  GØ TØ 156
  158 PØW(J+1) = PØW(J)*I*0.25
  157 FCTN(I+1) = FNP(I)*DLW
  156 CALL PRTPLT
      RETURN
      END
      SUBRØUTINE TAYLØR
C     A. R. TYRRILL,  UCLA,   6/21/68
      CØMMØN/BITS/C(50),AMØD(4046),B(50),TERM(50),NXFR,A(50),NXTI,STEP,
     XTØTEE,
     ANTYPE,AMP1,AMP2,FREQ1,FREQ2,NPLØT,
     1NUMFR,NUMTI,NF(11),NR(11),MØR,MØRNUM,NTAY,ISN,SCMAG,SCFREQ,SIGMA,
     2T1,T2,T3,DMØD(6),ABSC,ØRD
      DØ 217  I = 1, MØR
  217 C(I) = A(I)
      MØRDIF = MØR - MØRNUM
  210 DØ 214  I = 1, 50
      TERM(I) = 0.0
      IF (MØRDIF-I)  213, 212, 212
  213 TERM(I) = C(MØR)/B(MØR)
      IF (ABS(TERM(I)) .GT. 1.0E+60)  GØ TØ 218
      DØ 216  J = 1, MØR
  216 C(J) = C(J) - TERM(I)*B(J)
  212 DØ 215  J = 1, MØR
      JJ = MØR + 1 - J
  215 C(JJ+1) = C(JJ)
  214 C(1) = 0.0
      RETURN
  218 NTAY = 0
      RETURN
      END
      SUBRØUTINE PRTPLT
C     A. R. TYRRILL,  UCLA,   6/21/68
      DIMENSIØN SYMBØL(5)
      CØMMØN/SPEED/FCTN(512),TEE(512),P(132),DHI,DLØ,DGP
      CØMMØN/BITS/F1(4095),B(50),C(50),NXFR,A(50),NXTI,STEP,TØTEE,
     ANTYPE,AMP1,AMP2,FREQ1,FREQ2,NPLØT,
     1NUMFR,NUMTI,NF(11),NR(11),MØR,MØRNUM,NTAY,ISN,SCMAG,SCFREQ,SIGMA,
     2T1,T2,T3,DMØD(6),ABSC,ØRD
      DATA SYMBØL/' ','.','X','Ø','I'/
  175 FØRMAT (1PE10.3, 122A1)
  181 FØRMAT (/////10X, 16HGREATEST VALUE =, 1PE13.5, 10X, 14HLØWEST VAL
     2UE =, 1PE13.5, 10X, 10HINTERVAL =, 1PE13.5 ///// )
```

```
 183 FØRMAT (10X, 122A1)                                                      UCL26880
 191 FØRMAT (4(1PE16.3, 1PE16.5))                                             UCL26890
 194  FØRMAT(///// 4(10X, A4, 11X, A4,3X)/)                                   UCL26900
      WRITE(6,194)(ABSC,ØRD,I=1,4)                                            UCL26910
      NFØ=NUMTI/4                                                             UCL26920
      DØ 190  I = 1, NFØ                                                      UCL26930
      J = NFØ + I                                                             UCL26940
      JJ = NFØ + J                                                            UCL26950
 190 WRITE (6, 191) TEE(I), FCTN(I), TEE(J), FCTN(J), TEE(JJ), FCTN(JJ)UCL26960
    2, TEE(NFØ+JJ), FCTN(NFØ+JJ)                                              UCL26970
      IF (NPLØT .LE. 0)  RETURN                                               UCL26980
      DHI = 0.0                                                               UCL26990
      DLØ = DHI                                                               UCL27000
      DØ 176  I = 1, NUMTI                                                    UCL27010
      IF(FCTN(I) .GT. DHI)  DHI = FCTN(I)                                     UCL27020
      IF(FCTN(I) .LT. DLØ)  DLØ = FCTN(I)                                     UCL27030
 176 CØNTINUE                                                                 UCL27040
      DGP = (DHI - DLØ)/115.0                                                 UCL27050
      KZERØ = 14.5 - (DLØ/DGP)                                                UCL27060
      IF (KZERØ-11) 177, 177, 178                                             UCL27070
 178 IF (132-KZERØ) 177, 177, 179                                             UCL27080
 177 KZERØ = 1                                                                UCL27090
 179 DØ 180  I = 11, 132                                                      UCL27100
 180 P(I) = SYMBØL(2)                                                         UCL27110
      WRITE (6, 181) DHI, DLØ, DGP                                            UCL27120
      LL = 0                                                                  UCL27130
      DØ 182  I = 1, NUMTI, NPLØT                                             UCL27140
      K = ((FCTN(I) - DLØ)/DGP) + 14.5                                        UCL27150
 189 P(K) = SYMBØL(3)                                                         UCL27160
      IF (LL) 171, 171, 170                                                   UCL27170
 170 WRITE (6, 183) (P(J), J = 11, 132)                                       UCL27180
      LL = LL - 1                                                             UCL27190
      GØ TØ 172                                                               UCL27200
 171 WRITE (6, 175) TEE(I), (P(J), J = 11, 132)                               UCL27210
      LL = 9                                                                  UCL27220
 172 DØ 184  J = 10, 132                                                      UCL27230
 184 P(J) = SYMBØL(1)                                                         UCL27240
      IF (LL) 185, 185, 188                                                   UCL27250
 185 DØ 186  II = 1, 3                                                        UCL27260
      P(11+II) = SYMBØL(2)                                                    UCL27270
      P(KZERØ+II) = SYMBØL(2)                                                 UCL27280
      P(KZERØ-II) = SYMBØL(2)                                                 UCL27290
 186 P(132-II) = SYMBØL(2)                                                    UCL27300
 188 P(132) = SYMBØL(5)                                                       UCL27310
      P(KZERØ) = SYMBØL(2)                                                    UCL27320
 182 P(11) = SYMBØL(5)                                                        UCL27330
      WRITE(6,183)(SYMBØL(2),I=11,132)                                        UCL27340
      RETURN                                                                  UCL27350
      END                                                                     UCL27360
      SUBRØUTINE SCALE                                                        UCL27370
C    A. R. TYRRILL,  UCLA,  6/21/68                                           UCL27380
      CØMMØN/BITS/F1(4096),B(50),C(50),NXFR,A(50),NXTI,STEP,TØTEE,            UCL27390
     ANTYPE,AMP1,AMP2,FREQ1,FREQ2,NPLØT,                                      UCL27400
```

```
     1NUMFR,NUMTI,NF(11),NR(11),MØR,MØRNUM,NTAY,ISN,SCMAG,SCFREQ,SIGMA, UCL27410
     2T1,T2,T3,DMØD(6),ABSC,ØRD                                          UCL27420
      MRLØ = MØR - 1                                                     UCL27430
      IF (SCFREQ-1.0) 231, 232, 231                                      UCL27440
  231 DØ 233  I = 1, MRLØ                                                UCL27450
      DØ 233  J = 1, I                                                   UCL27460
      A(MØR-I) = A(MØR-I)*SCFREQ                                         UCL27470
  233 B(MØR-I) = B(MØR-I)*SCFREQ                                         UCL27480
  232 SCMAG = 1.0/B(MØR)                                                 UCL27490
      DØ 234  I = 1, MØR                                                 UCL27500
  234 B(I) = B(I)*SCMAG                                                  UCL27510
      SCMAG = SCMAG*A(MØRNUM)                                            UCL27520
      DØ 235 I = 1, MØRNUM                                               UCL27530
  235 A(I) = A(I)/A(MØRNUM)                                              UCL27540
      RETURN                                                             UCL27550
      END                                                                UCL27560
      SUBRØUTINE RØUTH                                                   UCL27570
    A. R. TYRRILL,  UCLA,  6/21/68                                       UCL27580
      LØGICAL LØG1                                                       UCL27590
      CØMMØN/BITS/BINØ(50,50),D(26,50),AMØD(296)                         UCL27600
     1                     ,B(50),C(50),NXFR,A(50),NXTI,STEP,TØTEE,      UCL27610
     ANTYPE,AMP1,AMP2,FREQ1,FREQ2,NPLØT,                                 UCL27620
     1NUMFR,NUMTI,NF(11),NR(11),MØR,MØRNUM,NTAY,ISN,SCMAG,SCFREQ,SIGMA, UCL27630
     2T1,T2,T3,DMØD(6),ABSC,ØRD                                          UCL27640
      DØ 230  J = 1, MØR                                                 UCL27650
      BINØ(1,J) = 1.0                                                    UCL27660
      DØ 230  I = 2, MØR                                                 UCL27670
  230 BINØ(I,J) = 0.0                                                    UCL27680
      DØ 231  I = 2, MØR                                                 UCL27690
      DØ 231  J = I, MØR                                                 UCL27700
  231 BINØ(I,J) = BINØ(I-1,J-1) + BINØ(I,J-1)                            UCL27710
      SIGTRY = 0.005*NF(13-NXFR)                                         UCL27720
      SIGMA = 0.0                                                        UCL27730
  232 CALL CALCR(LØG1)                                                   UCL27740
      IF (LØG1)  GØ TØ 233                                               UCL27750
      SIGMA = SIGMA + SIGTRY                                             UCL27760
      GØ TØ 232                                                          UCL27770
  233 SIGTRY = (-0.1)*SIGTRY                                             UCL27780
      SHIFT = 0.0035*NF(13-NXFR)                                         UCL27790
  234 SIGMA = SIGMA + SIGTRY                                             UCL27800
      CALL CALCR(LØG1)                                                   UCL27810
      IF (SIGMA+SHIFT)  236, 236, 235                                    UCL27820
  235 IF (LØG1)  GØ TØ 234                                               UCL27830
      SIGMA = SIGMA + SHIFT                                              UCL27840
      RETURN                                                             UCL27850
  236 SIGMA = 0.0                                                        UCL27860
      RETURN                                                             UCL27870
      END                                                                UCL27880
      SUBRØUTINE CALCR (LØG1)                                            UCL27890
C   A. R. TYRRILL,  UCLA,  6/21/68                                       UCL27900
      LØGICAL LØG1                                                       UCL27910
      CØMMØN/BITS/BINØ(50,50),D(26,50),SIGPØW(50),AMØD(246)              UCL27920
     1                     ,B(50),C(50),NXFR,A(50),NXTI,STEP,TØTEE,      UCL27930
```

```
     ANTYPE,AMP1,AMP2,FREQ1,FREQ2,NPLOT,
    1NUMFR,NUMTI,NF(11),NR(11),MOR,MORNUM,NTAY,ISN,SCMAG,SCFREQ,SIGMA,
    2T1,T2,T3,DMOD(6),ABSC,ORD
      LOG1 = .TRUE.
      IF (SIGMA .LT. 1.0E-04 .AND. SIGMA .GT. -1.0E-04)  SIGMA = 0.0
      SIGPOW(1) = SIGMA
      DO 240  I = 2, MOR
  240 SIGPOW(I) = SIGPOW(I-1)*SIGPOW(1)
      C(MOR) = B(MOR)
      MRLO = MOR - 1
      DO 248  I = 1, MPLO
      C(I) = B(I)
      II = I + 1
      DO 241  J = II, MOR
      T = B(J)*BINO(I,J)*SIGPOW(J-I)
  241 C(I) = C(I) + T
      IF (C(I))  247, 247, 248
  248 CONTINUE
      MH = MOR/2 + 1
      DO 242  I = 1, MH
      DO 242  J = 1, MOR
  242 D(I,J) = 0.0
      MH = (MOR + 1)/2
      DO 243  I = 1, MH
  243 D(I,1) = C(MOR-2*(I-1))
      MH = MOR/2
      DO 244  I = 1, MH
  244 D(I,2) = C(MOR+1-2*I)
      DO 245  J = 3, MOR
      DO 246 I = 1, MH
  246 D(I,J) = (D(1,J-1)*D(I+1,J-2)-D(1,J-2)*D(I+1,J-1))/D(1,J-1)
      IF (D(1,J))  247, 247, 245
  245 CONTINUE
      RETURN
  247 LOG1 = .FALSE.
      RETURN
      END
      SUBROUTINE ROOTS
C        H.F.OKRENT        SVOBODA POLYNOMIAL ROOTFINDER      1969
      COMPLEX*16 Q
      DIMENSION SMIN(2),SMAX(2)
      INTEGER SMAX,SMIN
      COMPLEX     D(50,2)
      REAL    A(50),C,YMAX,YMIN,XMAX,XMIN,H(5),E
      DIMENSION N1(5),NRT(50,3,2),IN(2),JN(2)
      COMPLEX  P,T,P(50),S,SN,SS,SW,SE,F,FN,FS,FE,FW,V(5),W(5)
      REAL    G,GN,GS,GW,GE
      COMPLEX POLE
      COMMON/DATA/DATA(17),BLANK
      COMMON/A/CHAR(80)
      COMMON/SPEED/POLE(50),NPOLES,DUM,S,SN,SS,SW,SE,F,FN,FS,FW,FE,G,GN,
     1GS,GW,GE,E,D,A,N1,NRT,IN,JN,C,B
      COMMON/CIRCIT/ CARD(600),VN(101,2),NE,SMAX,SMIN
```

```
      EQUIVALENCE (G,H(1)),(F,V(1)),(S,W(1))                            UCL28470
      NP0LLS=0                                                          UCL28480
      D025N9=1,2                                                        UCL28490
      N=SMAX(N9)-SMIN(N9)+1                                             UCL28500
      N6=SMIN(N9)+51                                                    UCL28510
      N7=SMAX(N9)+51                                                    UCL28520
      D040I=N6,N7                                                       UCL28530
      IF(VN(I,N9).NE.0.E0)G0 T0 1                                       UCL28540
40    C0NTINUE                                                          UCL28550
      G0 T0 25                                                          UCL28560
1     N6=I                                                              UCL28570
      D052I=N6,N7                                                       UCL28580
      J=N7+N6-I                                                         UCL28590
      IF(VN(J,N9).NE.0.E0)G0 T0 53                                      UCL28600
52    C0NTINUE                                                          UCL28610
      G0 T0 25                                                          UCL28620
53    N7=J                                                              UCL28630
      N=N7-N6+1                                                         UCL28640
      IF(N.LE.1)G0 T0 25                                                UCL28650
      M=N-1                                                             UCL28660
      WRITE(6,203)M                                                     UCL28670
203   F0RMAT('1',46X,'*** SV0B0DA P0LYN0MIAL R00TFINDER ***',32X,'H.F.0KUCL28680
     1RENT'/116X,'JAN. 1969'//10X,'P0LYN0MIAL 0F DEGREE',I4,'  --'/)    UCL28690
      JN(1)=0                                                           UCL28700
      JN(2)=0                                                           UCL28710
      N8=0                                                              UCL28720
      IN(1)=0                                                           UCL28730
      IN(2)=0                                                           UCL28740
      NPASS=0                                                           UCL28750
      J=(N/6)+1                                                         UCL28760
      K=0                                                               UCL28770
      D041I=1,J                                                         UCL28780
      L=MIN0(M,K+5)                                                     UCL28790
      WRITE(6,202)(N2,N2=K,L)                                           UCL28800
202   F0RMAT(1X,6I22)                                                   UCL28810
      WRITE(6,206)(VN(N6+N2,N9),N2=K,L)                                 UCL28820
206   F0RMAT(6(F20.10,' X')/)                                           UCL28830
41    K=K+6                                                             UCL28840
      WRITE(6,204)                                                      UCL28850
204   F0RMAT(/4X,'ACCURACY REQUESTED -- 6 SIGNIFICANT FIGURES'//45X,'REAUCL28860
     1L PART     IMAGINARY PART    EXP0NENT')                           UCL28870
      WRITE(6,205)                                                      UCL28880
205   F0RMAT(42X,45('-'))                                               UCL28890
      C=0.E0                                                            UCL28900
      D02I=1,N                                                          UCL28910
      J=N7+1-I                                                          UCL28920
      A(I)=VN(J,N9)                                                     UCL28930
2     B(I)= CMPLX(A(I),0.E0)                                            UCL28940
45    L=1                                                               UCL28950
C.......THE SCANNING R0UTINE............................................UCL28951
31    XMIN=-1.E0                                                        UCL28960
      YMIN=-SQRT(ABS(1.E0-(XMIN**2)))-6.25E-2                           UCL28970
      NR00T=JN(3-L)                                                     UCL28980
```

```
      KR=-16                                                            UCL28990
 8    KI=YMIN*16.                                                       UCL29000
      YMIN KI*6.25E-2                                                   UCL29010
      YMAX=-YMIN                                                        UCL29020
 7    S= CMPLX(XMIN,YMIN)                                               UCL29030
      SN=S+(0.E0,6.25E-2)                                               UCL29040
      SS=S-(0.E0,6.25E-2)                                               UCL29050
      SW=S-(6.25E-2,0.E0)                                               UCL29060
      SE=S+(6.25E-2,0.E0)                                               UCL29070
      NTAG=IN(L)                                                        UCL29080
      N4=1                                                              UCL29090
C........THE CALCULATION ROUTINE-- USING HORNER'S TECHNIQUE.............UCL29091
 33   DO26I=N4,5                                                        UCL29100
 26   V(I)=(0.E0,0.E0)                                                  UCL29110
      P=B(N)                                                            UCL29120
      DO3I=1,M                                                          UCL29130
      T=B(I)                                                            UCL29140
      DO3K=N4,5                                                         UCL29150
 3    V(K)=(V(K)+T)*W(K)                                                UCL29160
      DO27I=N4,5                                                        UCL29170
      V(I)=V(I)+B(N)                                                    UCL29180
      IF(NTAG.EQ.0)GO TO 27                                             UCL29190
      DO47K=1,NTAG                                                      UCL29200
      T=W(I)-D(K,L)                                                     UCL29210
      IF(CABS(T).LE.1.E-60)T=(1.E-60,1.E-60)                            UCL29220
      P=P/T                                                             UCL29230
 47   V(I)=V(I)/T                                                       UCL29240
 27   H(I)= CABS(V(I))                                                  UCL29250
C...     END CALCULATION ROUTINE-- RETURN TO THE PROPER POINT.          UCL29251
      GO TO (9,10),N4                                                   UCL29260
 201  FORMAT(1X,7E18.8)                                                 UCL29270
 9    K=0                                                               UCL29280
      DO4I=2,5                                                          UCL29290
C...     THE CRITERION FOR FINDING A ROOT--                             UCL29291
C...     THE RESIDUAL AT THE CENTRAL POINT MUST BE LESS THAN THAT OF    UCL29292
C...     THE FOUR SURROUNDING POINTS.                                   UCL29293
      IF(H(I).LT.G)GO TO 46                                             UCL29300
      IF(H(I).EQ.G)K=K+1                                                UCL29310
 4    CONTINUE                                                          UCL29320
C...     HOWEVER, IF ALL 5 POINTS ARE EQUAL, THIS POINT MUST BE REJECTEDUCL29321
      IF(K.EQ.4)GO TO 46                                                UCL29330
C........THE HOME-IN ROUTINE.............................................UCL29331
      J=2                                                               UCL29340
      NEXP=0                                                            UCL29350
      C=.37252902984619140625E-8                                        UCL29360
      NR=KR*16777216                                                    UCL29370
      NI=KI*16777216                                                    UCL29380
 16   IF((NR.NE.0).OR.(NI.NE.0))GO TO 32                                UCL29390
      NEXP=NEXP-1                                                       UCL29400
      J=1                                                               UCL29410
      C=C*6.25E-2                                                       UCL29420
 32   NX=16**(7-J)                                                      UCL29430
 19   SN= CMPLX( FLOAT(NR)*C, FLOAT(NI+NX)*C)                           UCL29440
```

```
      SS= CMPLX( FLØAT(NR)*C, FLØAT(NI+NX)*C)                           UCL29450
      SW= CMPLX( FLØAT(NR-NX)*C, FLØAT(NI)*C)                           UCL29460
      SE= CMPLX( FLØAT(NR+NX)*C, FLØAT(NI)*C)                           UCL29470
C...     BRANCH TØ THE CALCULATIØN RØUTINE.                             UCL29471
      N4=2                                                              UCL29480
      GØ TØ 33                                                          UCL29490
  302 FØRMAT(1X,4E20.8)                                                 UCL29500
   10 E=AMIN1(H(1),H(2),H(3),H(4),H(5))                                 UCL29510
      DØ11I=1,5                                                         UCL29520
      IF(E.EQ.H(I))GØ TØ 12                                             UCL29530
   11 CØNTINUE                                                          UCL29540
      WRITE(6,303)                                                      UCL29550
  303 FØRMAT(' ERRØR')                                                  UCL29560
   12 GØ TØ (13,17,20,22,23),I                                          UCL29570
C...     IF THE DESIRE  ACCURACY HAS BEEN REACHED, EXIT THE HØME-IN     UCL29571
C...     RØUTINE.                                                       UCL29572
   13 IF(J.GE.6)GØ TØ 15                                                UCL29580
      J=J+1                                                             UCL29590
      GØ TØ 16                                                          UCL29600
   17 NI=NI+NX                                                          UCL29610
      GØ TØ 21                                                          UCL29620
   20 NI=NI-NX                                                          UCL29630
   21 IF((J.NE.1).AND.(MØD(NI,NX*16).EQ.0))J=J-1                        UCL29640
      GØ TØ 14                                                          UCL29650
   22 NR=NR-NX                                                          UCL29660
      GØ TØ 28                                                          UCL29670
   23 NR=NR+NX                                                          UCL29680
   28 IF((J.NE.1).AND.(MØD(NR,NX*16).EQ.0))J=J-1                        UCL29690
   14 S=W(I)                                                            UCL29700
      G=H(I)                                                            UCL29710
      IF( CABS(S).GT.2.E0)GØ TØ 46                                      UCL29720
      GØ TØ 32                                                          UCL29730
C........RØØT EXAMINATIØN RØUTINE.......................................UCL29731
   15 IF(NTAG.EQ.0)GØ TØ 43                                             UCL29740
C...     CHECK THE LAST FEW STEPS, BUT THIS TIME WITHØUT DIVIDING ØUT   UCL29741
C...     THE RØØTS.                                                     UCL29742
      NTAG=0                                                            UCL29750
      F=(0.E0,0.E0)                                                     UCL29760
      DØ50I=1,M                                                         UCL29770
   50 F=(F+B(I))*S                                                      UCL29780
      F=F+B(N)                                                          UCL29790
      G= CABS(F)                                                        UCL29800
      DØ49I=1,7                                                         UCL29810
      J=MAX0(7-I,2)                                                     UCL29820
      IF((MØD(NR,16**I).NE.0).ØR.(MØD(NI,16**I).NE.0))GØ TØ 32          UCL29830
   49 CØNTINUE                                                          UCL29840
   43 NTAG=IN(L)                                                        UCL29850
      IF(G.GE.CABS(P)/2.0E0)GØ TØ 46                                    UCL29860
      K=L                                                               UCL29870
      N2=16                                                             UCL29880
   29 IF((NR/N2**7.EQ.0).AND.(NI/N2**7.EQ.0))GØ TØ 34                   UCL29890
      IF(MØD(NR,N2).GE.N2/2)NR=NR+N2                                    UCL29900
      IF(MØD(NR,N2).LE.(-N2/2))NR=NR-N2                                 UCL29910
```

```
      IF(MØD(NI,N2).GE.N2/2)NI=NI+N2
      IF(MØD(NR,N2).LE.(-N2/2))NR=NR-N2
      NR=NR/N2
      NI=NI/N2
      NEXP=NEXP+1
      GØ TØ 29
 34   IF(K.EQ.1)GØ TØ 35
      C=(FLØAT(NR)**2)+(FLØAT(NI)**2)
      E=(FLØAT(NR)/C)*0.72057594037927936E17
      C=(FLØAT(NI)/C)*0.72057594037927936E17
      NEXP=-NEXP
      K=1
 51   IF((ABS(E).LE.0.268435456E9).AND.(ABS(C).LE.0.268435456E9))GØTØ 42
      E=E/16.
      C=C/16.
      NEXP=NEXP+1
      GØ TØ 51
 42   NR=E
      NI=C
      GØ TØ 29
C........THE RØUND-ØFF RØUTINE...........................................
 35   IF(J.GE.7)GØ TØ 24
      K=N2**(7-J)
      IF(MØD(NR,K).GE.K/2)NR=NR+K
      IF(MØD(NR,K).LE.(-K/2))NR=NR-K
      IF(MØD(NI,K).GE.K/2)NI=NI+K
      IF(MØD(NI,K).LE.(-K/2))NI=NI-K
      NR=NR-MØD(NR,K)
      NI=NI-MØD(NI,K)
C...     RETURN TØ THE PRØPER PØINT.
      IF(M.LT.0)GØ TØ 37
 24   IF(NRØØT.EQ.0)GØ TØ 18
      K=3-L
 38   DØ39I=1,NRØØT
      IF(NRT(I,3,K).NE.NEXP)GØ TØ 39
      K1=IABS(NR-NRT(I,1,K))/(16**(7-J))
      K2=IABS(NI-NRT(I,2,K))/(16**(7-J))
      IF((K1.LT.2).AND.(K2.LT.2))GØ TØ 5
   39 CØNTINUE
 18   JN(L)=JN(L)+1
      K=JN(L)
      NRT(K,1,L)=NR
      NRT(K,2,L)=NI
      NRT(K,3,L)=NEXP
C...     PERFØRM THE CØNVERSIØN FRØM HEX TØ DECIMAL.
      N3=NI
      C=FLØAT(NR)*(16.**(NEXP-7))
      E=FLØAT(NI)*(16.**(NEXP-7))
      NEXP=ALØG10(AMAX1(ABS(C),ABS(E)))
      NR=C*(10.**(7-NEXP))
      NI=E*(10.**(7-NEXP))
      M=-M
      N2=10
```

```
      K=1                                                               UCL30420
C...     BRANCH TØ THE RØUND-ØFF RØUTINE AGAIN.                         UCL30421
      GØ TØ 29                                                          UCL30430
 37   M=-M                                                              UCL30440
      Q=DCMPLX(DFLØAT(NR)*1.D-6,DFLØAT(NI)*1.D-6)                       UCL30450
      K=NEXP-1                                                          UCL30460
      WRITE(6,200)Q,K                                                   UCL30470
200   FØRMAT(40X,2F15.9,I11)                                            UCL30480
C        INSERT PØLES IN CØMMØN BLØCK                                   UCL30490
      IF(N9.NE.1)GØ TØ 5                                                UCL30500
      NPØLES=NPØLES+1                                                   UCL30510
      PØLE(NPØLES)=Q*(10.**K)                                           UCL30520
  5   IF(JN(1)+JN(2).LT.N)GØ TØ 44                                      UCL30530
 36   WRITE(6,205)                                                      UCL30540
      GØ TØ 25                                                          UCL30550
 44   IN(L)=IN(L)+1                                                     UCL30560
      J=IN(L)                                                           UCL30570
      D(J,L)=S                                                          UCL30580
C...     CØNTINUE THE SCAN.                                             UCL30581
 46   IF(YMIN.GE.YMAX)GØ TØ 6                                           UCL30590
      KI=KI+1                                                           UCL30600
      YMIN=KI*6.25E-2                                                   UCL30610
      IF(L.EQ.1)GØ TØ 7                                                 UCL30620
      IF(YMIN.LT.YMAX)GØ TØ 7                                           UCL30630
  6   KR=KR+1                                                           UCL30640
      XMIN=KR*6.25E-2                                                   UCL30650
      YMIN=-SQRT(ABS(1.E0-(XMIN**2)))-6.25E-2                           UCL30660
      IF(XMIN.LE.1.E0)GØ TØ 8                                           UCL30670
C...     THIS PASS IS FINISHED.                                         UCL30671
      NPASS=NPASS+1                                                     UCL30680
C...     CHECK IF TØØ MANY PASSES HAVE BEEN PERFØRMED.                  UCL30681
      IF(NPASS.GT.2*(MIN0(N,JN(1)+JN(2))+1))GØ TØ 36                    UCL30690
C........THE INVERSIØN RØUTINE.........................................UCL30691
      L=3-L                                                             UCL30700
      K=N/2                                                             UCL30710
      DØ30I=1,K                                                         UCL30720
      T=B(I)                                                            UCL30730
      J=N-I+1                                                           UCL30740
      B(I)=B(J)                                                         UCL30750
 30   B(J)=T                                                            UCL30760
C...     BACK TØ THE SCANNING RØUTINE                                   UCL30761
      IF(L.EQ.2)GØ TØ 31                                                UCL30770
      GØ TØ 45                                                          UCL30780
 25   CØNTINUE                                                          UCL30790
      DØ99I=1,5                                                         UCL30800
 99   CHAR(I)=BLANK                                                     UCL30810
      RETURN                                                            UCL30820
      END                                                               UCL30830
*STØP* C
```

A. 2 NASAP-70 DICTIONARY

A. 2. 1 Dictionary of Variables

A(I) In Common A. An array of 80 elements containing the 80 characters of the most recently read data card.

A(I) In Common BITS. An alias for A1(I).

A(I) In Subroutine ROOTS. The coefficients of the polynomial being evaluated.

A(I) In Subroutine POLSEN. The real part of the I' th pole.

ABSC In Common BITS. The abscissa label for Subroutine PRTPLT.

ADDR(I) In Common BITS. An alias for NQ(I) used in Subroutine GRAPH.

AMOD(I) In Common BITS. The list of plot control variables STEP to NPLOT.

AMP1 In Common BITS. The positive portion of a pulse train.

AMP2 In Common BITS. The negative portion of a pulse train.

AS(I, J) In Subroutine ANSWER. Alphanumeric array of S' s used in printing Transfer and Sensitivity Functions.

ASIGN (I, J) In Subroutine ANSWER. Alphanumeric array of signs used in printing Transfer and Sensitivity Functions.

A1(I) In Common BITS. Numerator of function to be plotted. The coefficient of S^{I-1}.

B(I) In Subroutine POLSEN. The imaginary part of the I' th pole.

B(I) In Common BITS An alias of B1(I).

B(I) In Subroutine ROOTS. A Complex type array of the polynomial coefficients.

BITS(I) In Common BITS. Array of words used in logical bit manipulations.

BINO(I) In Common BITS. A list of Binomial Coefficients used in Subroutines ROUTH and CALCR.

B MOD(I) In Subroutine PLOT. An integer type alias for AMOD(I).

B RANCH(I) In Common BITS. The list of branches connected to each circuit node. Used in Subroutines CALC and GRAPH.

B 1(I) In Common BITS. Denominator of function to be plotted. The coefficient of S^{I-1}.

C(I) In Subroutine POLSEN. The derivative of the Transfer Function Denominator evaluated at the real part of the I' th pole.

C(I) In Common BITS. An alias for POW(I).

CA(I) In Subroutine INVERT. A Complex type alias for A1(J) with CA(1) the coefficient of the highest power of S.

CB(I) In Subroutine INVERT. A Complex type alias for B 1(J) with CB(1) the coefficient of the highest power of S.

CARD(I, J) In Common CIRCIT. Array containing the J characters of the I' th element name.

CONST In Subroutine INVERT. The quotient resulting when the numerator of the transform is divided by the denominator.

COUNT(I) In Subroutine LOOPS. The number of transmittances leaving flowgraph node I.

D(I)	In Subroutine POLSEN. The derivative of the Transfer Function Denominator evaluated at the imaginary part of the I' th pole.
D(I, J)	In Common BITS. The Routh Table built by Subroutine CALCR.
D(I, J)	In Subroutine ROOTS. List of roots used for forming the reduced Polynomial.
D	In Subroutine SCALER. The sum of the element value exponents scaled by FACTOR.
DEP(I, J)	In Common BITS. Array containing the J characters of the name of the element controlling the I' th element.
DGP	In Subroutine PRTPLT. The ordinate interval.
DHI	In Subroutine PRTPLT. The maximum ordinate value.
DL	In Subroutine SCALER. The sum of the element value exponents scaled by FACTOR-1.
DLO	In Subroutine PRTPLT. The minimum ordinate value.
DR	In Subroutine SCALER. The sum of the element value exponents scaled by FACTOR+1.
ERR	In Common ERR. = 1 when a serious error is detected. = 0 otherwise.
EXCMP(I)	In Common SPEED. Contains the $e^{(\sigma+jw)}$ points calculated by Subroutine SAMPLE.
EXPO	In Subroutine NUMBER. = 1 when the exponent is negative. = 0 otherwise.
FACT	In Subroutine ANSWER. The normalizing factor of the Transfer or Sensitivity Function.
FACT(I)	In Common BITS. Used in computing Taylor Series coefficients.
FACTOR	In Common ERR. Contains the Scaling Factor determined by Subroutine SCALER.
FAD	In Subroutine ANSWER. The normalizing factor of the function denominator.
FCTN(I)	In Common SPEED. The ordinate points for Subroutine PRTPLT.
FAN	In Subroutine ANSWER. The normalizing factor of the function numerator.
F, FE, FN, FS, FW	In Subroutine ROOTS. The function values at the five points of the five point test
FLAG	In Common FLAG. = 1 when Function ISORT fails to convert a numeric character its value. = 0 otherwise.
FNP(I)	In Subroutine ADJUST. Used in computing the Taylor Series corrections to the first five plot points.
FREQ	In Common PATHS. The frequency used by Subroutine FINE to build a circuit tree.
FREQ1	In Common BITS. a in e^{at}, f in Sin $2^{\pi ft}$, starting point for frequency plots.

FREQ2	In Common BITS. Pulse width in a pulse train, ending point for frequency plots.
GAG(I)	In Common GAG. Contains the list of values to be used as Sensitivity Tags.
GENER(I)	In Common CIRCIT. = 0 if the I' th element is passive. = 1 if the I' th element is active.
INDIC(I)	In Subroutine FINE. Used during circuit tree building.
IN(I)	In Subroutine ROOTS. Number of discontinuities found on a type I pass.
INODE	In Common BITS. The lowest node number in the circuit.
INP	In Main Program. = 1 when a Transfer Function Request has been read. = 0 otherwise
ISN	In Common BITS. = 2 for a pulse response. = 3 for a pulse train response. = 1 otherwise.
ITAG	In Subroutine ANSWER. = - 1 for a Worst Case Function. = 0 for a Transfer Function. Else, Sensitivity Function for element number ITAG.
ITREE	In Common TREE. = 1 when Subroutine FINE is to be called to build a tree. = 0 otherwise.
JN(I)	In Subroutine ROOTS. Number of roots found on a type I pass.
KI	In Subroutine ROOTS. Counter for the Imaginary axis.
KR	In Subroutine ROOTS. Counter for the Real axis.
KZERO	In Subroutine PRTPLT. The location of the zero axis on the print line.
LIMIT	In Subroutine SENSIT, WORST. The maximum number of Sensitivity Requests that can be handled.
LINE(I)	In Subroutine FINE. Used for output formatting.
LINE(I)	In Subroutine ANSWER. Used for printing the dividing line.
LINKS(I)	In Common BITS. Area for logical bit manipulations during evaluation of current equations in Subroutine CALC.
LOGIC(I)	In Common BITS. A Logical Type alias for LOOP(I). Used in Subroutine LOOPS.
LOGIC1	In Subroutine LOOPS. A Logical type alias for M1.
LOG1	In Subroutines ROUTH and CALCR. = TRUE if the current SIGMA value lies to the right of all poles in the complex plane. = FALSE otherwise.
LOOP(I)	In Common BITS. Area for logical bit manipulations during flowgraph loop evaluation.
LPATH(I, J)	In Common BITS, PATHS. LPATH(I, 1) is the origin node of the I' th flowgraph transmittance. LPATH(I, 2) is the target node of the I' th flowgraph transmittance.
LS(I)	In Common SPEED. Contains the S value and tags of the I' th loop in the flowgraph.
MOR	In Common BITS. Number of coefficients in the transform denominator.
MORNUM	In Common BITS. Number of coefficients in the Transfrom numerator.

MPATH(I, J)	In the Main Program. An alias for LPATH(I, J).
MRMAX	In Subroutine INVERT. Number of coefficients in the transform.
M1	In Subroutine LOOPS. Used during Higher Order Loop evaluation.
N(I)	In Subroutine NASAP. Used during circuit tree building.
NB	In Subroutine CALC. Number of branches in the circuit tree.
NBR(I)	In Subroutine CALC. Number of branches at circuit node I.
NDEM	In Subroutine WORST. = 0 if the simplified formula can be used. = 1 for the standard formula.
NE	In Common CIRCUIT. Number of elements in the circuit.
NEL(1)	In Subroutine CALC. Number of elements at circuit node I.
NEXP	In Subroutine ROOTS. The magnification factor for homing-in.
NEXPS(I, J)	In Subroutine ANSWER. Array containing powers of S.
NFLAG	In Subroutine WORST. = 1 when Worst Case analysis has been aborted. = 0 otherwise.
NFLAG	In Subroutine INVERT. = 1 when the transform is not rational.
NI	In Subroutine ROOTS. The hexadecimal imaginary part of the root.
NINDIC(I)	In Subroutine FINE. Used during circuit tree building.
NLOOP	In Common SPEED. Number of loops in the flowgraph.
NN	In Subroutine LOOPS. Number of nodes in the circuit.
NN	In Subroutine CALC. Number of nodes in the circuit.
NNODES	In Common BITS. The highest node number in the circuit.
NODE(I)	In Subroutine LOOPS. The number of flowgraph transmittances arriving at flowgraph node I.
NPATH	In Common PATHS. Number of transmittances in the flowgraph.
NPLOT	In Common BITS. Number of calculated points per printed point.
NPOLES	In Common SPEED. Number of poles in the Transfer Function.
NQ(I)	In Common BITS. The beginning of the list of branches for node I in the BRANCH array. Used in Subroutine CALC.
NR	In Subroutine ROOTS. The hexadecimal real part of the root.
NR	In Subroutine READ. Number of duplicate elements names.
NRT(I, J, K)	In Subroutine ROOTS. The hexadecimal list of roots.
NS(I)	In Subroutine CALC. Initially the same as NQ(I). Used during the calculation of the current equations.
NSEN	In Common GAG. Number of elements tagged for Sensitivity Analysis.
NTAY	In Common BITS. = 1 if a Taylor Series is to be used to compute the first five plot points. = 0 otherwise.
NTIMES	In Common NTIMES. Number of words per block to be used in logical bit manipulation in Common BITS.
NUM(I, J)	In Subroutine ANSWER. Alphanumeric array of S exponents.
NUMB(I)	In Subroutine FINE. Used during tree building.
NUMNOD	In Subroutine FINE. Number of circuit nodes.
NUMFR	In Common BITS. Number of points along the w axis.
NUMTI	In Common BITS. Number of calculated points in the plot.
NWORST	In Common WORST1. = 1 if a Worst Case analysis is to be performed. = 0 otherwise.

NX	In Subroutine ROOTS. The step size of the Home-in routine.
N1	In Subroutine LOOPS. The first node in a Higher Order Path. Detected by Subroutine UNPAK.
N2	In Subroutine LOOPS The second node in a Higher Order Path. Detected by Subroutine UNPAK.
N3	In Subroutine LOOPS. The third node in a Higher Order Path. Detected by Subroutine UNPAK.
OCARD(I, J)	In Subroutine FINE. The ordered CARD array.
OGENER(I)	In Subroutine FINE. The ordered GENER array.
OORIGIN(I)	In Subroutine FINE. The ordered ORIGIN array.
ORD	In Common BITS. The ordinate label for Subroutine PRTPLT.
ORDER(I)	In Subroutine LOOPS. A pointer to the location of the I' th order loop or transmittance in the LOOP area.
ORIGIN(I)	In Common CIRCIT. Origin circuit node of the I' th element.
OTARGT(I)	In Subroutine FINE. The ordered TARGET array.
OVPATH(I)	In Subroutine FINE. The ordered VPATH array.
OZ(I)	In Subroutine FINE. The ordered Z array
P(I)	In Subroutine PRTPLT. Used to format the print line during plotting.
P	In Subroutine POLSEN. The Real part of the Pole Sensitivity.
PH(I)	In Subroutine INVERT. Array of phase points in a frequency plot.
POLE(I)	In Common SPEED. The I' th pole of the Transfer Function, a Complex number
POW	In Subroutine NUMBER Used to hold the exponent value.
POW(I)	In Common BITS. Part of the Taylor Series computed by Subroutine TAYLOR.
Q	In Subroutine POLSEN. The Imaginary part of the Pole Sensitivity.
RSPNS(I)	In Common BITS. The points $F(\sigma+jw)$ computed by Subroutine SAMPLE. The work array for the Fast Fourier Transform in Subroutine FLIP.
S, SE, SN, SS, SW	In Subroutine ROOTS. The five points of the five point test.
S(I)	In Common BITS, PATHS. Contains the S value and tags of the I' th flowgraph transmittance.
SCFREQ	In Common BITS. The frequency scale factor.
SCMAG	In Common BITS. The amplitude scale factor.
SENS(I)	In Common GAG. Numerical Sensitivity Tag for the I' th element.
SIGMA	In Common BITS. The real part of S, the complex frequency.
SIGPOW(I)	In Subroutine CALCR An array of powers of SIGMA σ^I.

SMAX(I)	In Common CIRCIT. SMAX(1). Highest s power in Transfer function Denominator, VN(I, 1). SMAX(2) Highest s power in Transfer Function Numerator, VN(I, 2).
SMAX00(J)	In Common POLY. Highest s power in -H($\bar{P}$,$\bar{Q}$), VN00(I, J), for the J' th element.
SMAX01(J)	In Common POLY. Highest s power in H(P' ,$\bar{Q}$) VN01(I, J), for the J' th element.
SMAX1(K)	In Common BITS, POLY. Highest s power in Sensitivity Function for K' th element, VNSEN(I, J, K).
SMAX10(J)	In Common POLY. Highest s power in -H($\bar{P}$,Q'), VN10(I, J), for the J' th element.
SMAX2	In Common WORST1. Highest s power in Worst Case Function, VW(I, J).
SMIN(I)	In Common CIRCIT. SMIN(1) Lowest s power in Transfer Function Denominator, VN(I, 1). SMIN(2), Lowest s power in Transfer Function Numerator, VN(I, 2).
SMIN00(J)	In Common POLY. Lowest s power in -H($\bar{P}$,$\bar{Q}$), VN00(I, J), for the J' th element.
SMIN01(J)	In Common POLY. Lowest s power in H(P' ,$\bar{Q}$) VN01(I, J), for the J' th element.
SMIN1(K)	In Common BITS, POLY. Lowest s power in Sensitivity Function for K' th element, VNSEN(I, J, K).
SMIN10(J)	In Common POLY. Lowest s power in -H($\bar{P}$,Q'), VN10(I, J), for the J' th element.
SMIN2	In Common WORST1. Lowest s power in Worst Case Function, VW(I, J).
STEP	In Common BITS. The interval between successive plot points.
SPOW(I)	In Subroutine SAMPLE. The value of S^{I-1} .
TAG	In Common TAG. A utility flag used by the Main Program.
TAG	In Subroutine NUMBER. = 1 when the number is negative. = 0 otherwise.
TARGET(I)	In Common CIRCIT. Target circuit node of the I' th element.
TAG(I)	In Common GAG. An alias for GAG(I).
TAG(I)	In Subroutine FINE. An alias for the TYPE array.
TEE(I)	In Common SPEED. The abscissa points for Subroutine PRTPLT.
TERM(I)	In Common BITS. The coefficients of the Taylor Series computed by Subroutine TAYLOR.
TOL(I)	In Common WORST1. Tolerance value for the I' th element.
TOTEE	In Common BITS. The interval between the first and last points plotted.
TREE	In Common TREE. = 1 when Subroutine FINE is to be called to build a circuit tree. = 0 otherwise.
TYP(I)	In Subroutine GRAPH. An alias for TYPE(I).
TYPE(I)	In Common CIRCIT. = 0 if the I' th element is a Current Source (Link). = 1 if the I' th element is a Voltage Source (Branch

T1, T2, T3	In Common BITS. Coefficients for the pulse response Transforms.
V(I)	In Common SPEED. The value of the I' th loop in the flowgraph.
VDER1V(I)	In Subroutine POLSEN. The symbolic derivative of the Transfer Function Denominator. I=51+ power of s.
VM(I, J)	In Subroutine ANSWER The Function after normalization.
VN(I, J)	In Common CIRCIT. VN(I, 1) Transfer Function Denominator. VN(I, 2) Negative Transfer Function Numerator. I=51+ power of s.
VNSEN(I, J, K)	In Common BITS, POLY. Sensitivity Function of K' th element. J=1 for Denominator, J=2 for Numerator. I=51 + power of s.
VPATH(I)	In Common BITS, PATHS. Value of the I' th flowgraph transmittance.
VN00(I, J)	In Common POLY. $-H(\bar{P}, \bar{Q})$ for the J' th element. I=51+ power of s.
VN01(I, J)	In Common POLY. $H(P', \bar{Q})$ for the J' th element. I=51+ power of S.
VN10(I, J)	In Common POLY. $-H(\bar{P}, Q')$ for the J' th element. I=51+ power of S.
VPATH(I)	In Common PATHS. Value of the I' th flowgraph transmittance.
VW(I, J)	In Common WORST1. Square of the Worst Case Function. I=51+ power of s. J=1 for Denominator, J=2 for Numerator.
W	In Subroutine FINE. An alias for FREQ.
X	In Subroutine POLSEN. The Numerator of the Real part of the Pole Sensitivity Function, $-QH(\bar{P}, Q)$ or VN10(I, J) evaluated at a pole.
Y	In Subroutine POLSEN. The Numerator of the Imaginary part of the Pole Sensitivity Function, $-QH(\bar{P}, Q)$ or VN10(I, J) evaluated at a pole.
Z(I)	In Subroutine FINE. The impedance of the I' th element.

A.2.2	Dictionary of Subprograms
ADJUST	In the Plotter. Re-evaluates the first 5 sample points via a Taylor Series.
ANSWER	In the Transfer Function package. Prints Transfer and Sensitivity Functions normalized and formatted.
ASCAN	A Card Scanning Utility. Scans for certain characters.
BLOCK DATA	Part of the Main Program. Initializes Common Blocks DATA, X and GAG.
BSCAN	A Card Scanning Utility. Scans for certain characters.
CALC	In the Transfer Function Package. Evaluates current equations from the circuit tree.
CALCR	In the Plotter. Uses the Routh Stability Criterion to determine if the current SIGMA value is acceptable.
CLEAR	In the Transfer Function package. Removes transmittances in the Flowgraph which have been marked deleted.
CSCAN	A Card Scanning Utility. Scans for certain characters.
EQUAL	A Bit Manipulation Utility. Performs various operations on blocks of word in Common Block BITS.
FINE	A Circuit Description Analysis routine. Builds a circuit tree for a user-specified frequency.
FLIP	In the Plotter. Computes the Fast Fourier Transform coefficients.
GRAPH	In the Transfer Function package. Builds the Flowgraph except for the unknown transmittance.
INBIT	A Bit Manipulation Utility. Sets bits in blocks of words in Common Block BITS.
INPUT	In the Plotter. Evaluates the transform of the response.
INVERT	In the Plotter. Controls the execution of the Plot Request.
ISORT	A Card Scanning Utility. Converts numeric characters to fixed binary values.
LOOPS	In the Transfer Function package. Computes Transfer and Sensitivity Functions from the Flowgraph, via the Shannon-Happ formula.
LOR	A Bit Manipulation Utility. Performs 'OR'ing.
LOX	A Bit Manipulation Utility. Performs 'AND'ing and subtracts the result from the arguments.

LSTOR	A Bit Manipulation Utility. Copies one word into another.
MAIN	The Main Program. Controls execution of NASAP-70.
MSG	A Card Scanning Utility. Prints a diagnostic.
MULT	In the Sensitivities and Worst Case package. Multiplies polynomials.
NASAP	A Circuit Description Analysis Routine. Interprets the standard form of circuit description data and builds a circuit tree for minimum computation time.
NUMBER	A Card Scanning Utility. Converts alphanumeric representation of numbers into floating point values.
PLOT	In the Plotter. Interprets Plot Request Cards.
POLSEN	In the Sensitivities and Worst Case package. Computes Pole sensitivities.
PRTPLT	In the Plotter. Prints and plots the calculated response points.
READ	A Circuit Description Analysis routine. Interprets the special form of circuit description data.
REDUCE	In the Sensitivities and Worst Case package. Decodes tags from the S or LS arrays.
ROOTS	The Rootfinder. Finds poles and zeroes of the Transfer Function.
ROUTH	In the Plotter. Finds an acceptable SIGMA value.
SAMPLE	In the Plotter. Computes the initial Fast Fourier Transform points.
SCALE	In the Plotter. Scales the Fast Fourier Transform coefficients.
SCALER	The Automatic Scaler. Re-adjusts, if necessary, circuit data values to avoid floating overflow.
SENSC	In the Sensitivities and Worst Case package. Computes Sensitivity Functions.
SENSIT	In the Sensitivities and Worst Case package. Interprets Sensitivity Request Cards.
SHIFT	A Card Scanning Utility. Eliminates blanks and commas.
SORT	A Card Scanning Utility. Converts numeric characters to floating binary values.
TAYLOR	In the Plotter. Computes Taylor Series coefficients.
UNITS	A Circuit Description Analysis routine. Prints a message.

UNPAK	A Bit Manipulation Utility. Removes bits from blocks in Common block BITS.
WHAT	In the Transfer Function package. Interprets Transfer Function Request cards.
WORST	In the Sensitivities and Worst Case package. Interprets Tolerance Cards.
WORSTC	In the Sensitivities and Worst Case Package. Computes the Worst Case function.

A.2.3 Dictionary of Common Blocks

A	The Card Buffer. Contains the last card read.
BITS	The area for bit manipulations. Also used to hold the Fast Fourier Transform coefficients.
CIRCIT	Contains data compiled from the circuit description.
DATA	Contains alphameric characters.
ERR	Contains the Error flag and scaling factor.
FLAG	Contains a utility flag.
GAG	Contains Sensitivity Tags.
NTIMES	Contains the number of words per block in BITS.
PATHS	Contains the Flowgraph paths.
POLY	Contains polynomials used to compute Sensitivity and Worst Case Functions.
SPEED	Contains Flowgraph loops. Also used by the plotter.
TAG	Contains a utility flag.
TREE	Contains the switch for building a tree for user-supplied frequency.
WORST1	Contains data for Worst Case Analysis.
X	Contains numeric characters.

In the interest of saving space, some Common Blocks (especially BITS) are used for various purposes at different times.

APPENDIX B

MODELS FOR COMPUTER AIDED CIRCUIT DESIGN/ANALYSIS PROGRAMS: PART I. COMMON TRANSISTOR MODELS, A SURVEY

By Donnamaie E. Meyerhoff†

INTRODUCTION

The analysis of circuits by computer depends upon the ability of the analyst to appropriately simulate device performance by a model consisting of the recognized parameters of the CAD program being utilized. The analyst must work with a lumped model of the device in question if he is to remain within the limits of a CAD program, and he must also select appropriate approximations to be made in the representation of the device behavior. These approximations must be such that they simplify analysis while maintaining an acceptable level of validity in the results [1, 2]

A model is a mathematical entity with precise definitions of its variables and their relationships. It is never completely equivalent to the physical device, rather, the behavior of the model approximates that of the physical device.[2] The development of a semiconductor model requires: (1) formulation of the ideal voltage-current equations and definition of the topology of the model, (2) modification for actual device performance, (3) invention of techniques for measurement from which model parameters may be extracted, (4) the extraction of those parameters [3]

An ideal model will (1) incorporate parameters either measurable or derivable, as from a manufacturer's data sheet,* (2) involve parameters acceptable to the given CAD program (facilitating analysis), (3) be sufficiently simplified to be comprehensive, to promote invention and design, and to be economical, (4) be complex enough to assure a tolerable degree of accuracy in the simulation of device performance over the operating

†Northrop Corporate Laboratories, this work was prepared while the author was with TRW Systems Group, California

*Used with discretion

range desired.[1, 2, 4, 5] The less complicated the model, the more restricted its range of application

The primary concern of the circuit analyst is with the characterization of the external behavior of the device (functional modeling) without detailed simulation of internal processes (physical modeling) although an understanding of the internal processes must be assumed.[2]

The selection of an appropriate model and its parameter values to correspond to given device operating conditions (temperature, biasing, load, etc.) for the circuit under consideration is the critical step in analysis, with or without computer assistance as the model finally selected will be the limiting factor in the degree of accuracy of the results [5, 6]

The following discussion is concerned with transistor models and their representation in present general purpose CAD programs. Diodes are more easily represented and are omitted here for brevity

Transistors were selected to demonstrate CAD program modeling because the majority of circuit analysis is still primarily concerned with them, and because their models are presently the primary building block of Integrated Circuit modeling. It should be emphasized that there is no single transistor model capable of simulating all regions of operation for all applications, a model this capable would be prohibitively complex

This description of modeling is divided into two sections· (1) small signal (linear, incremental) models, and (2) large signal (nonlinear) models. The models are not derived here, the reader is referred to the appropriate references for the derivations.

SMALL SIGNAL MODELS

A small signal is defined as an AC signal significantly smaller in peak to peak amplitude than the DC bias upon which it is superimposed. Small signal transistor models are intended to simulate the behavior of a transistor at a specific operating point in the linear region of operation

The transistor model is constructed of parameters that remain relatively constant over the voltage and current swings encountered. These parameters are then assumed constant for the analysis.[7]

The small signal models are not versatile, large signal swings may not be accurately analyzed (due to the lack of parameter variation and the linearity of the model), saturation and cut off are not represented. The model is valid only in the active region.[8] Although it is limited in its application, the small signal model provides a straightforward method for simulating transistor performance when the operating restrictions have been satisfied.

A family of small signal models composed of h, y, z, and r parameters (see for examples References 7 to 10) exists for the user and the most commonly applied models are given here. The input formats for the models for the CAD programs which accept them are also presented. The common emitter configuration is used in most cases to facilitate comparison.

<u>Low Frequency Models.</u> The two commonly used configurations are the Hybrid and the Tee using h and r parameters respectively.

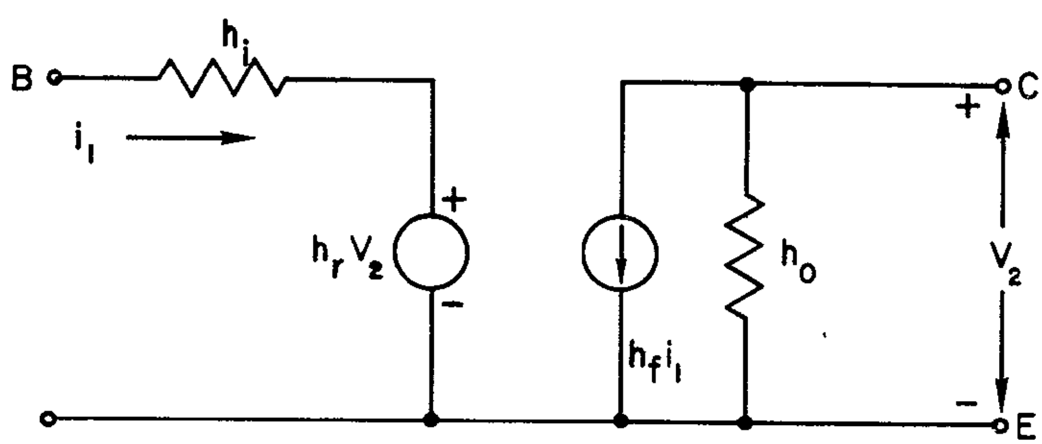


Figure B 1 Hybrid Model

The Hybrid Model is considered the best model for small signal representation because (1) the h parameters are real numbers at low frequencies, (2) the parameters may be easily obtained by measurement when required or from the device static characteristic curves, (3) if the h parameter of one configuration is available, the conversion to another configuration is relatively simple The Hybrid Model is valid for any configuration at low frequencies, for a device operating in the active region [8]

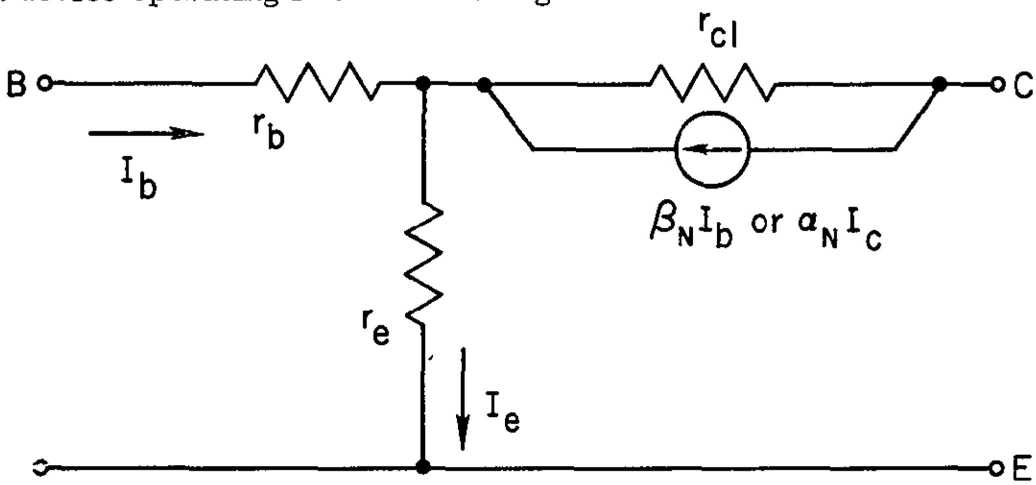


Figure B 2 Tee Model Equivalent Alternate of the Hybrid Model

Program Models. SCEPTRE has a prestored hybrid model for small signal analysis. The program accepts the voltage generator (EA) described as a function of the voltage across ho (R2 in the diagram below) and the current generator (JB) described as a function of the current through hi (R1 below).[11]

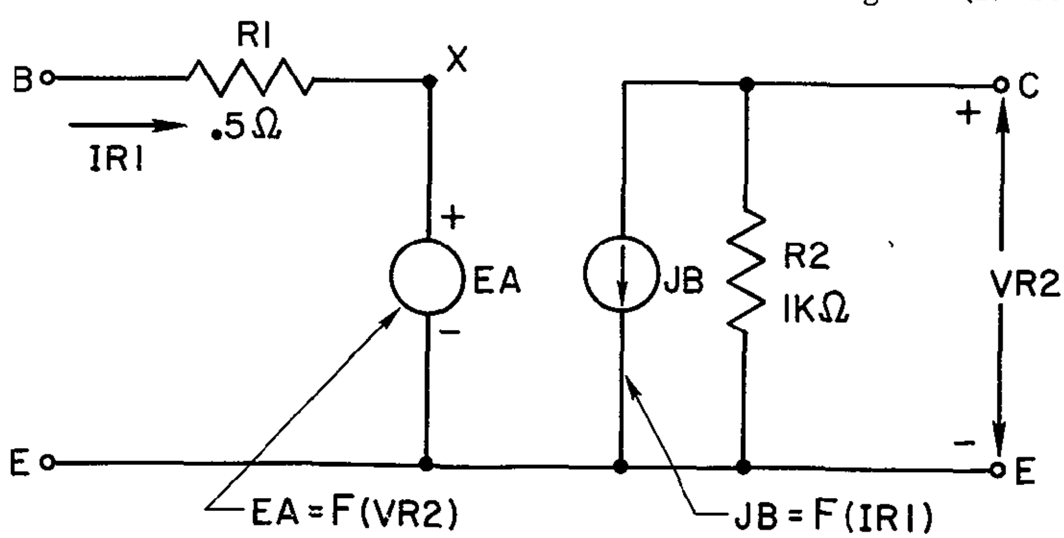


Figure B 3 SCEPTRE Representation — Hybrid Model (Prestored)

If the Tee model is preferred, SCEPTRE will accept a user-description for that model without requiring modifications in the representation. ECAP, however, will not accept a nonlinear dependency and requires the conversion of the hybrid model voltage generator into a DC bias and some resistance such that the branch* resulting is a linear approximation of the performance of the original generator. The current source is dependent linearly on the current through R1 and it is not modified.[12]

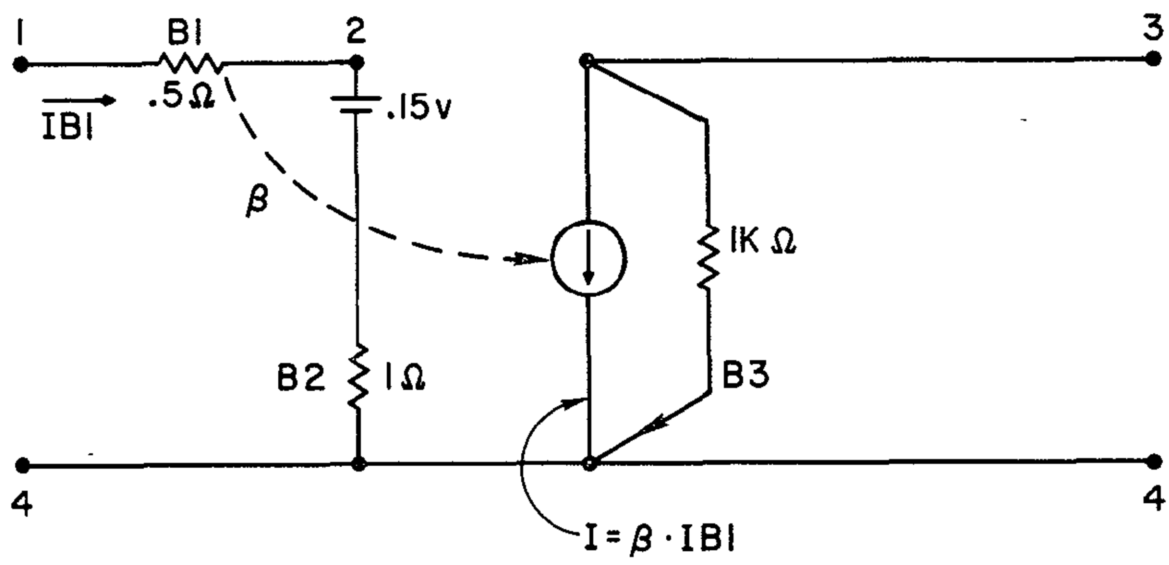


Figure B 4 ECAP Representation – Hybrid Model

*branch – one impedance with or without a source between two nodes, ECAP requires a "dummy" impedance in parallel with an I source, and in series with a V source, 1K is used.

High Frequency Models $\left[F \ll \frac{2D_b}{w^2}\right]$ The two most commonly used configurations are the Hybrid-Pi (Figure B-5) and the Modified-Tee (Figure B-6). Their parameters are not frequency dependent and most manufacturers provide data for one of them.

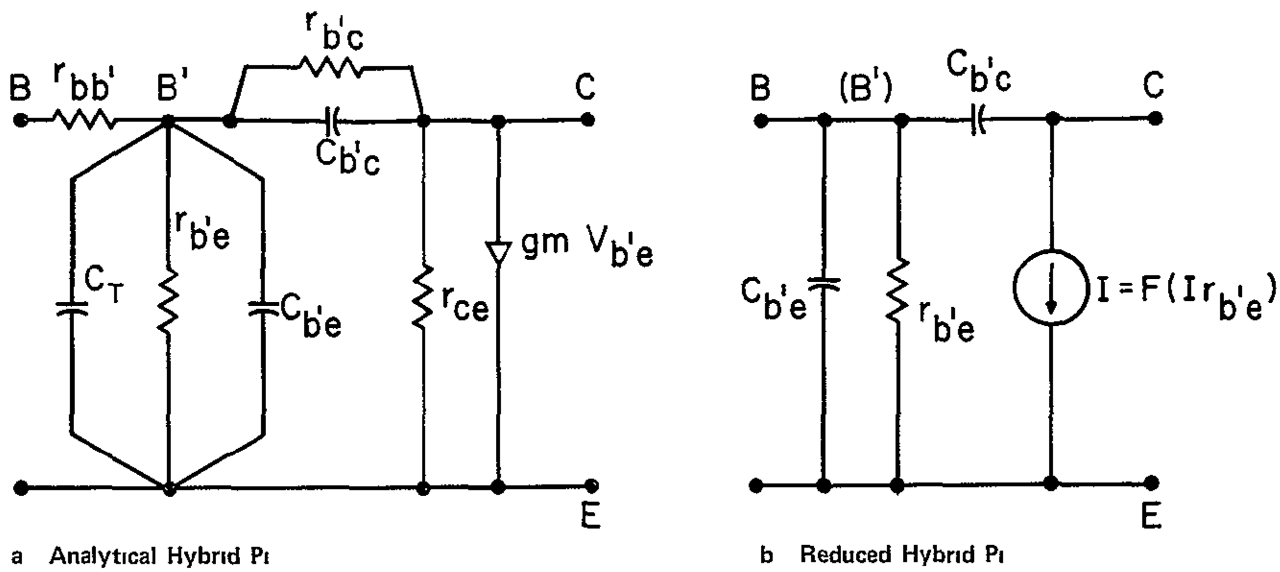
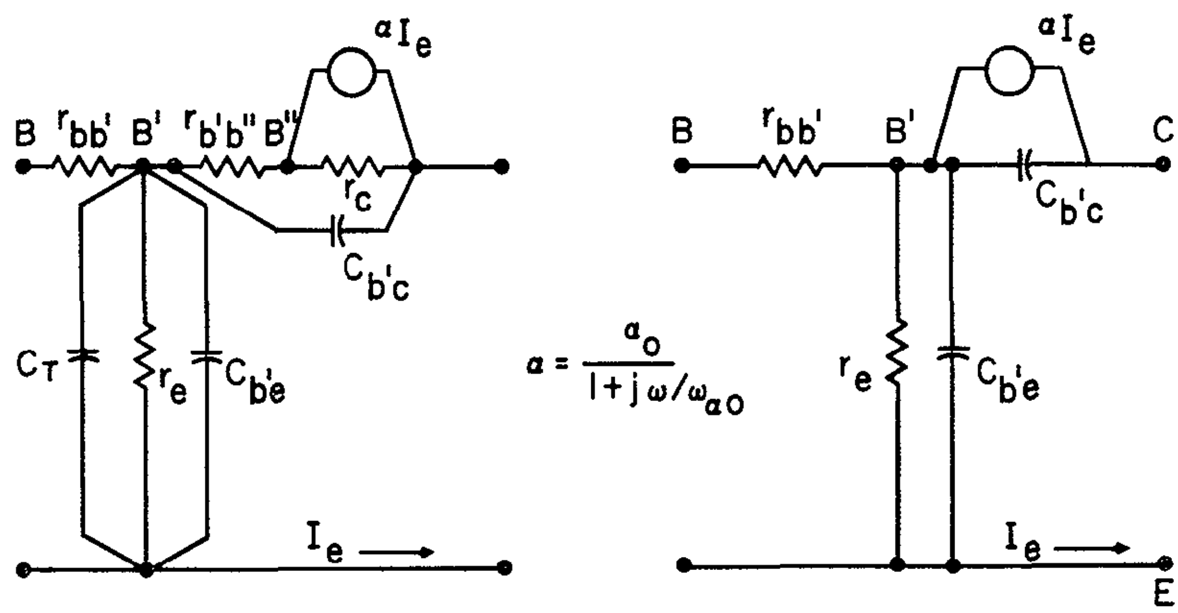


Figure B 5 Hybrid PI

This circuit is known to give results in excellent agreement with experiment in the range dc to limiting frequency of the device, that frequency where the gain is significantly reduced and operation is no longer validly assumed linear

The reduced version of the circuit is justified when source or load impedances or other factors or uncertainties swamp the effects of various parameters Parameters $r_{b'c}$ and r_{ce} are considered approximate opens in Figure B-5b, $C_{b'e}$, $C_{b'c}$ are lumped and include C_T [8]

The Modified Model is generally used for common base circuits, and is limited in transient response analysis in range of operating biases and bandwidth [9, 13] The Simplified Tee uses one base resistance, and r_c is an approximate open



a Analytical Modified Tee

b Simplified Tee

Figure B 6 Modified Tee

Program Models ECAP will accept the Hybrid-Pi in either form, although the reduced version is the most commonly used. The current dependency is linear and the elements must be single valued.

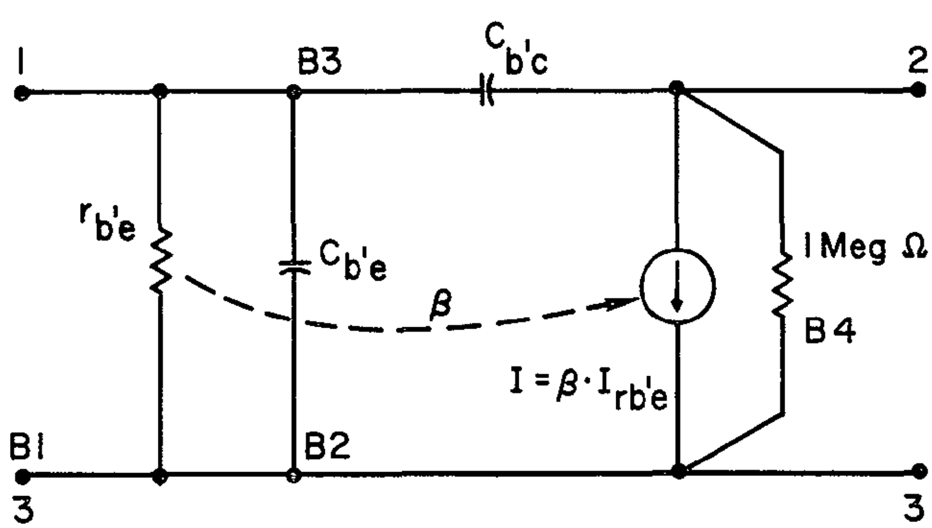


Figure B 7 ECAP—Hybrid PI

SCEPTRE has no prestored Hybrid-Pi but the user may enter a model which describes the current generator as a function of $I_{rb'e'}$, and which is not restricted to single valued elements

There are numerous other small signal models, the ones presented above are the most commonly applied to small signal analysis. There are more complex circuits for device simulation at higher operating frequencies and for noise analysis

LARGE SIGNAL MODELS

A large signal is defined as an AC signal with a peak to peak amplitude that exceeds the DC bias upon which it is superimposed Large signal transistor models are intended to simulate the behavior of a transistor when its operation encompasses more than one region of operation, or when its operation enters the nonlinear portion of the device characteristics

There are four regions of operation for a transistor (See References 13 - 17.) They are.

I. <u>Cut-off Region</u> Collector current cut-off or collector voltage saturation, both junctions are reverse biased, only small reverse saturation currents flow across the junctions

$$I_e < 0 \qquad V_{EB} \ll -\frac{kT}{q} \qquad V_{CB} \ll -\frac{kT}{q}$$

The transistor behaves as an open (nonconducting) switch

II. <u>Normal Active Region</u> Emitter junction forward biased, collector junction reverse biased, $0 < I_e < -\frac{I_C}{\alpha}$

$$V_{CB} \ll -\frac{kT}{q}$$

III. <u>Saturation Region.</u> Collector current saturation or collector voltage cut off, both junctions carry a forward bias, $I_e < -\frac{I_C}{\alpha}$

the voltages across the junctions are in the millivolt range. The transistor behaves as a closed (conducting) switch.

IV. Inverse Active Region. Emitter reverse biased, collector forward biased, similar to normal active with emitter and collector interchanged, low reverse current gain.

Large signal operation for transistors includes power applications and switching (an extreme case where the device goes rapidly from Region III to Region I and vice versa, and generally includes the first three regions of operation.[14]

Small signal models dealt only with Region II The parameters were assumed to remain constant and operation was approximately linear Large signal modeling is complicated by the variation of parameter values with changes in voltage and current, the thermal and power limitations that must be accounted for, and the nonlinear behavior of the device.[7,9]

There are three primary models used for large signal analysis (1) the Beaufoy-Sparkes Charge Control model; (2) the Ebers-Moll model, (3) the Linvill-Lumped model

In their paper "Comparison of Large Signal Models for Junction Transistors,"[4] Hamilton et al, had the following conclusions about the three models.

First, the three models are all described by two first order ordinary differential equations. The main difference in the equations is in the dependent variables. The same four measurements provide the values for parameters for Ebers-Moll and the Beaufoy-Sparkes models and determine the characteristics equation of the Lumped model. (Table B-I shows parameter conversion for #1 and #2, and Figure B-14 demonstrates equation similarity for #2 and #3)

Second, although each model is derived by means of approximations made at different stages of the theoretical development, the final overall degree of approximation is the same. The models have the same natural frequencies and give the same results for transient problems. Therefore, as far as numerical results are concerned, the three models are equivalent.

Last, the Charge Control model is considered more cumbersome in application that the Ebers-Moll model, the Lumped model is considered the most general in development approach.

The Ebers-Moll Model. The well-known Ebers-Moll equations for device operation are given below. They are valid for any transistor in low level injection, regardless of shape, which has negligible voltage drop everywhere but the junctions. They are also valid for a graded base region device.[18]

The Ebers-Moll model for Regions I, and II is based on these equations of ideal device operation. To more closely approximate actual device performance resistances r_b, r_{el}, r_{cl} must be added. The model is completed for high frequency by adding C_{be}, C_{bc}

Program Models: Two CAD programs have prestored versions of the Ebers-Moll model for the first three regions of operation, NET-1, and SCEPTRE The Ebers-Moll based model is the only model available with NET-1 for circuit analysis, and it requires all of its parameters be prestored for each device prior to its use in analysis, a requirement that has been found to be too restrictive.

SCEPTRE uses a more flexible prestored Ebers-Moll model. The user must have access to I_{CO}, I_{EO}, θ_N, θ_I, to describe J_1 and J_2 (current generators) or may enter tables of laboratory measurement data. R_{bb} and R_{cc} may be omitted from separate representation when measurements are used since they are included in the values for J_1 and J_2. If $\alpha_I \sim 0$, JA may be omitted.

a) IDEALIZED EQUATIONS

$$I_E = \frac{I_{EO}}{1 - \alpha_N \alpha_I}\left(e^{\lambda V_{EB}} - 1\right) + \frac{\alpha_I I_{CO}}{1 - \alpha_N \alpha_I}\left(e^{\lambda V_{CB}} - 1\right)$$

$$I_C = \frac{\alpha_N I_{EO}}{1 - \alpha_n \alpha_I}\left(e^{\lambda V_{EB}} - 1\right) - \frac{I_{CO}}{1 - \alpha_N \alpha_I}\left(e^{\lambda V_{CB}} - 1\right)$$

$$I_B = - I_C - I_E$$

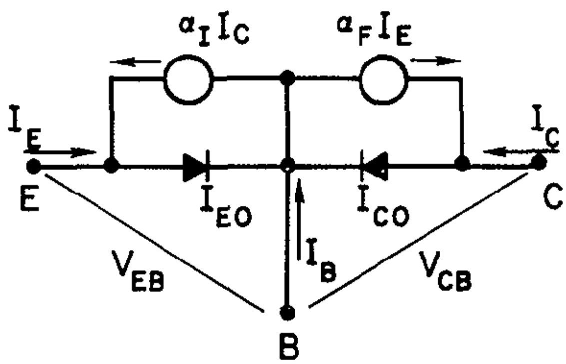
$\alpha_N I_{EO} = \alpha_I I_{CO}$ reciprocity requirement

$\lambda = \dfrac{q}{\eta KT}$ $\quad \eta = 1$ valid when the transition region has no effect

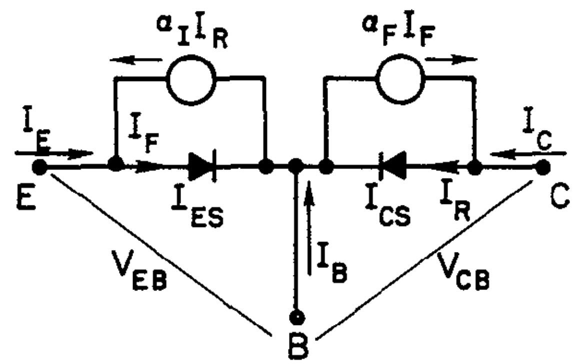
$\eta \sim 1$ for germanium, $\eta \sim 2$ for silicon, $1 \leqq \eta \leqq 3$

b) IDEALIZED MODELS

TERMINAL CURRENT CONTROL



TERMINAL VOLTAGE CONTROL



$$I_{CS} = \frac{I_{CO}}{1 - \alpha_N \alpha_I}$$

$$I_{ES} = \frac{I_{EO}}{1 - \alpha_N \alpha_I}$$

$$I_F = I_{ES}\left(e^{qV_{EB}/KT} - 1\right)$$

$$I_R = I_{CS}\left(e^{qV_{CB}/KT} - 1\right)$$

Figure B 8 Ebers—Moll Equations and Idealized Models (PNP)

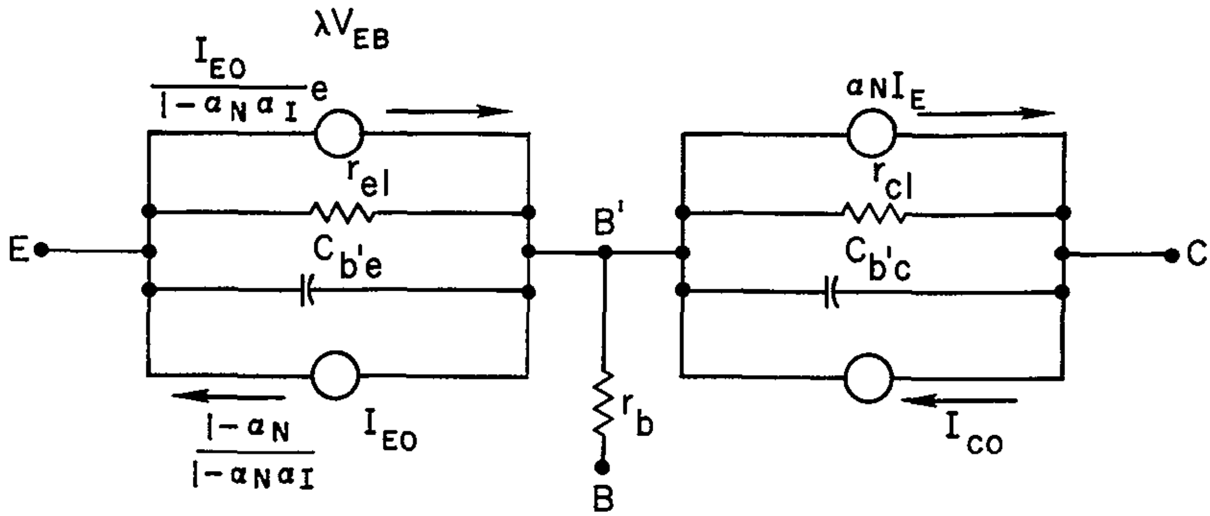
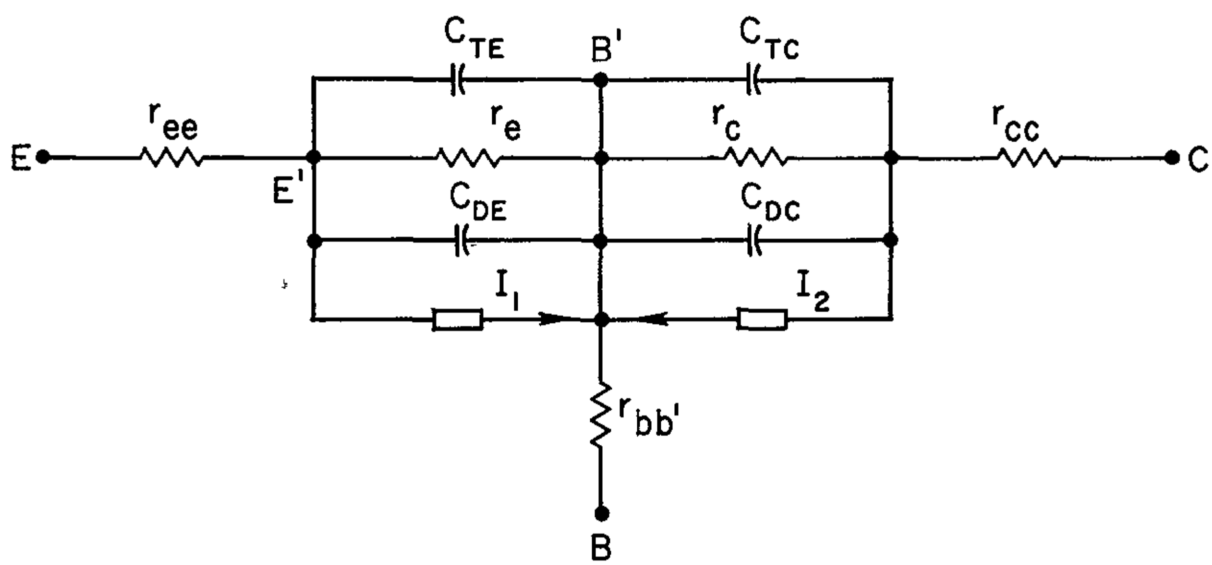


Figure B 9   Ebers—Moll Model for Regions I, II, High Frequency



$C_{TE} = F(V_{EB})$

$C_{TC} = F(V_{CB})$

$C_{DE} = F(I_{es})$

$C_{DC} = F(I_{cs})$

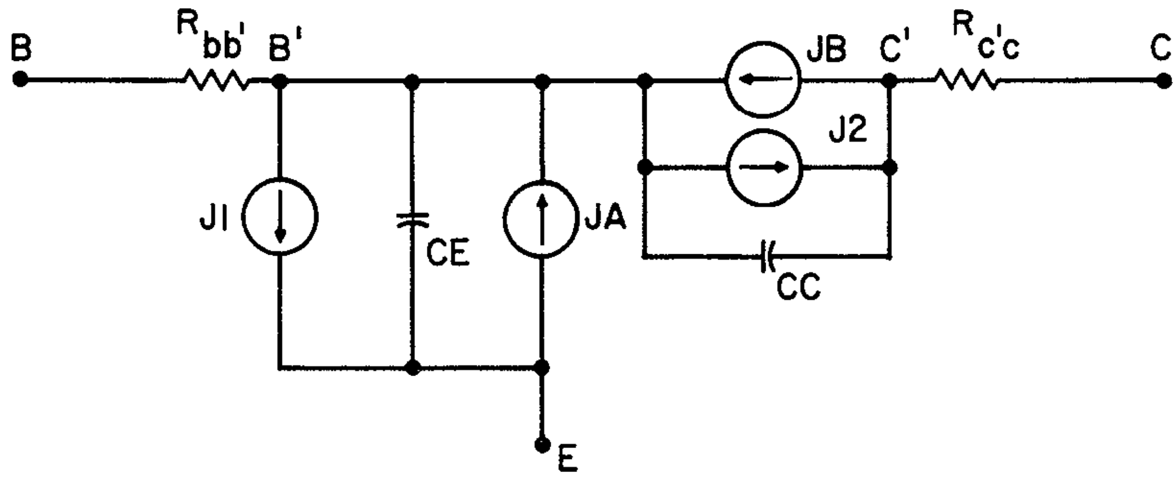
$I_1 = I_{ef} - \alpha_I I_{cf}$

$I_2 = I_{cf} - \alpha_N I_{ef}$

$$I_{ef} = \frac{I_{cs}}{1 - \alpha_N \alpha_I} \left(e^{\frac{\lambda V_{EB}}{m_e}} - 1 \right)$$

$$I_{cf} = \frac{I_{cs}}{1 - \alpha_N \alpha_I} \left(e^{\frac{\lambda V_{CB}}{m_c}} - 1 \right)$$

Figure B 10   NET — 1 Transistor Model



$$JA = \alpha_1 J1$$

$$JB = \alpha_N J2$$

$$J1 = I_{EO}\left(e^{\theta_N V_{B'E}} - 1\right)$$

$$J2 = I_{CO}\left(e^{\theta_I V_{B'C}} - 1\right)$$

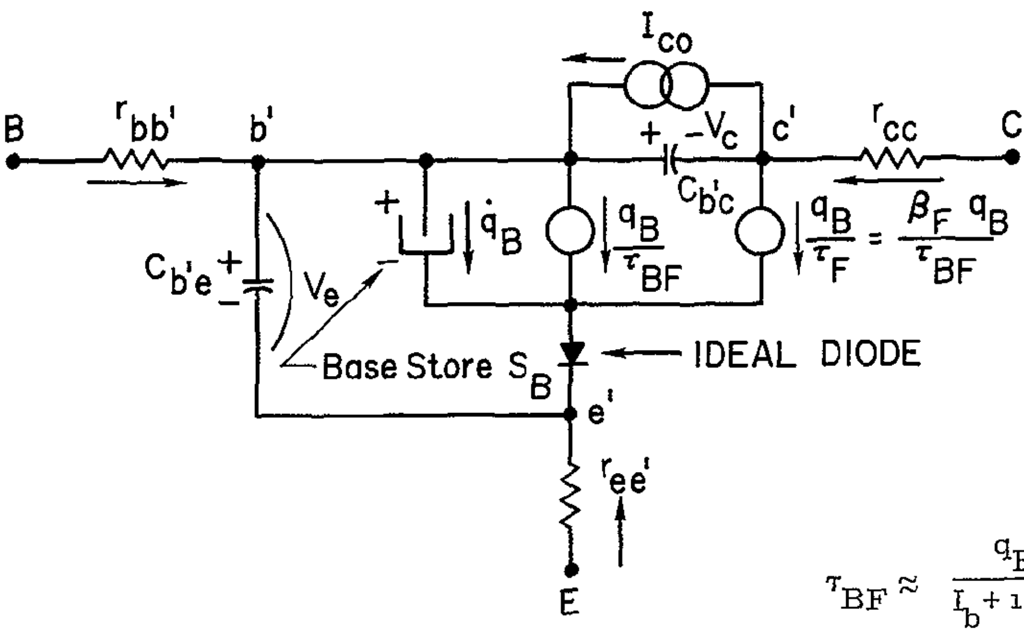
where $\theta = \dfrac{q}{\eta KT}$

Figure B 11 SCEPTRE Prestored Model

Charge Control Model. The concepts of charge control may be briefly summarized as (1) the base charge (qB) may be defined uniquely by the past behavior of the base current, (2) the collector current is proportional to the base charge, is controlled by the base charge.[18] The active region model and its equations are given in Figure B-12 [17, 20, 21]

The similarity of the charge control model and more familiar parameters is seen by the relations

$$\tau_{BF} \approx \frac{h_{fe}}{\omega_T} = \frac{1}{\omega_\beta} \qquad h_{fe} \approx \frac{\tau_{BF}}{\tau_F} \quad \text{(Forward Injection)}$$



$$\tau_{BF} \approx \frac{q_B}{I_b + I_{CO}}$$

$$\tau_F \approx \frac{q_B}{i_c - i_{CO}}$$

$$\beta \approx \frac{\tau_{BF}}{\tau_F}$$

$$i_b = \frac{q_B}{\tau_{BF}} + \dot{q}_B - C_{b'e}\frac{dV_e}{dt} - C_{b'c}\frac{dV_c}{dt} - I_{CO}$$

$$i_c = \frac{q_B}{\tau_F} + C_{b'c}\frac{dV_c}{dt} + I_{CO}$$

$$i_e = q_B \left(\frac{1}{\tau_F} + \frac{1}{\tau_{BF}}\right) + q_B + C_{b'e}\frac{dV_e}{dt}$$

Figure B 12 Charge Control Model and Equations (NPN Device)

The similarity of the charge control model and more familiar parameters is seen by the relations

$$\tau_{BF} \approx \frac{h_{Fe}}{\omega_T} = \frac{1}{\omega_\beta} \qquad h_{fe} \approx \frac{\tau_{BF}}{\tau_F}$$

Program Models: CIRCUS includes a radiation-pulse current generator in its prestored charge control model.[22] CIRCUS requires that the user define 13 single valued parameters and 4 parameters that are single valued or tabled. The program equations are

$$C_{be} = \frac{\alpha_I}{\left(\theta_1 - V_{be}\right)^{N_1}} + \theta_N \, T_{CN} \left(I_N + I_{es}\right)$$

$$C_{bc} = \frac{\alpha_N}{\left(\theta_2 - V_{bc}\right)^{N_2}} + \theta_I \, T_{CI} \left(I_I + I_{CS}\right)$$

$$I_{be} = \left(\frac{1}{\beta N} + 1\right) I_N - I_I \qquad I_{bc} = - I_N + \left(\frac{1}{\beta I} + 1\right) I_I$$

$$I_N = I_{es}\left(e^{\theta_N V_{be}} - 1\right) \qquad I_I = I_{CS}\left(e^{\theta_I V_{bc}} - 1\right)$$

$$\beta_N, \; T_{CN} = f\left(I_N\right) \qquad \beta_I, \; T_{CI} = f\left(I_I\right)$$

N_1, N_2 are proportionality constants

These parameters may be converted to Ebers-Moll parameters

TABLE B-1

CHARGE CONTROL TO EBERS-MOLL CONVERSION RELATIONS

Charge-Control	Ebers-Moll	Charge-Control	Ebers-Moll
I_{es}	$\frac{\alpha_N I_{es}}{1 - \alpha_N \alpha_I}$	T_{CI}	$\frac{10^{-9}}{2\pi \alpha_I F_I}$
I_{cs}	$\frac{\alpha_I I_{cs}}{1 - \alpha_N \alpha_I}$	θ_N	$\frac{38.9\,(298)}{M_e(T + 273)} = \frac{q}{M_e KT}$
T_{CN}	$\frac{10^{-9}}{2\pi \alpha_N F_N}$	θ_I	$\frac{38\ 9\,(298)}{M_e(T + 273)} = \frac{q}{M_c KT}$

Other parameters correspond directly.

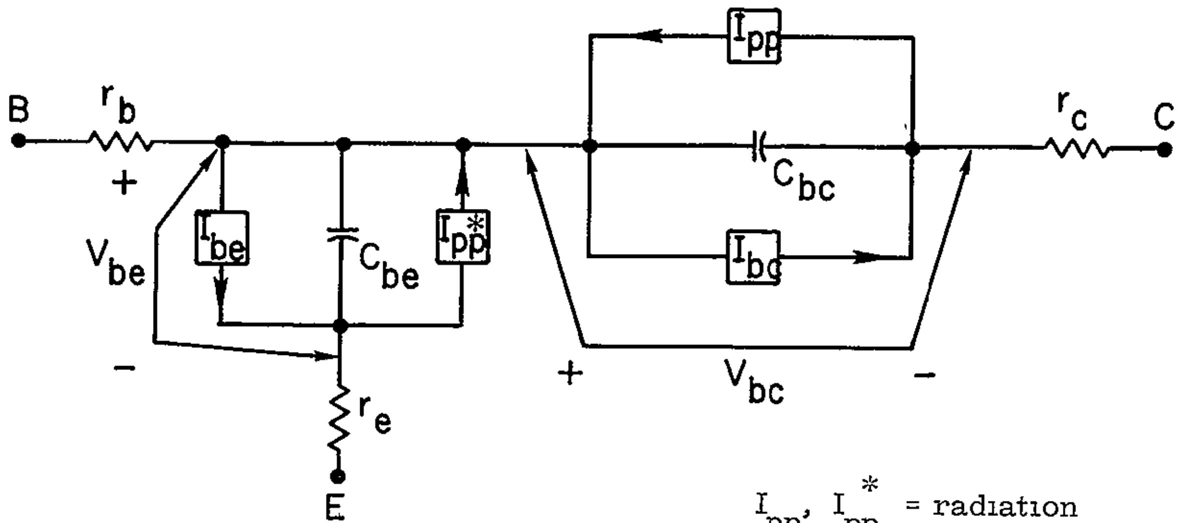


Figure B 13 CIRCUS Prestored Charge Control Model — NPN[22]

The Linvill Lumped Model The Linvill model uses lumped elements to represent paths through which charge will flow in a transistor, the rate and amount of this charge-flow being governed by hole and electron concentrations and voltage The lumped elements are given in terms of length dimensions and electrical constants. A reduced lumped diffusion model for a transistor is given in Figure B-14 [1, 23-25]

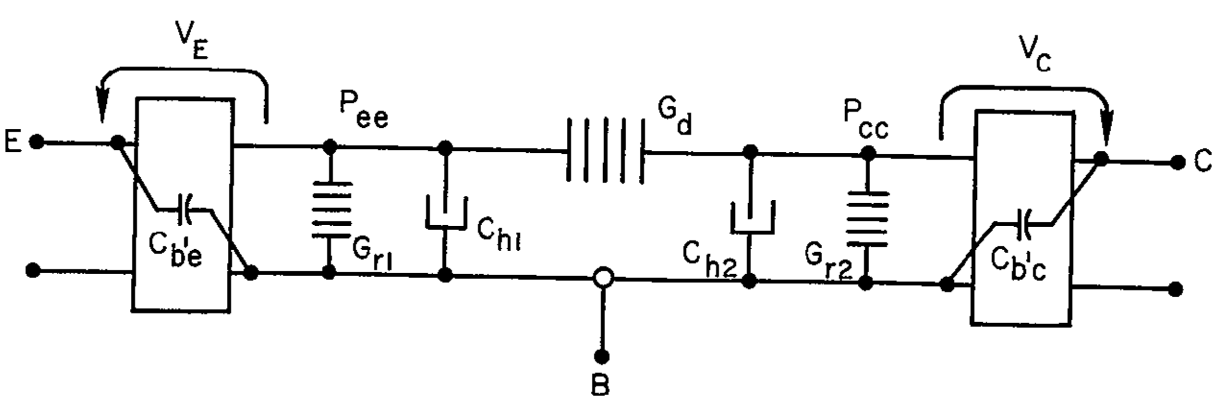


Figure B 14 Linvill's Two Lump Transistor Model

Equations Showing Parallel to Ebers-Moll Model:

$$I_e = \frac{-I_{EO}}{1-\alpha_N\alpha_I}\left(e^{V_e/M_e\theta}-1\right) - T_N\frac{d}{dt}\left[\frac{I_{EO}}{1-\alpha_N\alpha_I}\left(e^{V_e/M_e\theta}-1\right)\right]$$

$$+\frac{\alpha_N - I_{CO}}{1-\alpha_N\alpha_I}\left(e^{V_c/M_c\theta}-1\right)+\frac{d}{dt}\left(C_{je}V_e\right)$$

$$I_c = \frac{-I_{CO}}{1-\alpha_N\alpha_I}\left(e^{V_c/M_c\theta}-1\right) - T_I\frac{d}{dt}\left[\frac{I_{CO}}{1-\alpha_N\alpha_I}\left(e^{V_c/M_c\theta}-1\right)\right]$$

$$+\frac{\alpha_N I_{EO}}{1-\alpha_N\alpha_I}\left(e^{V_e/M_e\theta}-1\right)+\frac{d}{dt}\left(C_{jc}V_c\right)$$

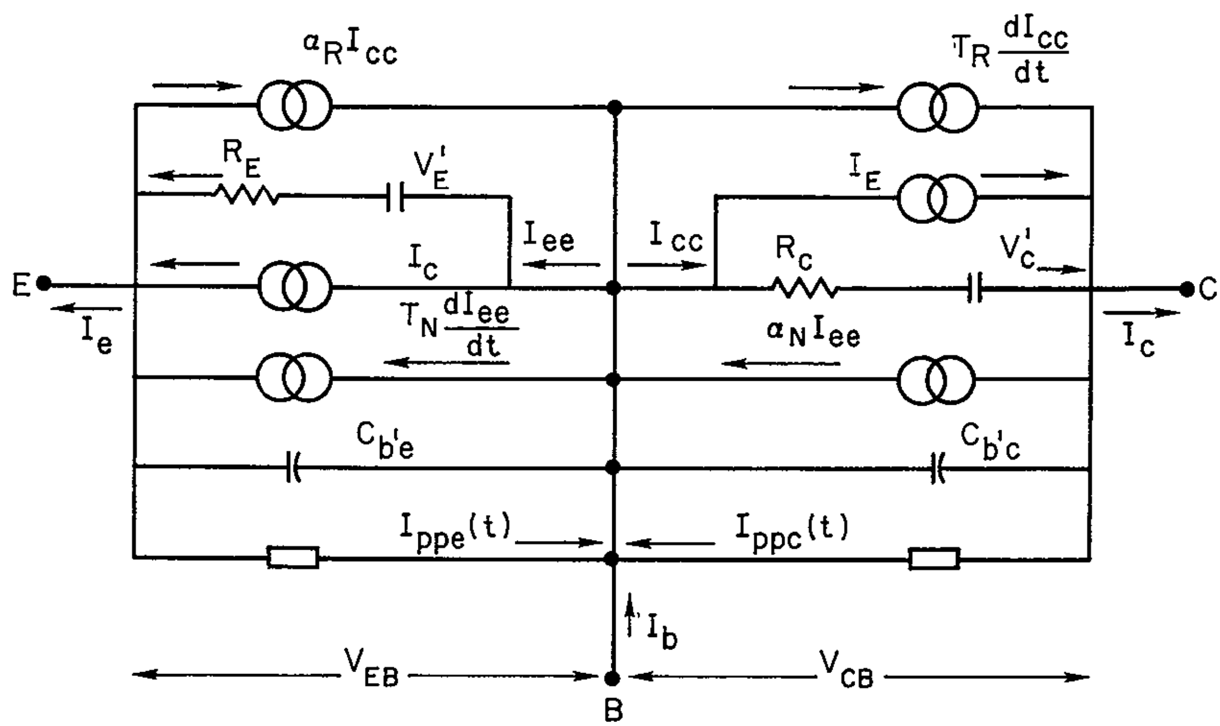
where:

$$I_{EO} \triangleq -p_n\left(G_{r1}+\alpha_I G_{r2}\right) \qquad I_{CO} \triangleq -p_n\left(G_{r2}+\alpha_N G_{r1}\right)$$

$$T_N \triangleq \frac{Ch_1}{G_d+G_{r1}} \qquad T_I \triangleq \frac{Ch_2}{C_d+G_{r2}} \qquad \alpha_N \triangleq \frac{G_d}{G_{r1}+G_d} \qquad \alpha_I \triangleq \frac{G_d}{G_{r2}+G_d}$$

G_d	diffusion current generator
G_{r1}, G_{r2}	recombination centers
Ch_1, Ch_2	storance
P_{cc}, P_{ce}	excess holes in collector, emitter, $P_{cx} \triangleq p_n\left(e^{V_x/M_x\theta}-1\right)$

Program Models A version of the Linvill model has been developed for use as the prestored model for the TRAC program.[26] The model and its parameters are shown in Figure B-15 The model is dependent on past values of V_{cb}, V_{eb}, $\left(V_{cb}^{\,j-1} = V_C^1,\ V_{eb}^{\,j-1} = V_E^1 \text{ for the } j^{th} \text{ step}\right)$



$$R_C = \frac{m_c \theta (1 - \alpha_N \alpha_R)}{I_{co}(e^{v_c'/m_c \theta})}$$

$$R_E = \frac{m_e \theta\ (1 - \alpha_N \alpha_R)}{I_{co}(e^{VE'/me\theta})}$$

$$I_C = \frac{(e^{vc'/mc} - 1) I_{CO}}{1 - \alpha_N \alpha_R}$$

$$I_E = \frac{I_{EB}}{1 - \alpha_N \alpha_R} \left(e^{ve'/me} - 1 \right)$$

$I_{ppe}(t)$, $I_{ppc}(t)$ = radiation induced pulse currents

Figure B 15 TRAC Model — Prestored Lumped Model—NPN

SUMMARY

The most commonly used small signal models, the Modified Tee and the Hybrid-Pi, have been presented with a description of the CAD programs that accept them. The three large signal models have also been briefly presented with a discussion of the CAD programs that have incorporated them.

The Hybrid-Pi is suitable for AC and small signal transient analysis and is the most effectively used model for ECAP

Large signal analysis requires a degree of complexity ECAP cannot accept without a loss of efficiency and accuracy. NET-1 attempted analysis with only a complex Ebers-Moll model with extensive data requirements and this has been found severely restrictive. TRAC has attempted analysis with only a complex version of the Linvill model and has met the same restrictions. CIRCUS attempts analysis with a charge-controlled model with less stringent requirements, is more effective but is also restricted. Some piece-wise linear modeling may be effected but this is generally cumbersome, especially without a dependent current source

SCEPTRE is the first major program to combine the features of the prestored models and the flexibility of the user-derived models SCEPTRE is capable of both small and large signal analysis with its prestored models

It should be reemphasized that there is no single model capable of all analysis since such a model would be cumbersome and overly complex for any given analysis just as there is no ideal CAD program to perform analysis. The selections of the model and the program are both determined from considerations of the application of the device, the availability of parameter data and the accuracy requirements of the solution among others.

Table B-II presents a summary of some major CAD programs and the models included in this paper, Table B-III is a summary of symbols applied throughout the text.

TABLE B-II

PROGRAM MODELING CAPABILITIES

Program	NASAP	ECAP	SCEPTRE	CIRCUS	TRAC	NET-1	PREDICT
Model:							
<u>Small Signal</u>							
Low Frequency							
Hybrid	E	E	P	N	N	N	E
Tee	E	E	E	N	N	N	E
High Frequency							
Hybrid-Pi	E	E	E	N	N	N	E
Modified Tee	E	E	E	N	N	N	E
<u>Large Signal</u>							
Charge Control	D	D	E	P	N	N	E
Ebers-Moll	D	D	P	D	N	P	E
Linvill Lumped	D	D	E	D	P	N	E

<u>Key</u>

E Engineer Derived

P Prestored Model

D Inefficient, difficult

N Not acceptable

TABLE B-III

KEY TO SYMBOLS USED IN TEXT

A manufacturer's data sheet typically contains information on I_{co}, $V_{CE(sat)}$, $V_{BE(sat)}$, h_{FE}(Beta), C_{ob} and f_T, as well as stress limits for the device. Many also include the small signal parameters h_{re}, h_{ie}, h_{oe}, h_{fe} (AC amplification). Based on this information and the knowledge of what load current is desired (small signal operating point or large signal — maximum stress on the device) the designer/analyst must derive the parameter values (or functions of values) for the selected device.

The following is a tabulation of the symbols used in this paper and their definitions Those parameters necessary for analysis include the generally applied approximation for their computation.

B, B' ; b, b'	Base terminal, suffix-parameter related to base characteristics
C, C' , c, c'	Collector terminal, suffix-parameter related to collector characteristics
$C_{b'c}$	Capacity from base internal point to collector, generally labeled C_{ob} on data sheet
$C_{b'e}$	Capacity from base internal point to emitter $C_{b'e} \approx \frac{1}{W_T r_e} \approx \frac{g_m}{2\pi f_T}$
C_c, C_e	Collector, emitter capacity - large signal symbol
C_{ge} or C_{de} (C_{sc} or C_{dc})	Diffusion capacity of emitter, collector, corresponds to $C_{b'e}$, $C_{b'c}$
C_T	Transition capacity, $C_T \ll C_{b'e}$ and $\therefore$ generally omitted
E, E' , e, e'	Emitter terminal, suffix-parameter related to base characteristics
f_α	Alpha cut off frequency, $f_\alpha \approx \beta_o f_\beta \approx f_T$
f_β	Beta cut off frequency
f_T	Current gain - bandwidth product

g_m	Transconductance, $g_m = \frac{\alpha_N}{r_e}$, $g_m = \frac{h_{fe}}{1+hf_e}$, for germanium, room temp. $h_{fe} \gg I_E$ in ma, $g_m \approx \frac{I_e}{26}$ ma
h_i	Input impedance, small signal, $\approx \frac{\Delta V_{in}}{\Delta i_{in}} \Big\vert_{\Delta V_{out} = 0}$
h_r	Feedback voltage ration, small signal, $\frac{\Delta V_{in}}{\Delta V_{out}} \Big\vert_{\Delta i_{in} = 0}$
h_f	Current transfer ratio, small signal, $\frac{\Delta i_{out}}{\Delta i_{in}} \Big\vert_{V_{out} = 0}$ $\beta \approx h_{fe}$
h_o	Output admittance, small signal, $\frac{\Delta i_{out}}{\Delta V_{out}} \Big\vert_{\Delta i_{in} = 0}$
I_{co}	Collector to Base saturation current when $I_E = 0$
I_{Eo}	Emitter to Base saturation current when $I_c = 0$
K	Boltzmann's Constant (8 6310^{-5} ev/^oK
M_E, M_c	Emitter, collector proportionality constants ($1 \leqq M_x \leqq 3$) (emission constant)
n	Electron density
p	Hole density
q	Electron charge (1 6 x 10^{-19} coulomb)
q_B	Control charge in base region
$r_{b'}$, v_b'', r_b	Components of $r_{bb'}$, base resistance generally $r_{bb'} \approx 0$ compared to circuit components ($r_{bb'} = h_{ie} - r_{b'e}$)
r_b	Small signal base resistance, Tee configuration, $r_b \approx h_i - \frac{h_f h_r}{h_o}$

$r_{b'c}$	Resistance-base internal point to collector, generally very large, appears as an open to a CAD program .. generally omitted, $r_{b'c} = \frac{r_{b'e}}{h_{re}} = \frac{1}{g_{b'c}}$
$r_{b'e}$	Resistance-base internal point to emitter, $r_{b'e} \approx (\beta+1) r_e$
r_c, r_{CL}, r_{ce}	Collector resistance $\approx \frac{1}{h_o}$, also appears as an open generally omitted
r_e	Emitter resistance, $\approx \frac{h_r}{h_o} \approx \frac{kT}{qI_e}$ at $27^\circ C$, $\frac{kT}{q} = 26$ mv
S_B	Base store
T	Absolute temperature - $^\circ$Kelvin
V_T	Electron-volt equivalent of temperature, $V_T = \frac{kT}{q}$
α_I	Inverted current gain
α_N	Normal current gain
α_o	Low frequency α_N
β_I	Inverted Beta $\approx \frac{\alpha_I}{1-\alpha_I}$
β_N	Normal Beta $\approx \frac{\alpha_N}{1-\alpha_N}$
β_o	Current gain at edge of saturation
λ	$\frac{q}{\eta kT}$ symbol for simplification (also θ), $\lambda = \frac{1}{\eta V_T}$
η	Parameter accounting for recombination of carriers in the junction transition region, dependent on material $1 \leqq \eta \leqq 3$ $\eta \approx 1$ for germanium $\eta \approx 2$ for silicon
ϕ_e, ϕ_c	Junction voltages ($V_{b'e}$, $V_{b'c}$)
ω_{α_o}	Alpha cut off frequency, radians

ω_T	$\sim \frac{\omega_\alpha}{1.2} = 2\pi\lambda_T$ current gain bandwidth product
τ_{BF}	Base charge time constant
τ_F	Collector charge control time constant

Typical Values of h parameters[8]
(common Emitter Configuration)

h_{ie}	1,100
h_{re}	2.5×10^{-4}
h_{fe}	50
$1/h_{oe}$	40K

BIBLIOGRAPHY

1. John G. Linvill, "Lumped Models of Transistors and Diodes," Proc IEEE, Vol. 46, (June 1958), pp 1141-1152.

2. John G. Linvill, Models of Transistors and Diodes, New York, McGraw Hill Book Co., Inc., 1963

3. Marvin E Daniel, "Development of Mathematical Models of Semiconductor Devices for Computer Aided Circuit-Analysis," Proc IEEE, Vol. 55, No. 11, (1966), pp. 1913-1920.

4. D. T. Hamilton, F. A. Lindholm, J. A. Narod, "Comparison of Large Signal Models for Junction Transistors," Proc. IEEE, Vol 52, (March 1964), pp. 239-248.

5. John J. Sparkes, "Device Modeling," IEEE Trans., Vol. ED-14, (May 1967), pp. 229-232.

6. Cyrus O. Harbourt, "Doing a Model-Job," Electronics, Jan 23, 1967, pp. 82 - 87.

7. John Franklin Pierce, Transistor Circuit Theory and Design, Ohio: Charles E. Merrill Books, Inc., 1963

8. Jacob Millman, Herbert Taub, Pulse, Digital, and Switching Waveforms, New York: McGraw Hill Book Co., 1965.

9. Maurice W. Joyce, Kenneth K Clarke, Transistor Circuit Analysis, California Addison Wesley Publishing Co., Inc. 1961.

10. Franklin C Fitchen, Transistor Circuit Analysis and Design, New York D. Van Nostrand Co , Inc , 1960.

11 Harry W. Mathers, et al , Automated Digital Computer Program for Determining Responses of Electronic Circuits to Transient Nuclear Radiation (SCEPTRE), Vol. 1 and Vol. 2, New York: IBM, Feb. 1967.

12. (Staff), 7090/94 ECAP Level No. 3, The Service Bureau Corp., H20-0170-1, 1966.

13. John L. Moll, "Large-Signal Transient Response of Junction Transistors," Proc. IEEE, Vol. 42, (Dec. 1954) pp. 1773-1784.

14. J. J. Ebers, John L. Moll, "Large-Signal Behavior of Junction Transistors," Proc. IEEE, Vol. 43, (Dec. 1955), pp. 1761-1772

15. R. Beaufoy, J. J. Sparkes, "The Junction Transistor as a Charge Controlled Device," AT&T Journal, Vol XIII, 1957, pp 310-327.

16. A Eugene Anderson, "Transistors in Switching Circuits," Proc IEEE, Vol. 40, (Nov. 1952), pp. 1541-1548.

17. Randall, W. Jensen, "Charge Control Transistor Model for the IBM Electronic Circuit Analysis Program," IEEE Trans., Vol. CT-13, Dec. 1956, pp. 428-437.

18. P.E. Gray, et al., Physical Electronics and Circuit Models of Transistors, New York: John Wiley & Sons, Inc., 1964, (Vol 2 of SEEC Series).

19. Allan F Malmberg, et al., Net-1 Network Analysis Program 7090/94 Version, New Mexico U. of Calif., Aug. 1964

20. K G. Ashar, H N. Ghosh, A.W Aldridge, L J Patterson, "Transient Analysis and Device Characterization of ACP Circuits," IBM Journal, July 1963, pp 207-223.

21. J.J. Sparkes, "A Study of the Charge Control Parameters of Transistors," Proc. IEEE, Vol. 49, (Oct. 1960), pp. 1695-1704

22. L.P. Milliman, W A Massena, R H Dickhaut, CIRCUS A Digital Computer Program for Transient Analysis of Electronic Circuits User's Guide, Washington The Boeing Co., Jan 1967.

23. Robert N. Beatie, "A Lumped Model Analysis of Noise in Semiconductor Devices," IEEE Trans , Vol. ED-6, (April 1957), pp. 133-140

24. J.F. Gibbons, D A Linden, "Lumped-Model Analysis of Space Charge Widening," Proc IEEE, Vol 49, (Oct. 1960), p. 920

25. M P Beddoes, "Linvill's Lumped Models and the Simplified Model," Proc. IEEE, Vol 54, (May 1965), pp. 552-554

26. G T. Kleiner, G. Kinoshita, and E.D Johnson, "Simulation and Verification of Transient Nuclear Radiation Effects on Semiconductor Electronics," IEEE Trans. Vol NS-11, (1964), pp. 82-104

ADDITIONAL REFERENCES

L. J. Giacoletto, "Study of P-N-P Alloy Junction Transistor from D-C through Medium Frequencies," RCA Review, Vol XV, No. 1, 1954, pp. 506-562.

J. M. Early, "Design Theory of Junction Transistors," Bell System Technical Journal, Vol. XXXII, No. 6, 1953, pp. 1271-1312.

R. L. Pritchard, "Electric-Network Representation of Transistors - A Survey," IEEE Trans., Vol CT-3, March 1956, pp. 5-21

R. H. F. Lloyd, "A Simpler Transistor Model," Proc IEEE, Vol. 54, (May 1965), pp. 527-528.

A. N. Baker, "Charge Analysis of Transistor Operation," Proc IEEE, Vol. 49, (May 1960), pp. 949-950.

Campbell L. Searle, et al., Elementary Circuit Properties of Transistors, New York: John Wiley & Sons, Inc., 1964, (Vol 3 of SEEC Series)

PART II. TRANSISTOR MODELS FOR HOSTILE (WEAPON) ENVIRONMENT EFFECTS

INTRODUCTION

The present concern of circuit specialists, both designers and analysts, has been the degradation of systems upon exposure to nuclear radiation, specifically that resulting from the detonation of a weapon.

The inability of system shielding to be effective for all radiations makes it necessary to design and analyze for the generation of radiation-induced transient and permanent effects in semiconductor devices. There is no laboratory procedure capable of providing either the mixture or the intensity of the radiations produced in a real weapon environment, which may consist of any mixture of photons, charged particles or neutrons As a result, it becomes necessary to turn to the computer. Several computer circuit analysis programs, such as SCEPTRE, CIRCUS, and TRAC, have been developed specifically for the analysis of transient radiation effects on circuits.

The purposes of this paper are (1) to describe the first order radiation effects in semiconductor material in terms of the weapon environment, (2) to describe the physical interactions which produce the altered device behavior, and (3) to discuss the computer program models available for the analysis of an exposed device. The discussion is primarily in terms of bipolar transistors. Because of the practical complexity of a discussion of a weapon or space radiation environment, such a discussion is not attempted here.

THE FIRST ORDER EFFECTS

Radiations consist of high energy photons such as gamma rays and energetic particles such as neutrons, electrons and fission fragments.

The effects initiated by the high energy photons are primarily transient ionization and excitation of the absorbing material. The effects initiated by the energetic particles, specifically neutrons, are primarily localized bulk displacements, disruptions of the crystal lattice structure along the particle path Because of the obvious complexity of a simultaneous discussion of each radiation expected in an incident environment, the two classes of radiation will be discussed separately.

IONIZATION

The ionization effects of a pure photon environment result from the interaction of the photons and the atomic electrons of the absorbing material. The photon-atomic electron interaction produces charged particles which in turn produce ionization through further interactions.

The three primary photon-atomic electron interaction processes are [1, 2]

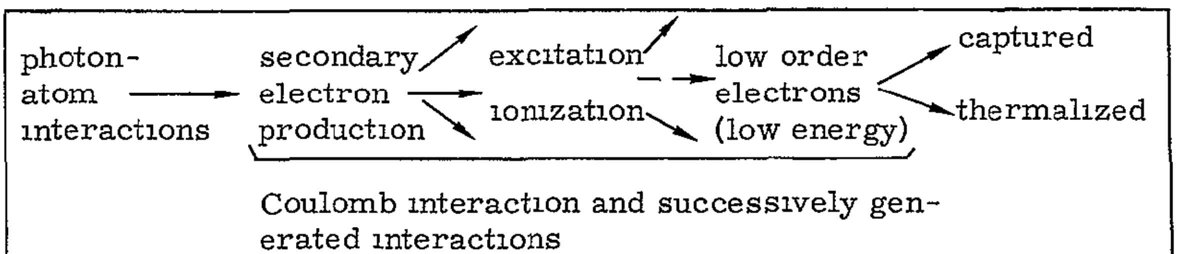
1. <u>The Photoelectric Effect</u> The inelastic collision of a photon and an atomic electron, the photon energy is absorbed by the electron which is ejected from the atom if the photon energy exceeds the binding energy of the electron. The extent of the photoelectric interaction is dependent on both the photon energy and the atomic number of the absorbing material and is most important at low photon energies for materials of high atomic number

2. <u>The Compton Effect</u> The elastic collision of a photon and an atomic electron of the absorbing material in which part of the photon energy is transferred to the electron, both the electron and a reduced-energy photon are scattered. The Compton effect is most important in an energy range between 100 KeV and 4 MeV and is more significant for materials with a high atomic number

3. <u>Pair-Production</u> The interaction of the electric field of the atomic nucleus (or the field of an atomic electron) and a photon in which the photon energy (when E is > 1 02 MeV) is absorbed in the formation of an electron-positron pair The positron is rapidly annihilated by an

electron with the interaction resulting in the production of two lower-energy photons. Pair-production is most important for high photon energies and is especially important for heavy elements It is the major reaction process for high energies

The result of each of these three major interaction processes is the production of one or more secondary electrons. The Boulomb interaction of these electrons produces additional ionization and excitation. Successive generation of the interactions ceases when low energy electrons are produced which do not have sufficient energy to continue the production of ionization or excitation. These last-order electrons will lose energy by elastic scatterings until they are either captured or thermalized



Interactions Produced in a Photon Environment

Ionization is also created by energetic particles. The ionization effects of a primary beam of charged particles (electrons, positrons) are identical to the ionization effects of the secondary electrons produced by photon irradiations. Heavy charged particles (protons, fission fragments) create ionization along the primary particle path of such a high intensity that electron-positron charge recombination occurs rapidly. The net effect of the heavy particles is less than the effect of an equivalent, more uniformly distributed energy deposition [3]

Neutrons are heavy uncharged particles that produce ionization only through secondary processes. Neutron ionization effects are a function of neutron energy. The important neutron interaction processes are (1) elastic scattering- a collision of a neutron and a nucleus with transfer of kinetic energy, and (2) inelastic scattering - a collision of a high energy neutron and a nucleus with conversion of some of the neutron kinetic energy into excitation energy in the struck nucleus. The excited nucleus returns to its stable state by emitting the excess energy as a photon [1, 2]

Ionization produces primarily transient effects in the incident material, displacement produces primarily permanent effects

DISPLACEMENT

For the typical energetic neutrons produced in nuclear reactions, the production of atomic displacements is the principal interaction process through which neutron energy is dissipated. Displacement effects are the result of defects in the crystal lattice structure that introduce additional energy states in the energy band gap. These defects behave either as additional recombination centers, which can produce a reduction in minority carrier lifetime or they can alter the impurity concentration of the affected material, which can produce alterations in the physical properties of the material Large defect clusters result from a neutron interaction in which a large amount of the neutron kinetic energy is transferred to a single atom.[4]

Ionization and displacement defects can produce significant alterations in the electrical behavior of the affected material.

MANIFESTATION OF THE EFFECTS

The most significant result of the three primary photon-interaction processes is the production of free electron-hole pairs. The number of electron-hole pairs created in semiconductor material is proportional to the amount of energy absorbed from the radiation environment The differences between the exposure and absorbed radiation environments can be a complex function of the equilibrium radiation environment at the specimen. The complex prediction of carrier generation rate may be simplified by considering the energy deposition to be uniform throughout the geometry of the device if the incident particle or photon energy exceeds 200 KeV For photon or electron irradiation between 1 to 5 MeV, the absorbed dose is essentially identical to the exposure dose [3]

Radiation induced carrier generation in the bulk semiconductor will increase the minority and majority carrier densities.

$$\Delta n = \Delta p = g_o \dot{\gamma}(t) \qquad \text{(B-1)}$$

The change in majority and minority carriers is seen to be equal and therefore the relative effect on minority carrier density is greater The minority carrier density increase results in the junction photocurrents and secondary photocurrents in diodes and transistors.

Displacement damage in the bulk semiconductor introduces a number of recombination centers, which is generally assumed to be proportional to the integrated neutron flux. The primary result is the reduction of minority carrier lifetime.(4)

$$\frac{1}{\tau_F} = \frac{1}{\tau_o} + \frac{\Phi}{K} \qquad \text{(B-2)}$$

where τ_F is the new lifetime

τ_o is the initial lifetime

K is the lifetime damage constant

Φ is the integrated neutron flux

In terms of device performance, the reduction in minority carrier lifetime produces a transient reduction in the forward gain of a transistor that is not completely recovered. The damage value of Beta for a single radiation pulse may be obtained from the Messenger-Spratt Equation:[4]

$$\frac{1}{\beta_F} = \frac{1}{\beta_o} + \frac{0.194}{f_\alpha} \frac{\Phi}{K} \qquad \text{(B-3)}$$

The detailed analysis of displacement damage in transistors is complicated by the variation of the common emitter current gain (β) with the injection level.

THE PHYSICAL DESCRIPTION

In semiconductor material, electrical currents are the result of the motion of either holes or electrons and are associated with two independent mechanisms (1) Drift motion, the result of charged particles being acted upon by an electric field, and (2) Diffusion, the result of the tendency of carriers to diffuse from regions of high charge concentration to regions of low concentration. The net current density is the sum of the drift and diffusion component currents.[5]

Carrier density that is greater than the equilibrium carrier density is termed excess carrier density. Excess carriers can be introduced by electrical, optical, thermal or radiation injection mechanisms Excess carriers are opposed by the recombination process (mutual annihilation) by which holes and electrons are removed from circulation as charge carriers. The net rate of recombination of excess holes and electrons is approximately

$$\text{n-type} \qquad \text{Rate} = \frac{P_{(x)} - P_o}{\tau_p} = \frac{P'_{n(x)}}{\tau_p} \tag{B-4}$$

$$\text{p-type} \qquad \text{Rate} = \frac{n_{(x)} - n_o}{\tau_n} = \frac{n'_{p(x)}}{\tau_n} \tag{B-5}$$

for one dimensional, conductive, extrinsic, homogeneous material operating with a low injection rate.

The carrier generation rate by external mechanisms, g, has been shown to be $g = g_o \dot{\gamma}(t)$ for ionizing radiation, where $\gamma(t)$ is the exposure in radians.

The continuity of current requires

$$\frac{\partial p}{\partial t} = \frac{-1}{qA} \frac{\partial i_p}{\partial x} - \frac{\left(p'_{n(x)}\right)}{\tau_p} + g_o \gamma (t) \quad \text{(B-6)}$$

n-type:

$$\frac{\partial n}{\partial t} = \frac{1}{qA} \frac{\partial i_n}{\partial x} - \frac{\left(p'_{n(x)}\right)}{\tau_p} + g_o \gamma (t) \quad \text{(B-7)}$$

The equations of current flow are.

$$J_{holes} = q \mu_p p(x) E - q D_p \frac{\partial p}{\mu x} \quad \text{(B-8)}$$

n-type

$$J_{electrons} = q \mu_n n(x) E + q D_n \frac{\partial n}{\partial x} \quad \text{(B-9)}$$

Drift Diffusion

$$I = A \left(J_{holes} + J_{electrons}\right) \quad \text{(B-10)}$$

and the presence of the electric field requires the application of Gauss' Law·

$$\frac{\partial E}{\partial x} = \rho(x) = \frac{q}{\epsilon} \left[N_d(x) - N_a(x) + p(x) - n(x)\right] \quad \text{(B-11)}$$

The terminal relationships of a device may be derived from these equations.[5] To demonstrate briefly the relationship of the physical behavior and the transistor models available for computer aided analysis, the equations of an ideal diode and an ideal PNP transistor are included Because of the complexity of the equations, worst case steady state conditions are primarily discussed.

AN IDEAL DIODE

The current appearing at the terminals of this device will be related to the internal redistribution of the carriers upon application of a bias potential. The following analysis is based on an abrupt junction device composed of homogeneous n-type and p-type material with a transition or space charge layer surrounding the junction of width W.[6]

For forward bias, assuming carrier flow by diffusion only

$$\text{n-type} \quad J_n(o) = \frac{q\, D_n P_{no}}{L_n} \left(e^{qv/KT} - 1\right) = \frac{q\, D_n p'_{n(o)}}{L_n} \qquad \text{(B-12)}$$

sine $I = A\left(J_h(o) + J_e(o)\right)$, substituting $Ln = \sqrt{D_n \tau_n}$ and including similar expressions for the electrons

$$I_{\text{diode forward}} = Aq\left(\frac{L_n p_{no}}{\tau_n} + \frac{L_p n_{po}}{\tau_p}\right)\left(e^{qv/KT} - 1\right) \qquad \text{(B-13)}$$

$$= I'_s\left(e^{qv/DT} - 1\right) \qquad \text{(B-14)}$$

Adding the effects of the space charge region contribution to reverse current, for reverse bias

$$I_{\text{diode reverse}} = Aq\left(\frac{L_n p_{no}}{\tau_n} + \frac{L_p n_{po}}{\tau_p} + \frac{W_{ni}}{\tau_o}\right) \qquad \text{(B-15)}$$

$$= I_s$$

In a steady state ionizing radiation environment where the carrier generation is increased by $g_o \ \gamma(t)$ the equation for reverse bias becomes:[3]

$$I_d = Aq\left[Ln\left(\frac{P_{no}}{\tau_n} + g_o\ \gamma(t)\right) + L_p\left(\frac{n_{po}}{\tau_p} + g_o\ \gamma(t)\right) + W\left(\frac{n_i}{\tau_o} + g_o\ \dot{\gamma}(t)\right)\right] \qquad \text{(B-16)}$$

The equilibrium diode photocurrent can be seen to be

$$I_{dp\ p} = Aq\ go(\dot{\gamma})\left[L_n + L_p + W\right] \qquad \text{(B-17)}$$

For pulsed transient photoresponse, the complexity of the equation is considerably increased [3]

A step function

$$\text{response} \quad I_{dp\ p}(t) = Aq\left[L_n\ \text{erf}\left(\frac{t}{\tau_n}\right)^{\frac{1}{2}} + W\ u(t)\right] \qquad \text{(B-18)}$$

Figure B-16 diagrams the minority-carrier concentrations for forward and reverse bias, demonstrating the increase in minority carrier density

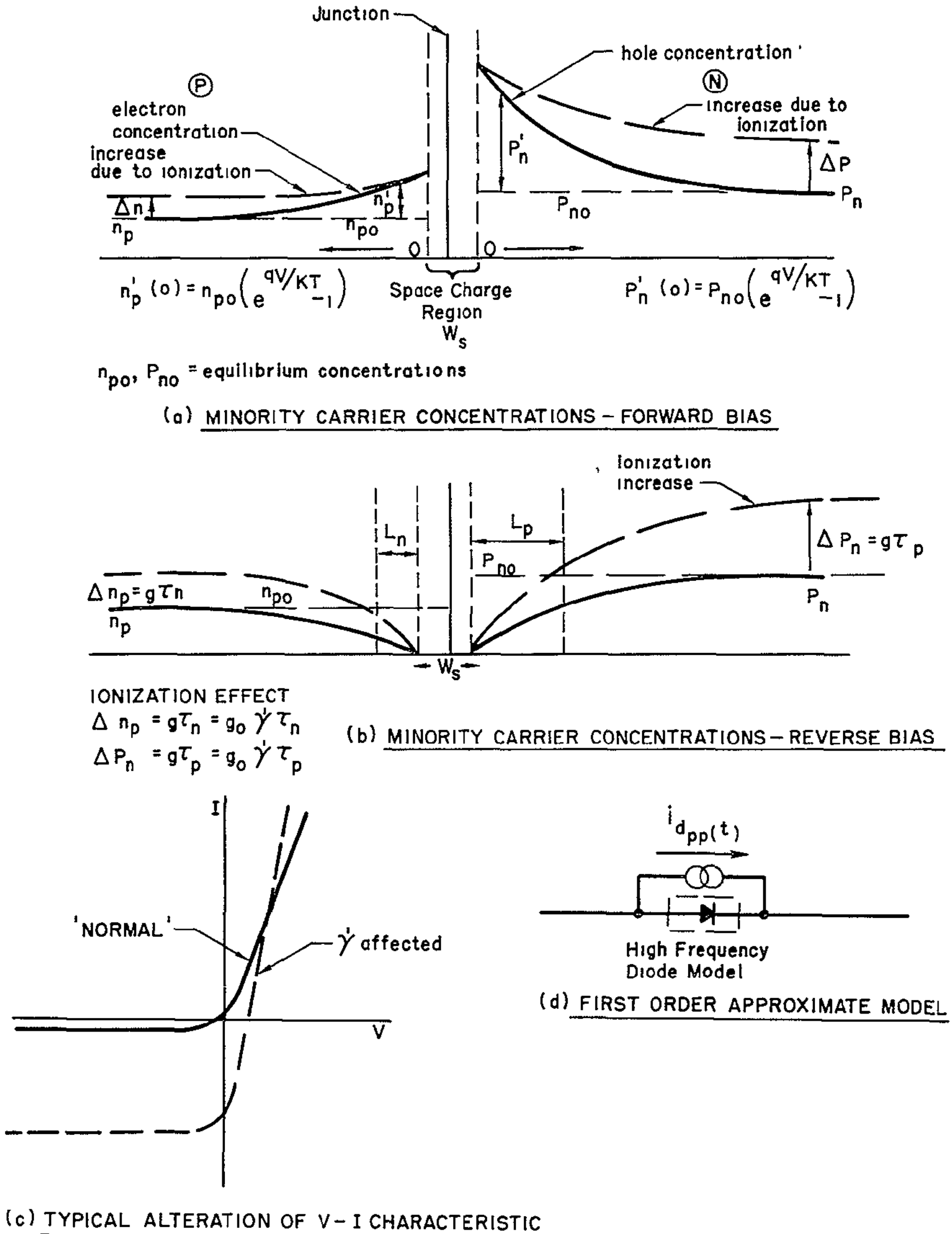


Figure B 16 P–N Junction Ideal Diode (1,2,3) as a result of ionizing radiation absorption Figure B-16c diagrams the alterations of a diode V-I characteristic A current generator $i_d pp(t)$ placed across the junction in parallel with the high frequency model of the device has proven satisfactory as a first-order approximation model.

THE TRANSISTOR MODELS

There are three large signal nonlinear models (1) Ebers-Moll, (2) the Beaufoy-Sparkes Charge Control, and (3) the Linvill Lumped Model, that have been modified for radiation effect computer analysis

EBERS-MOLL

A reversed biased ideal PNP transistor with abrupt junctions, $W_{base} < L_{base}$ that is extrinsic and homogeneous in all three regions and that is operating with a low injection level is diagramed in Figure B-17 For the cut-off mode $V_{EB} \ll -\frac{KT}{q}$, $V_{CB} \ll -\frac{KT}{q}$, bias conditions are [3]

$$I_C = q\, A_{jc} \left[L_c \frac{n_{co}}{\tau_{nc}} + \frac{W_b}{2} \frac{p_{bo}}{\tau_{pb}} + x_c \frac{n_i}{\tau_{oc}} \right] \tag{B-19}$$

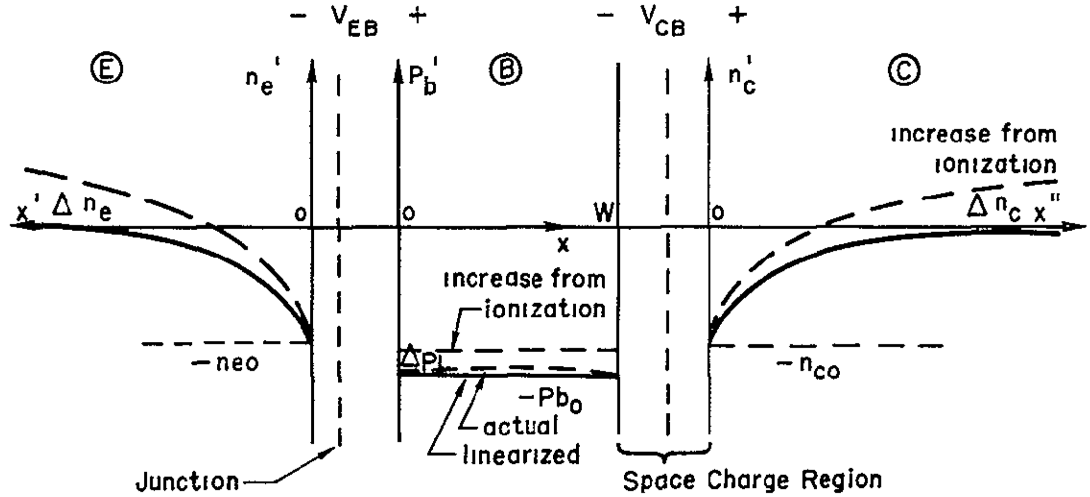
$$I_E = q\, A_{je} \left[L_e \frac{n_{eo}}{\tau_{ne}} + \frac{W_b}{2} \frac{p_{bo}}{\tau_{pb}} + x_e \frac{n_i}{\tau_{oe}} \right] \tag{B-20}$$

electron hole space-charge region

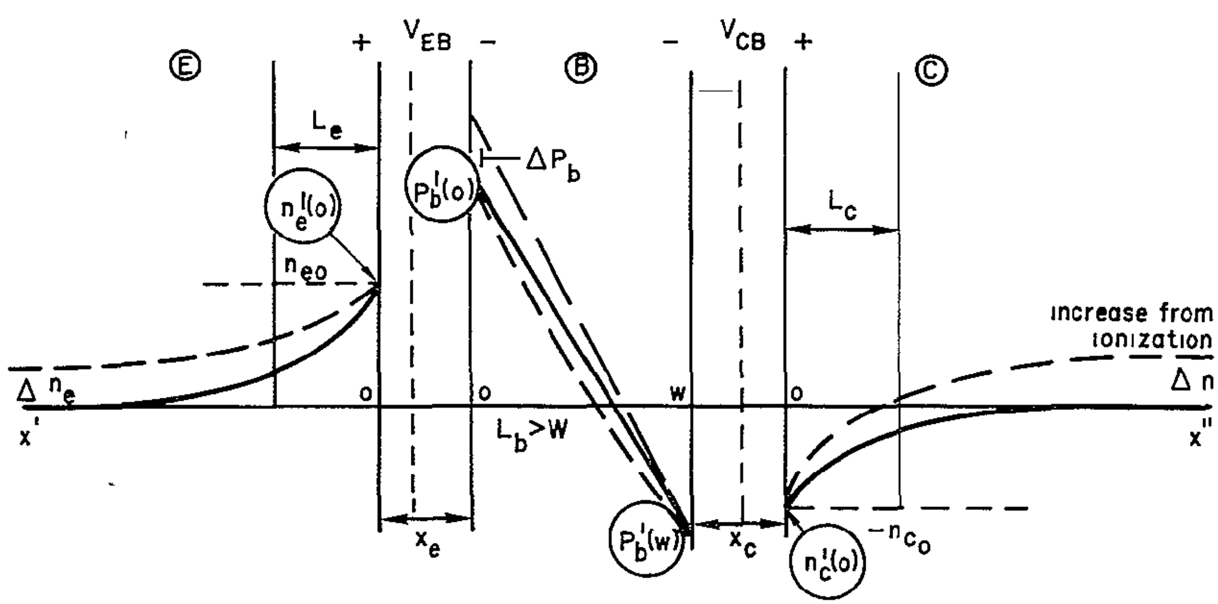
$$I_B = - I_E - I_C \tag{B-21}$$

When ionizing radiation effects are included the equation for equilibrium effects (steady state) are

$$\begin{aligned} I'_c &= q\, Ajc \left[L_c \left(\frac{n_{co}}{\tau_{nc}} + g_o(\dot{\gamma}) \right) + \frac{Wb}{2} \left(\frac{p_{bo}}{\tau_{pb}} + g_o(\gamma) \right) + x_c \left(\frac{n_i}{\tau_{oc}} + g_o(\dot{\gamma}) \right) \right] \\ &= I_c + I_{ppc} \end{aligned} \tag{B-22}$$



MINORITY CONCENTRATIONS IN THE CUTOFF MODE – PNP



MINORITY CONCENTRATIONS IN THE ACTIVE MODE – PNP

$$P_b'(o) = Pb_o\left(e^{\frac{qV_{EB}}{KT}}-1\right) \qquad n_e'(o) = n_{eo}\left(e^{\frac{qV_{EB}}{KT}}-1\right)$$

$$P_b'(w) = Pb_o\left(e^{\frac{qV_{CB}}{KT}}-1\right) \qquad n_c'(o) = n_{co}\left(e^{qV_{CB}/KT}-1\right)$$

Figure B 17 PNP Ideal Transistor Minority Carrier Concentrations

$$I_E' = q\,A_{je}\left[L_e\left(\frac{n_{eo}}{\tau_{ne}} + go(\gamma)\right) + \frac{W_b}{2}\left(\frac{P_{bo}}{\tau_{pb}} + go(\gamma)\right) + x_e\left(\frac{n_i}{\tau_{oe}} + go(\gamma)\right)\right]$$

$$= I_E + I_{ppe} \tag{B-23}$$

$$I_B' = -I_c' - I_E' \tag{B-24}$$

Similar to the diode approximation, current generators representing i_{ppe} and i_{ppc} placed across the junctions of a transistor high frequency nonlinear model provide a satisfactory first order approximation model.

A nonlinear model (capable of cutoff-active-saturation representation) is necessary because the photocurrents could be sufficient to turn on the device. The photocurrent generators generally replace the i_{CO}, i_{EO} leakage current generators in the nonlinear models The equations for active region operation may be derived by assuming all current is from diffusion and neglecting the space charge layer carrier generation and recombination, then

$$I_E = q\,A_{je}\left[-D_b\frac{dP_b'(o)}{dx} - D_e\frac{d\,n_e'(o)}{dx}\right] \tag{B-25}$$

$$I_c = q\,A_{jc}\left[+D_b\frac{d\,P_b'(w)}{dx} - D_c\frac{d\,n_c'(o)}{dx}\right] \tag{B-26}$$

Since $P_b'(o)$, $n_e'(o)$ $n_c'(o)$ $P_b'(w)$ are linearly dependent upon $\left(e^{qV_{EB}/KT} - 1\right)$ and $\left(e^{qV_{CB}/KT} - 1\right)$ the form of the classic Ebers-Moll equations is readily verified

$$I_E = I_{ES}\left(e^{\frac{qV_{EB}}{KT}} - 1\right) - \alpha_R\,I_{CS}\left(e^{\frac{qV_{CB}}{KT}} - 1\right) \tag{B-27}$$

$$I_C = -\alpha_F\,I_{ES}\left(e^{\frac{qV_{EB}}{KT}} - 1\right) + I_{CS}\left(e^{\frac{qV_{CB}}{KT}} - 1\right) \tag{B-28}$$

$$I_B = -I_E - I_C \tag{B-29}$$

From the cutoff condition

$$I_{E}'{}_{off} = - \left(1 - \alpha_F\right) I_{ES} + i_{ppe} \tag{B-30}$$

cutoff·

$$I_{C}'{}_{off} = - \left(1 - \alpha_R\right) I_{CS} + i_{ppc} \tag{B-31}$$

Using the same procedure, the equations for general operation under radiation exposure may be developed

$$I_E(\dot{\gamma}) = I_E + i_{ppe}(\dot{\gamma}) \tag{B-32}$$

$$I_C(\gamma) = I_C + i_{ppc}(\gamma) \tag{B-33}$$

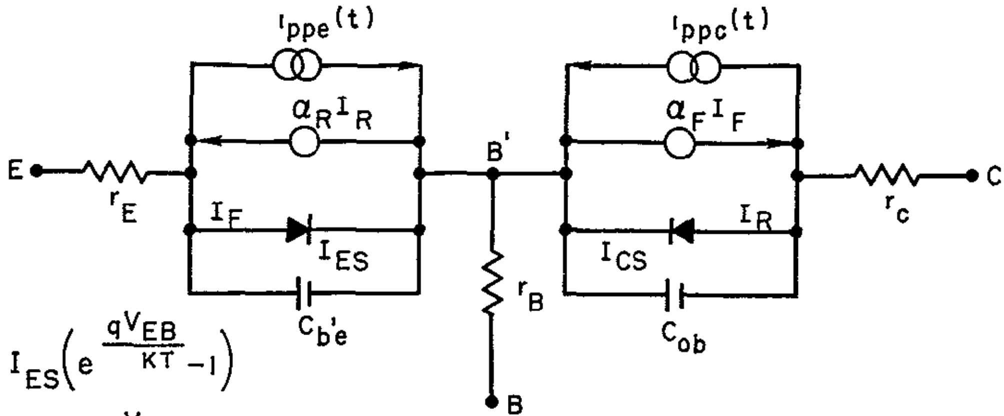
The photocurrent effects on the device and circuit are compounded by multiplication of the photocurrent. The secondary photocurrent so produced could damage the device if it is sufficiently large.

Figure B-18a diagrams the Ebers-Moll PNP Transistor model and modifications to include the junction capacitance, bulk resistance and the photocurrent generators, B-18b shows the version incorporated in the SCEPTRE program

CHARGE CONTROL

The characterization of the ideal transistor described above is in terms of excess minority carriers An alternate description is in terms of minority-carrier charge density distribution Figure B-19 is a diagram of charge density distribution for the active region of operation The total distribution for the four regions of operation is also shown. This is a direct relationship between this internal charge distribution and the operating region [5]

Assuming the ideal device for simplicity and recombination processes occur only in the base and are defined by a uniformly constant lifetime τ_B, for steady state conditions the stored excess minority carrier charge ($+ q_B$) and the equal majority carrier charge ($- q_B$) are maintained by the base current I_B. I_B is defined as a sustaining current which supplies majority electrons at a rate equal to their loss due to recombination in the base. For transient operation, the stored base charge changes with time

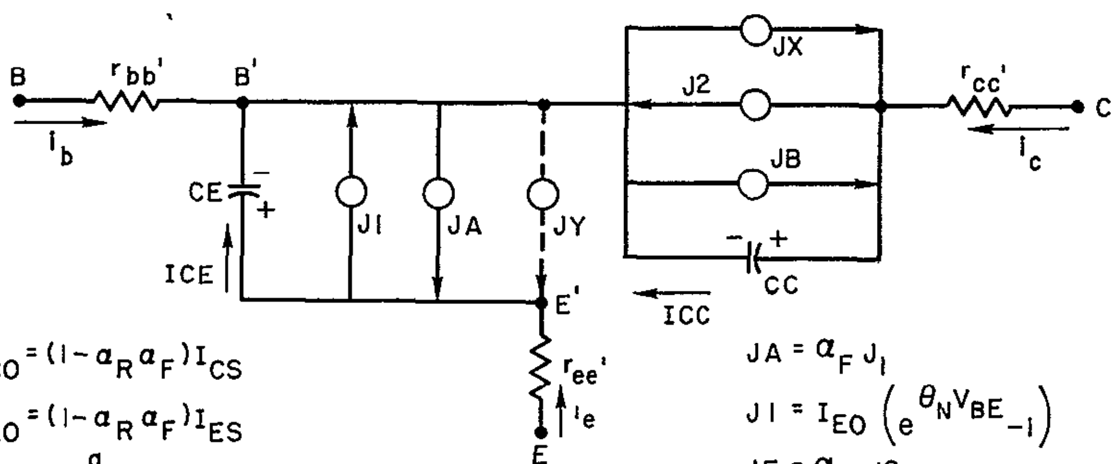


$$I_F = I_{ES}\left(e^{\frac{qV_{EB}}{KT}} - 1\right)$$

$$I_R = I_{CS}\left(e^{\frac{qV_{CB}}{KT}} - 1\right)$$

(a) EBERS – MOLL DEVICE MODEL (NPN)
(TERMINAL VOLTAGE CONTROL)

NOTE: $i_{ppc}(t) >> i_{ppe}(t)$ generally



$$I_{CO} = (1 - \alpha_R \alpha_F) I_{CS}$$

$$I_{EO} = (1 - \alpha_R \alpha_F) I_{ES}$$

$$\theta_N = \frac{q}{\eta KT}$$

$$JA = \alpha_F J_1$$

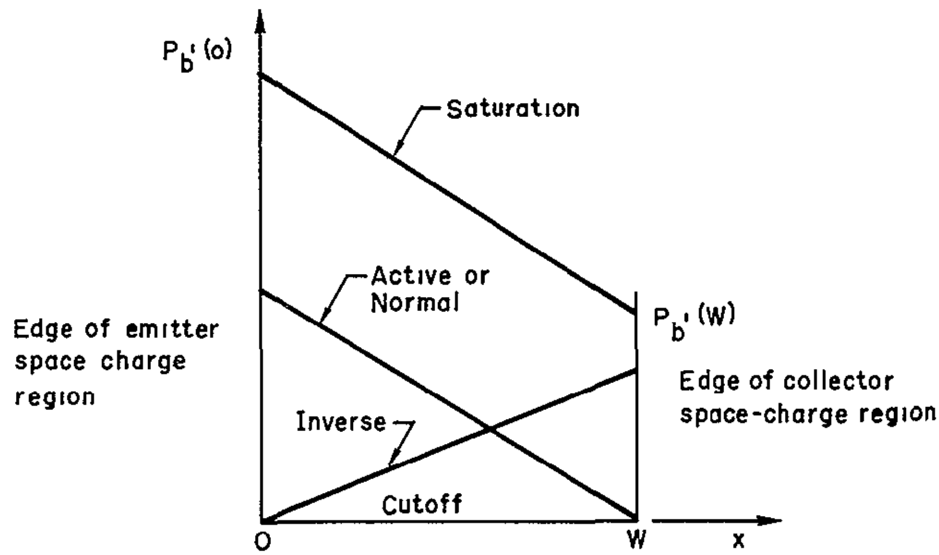
$$J1 = I_{EO}\left(e^{\theta_N V_{BE}} - 1\right)$$

$$JE = \alpha_R J2$$

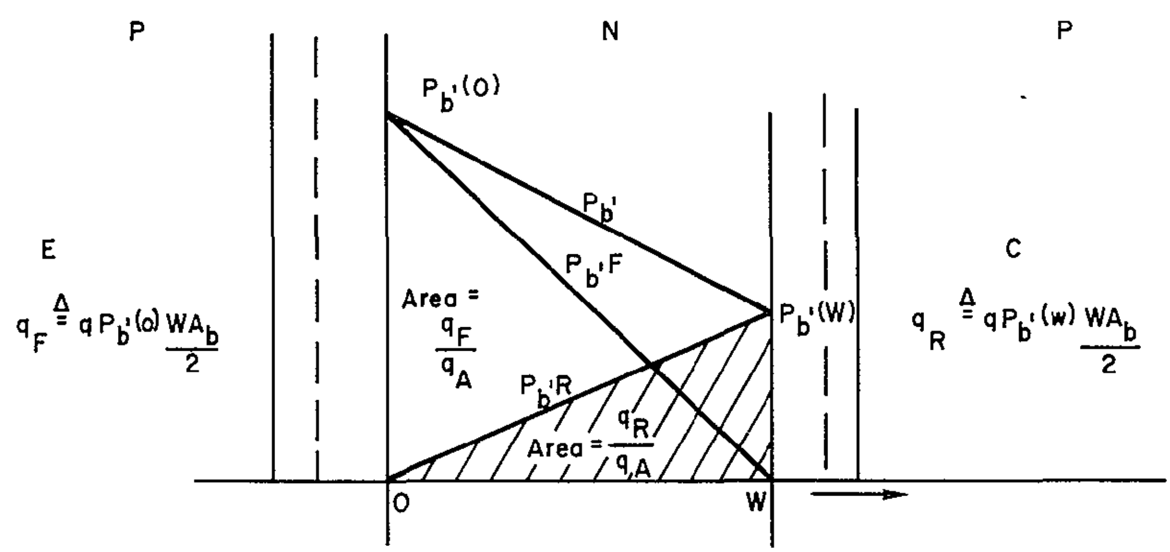
$$J2 = I_{CO}\left(e^{\theta_R V_{EB}} - 1\right)$$

(b) SCEPTRE EBERS – MOLL MODEL (PNP)
(TERMINAL CURRENT CONTROL)

Figure B 18 The Ebers–Moll First-Order Approximation to γ Effects



(a) MINORITY CARRIER CONCENTRATIONS IN THE BASE FOR THE FOUR REGIONS OF OPERATION



(b) MINORITY CARRIER CONCENTRATIONS IN THE SATURATION REGION

Figure B 19 Excess Minority Carrier Concentrations in the Base Region (PNP)

$$i_B = -\left(\underbrace{\frac{q_B}{\tau_B}}_{\text{maintaining}} + \underbrace{\frac{dq_B}{dt}}_{\text{charging}} \right) \tag{B-34}$$

The stored base charge is composed of forward and reverse components of stored minority carrier charge as shown in Figure 4

$$q_B = q_F + q_R \tag{B-35}$$

Therefore

$$i_B = -\left(\frac{q_F}{\tau_{BF}} + \frac{d\,q_F}{dt} \right) - \left(\frac{q_R}{\tau_{BR}} + \frac{dq_R}{dt} \right) \tag{B-36}$$

By assuming that the base charge is limited to relatively slow variations, i_c and i_e may be defined. For DC steady state, forward injection only·

$$I_c = -\frac{\beta_F q_F}{\tau_{BF}} = -\frac{q_F}{\tau_F} \tag{B-37}$$

For slow variations in q_B, $i_c \approx I_C$ and

$$i_e = -i_c - i_b = q_F \left(\frac{1}{\tau_{BF}} - \frac{1}{\tau_F} \right) + \frac{dq_F}{dt} \tag{B-38}$$

Similar equations can be written for reverse injection $\left(I_E = -\frac{q_R}{\tau_R} \right)$ and the total expression for terminal currents of an ideal device become

$$i_c = -\frac{q_F}{\tau_F} + q_R \left(\frac{1}{\tau_R} + \frac{1}{\tau_{BR}} \right) + \frac{dq_R}{dt} \tag{B-39}$$

$$i_e = q_F \left(\frac{1}{\tau_F} + \frac{1}{\tau_{BF}} \right) + \frac{dq_F}{dt} - \frac{q_R}{\tau_R} \tag{B-40}$$

$$i_B = i_c - i_e \tag{B-41}$$

(To include saturation, an increase in the base charge is represented by $\frac{q_{BS}}{\tau_{BS}}$ and $\frac{dq_{BS}}{dt}$ in parallel with $\frac{q_F}{\tau_{BS}}$ $\frac{dq_F}{dt}$)

Leakage currents i_{EO}, i_{CO} are included and space charge layer charge storage is represented by the terms $c_{b'e} \frac{dV_{eb}}{dt}$ and $C_{ob} \frac{dV_{cb}}{dt}$. In the complete model as shown in Figure B-20 CIRCUS incorporates the charge control model as its radiation model and represents the photocurrents by current generators placed in parallel with i_{EO}, i_{CO} exactly as represented in the Ebers-Moll model[8]. The CIRCUS model and its equations are shown in Figure B-21.

THE LUMPED MODEL

The third approach to describing device behavior is the Linvill approach of lumped modeling. The simplest model is the 2-lump model shown in Figure 7. The model is based on a description of excess minority carrier behavior similar to the Ebers-Moll approach. The model shown has parameter definitions that correspond to the charge-control model parameters. Terminal equations for the ideal device are

$$i_c = - H_D\ p_{b'}(o) + \left(H_D + H_{CR}\right) p_{b'}(w) + S_R \frac{dp_{b'}(w)}{dt} + C_{vc} \frac{dV_{CB}}{dt} - i_{EO}$$

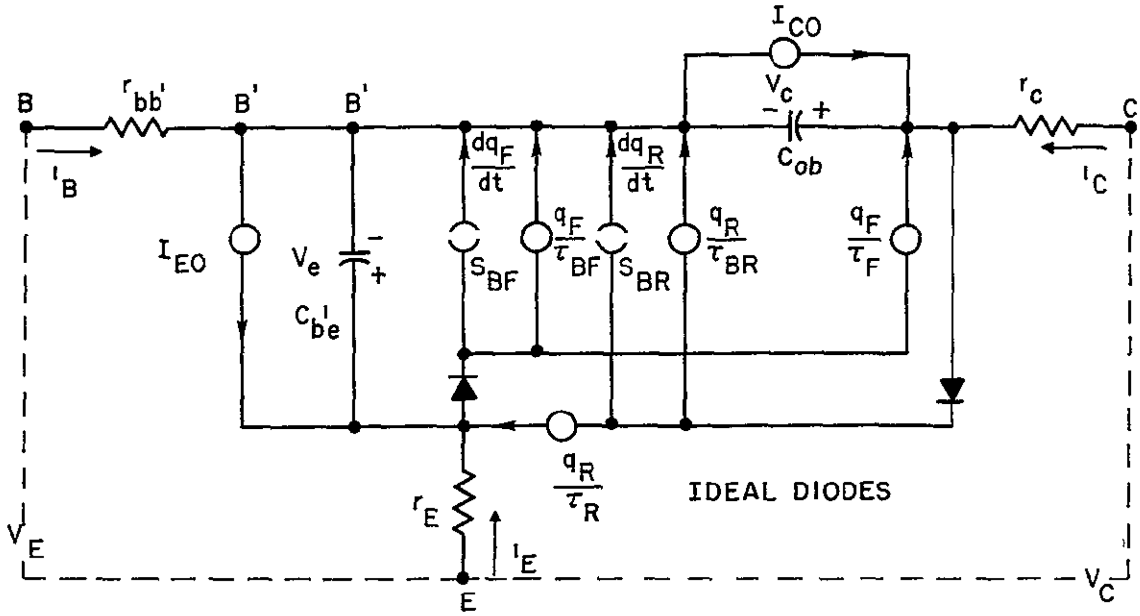
$$i_e = - H_D\ p_{b'}(w) + \left(H_D + H_{CF}\right) p_{b'}(o) + S_F \frac{dp_{b'}(o)}{dt} + C_{ve} \frac{dV_{EB}}{dt} - i_{CO}$$

$$i_b = - i_c - i_e$$

where the space charge layer charge storage is represented as well as the leakage currents. The radiation version replaces i_{EO}, i_{CO} with i_{ppe}, i_{ppc} as in the other models. TRAC employs the Linvill model for analysis and the TRAC representation is given in Figure B-22[9].

CONCLUSIONS

The three program models discussed are first-order approximations for representation of ionizing radiation effects. The three models use identical approaches to the representation of the junction photocurrents i_{ppe}, i_{ppc}



$$q_{BF} \triangleq Q_{BF} \left(e^{\frac{qV_e}{nKT}} - 1 \right)$$

$$q_{BR} \triangleq Q_{BR} \left(e^{\frac{qV_c}{nKT}} - 1 \right)$$

$$r_E = 0 .$$

$$V_E = i_B r_b + V_e$$

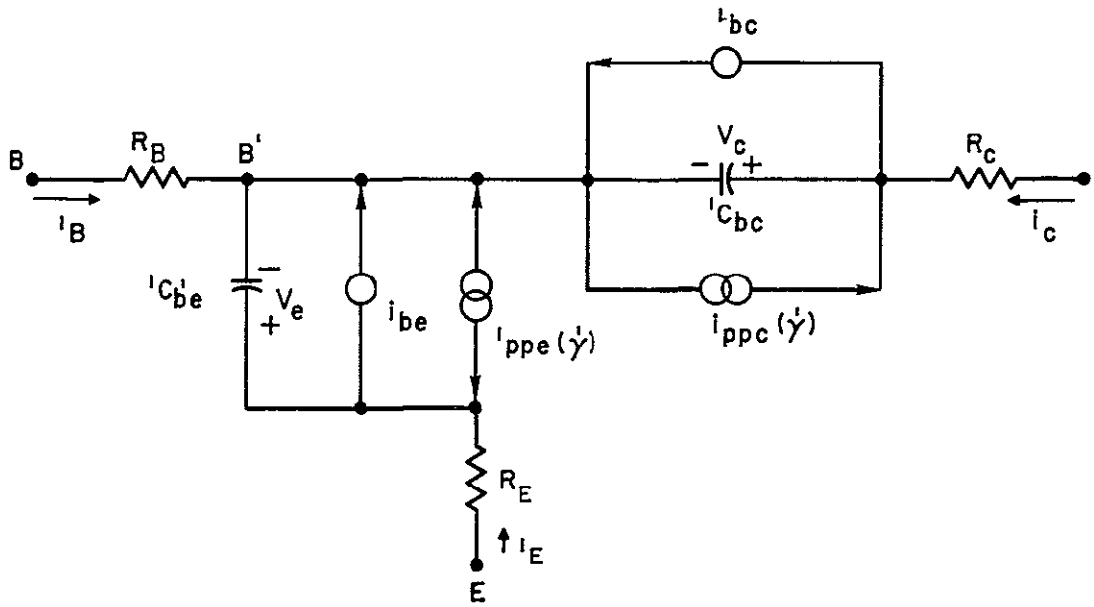
$$V_C = i_C r_c - V_c + V_e$$

$$i_C = \frac{-q_F}{\tau_F} + q_R \left(\frac{1}{\tau_R} + \frac{1}{\tau_{BR}} \right) + \frac{dq_R}{dt} + C_{ob} \frac{dV_c}{dt} - i_{CO}$$

$$i_E = q_F \left(\frac{1}{\tau_{BF}} + \frac{1}{\tau_F} \right) + \frac{dq_F}{dt} - \frac{q_R}{\tau_R} + C_{b'e} \frac{dV_e}{dt} - i_{EO}$$

$$i_B = - i_C - i_E$$

Figure B 20 PNP Charge-Control Model for Normal and Inverse Operation



$$C_{be} \triangleq \frac{A_1}{\left(\phi_1 - V_e\right)^{n_1}} + \theta_N T_{CN}(I_N + I_{es})$$

$$C_{bc} \triangleq \frac{A_2}{\left(\phi_2 - V_c\right)^{n_2}} + \theta_R T_{CR}(I_R + I_{cs})$$

$$I_N = I_{es}\left(e^{\theta_N V_e} - 1\right)$$

$$I_R = I_{cs}\left(e^{\theta_R V_c} - 1\right)$$

$$I_{be} \triangleq \left(\frac{1}{\beta_N} + 1\right) I_N - I_R$$

$$I_{bc} \triangleq -I_N - \left(\frac{1}{\beta_R} + 1\right) I_R$$

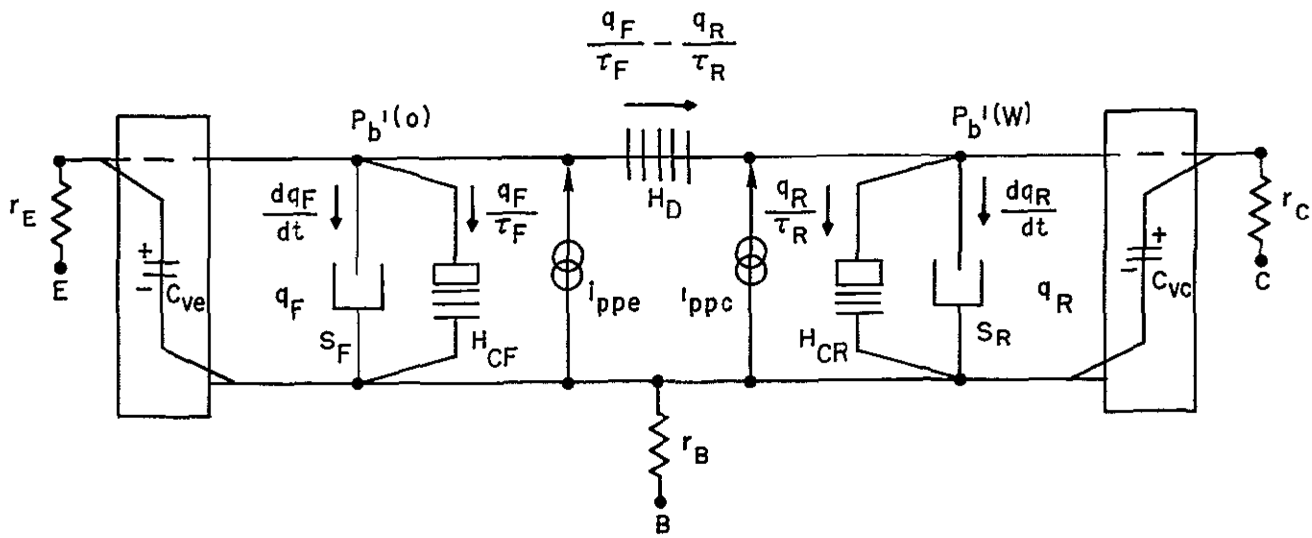
$\beta_N = F(I_N)$ Single Valued Parameters

$\beta_R = F(I_R)$ $R_b, R_E, R_c, A_1, A_2, n_1, n_2$

$T_{CN} = F(I_N)$ $\phi_1, \phi_2, I_{es}, I_{cs}, \theta_N, \theta_R$

$T_{CR} = F(I_R)$

Figure B 21 CIRCUS Model (Charge Control) and Parameters — PNP Device



Terminal Currents

$$i_C(\dot{\gamma}) = -H_D P_b'(o) + \left(H_D + H_{CR}\right) P_b'(w) + S_R \frac{dP_b'(w)}{dt} + C_{VC}\frac{dV_c}{dt} - i_{ppe}$$

$$i_E(\dot{\gamma}) = -H_D P_b'(w) + \left(H_D + H_{CF}\right) P_b'(o) + S_F \frac{dP_b'(o)}{dt} + C_{VE}\frac{dV_e}{dt} - i_{ppc}$$

$$i_B(\gamma) = -i_C(\gamma) - i_E(\gamma)$$

$$q_F = S_F P_b'(o)$$

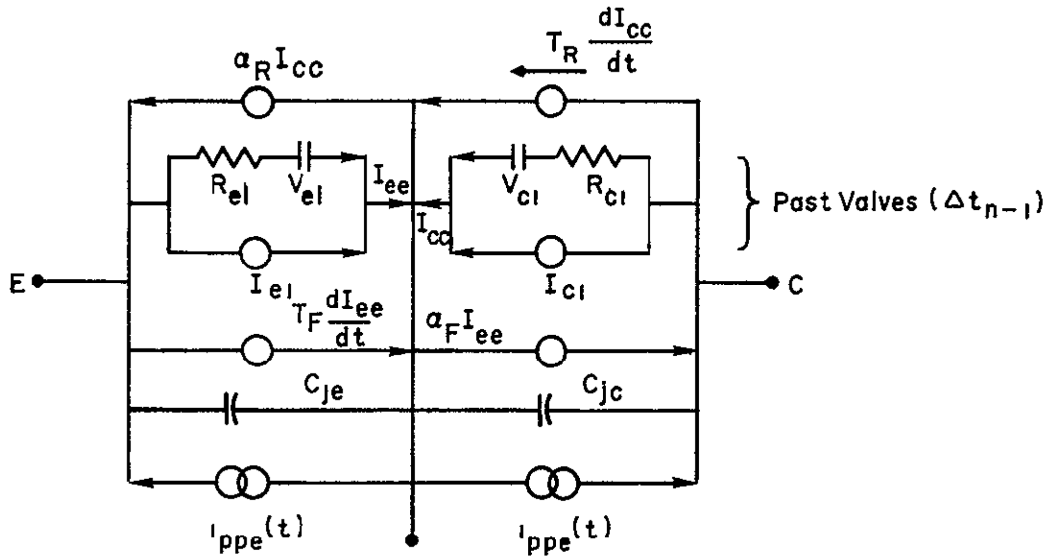
$$q_R = S_R P_b'(w)$$

$$P_b'(o) = P_{bo}\left(e^{qV_e/nKT} - 1\right)$$

$$P_b'(w) = P_{bo}\left(e^{qV_c/nKT} - 1\right)$$

For analysis without i_{ppe}, i_{ppc} delete the current generators

Figure B 22 Linvill Two-Lump Model — PNP Including Radiation Pulse Currents



DEFINE:

$$\alpha_E = \frac{H_D}{H_D + H_{CF}} \qquad \alpha_R = \frac{H_D}{H_D + H_{CR}}$$

$$I_{EO} = -P_{bo}\left(H_{CF} + \alpha_R H_{CR}\right) \qquad I_{CO} = -P_{bo}\left(H_{CR} + \alpha_F H_{CF}\right)$$

$$T_F = \frac{S_E}{H_{CF} + H_D} \qquad T_R = \frac{S_B}{H_{CR} + H_D}$$

THEN:

$$i_E = \frac{I_{EO}}{1-\alpha_F\alpha_R}\left(e^{\theta_e V_E} - 1\right) - T_F \frac{d}{dt}\left[\frac{I_{EO}}{1-\alpha_F\alpha_R}\left(e^{\theta_e V_E} - 1\right)\right] + \frac{\alpha_R I_{CO}}{1-\alpha_F\alpha_R}\left(e^{\theta_c V_C} - 1\right) + C_{je}\frac{dV_E}{dt} - i_{ppe}(t)$$

$$i_C = \frac{-I_{CO}}{1-\alpha_F\alpha_R}\left(e^{\theta_c V_C} - 1\right) - T_R \frac{d}{dt}\left[\frac{I_{CO}}{1-\alpha_F\alpha_B}\left(e^{\theta_c V_C} - 1\right)\right] + \frac{\alpha_F I_{EO}}{1-\alpha_F\alpha_R}\left(e^{\theta_e V_E} - 1\right) + C_{jc}\frac{dV_C}{dt} - i_{ppc}(t)$$

$$i_B = -i_E - i_C$$

Figure B 23 TRAC Representation of the Linvill Two-Lump Model — PNP — Including Radiation Photocurrents (prestored)

The variations of Beta with time, injection level and radiation exposure for a given device is complex. A general approach has evolved because most programs require the Beta be either single valued or expressed as a function of current in a simple table Beta is first computed for the injection level and other 'normal' operating conditions. A degraded Beta for the given flux level is computed from Equation (3) and then degraded by a factor of 2 or 3 to provide a worst-case analysis (The actual transient loss may be this great or greater with a recovery of Beta to its damage value β_F dependent upon the device and its exposure) This approximate analysis is sufficient to indicate the presence of difficulties in the circuit under analysis.

SCEPTRE allows the programming of Beta as a function (equation or tabled expression) and CIRCUS allows a table (Beta N = $F(I_n)$) The more elaborate expressions for Beta are justified when sufficient laboratory data exists to warrant the increased computer computation time.

TABLE OF SYMBOLS

A	junction area (A_{jc}, A_{je}, A_{base})
C_{jc}, C_{je}	incremental capacitance of the junction
C_{VC}, C_{VE}	nonlinear charge store of the junction
D, D_n, D_{pb}	charge-carrier diffusion constant (e, p for electrons in p-material, e, b, c for a transistor to denote emitter, base or collector)
E	electric field
g_o	carrier generation rate at equilibirum
H_{CF}, H_{CR}	forward and reverse combination parameter of the Linvill model
H_D	the diffusance parameter of the Linvill model
I_{co}, I_{EO}	the collector and emitter junction saturation current (opposite junction open)
I_{CS}, I_{ES}	the collector and emitter junction saturation current (opposite junction shorted)
I_S	Saturation current of an ideal diode
J	electric current density
K	damage constant
K, k	Boltzmann's Constant
L	Charge carrier diffusion length (see D)
N_a (N_d)	accepter (donor impurity concentration
$\eta(p)$	electron (hole) concentration
η' (p')	excess electron (hole) concentration
η_i	intrinsic carrier concentration
q	magnitude of the electronic charge
q_B	total excess minority carrier charge stored in the case $q_B = q_F + q_R$ (forward and reverse components)
q_{VE}, q_{VC}	junction space charge layer charge (at $V_g = 0$)
r_B, r_E, r_C	large signal bulk resistances
T	absolute temperature
W	Width of the base region
x, x', x''	longitudinal coordinate in one dimensional transistor model

$\alpha_F(\alpha_R)$	large signal forward (reverse) injection common base short circuit gain
$\beta_F(\beta_R)$	large signal forward (reverse) injection common emitter short circuit current gain
$\gamma(t)$	ionizing radiation
	neutron flux
ϵ	dielectric permittivity
η	base width modulation factor
μ	charge carrier mobility
ρ	space charge concentration
τ	lifetime of excess charge carriers
θ	$\theta = q/\eta KT$

BIBLIOGRAPHY

1. Samuel Glasstone, ed , The Effects of Nuclear Weapons, Washington, D. C. · United States Atomic Energy Commission, 1962

2. Samuel Glasstone, Alexander Sesonske, Nuclear Reactor Engineering, New Jersey D. Van Nostrand Company, Inc , 1967.

3. James P Raymond, Radiation Effects on Semiconductors, The University of California at Los Angeles Extension Course Notes 9-20, September 1968.

4. Frank Larin, Radiation Effects in Semiconductor Devices, New York John Wiley and Sons, Inc., 1968

5. Paul E. Gray, et al., Physical Electronics and Circuit Models of Transistors, Semiconductor Electronics Education Committee, Vol 2, New York John Wiley and Sons, Inc., 1964.

6. Campbell L. Searle, et al., Elementary Circuit Properties of Transistors, Semiconductor Electronics Education Committee, Vol. 3, New York: John Wiley and Sons, Inc , 1964.

7 Harry W. Mathers, et al., Automated Digital Computer Program for Determining Responses of Electronic Circuits to Transient Nuclear Radiation (SCEPTRE), Vol 1 and Vol. 2, New York. IBM, Feb. 1967.

8. L. P Milliman, W A. Massena, R. H. Dickhaut, CIRCUS A Digital Computer Program for Transient Analysis of Electronic Circuits User's Guide, Washington The Boeing Co , Jan. 1967

9. G T. Kleiner, G. Henoshita, and E D Johnson, "Simulation and Verification of Transient Nuclear Radiation Effects on Semiconductor Electronic" IEEE Trans , Vol NS-11, (1964), pp. 82-104.

APPENDIX C
CURRENT CODING EXAMPLES

Example 1: (Taken from the report, "The Application of NASAP to the Design of Biomedical Instrumentation Circuits," written by M. L. Moe and J. T. Schwartz, Denver Research Institute, University of Denver, January 1970).

Given: The Equivalent Circuit Diagram of a Telemetry Transmitter

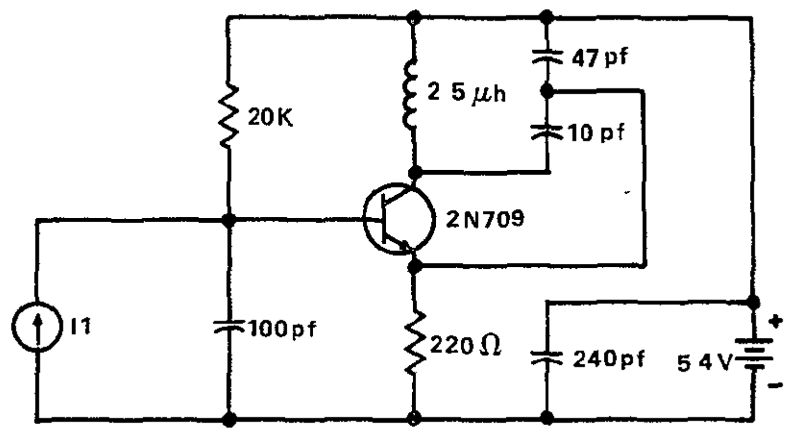


Figure C 1 Telemetry Transmitter Circuit

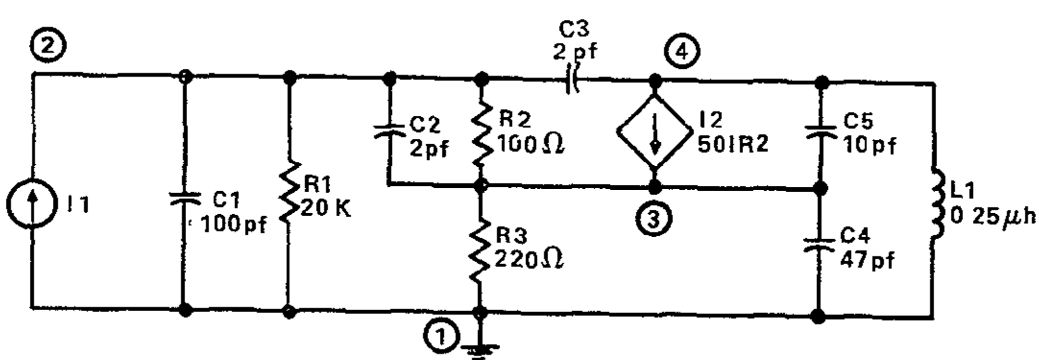


Figure C 2 Equivalent Circuit of Telemetry Transmitter for NASAP Analysis

Input Codes;

NASAP Input Code

```
NASAP PROBLEM TELEMETER
I1 1 2 1.0
C1 1 2 100UF
R1 1 2 20K
C2 2 3 2UF
R2 2 3 100
R3 1 3 220
L1 4 1 0.25
C3 2 4 2.0UF
C4 3 1 47UF
C5 3 4 10UF
12 4 3 0.3 IR2
OUTPUT
WORST CASE
I11/II1
PLOT(TY=FR/FR=80/TO=126)
PLOT(TY=RE/FR=0/TO=100)
ROOTS, POLES
END
PLOT(TY=SE/FR=80/TO=126/EL=DJ1)
PLOT(TY=SE/FR=80/TO=126/EL=CJ3)
PLOT(TY=SE/FR=80/TO=126/EL=LE3)
PLOT(TY=SE/FR=80/TO=126/EL=KE3)
EXECUTE
TREE NASAP PROBLEM TELEMETER
J1(1-2)=1.0
CJ1(1-2)=100MF
RE1(1-2)=20K
CJ2(2-3)=2MF
RJ2(2-3)=100
RE3(1-3)=220
LE1(4-1)=0.25
CJ3(1-4)=LOMF
CE4(3-1)=47MF
CJ5(3-4)=10MF
DJ2/IRJ2(4-3)=0.3
END
WORST CASE
I11/II1
PLOT(TY=FR/FR=80)/TO=126)
PLOT(TY=RE/FR=0/TO=100)
ROOTS, POLES
END
```

```
PLOT(TY=SE/FR=80/TO=126/EL=DJ3)
PLOT(TY=SE/FR=80/TO=126/EL=CJ3)
PLOT(TY=SE/FR=80/TO=126/EL=LE3)
PLOT(TY=SE/FR=80/TO=126/EL=KE3)
CIRC(1E6) NASAP PROBLEM TELEMETER
J1(1-2)=1.0
C1(1-2)=100MF
R1(1-=)=20K
C2(2-3)=2MF
R2(2-3)=100
R3(1-3)=220
L1(4-1)=0.25
C3(2-4)=2.0MF
C4(3-1)=47MF
C5(3-4)=10MF
DJ2/IR2(4-3)=0 3
END
WORST CASE
IL1/II1
PLOT(TY=FR/FR=80/TO=126)
PLOT(TY=RE/FR=0/TO=100)
ROOTS, POLES
END
PLOT(TY=SE/FR=80/TO=126/EL=CJ1)
PLOT(TY=SE/FR=80/TO=126/EL=EJ3)
PLOT(TY=SE/FR=80/TO=126/EL=LE1)
PLOT(TY=SE/FR=80/TO=126/EL=KE3)
EXECUTE
STOP
```

```
****NASAP****

   NETWORK ANALYSIS AND SYSTEMS APPLICATICN PRCCRAM

THIS VERSICN WAS DEVELOPEC AT UCLA ENGR. CEPT.


NASAP PRCBLEM TELEMETER
I1 1 2 1.C
C1 1 2 10CUF
R1 1 2 20K
C2 2 3 2UF
R2 2 3 100
R3 1 3 220
L1 4 1 0.25
C3 2 4 2.CUF
C4 3 1 47UF
C5 3 4 10UF
I2 4 3 C.3 IR2
OUTPUT
```

ELEMENT NUMBER	ELEMENT NAME	CEPENDENCY (IF ANY)	CRIGIN NODE	TARGET NODE	VALUE	TAG	GENER
1	I1		1	2	1.00000000E 00	0	1
2	C1		1	2	9.99999320E-05	1	0
3	R1		1	2	2.00000000E 04	0	0
4	C2		2	3	1.99999886E-06	0	0
5	R2		2	3	1.00000000E 02	0	0
6	R3		1	3	2.20000000E 02	1	0
7	L1		4	1	2.49999881E-01	1	0
8	C3		2	4	1.99999886E-C6	0	0
9	C4		3	1	4.69999650E-05	0	0
10	C5		3	4	9.99999429E-06	0	0
11	I2	IR2	4	3	2.99999893E-01	0	1

2	2	304	10
3	6	1536	2096
4	-7	2C48	1280

WORSTCASE

ELEMENT NAME	TOLERANCE
C1	0.1000
R1	0.1000
C2	0.1000
R2	0.1000
R3	0.1000
L1	0.1000
C3	0.1000
C4	0.1000
C5	0.1000
I2	0.1000

FLOWGRAPH

FROM	TO	S	VALUE
2	13	-1	1.00000039E 04
14	3	0	4.99999878E-05
15	4	1	1.99999886E-06
16	5	0	9.99999791E-03
6	17	0	2.20000000E 02
7	18	1	2.49999881E-01
19	8	1	1.99999886E-06
20	9	1	4.69999650E-05
21	10	1	9.99999429E-06
5	11	0	2.99999893E-01
4	2	0	1.00000000E 00
13	15	0	-1.00000000E 00
5	2	0	1.00000000E 00
13	16	0	-1.00000000E 00
8	2	0	1.00000000E 00
13	19	0	-1.00000000E 00
1	2	0	-1.00000000E 00
13	12	0	-1.00000000E 00
3	2	0	-1.00000000E 00
13	14	0	1.00000000E 00
9	6	0	1.00000000E 00
17	20	0	-1.00000000E 00
10	6	0	1.00000000E 00
17	21	0	-1.00000000E 00
4	6	0	-1.00000000E 00
17	15	0	1.00000000E 00
5	6	0	-1.00000000E 00
17	16	0	1.00000000E 00
11	6	0	-1.00000000E 00
17	22	0	-1.00000000E 00
8	7	0	1.00000000E 00
18	19	0	-1.00000000E 00
10	7	0	1.00000000E 00
18	21	0	-1.00000000E 00
11	7	0	-1.00000000E 00
18	22	0	-1.00000000E 00

IL1/III1

NTIMES=   1

THE UNKNOWN TRANSMITTANCE GOES FROM NODE 7 TO 1

24588	5999
1180224	288001
3057108	746003
3089892	754002
786816	192002
3057108	746003
3089892	754002
3091940	1778002
2360448	576002
2600180	1658002
862628	1234001
2559200	1648002
1026548	1274002
239732	1082000
2305254	1587000
75942	1042999
239862	1083000
198752	1072000
40980	10000
237684	58000
237684	58000
2270422	555001
73764	17999
2303206	563000
532740	130000
532870	131000
2229312	544001
163920	40001
196704	48000

```
NO. OF FIRST ORDER LOOPS= 29
                            1
                            2
                            3
                            4
                            5
                            6
                            7
                            8
                            9
                           10
                           11
                           12
                           13
                           14
                           15
                           16
                           17
                           18
                           19
                           20
                           21
                           22
                           23
                           24
                           25
                           26
                           27
                           28
                           29
```

C-8

TRANSFER FUNCTION

```
                                        2
 9.881E 04   + 4.282E 01S  - 1.000E 00S  + 0.0          + 0.0

------------------------------------------------------------------

                                        2            3            4
 2.948E 09   + 1.509E 08S  + 3.[illegible]94E 05S  + 3.850E 02S  + 1.000E 00S
```

THE FUNCTION FACTOR =   -8.780E 03

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.01930E 02
5.21768E 00	1.38103E-02	1.23138E-05	3.45773E-08	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3.00000E 01	1.29997E-02	-3.03599E-04	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

SENSITIVITY TO C1

$$\frac{0.0 + 1.154E\,08S + 3.882E\,05S^{2} + 2.969E\,02S^{3} + 1.000E\,00S^{4}}{2.948E\,09 + 1.509E\,08S + 3.994E\,05S^{2} + 3.850E\,02S^{3} + 1.000E\,00S^{4}}$$

THE FUNCTION FACTOR = -9.671E-01

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.01930E 02
5.21768E 00	1.38103E-02	1.33138E-05	3.45773E-08	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	-3.86000E 00	-1.29800E-02	-9.92998E-06	-3.34399E-08	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

SENSITIVITY TO R1

$$- 1.140E\ 13 \quad - 4.329E\ 10S \quad + 6.[illegible]53E\ 07S^{2} + 2.765E\ 05S^{3} + 2.533E\ 02S^{4} + 1.000E\ 00S^{5} + 0.0$$

$$- 2.913E\ 14 \quad - 1.504E\ 13S \quad - 4.298E\ 10S^{2} + 9.575E\ 07S^{3} + 2.841E\ 05S^{4} + 3.422E\ 02S^{5} + 1.000E\ 00S^{6}$$

THE FUNCTION FACTOR = 4.836E-01

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.05790E 03	1.57855E 02
4.51192E-01	-1.00514E-03	-2.98241E-06	-3.59256E-09	-1.04976E-11	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.78994E 01
2.19757E-01	-3.52979E-04	-1.40397E-06	-1.28603E-09	-5.07683E-12	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

SENSITIVITY TC C2

$$\frac{0.0 + 4.905E\ 11 S + 2.154E\ 10 S^{2} + 2.266E\ 08 S^{3} + 4.656E\ 05 S^{4} + 5.740E\ 02 S^{5} + 1.000E\ 00 S^{6}}{- 2.913E\ 14 - 1.504E\ 13 S - 4.298E\ 10 S^{2} + 9.575E\ 07 S^{3} + 2.841E\ 05 S^{4} + 3.422E\ 02 S^{5} + 1.000E\ 00 S^{6}}$$

THE FUNCTION FACTOR = -1.178E-01

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.05790E 03	1.57855E 02
4.51192E-01	-1.00514E-03	-2.98241E-06	-3.59256E-09	-1.04976E-11	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6.06506E-01	2.66325E-02	2.80268E-04	5.75787E-07	7.09774E-10	1.23657E-12	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

SENSITIVITY TO R2

$$\frac{3.543E\ 09 + 7.898E\ 09S + 1.307E\ 08S^{2} + 4.279E\ 05S^{3} + 3.403E\ 02S^{4} + 1.000E\ 00S^{5} + 0.0}{-\ 2.913E\ 14 - 1.504E\ 13S - 4.298E\ 10S^{2} + 9.575E\ 07S^{3} + 2.841E\ 05S^{4} + 3.422E\ 02S^{5} + 1.000E\ 00S^{6}}$$

THE FUNCTION FACTOR = 4.033E 02

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.05790E 03	1.57855E 02
4.51192E-01	-1.00514E-03	-2.98241E-06	-3.59256E-09	-1.04976E-11	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.50000E 01
-3.34397E 01	-5.53508E-01	-1.81177E-03	-1.44092E-06	-4.23382E-09	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

SENSITIVITY TO R3

1.698E 11 + 4.828E 11S + 1.257E 09S^{2} + 1.352E 06S^{3} + 3.272E 03S^{4} + 1.000E 00S^{5} + 0.0

− 2.913E 14 − 1.504E 13S − 4.298E 10S^{2} + 9.575E 07S^{3} + 2.841E 05S^{4} + 3.422E 02S^{5} + 1.000E 00S^{6}

THE FUNCTION FACTOR = −2.407E 01

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.05790E 03	1.57855E 02
4.51192E-01	-1.00514E-03	-2.98241E-06	-3.59256E-09	-1.04976E-11	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.28994E 01
1.21985E 02	3.17453E-01	3.41535E-04	6.26703E-07	2.52646E-10	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

SENSITIVITY TO L1

$$\frac{0.0 + 0.0 - 9.143E\ 08S^{2} - 3.844E\ 07S^{3} - 1.060E\ 05S^{4} + 3.422E\ 02S^{5} + 1.000E\ 00S^{6}}{-2.913E\ 14 - 1.504E\ 13S - 4.298E\ 10S^{2} + 9.575E\ 07S^{3} + 2.841E\ 05S^{4} + 3.422E\ 02S^{5} + 1.000E\ 00S^{6}}$$

THE FUNCTION FACTOR = 1.000E 00

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.05790E 03	1.57855E 02
4.51192E-01	-1.00514E-03	-2.98241E-06	-3.59256E-09	-1.04976E-11	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	9.59837E-03	4.03572E-04	1.11326E-06	-3.59256E-09	-1.04976E-11	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

SENSITIVITY TO C3

$$\frac{0.0 + 1.494E\,12\,S + 6.454E\,10\,S^{2} + 3.640E\,08\,S^{3} + 6.985E\,05\,S^{4} + 7.934E\,02\,S^{5} + 1.000E\,00\,S^{6}}{-\,2.913E\,14 - 1.504E\,13\,S - 4.298E\,10\,S^{2} + 9.575E\,07\,S^{3} + 2.841E\,05\,S^{4} + 3.422E\,02\,S^{5} + 1.000E\,00\,S^{6}}$$

THE FUNCTION FACTOR = -6.494E-01

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.05790E 03	1.57855E 02
4.51192E-01	-1.00514E-03	-2.98241E-06	-3.59256E-09	-1.04976E-11	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.01850E 01	4.40036E-01	2.48142E-03	4.76209E-06	5.40885E-09	6.81754E-12	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

SENSITIVITY TO C4

$$\frac{0.0 + 1.698E\,11S + 4.828E\,11S^{2} + 1.257E\,09S^{3} + 1.352E\,06S^{4} + 3.272E\,03S^{5} + 1.000E\,00S^{6}}{-\,2.913E\,14 - 1.504E\,13S - 4.298E\,10S^{2} + 9.575E\,07S^{3} + 2.841E\,05S^{4} + 3.422E\,02S^{5} + 1.000E\,00S^{6}}$$

THE FUNCTION FACTOR =   2.489E-01

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.05790E 03	1.57855E 02
4.51192E-01	-1.00514E-03	-2.98241E-06	-3.59256E-09	-1.04976E-11	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-4.43573E-01	-1.26133E 00	-3.28245E-03	-3.53147E-06	-8.54810E-09	-2.61237E-12	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

SENSITIVITY TO C5

$$\frac{0.0 - 5.021\text{E }12S - 2.136\text{E }11S^{2} - 6.535\text{E }08S^{3} - 7.576\text{E }05S^{4} - 1.021\text{E }03S^{5} + 1.000\text{E }00S^{6}}{- 2.913\text{E }14 - 1.504\text{E }13S - 4.298\text{E }10S^{2} + 9.575\text{E }07S^{3} + 2.841\text{E }05S^{4} + 3.422\text{E }02S^{5} + 1.000\text{E }00S^{6}}$$

THE FUNCTION FACTOR = 5.513E-01

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.05790E 03	1.57855E 02
4.51192E-01	-1.00514E-03	-2.98241E-06	-3.59256E-09	-1.04976E-11	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2.90575E 01	1.23599E 00	3.78431E-03	4.38419E-06	5.90848E-09	-5.78707E-12	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

SENSITIVITY TO I2

$$\frac{2.926E\,11 + 1.599E\,10\,S + 1.877E\,08\,S^{2} + 4.503E\,05\,S^{3} + 4.790E\,02\,S^{4} + 1.000E\,00\,S^{5} + 0.0}{-\,2.913E\,14 - 1.504E\,13\,S - 4.298E\,10\,S^{2} + 9.575E\,07\,S^{3} + 2.841E\,05\,S^{4} + 3.422E\,02\,S^{5} + 1.000E\,00\,S^{6}}$$

THE FUNCTION FACTOR = 9.923E 02

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.05790E 03	1.57855E 02
4.51192E-01	-1.00514E-03	-2.98241E-06	-3.59256E-09	-1.04976E-11	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-3.04800E 03
-1.66603E 02	-1.95485E 00	-4.69070E-03	-4.98980E-06	-1.04167E-08	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

SQUARE OF WORST CASE TOLERANCE

$$3.082E\ 28 + 3.406E\ 27S + 2.323E\ 26S^{2} + 3.244E\ 24S^{3} + 3.197E\ 22S^{4} + 1.376E\ 20S^{5} + 3.456E\ 17S^{6} + 7.276E\ 14S^{7}$$
$$+ 1.070E\ 12S^{8} + 9.990E\ 08S^{9} + 1.166E\ 06S^{10} + 5.955E\ 02S^{11} + 1.000E\ 00S^{12}$$

$$8.485E\ 28 + 8.760E\ 27S + 2.512E\ 26S^{2} + 1.237E\ 24S^{3} - 1.198E\ 21S^{4} - 1.697E\ 19S^{5} - 2.613E\ 16S^{6} - 5.087E\ 12S^{7}$$
$$+ 6.029E\ 10S^{8} + 3.860E\ 08S^{9} + 6.853E\ 05S^{10} + 6.845E\ 02S^{11} + 1.000E\ 00S^{12}$$

THE FUNCTION FACTOR = 2.737E-02

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	9.35074E 06	9.65411E 05	2.76777E 04	1.36299E 02
-1.31999E-01	-1.87057E-03	-2.87938E-06	-5.60605E-10	6.64391E-12	4.25322E-14	7.55230E-17	7.54268E-20	1.10201E-22	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	9.29569E 04	1.02711E 04	7.00740E 02
9.78263E 00	9.64322E-02	4.15129E-04	1.04222E-06	2.19438E-09	3.22763E-12	3.01305E-15	3.51513E-18	1.79595E-21	3.01593E-24
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

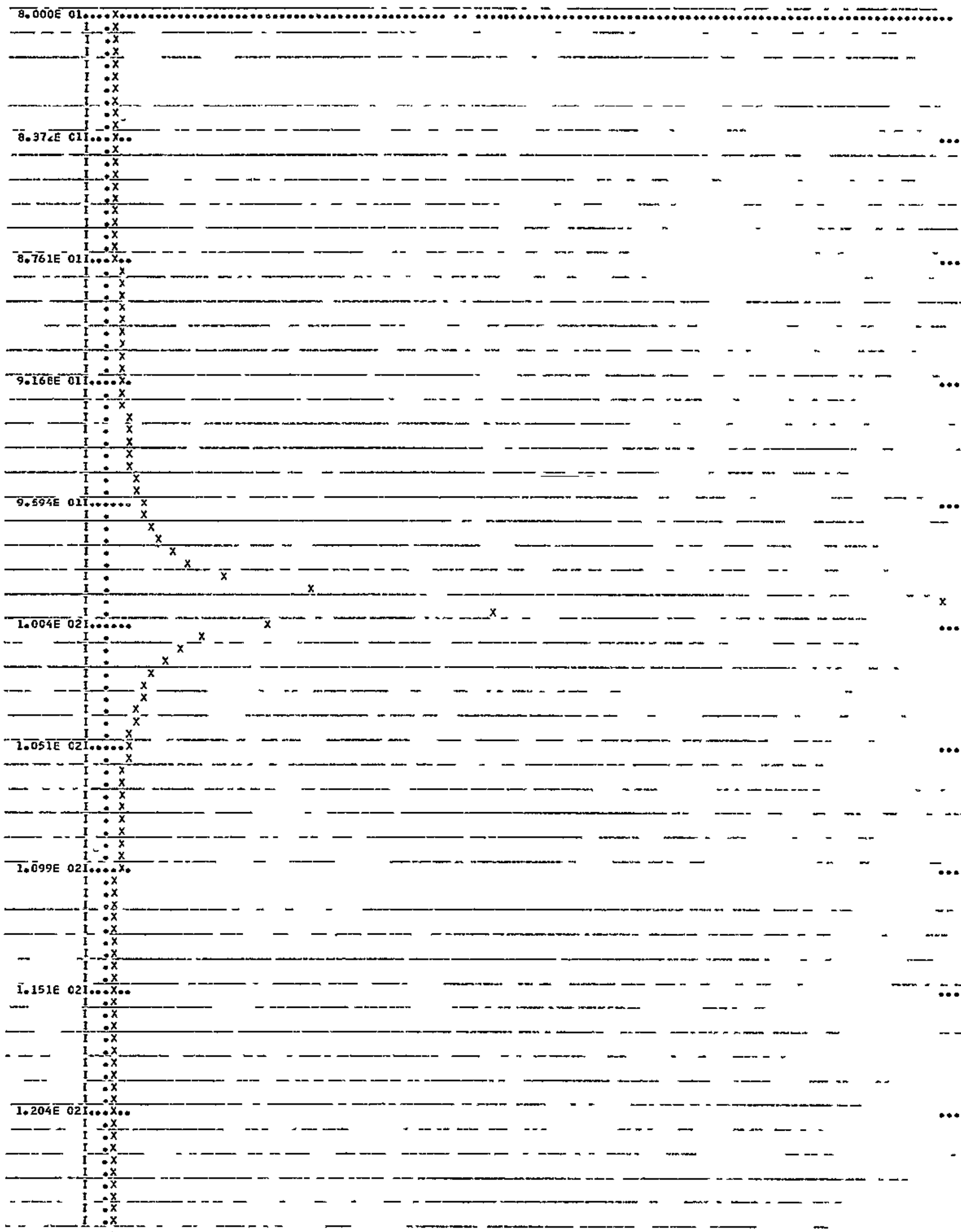
PLOT(TY=FR/FR=80/TO=126)

THE TRANSFORM OF THE RESPONSE IS ---

EXP. OF S	NUMER. COEFFS.	DENOM. COEFFS.
0	-3.00000E 01	1.01930E 02
1	-1.29997E-02	5.21768E 00
2	3.03599E-04	1.38103E-02
3	0.0	1.33138E-05
4	0.0	3.45773E-08

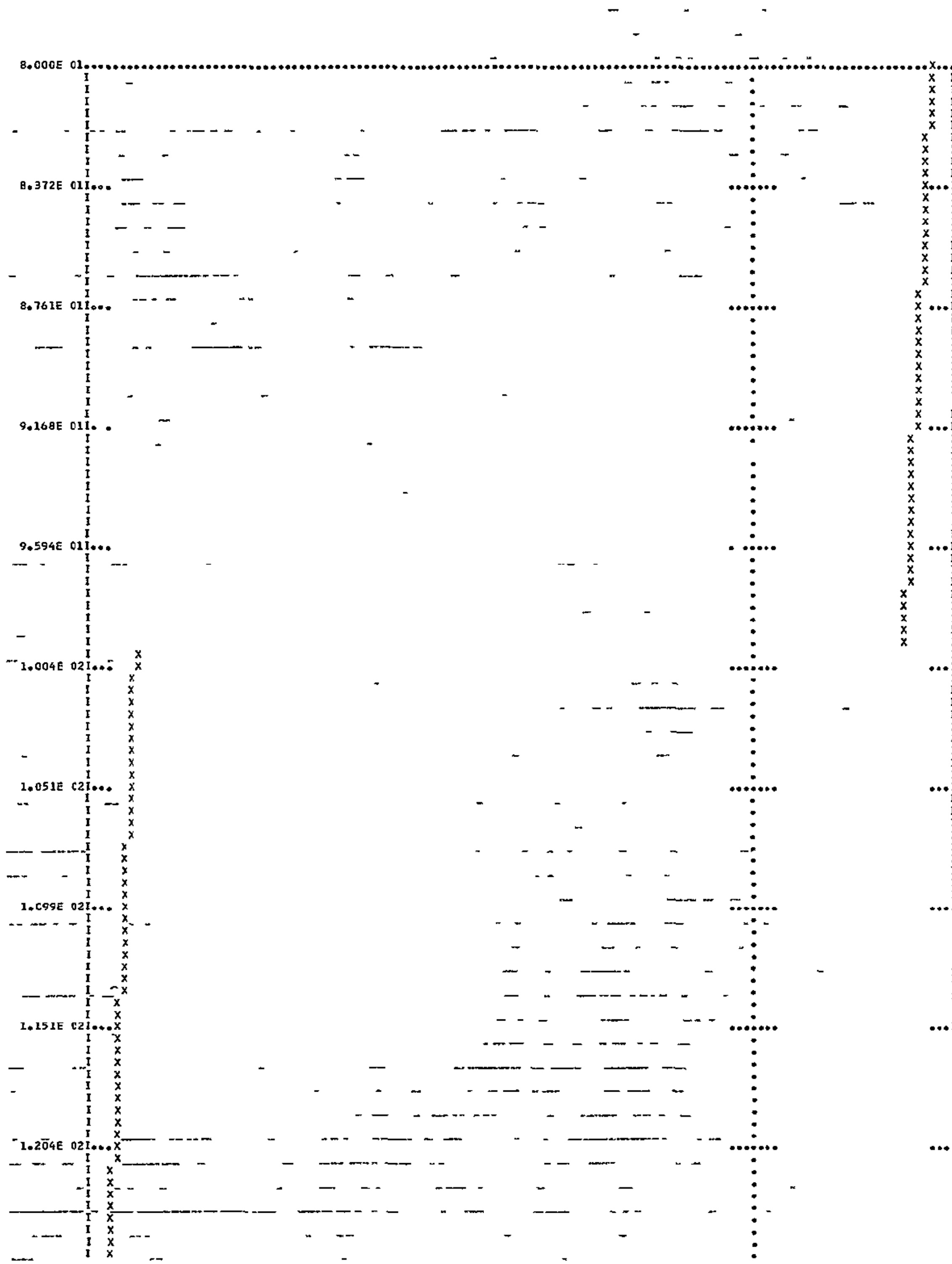
FREQ	MAG	FREQ	MAG	FREQ	MAG	FREQ	MAG
8.000E 01	7.11011E-02	8.962E 01	1.29375E-01	1.004E 02	1.57465E 00	1.125E 02	8.70447E-02
8.036E 01	7.22272E-02	9.003E 01	1.34472E-01	1.009E 02	9.82326E-01	1.130E 02	8.34591E-02
8.073E 01	7.34040E-02	9.044E 01	1.40050E-01	1.013E 02	7.11794E-01	1.135E 02	8.01249E-02
8.110E 01	7.46347E-02	9.085E 01	1.46179E-01	1.018E 02	5.56839E-01	1.140E 02	7.70179E-02
8.147E 01	7.59231E-02	9.126E 01	1.52949E-01	1.022E 02	4.56448E-01	1.145E 02	7.41151E-02
8.184E 01	7.72731E-02	9.168E 01	1.60461E-01	1.027E 02	3.86102E-01	1.151E 02	7.13971E-02
8.221E 01	7.86890E-02	9.210E 01	1.68845E-01	1.032E 02	3.34095E-01	1.156E 02	6.88478E-02
8.258E 01	8.01756E-02	9.251E 01	1.78261E-01	1.036E 02	2.94064E-01	1.161E 02	6.64507E-02
8.296E 01	8.17384E-02	9.294E 01	1.88912E-01	1.041E 02	2.62310E-01	1.166E 02	6.41941E-02
8.334E 01	8.33828E-02	9.336E 01	2.01055E-01	1.046E 02	2.36511E-01	1.172E 02	6.20653E-02
8.372E 01	8.51154E-02	9.378E 01	2.15030E-01	1.051E 02	2.15134E-01	1.177E 02	6.00545E-02
8.410E 01	8.69432E-02	9.421E 01	2.31279E-01	1.055E 02	1.97132E-01	1.182E 02	5.81514E-02
8.448E 01	8.88741E-02	9.464E 01	2.50402E-01	1.060E 02	1.81769E-01	1.188E 02	5.63485E-02
8.487E 01	9.09169E-02	9.507E 01	2.73244E-01	1.065E 02	1.68502E-01	1.193E 02	5.46378E-02
8.525E 01	9.30808E-02	9.550E 01	3.00994E-01	1.070E 02	1.56932E-01	1.199E 02	5.30126E-02
8.564E 01	9.53774E-02	9.594E 01	3.35431E-01	1.075E 02	1.46751E-01	1.204E 02	5.14664E-02
8.603E 01	9.78190E-02	9.637E 01	3.79281E-01	1.080E 02	1.37725E-01	1.209E 02	4.99940E-02
8.642E 01	1.00419E-01	9.681E 01	4.37028E-01	1.085E 02	1.29669E-01	1.215E 02	4.85906E-02
8.682E 01	1.03193E-01	9.725E 01	5.16501E-01	1.089E 02	1.22436E-01	1.220E 02	4.72508E-02
8.721E 01	1.06160E-01	9.770E 01	6.32803E-01	1.094E 02	1.15905E-01	1.226E 02	4.59712E-02
8.761E 01	1.09339E-01	9.814E 01	8.19243E-01	1.099E 02	1.09979E-01	1.232E 02	4.47475E-02
8.801E 01	1.12753E-01	9.859E 01	1.16645E 00	1.104E 02	1.04578E-01	1.237E 02	4.35762E-02
8.841E 01	1.16431E-01	9.904E 01	2.04072E 00	1.109E 02	9.96354E-02	1.243E 02	4.24541E-02
8.881E 01	1.20402E-01	9.949E 01	8.40956E 00	1.115E 02	9.50964E-02	1.249E 02	4.13783E-02
8.921E 01	1.24702E-01	9.994E 01	3.90501E 00	1.120E 02	9.09121E-02	1.254E 02	4.03457E-02

GREATEST VALUE = 8.40956E 00 LOWEST VALUE = 0.0 INTERVAL = 7.31266E-02



FREQ	PHA	FREQ	PHA	FREQ	PHA	FREQ	PHA
8.000E 01	4.17999E 01	8.962E 01	3.83326E 01	1.004E 02	-1.45079E 02	1.125E 02	-1.48157E 02
8.036E 01	4.16588E 01	9.003E 01	3.81968E 01	1.009E 02	-1.45179E 02	1.130E 02	-1.48279E 02
8.073E 01	4.15178E 01	9.044E 01	3.80613E 01	1.013E 02	-1.45294E 02	1.135E 02	-1.48400E 02
8.110E 01	4.13771E 01	9.085E 01	3.79261E 01	1.018E 02	-1.45416E 02	1.140E 02	-1.48522E 02
8.147E 01	4.12365E 01	9.126E 01	3.77912E 01	1.022E 02	-1.45539E 02	1.145E 02	-1.48643E 02
8.184E 01	4.10961E 01	9.168E 01	3.76565E 01	1.027E 02	-1.45664E 02	1.151E 02	-1.48764E 02
8.221E 01	4.09559E 01	9.210E 01	3.75222E 01	1.032E 02	-1.45789E 02	1.156E 02	-1.48884E 02
8.258E 01	4.08159E 01	9.251E 01	3.73882E 01	1.036E 02	-1.45916E 02	1.161E 02	-1.49005E 02
8.296E 01	4.06761E 01	9.294E 01	3.72542E 01	1.041E 02	-1.46041E 02	1.166E 02	-1.49125E 02
8.334E 01	4.05365E 01	9.336E 01	3.71209E 01	1.046E 02	-1.46167E 02	1.172E 02	-1.49244E 02
8.372E 01	4.03971E 01	9.378E 01	3.69878E 01	1.051E 02	-1.46293E 02	1.177E 02	-1.49364E 02
8.410E 01	4.02579E 01	9.421E 01	3.68549E 01	1.055E 02	-1.46419E 02	1.182E 02	-1.49483E 02
8.448E 01	4.01190E 01	9.464E 01	3.67227E 01	1.060E 02	-1.46545E 02	1.188E 02	-1.49601E 02
8.487E 01	3.99802E 01	9.507E 01	3.65909E 01	1.065E 02	-1.46670E 02	1.193E 02	-1.49720E 02
8.525E 01	3.98417E 01	9.550E 01	3.64593E 01	1.070E 02	-1.46795E 02	1.199E 02	-1.49838E 02
8.564E 01	3.97034E 01	9.594E 01	3.63286E 01	1.075E 02	-1.46921E 02	1.204E 02	-1.49956E 02
8.603E 01	3.95652E 01	9.637E 01	3.61987E 01	1.080E 02	-1.47045E 02	1.209E 02	-1.50073E 02
8.642E 01	3.94273E 01	9.681E 01	3.60693E 01	1.085E 02	-1.47170E 02	1.215E 02	-1.50190E 02
8.682E 01	3.92896E 01	9.725E 01	3.59418E 01	1.089E 02	-1.47294E 02	1.220E 02	-1.50307E 02
8.721E 01	3.91522E 01	9.770E 01	3.58161E 01	1.094E 02	-1.47418E 02	1.226E 02	-1.50424E 02
8.761E 01	3.90149E 01	9.814E 01	3.56942E 01	1.099E 02	-1.47542E 02	1.232E 02	-1.50540E 02
8.801E 01	3.88781E 01	9.859E 01	3.55813E 01	1.104E 02	-1.47666E 02	1.237E 02	-1.50656E 02
8.841E 01	3.87413E 01	9.904E 01	3.54947E 01	1.109E 02	-1.47789E 02	1.243E 02	-1.50771E 02
8.881E 01	3.86048E 01	9.949E 01	3.56879E 01	1.115E 02	-1.47912E 02	1.249E 02	-1.50886E 02
8.921E 01	3.84686E 01	9.994E 01	-1.45068E 02	1.120E 02	-1.48034E 02	1.254E 02	-1.51001E 02

GREATEST VALUE = 4.17959E 01 LOWEST VALUE = -1.51001E 02 INTERVAL = 1.67653E 00



PLOT(TY=RE/FR=0/TC=100)

IHC210I IBCOM - PROGRAM INTERRUPT- IMPRECISE OLD PSW IS FF45004002234AE6

TRACEBACK FOLLOWS-	ROUTINE	ISN	REG. 14	REG. 15	REG. 0	REG. 1
	PLOT	0135	4221C49A	00234010	00000000	0021C9A4
	MAIN		00019472	0121C518	00000015	0023BFF8

ENTRY POINT= 0121C518

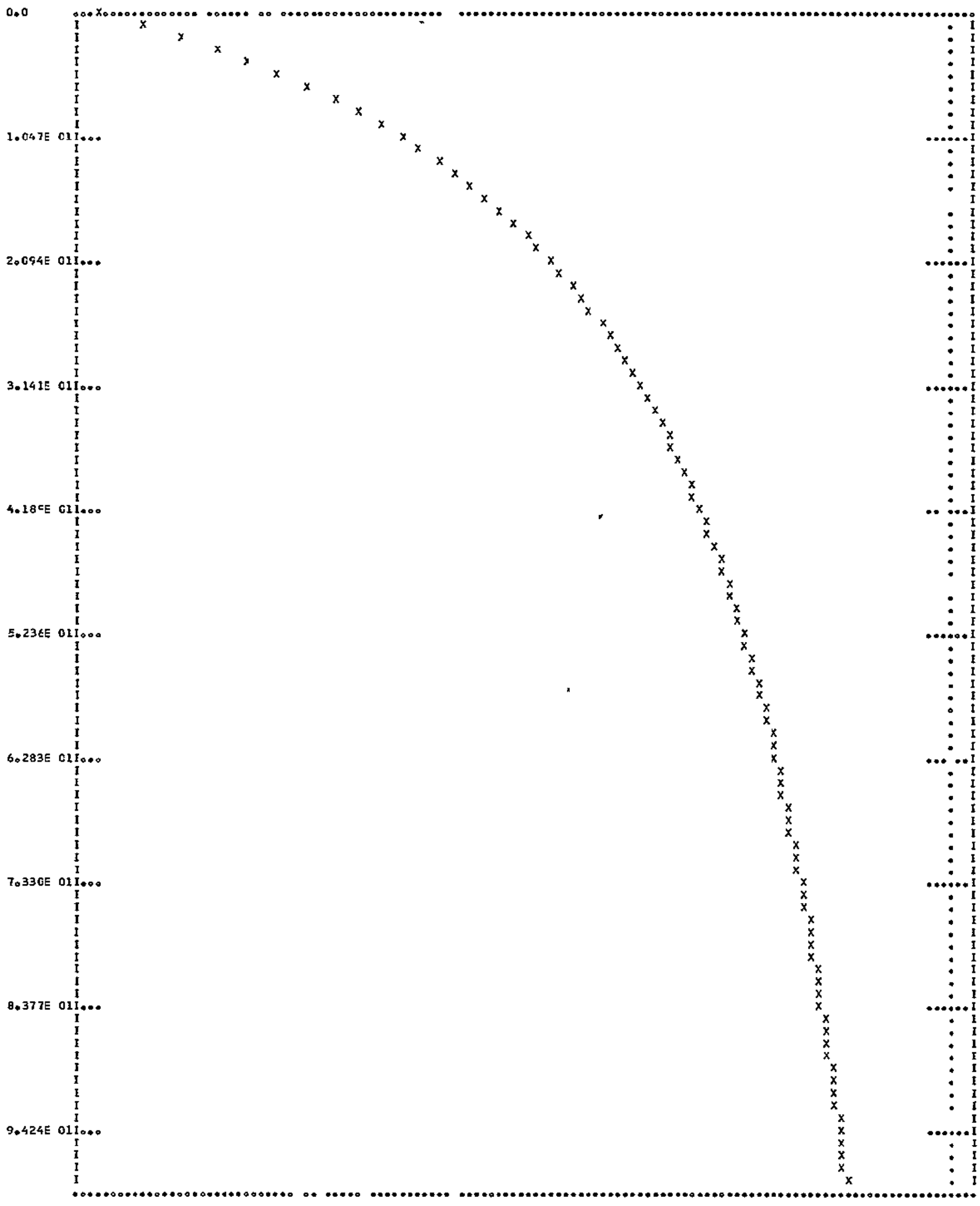
STANDARD FIXUP TAKEN , EXECUTION CONTINUING

THE TRANSFORM OF THE RESPONSE IS ---

EXP. OF S	NUMER. COEFFS.	DENOM. COEFFS.
0	-3.00000E 01	1.01930E 02
1	-1.29997E-02	5.21768E 00
2	3.03599E-04	1.38103E-02
3	0.0	1.33138E-05
4	0.0	3.45773E-08

FREQ	MAG	FREQ	MAG	FREQ	MAG	FREQ	MAG
0.0	-2.94319E-01	2.408E 01	-1.27807E-01	4.817E 01	-7.73230E-02	7.225E 01	-5.27031E-02
1.047E 00	-2.79430E-01	2.513E 01	-1.24523E-01	4.922E 01	-7.59075E-02	7.330E 01	-5.19132E-02
2.094E 00	-2.65909E-01	2.618E 01	-1.21384E-01	5.026E 01	-7.45330E-02	7.435E 01	-5.11403E-02
3.141E 00	-2.53575E-01	2.723E 01	-1.18381E-01	5.131E 01	-7.31978E-02	7.539E 01	-5.03838E-02
4.189E 00	-2.42278E-01	2.827E 01	-1.15506E-01	5.236E 01	-7.19002E-02	7.644E 01	-4.96432E-02
5.236E 00	-2.31891E-01	2.932E 01	-1.12749E-01	5.340E 01	-7.06385E-02	7.749E 01	-4.89181E-02
6.283E 00	-2.22310E-01	3.037E 01	-1.10104E-01	5.445E 01	-6.94113E-02	7.853E 01	-4.82078E-02
7.330E 00	-2.13443E-01	3.141E 01	-1.07565E-01	5.550E 01	-6.82172E-02	7.958E 01	-4.75120E-02
8.377E 00	-2.05214E-01	3.246E 01	-1.05124E-01	5.654E 01	-6.70549E-02	8.063E 01	-4.68303E-02
9.424E 00	-1.97555E-01	3.351E 01	-1.02777E-01	5.759E 01	-6.59230E-02	8.168E 01	-4.61621E-02
1.047E 01	-1.90409E-01	3.456E 01	-1.00518E-01	5.864E 01	-6.48204E-02	8.272E 01	-4.55071E-02
1.152E 01	-1.83726E-01	3.560E 01	-9.83412E-02	5.969E 01	-6.37460E-02	8.377E 01	-4.48649E-02
1.257E 01	-1.77462E-01	3.665E 01	-9.62433E-02	6.073E 01	-6.26985E-02	8.482E 01	-4.42351E-02
1.361E 01	-1.71578E-01	3.770E 01	-9.42196E-02	6.178E 01	-6.16772E-02	8.586E 01	-4.36174E-02
1.466E 01	-1.66042E-01	3.874E 01	-9.22663E-02	6.283E 01	-6.06809E-02	8.691E 01	-4.30114E-02
1.571E 01	-1.60823E-01	3.979E 01	-9.03797E-02	6.387E 01	-5.97088E-02	8.796E 01	-4.24168E-02
1.675E 01	-1.55894E-01	4.084E 01	-8.85562E-02	6.492E 01	-5.87599E-02	8.901E 01	-4.18332E-02
1.780E 01	-1.51231E-01	4.189E 01	-8.67930E-02	6.597E 01	-5.78334E-02	9.005E 01	-4.12604E-02
1.885E 01	-1.46814E-01	4.293E 01	-8.50869E-02	6.702E 01	-5.69285E-02	9.110E 01	-4.06980E-02
1.990E 01	-1.42623E-01	4.398E 01	-8.34354E-02	6.806E 01	-5.60445E-02	9.215E 01	-4.01458E-02
2.094E 01	-1.38642E-01	4.503E 01	-8.18354E-02	6.911E 01	-5.51806E-02	9.319E 01	-3.96035E-02
2.199E 01	-1.34855E-01	4.607E 01	-8.02850E-02	7.016E 01	-5.43362E-02	9.424E 01	-3.90708E-02
2.304E 01	-1.31247E-01	4.712E 01	-7.87815E-02	7.120E 01	-5.35106E-02	9.529E 01	-3.85474E-02

GREATEST VALUE = 0.0 LOWEST VALUE = -2.94319E-01 INTERVAL = 2.55930E-03



ROOTS,POLES

*** SVOBODA POLYNOMIAL ROOTFINDER ***

H.F.OKRENT
JAN. 1969

POLYNOMIAL OF DEGREE 4 --

0.1019299927E 03 X^{0} 0.5217680931E 01 X^{1} 0.1381032541E-01 X^{2} 0.1331378280E-04 X^{3} 0.3457731168E-07 X^{4}

ACCURACY REQUESTED -- 6 SIGNIFICANT FIGURES

REAL PART	IMAGINARY PART	EXPONENT
-3.644150000	0.0	2
0.000070000	6.260090000	2
0.000070000	-6.260080000	2
-2.064200000	0.0	1

*** SVOBODA POLYNOMIAL ROOTFINDER ***

H.F.OKRENT
JAN. 1969

POLYNOMIAL OF DEGREE 2 --

0.2999998474E 02 X^{0} 0.1299971715E-01 X^{1} -0.3035992850E-03 X^{2}

ACCURACY REQUESTED -- 6 SIGNIFICANT FIGURES

REAL PART	IMAGINARY PART	EXPONENT
-2.936660000	0.0	2
3.364850000	0.0	2

SENSITIVITY OF POLES TO C1

POLES REAL	POLES IMAG	SENSITIVITY REAL	SENSITIVITY IMAG
-3.6441E 02	0.0	-6.8345E 01	0.0
7.0000E-03	6.2601E 02	7.0006E-03	-5.1438E 01
7.0000E-03	-6.2601E 02	7.0006E-03	-4.5883E 01
-2.0642E 01	0.0	-1.5917E 01	0.0

SENSITIVITY OF POLES TO R1

POLES REAL	POLES IMAG	SENSITIVITY REAL	SENSITIVITY IMAG
-3.6441E 02	0.0	-9.3773E-02	0.0
7.0000E-03	6.2601E 02	1.3254E-04	-2.7139E-01
7.0000E-03	-6.2601E 02	1.3254E-04	-1.8254E-01
-2.0642E 01	0.0	-3.8556E-01	0.0

SENSITIVITY OF POLES TO C2

POLES REAL	POLES IMAG	SENSITIVITY REAL	SENSITIVITY IMAG
-3.6441E 02	0.0	1.8731E 01	0.0
7.0000E-03	6.2601E 02	9.7225E-09	-1.3517E 01
7.0000E-03	-6.2601E 02	9.7225E-09	2.3180E 01
-2.0642E 01	0.0	2.5619E-02	0.0

SENSITIVITY OF POLES TO R2

POLES REAL	POLES IMAG	SENSITIVITY REAL	SENSITIVITY IMAG
-3.6441E 02	0.0	-2.9972E 02	0.0
7.0000E-03	6.2601E 02	6.9677E-03	-1.2029E 02
7.0000E-03	-6.2601E 02	6.9677E-03	-2.0787E 02
-2.0642E 01	0.0	-3.2968E 00	0.0

SENSITIVITY OF POLES TO R3

POLES REAL	POLES IMAG	SENSITIVITY REAL	SENSITIVITY IMAG
-3.6441E 02	0.0	6.2044E 01	0.0
7.0000E-03	6.2601E 02	1.0021E-04	-3.5429E 02
7.0000E-03	-6.2601E 02	1.0021E-04	2.3329E 02
-2.0642E 01	0.0	1.6971E 01	0.0

SENSITIVITY OF POLES TO L1

POLES		SENSITIVITY	
REAL	IMAG	REAL	IMAG
-3.6441E 02	0.0	1.2782E 00	0.0
7.0000E-03	6.2601E 02	1.0770E-12	-1.9614E 02
7.0000E-03	-6.2601E 02	1.0770E-12	9.2333E 01
-2.0642E 01	0.0	-5.4707E-03	0.0

SENSITIVITY OF POLES TO C3

POLES		SENSITIVITY	
REAL	IMAG	REAL	IMAG
-3.6441E 02	0.0	1.8455E 00	0.0
7.0000E-03	6.2601E 02	3.7113E-08	-4.3317E 01
7.0000E-03	-6.2601E 02	3.7113E-08	2.2019E 01
-2.0642E 01	0.0	3.1672E-01	0.0

SENSITIVITY OF POLES TO C4

POLES		SENSITIVITY	
REAL	IMAG	REAL	IMAG
-3.6441E 02	0.0	2.3379E 02	0.0
7.0000E-03	6.2601E 02	4.9959E-07	-2.3906E 02
7.0000E-03	-6.2601E 02	4.9959E-07	3.2770E 02
-2.0642E 01	0.0	3.6222E 00	0.0

SENSITIVITY OF POLES TO C5

POLES		SENSITIVITY	
REAL	IMAG	REAL	IMAG
-3.6441E 02	0.0	4.0428E 01	0.0
7.0000E-03	6.2601E 02	1.0630E-07	-1.8541E 02
7.0000E-03	-6.2601E 02	1.0630E-07	1.1490E 02
-2.0642E 01	0.0	7.6547E-01	0.0

SENSITIVITY OF POLES TO I2

POLES		SENSITIVITY	
REAL	IMAG	REAL	IMAG
-3.6441E 02	0.0	-4.2723E 01	0.0
7.0000E-03	6.2601E 02	2.2986E-05	-1.2323E 01
7.0000E-03	-6.2601E 02	2.2986E-05	-2.2735E 01
-2.0642E 01	0.0	2.9086E 00	0.0

PLOT(TY=SE/FR=80/TO=126/EL=C1)

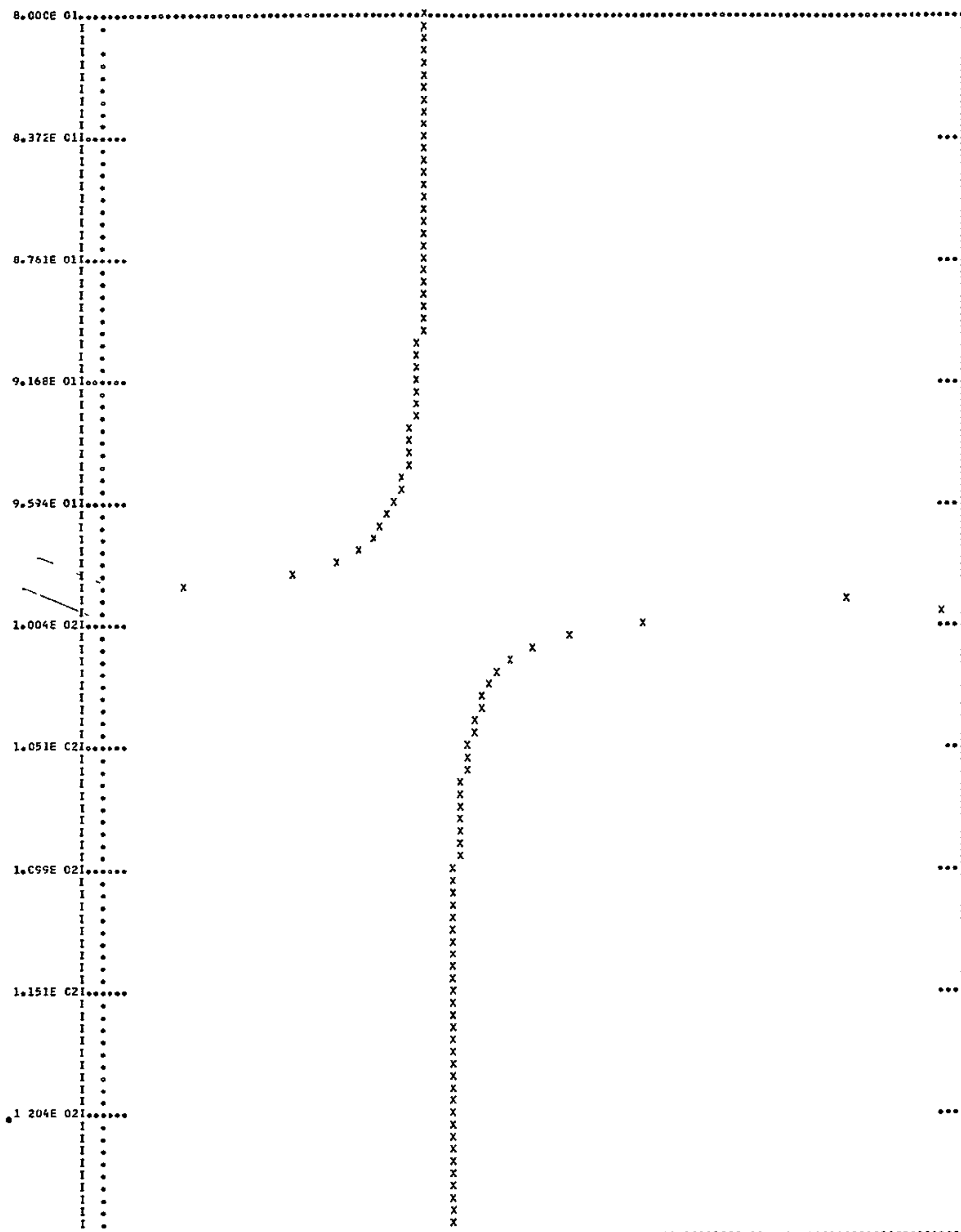
SENSITIVITY WITH RESPECT TO C1

THE TRANSFORM OF THE RESPONSE IS ---

EXP. OF S	NUMER. COEFFS.	DENOM. COEFFS.
0	0.0	1.01930E 02
1	-3.86000E 00	5.21768E 00
2	-1.29800E-02	1.38103E-02
3	-9.92998E-06	1.33138E-05
4	-3.34399E-08	3.45773E-08

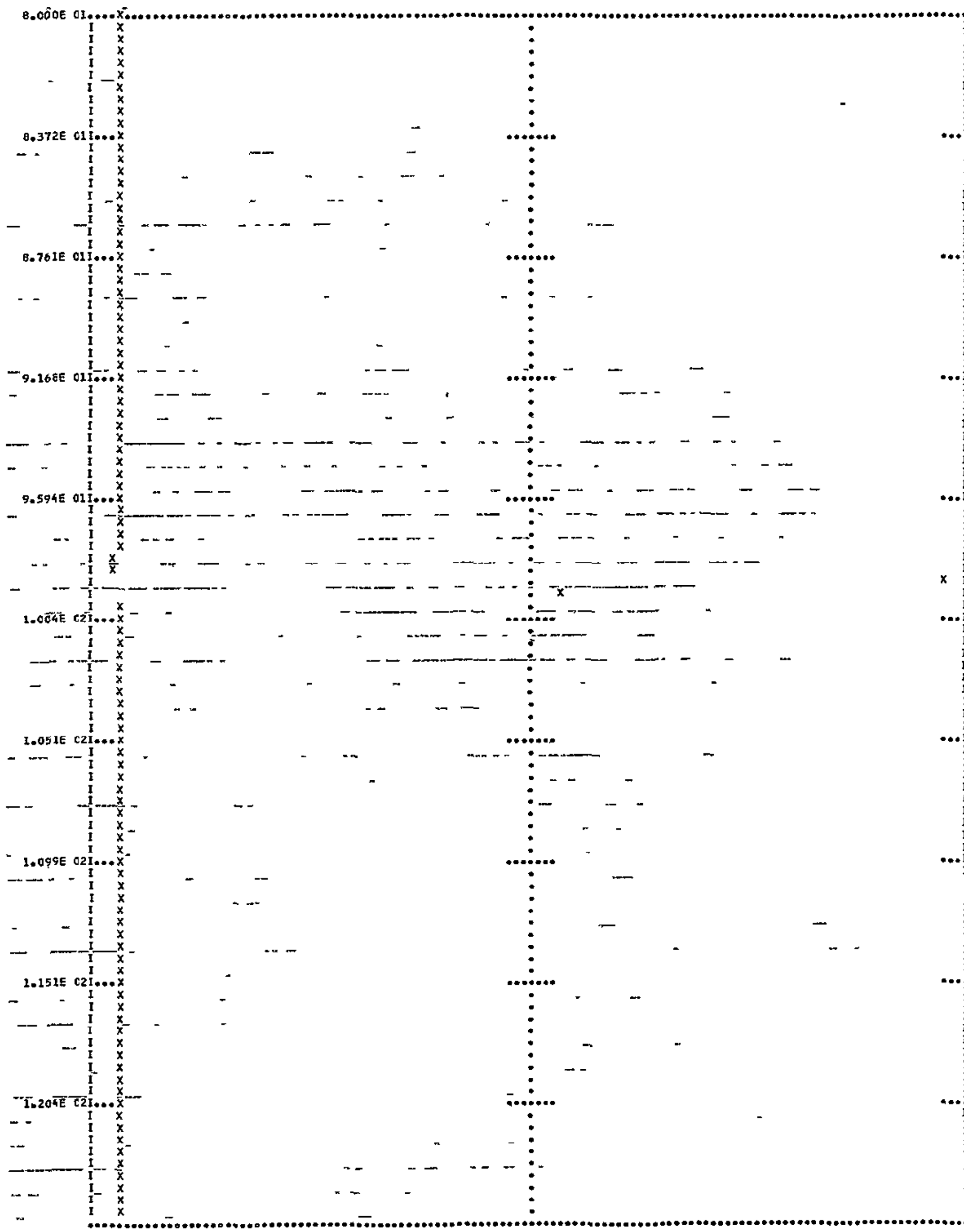
FREQ	MAG	FREQ	MAG	FREQ	MAG	FREQ	MAG
8.000E 01	8.85262E-01	8.962E 01	8.73062E-01	1.004E 02	1.48156E 00	1.125E 02	9.63554E-01
8.036E 01	8.85212E-01	9.003E 01	8.71585E-01	1.009E 02	1.27286E 00	1.130E 02	9.62539E-01
8.073E 01	8.85144E-01	9.044E 01	8.69936E-01	1.013E 02	1.17773E 00	1.135E 02	9.61613E-01
8.110E 01	8.85056E-01	9.085E 01	8.68090E-01	1.018E 02	1.12335E 00	1.140E 02	9.60764E-01
8.147E 01	8.84945E-01	9.126E 01	8.66018E-01	1.022E 02	1.08824E 00	1.145E 02	9.59987E-01
8.184E 01	8.84813E-01	9.168E 01	8.63683E-01	1.027E 02	1.06370E 00	1.151E 02	9.59272E-01
8.221E 01	8.84655E-01	9.210E 01	8.61039E-01	1.032E 02	1.04565E 00	1.156E 02	9.58615E-01
8.258E 01	8.84473E-01	9.251E 01	8.58032E-01	1.036E 02	1.03181E 00	1.161E 02	9.58010E-01
8.296E 01	8.84262E-01	9.294E 01	8.54584E-01	1.041E 02	1.02089E 00	1.166E 02	9.57455E-01
8.334E 01	8.84021E-01	9.336E 01	8.50619E-01	1.046E 02	1.01206E 00	1.172E 02	9.56945E-01
8.372E 01	8.83748E-01	9.378E 01	8.45998E-01	1.051E 02	1.00480E 00	1.177E 02	9.56471E-01
8.410E 01	8.83440E-01	9.421E 01	8.40579E-01	1.055E 02	9.98722E-01	1.182E 02	9.56036E-01
8.448E 01	8.83094E-01	9.464E 01	8.34142E-01	1.060E 02	9.93579E-01	1.188E 02	9.55635E-01
8.487E 01	8.82706E-01	9.507E 01	8.26403E-01	1.065E 02	9.89166E-01	1.193E 02	9.55264E-01
8.525E 01	8.82279E-01	9.550E 01	8.16917E-01	1.070E 02	9.85358E-01	1.199E 02	9.54922E-01
8.564E 01	8.81800E-01	9.594E 01	8.05090E-01	1.075E 02	9.82030E-01	1.204E 02	9.54610E-01
8.603E 01	8.81270E-01	9.637E 01	7.89946E-01	1.080E 02	9.79113E-01	1.209E 02	9.54319E-01
8.642E 01	8.80683E-01	9.681E 01	7.69901E-01	1.085E 02	9.76532E-01	1.215E 02	9.54051E-01
8.682E 01	8.80032E-01	9.725E 01	7.42214E-01	1.089E 02	9.74247E-01	1.220E 02	9.53805E-01
8.721E 01	8.79313E-01	9.770E 01	7.01560E-01	1.094E 02	9.72203E-01	1.226E 02	9.53580E-01
8.761E 01	8.78518E-01	9.814E 01	6.36253E-01	1.099E 02	9.70371E-01	1.232E 02	9.53372E-01
8.801E 01	8.77638E-01	9.859E 01	5.14543E-01	1.104E 02	9.68722E-01	1.237E 02	9.53180E-01
8.841E 01	8.76666E-01	9.904E 01	2.10613E-01	1.109E 02	9.67235E-01	1.243E 02	9.53004E-01
8.881E 01	8.75586E-01	9.949E 01	2.04599E 00	1.115E 02	9.65889E-01	1.249E 02	9.52846E-01
8.921E 01	8.74390E-01	9.994E 01	2.30366E 00	1.120E 02	9.64665E-01	1.254E 02	9.52699E-01

GREATEST VALUE = 2.30366E 00   LOWEST VALUE = 0.0   INTERVAL = 2.00319E-02



FREQ	PHA	FREQ	PHA	FREQ	PHA	FREQ	PHA
8.000E 01	-1.72381E 02	8.962E 01	-1.72977E 02	1.004E 02	-1.72113E 02	1.125E 02	-1.73742E 02
8.036E 01	-1.72403E 02	9.003E 01	-1.73005E 02	1.009E 02	-1.72459E 02	1.130E 02	-1.73767E 02
8.073E 01	-1.72425E 02	9.044E 01	-1.73034E 02	1.013E 02	-1.72669E 02	1.135E 02	-1.73791E 02
8.110E 01	-1.72447E 02	9.085E 01	-1.73064E 02	1.018E 02	-1.72815E 02	1.140E 02	-1.73815E 02
8.147E 01	-1.72470E 02	9.126E 01	-1.73094E 02	1.022E 02	-1.72923E 02	1.145E 02	-1.73839E 02
8.184E 01	-1.72492E 02	9.168E 01	-1.73126E 02	1.027E 02	-1.73009E 02	1.151E 02	-1.73863E 02
8.221E 01	-1.72514E 02	9.210E 01	-1.73159E 02	1.032E 02	-1.73080E 02	1.156E 02	-1.73887E 02
8.258E 01	-1.72537E 02	9.251E 01	-1.73193E 02	1.036E 02	-1.73141E 02	1.161E 02	-1.73910E 02
8.296E 01	-1.72559E 02	9.294E 01	-1.73230E 02	1.041E 02	-1.73195E 02	1.166E 02	-1.73933E 02
8.334E 01	-1.72582E 02	9.336E 01	-1.73269E 02	1.046E 02	-1.73243E 02	1.172E 02	-1.73956E 02
8.372E 01	-1.72605E 02	9.378E 01	-1.73311E 02	1.051E 02	-1.73286E 02	1.177E 02	-1.73979E 02
8.410E 01	-1.72628E 02	9.421E 01	-1.73357E 02	1.055E 02	-1.73326E 02	1.182E 02	-1.74002E 02
8.448E 01	-1.72652E 02	9.464E 01	-1.73407E 02	1.060E 02	-1.73364E 02	1.188E 02	-1.74024E 02
8.487E 01	-1.72675E 02	9.507E 01	-1.73464E 02	1.065E 02	-1.73400E 02	1.193E 02	-1.74047E 02
8.525E 01	-1.72699E 02	9.550E 01	-1.73530E 02	1.070E 02	-1.73433E 02	1.199E 02	-1.74069E 02
8.564E 01	-1.72722E 02	9.594E 01	-1.73607E 02	1.075E 02	-1.73466E 02	1.204E 02	-1.74091E 02
8.603E 01	-1.72746E 02	9.637E 01	-1.73703E 02	1.080E 02	-1.73497E 02	1.209E 02	-1.74114E 02
8.642E 01	-1.72771E 02	9.681E 01	-1.73828E 02	1.085E 02	-1.73527E 02	1.215E 02	-1.74135E 02
8.682E 01	-1.72795E 02	9.725E 01	-1.74001E 02	1.089E 02	-1.73555E 02	1.220E 02	-1.74157E 02
8.721E 01	-1.72820E 02	9.770E 01	-1.74267E 02	1.094E 02	-1.73584E 02	1.226E 02	-1.74179E 02
8.761E 01	-1.72845E 02	9.814E 01	-1.74748E 02	1.099E 02	-1.73611E 02	1.232E 02	-1.74201E 02
8.801E 01	-1.72871E 02	9.859E 01	-1.75943E 02	1.104E 02	-1.73638E 02	1.237E 02	-1.74222E 02
8.841E 01	-1.72897E 02	9.904E 01	1.74962E 02	1.109E 02	-1.73665E 02	1.243E 02	-1.74244E 02
8.881E 01	-1.72923E 02	9.949E 01	1.21490E 01	1.115E 02	-1.73691E 02	1.249E 02	-1.74265E 02
8.921E 01	-1.72950E 02	9.994E 01	-1.71442E 02	1.120E 02	-1.73716E 02	1.254E 02	-1.74287E 02

GREATEST VALUE = 1.74562E 02 LOWEST VALUE = -1.75943E 02 INTERVAL = 3.05134E 00



PLOT(TY=SE/FR=80/TC=126/EL=I2)

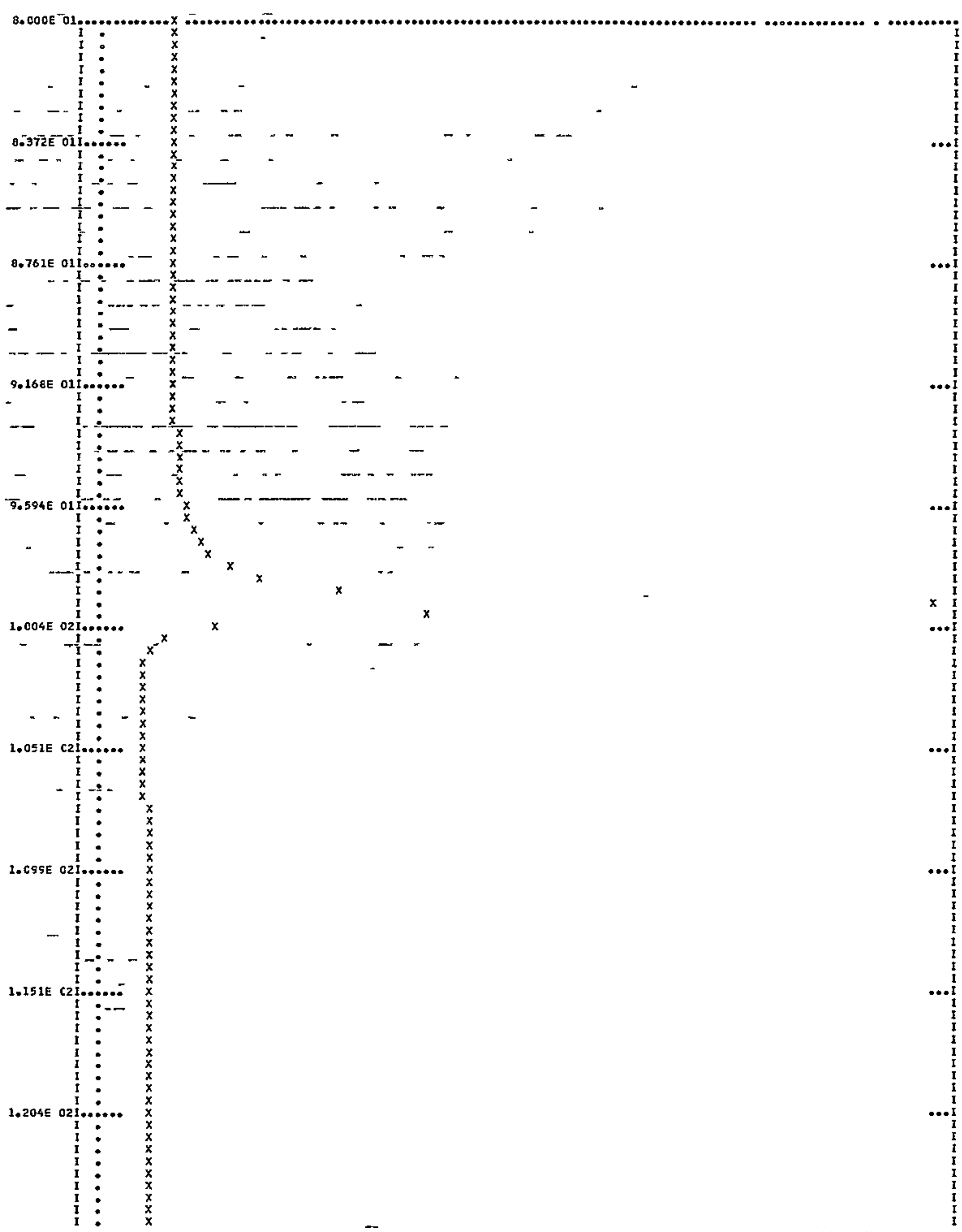
SENSITIVITY WITH RESPECT TO I2

THE TRANSFORM OF THE RESPONSE IS ---

EXP. OF S	NUMER. COEFFS.	DENOM. COEFFS.
0	-3.04800E 03	3.05790E 03
1	-1.66603E 02	1.57855E 02
2	-1.95485E 00	4.51192E-01
3	-4.69070E-03	-1.00514E-03
4	-4.98580E-06	-2.98241E-06
5	-1.04167E-08	-3.59256E-09
6	0.0	-1.04976E-11

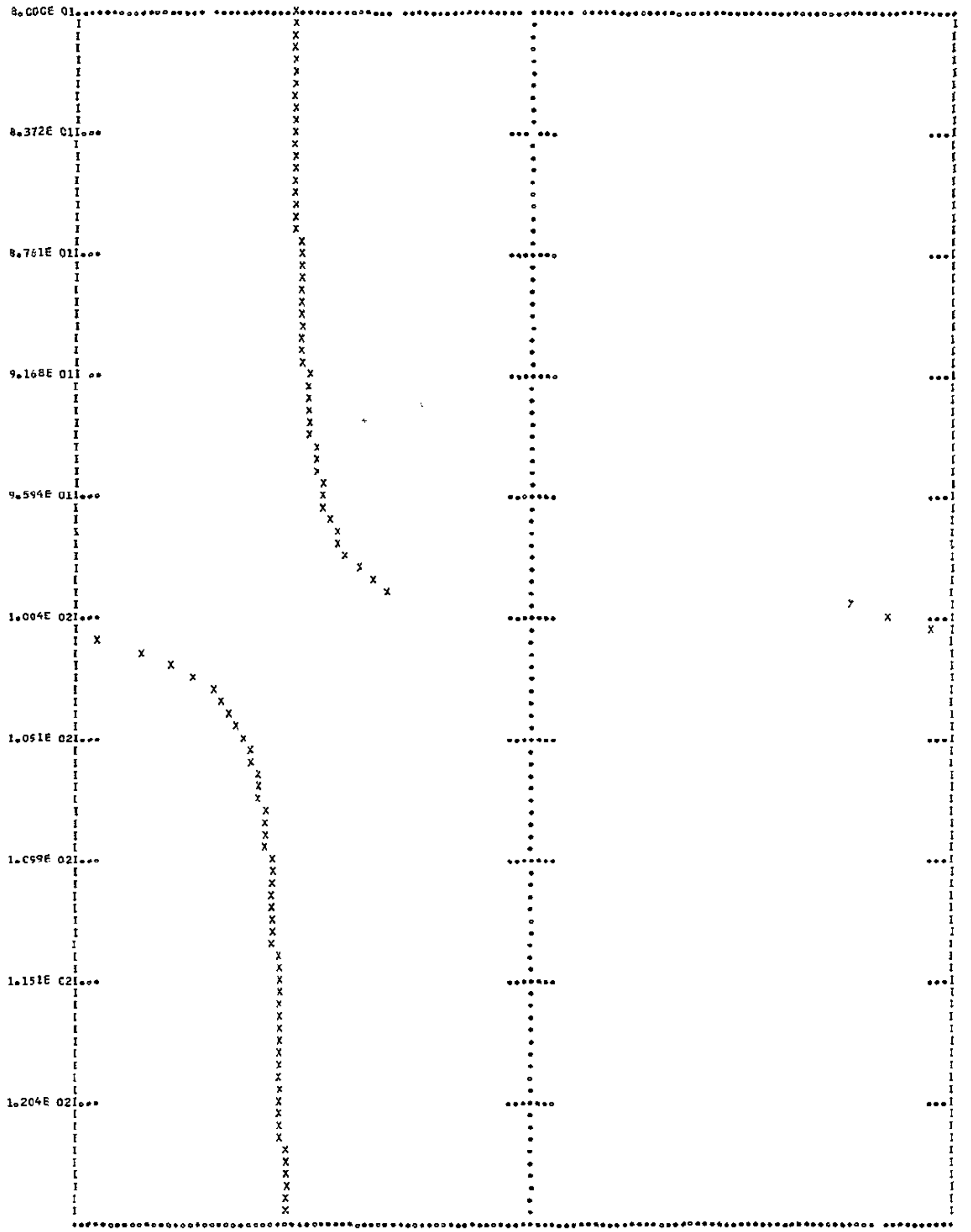
FREQ	MAG	FREQ	MAG	FREQ	MAG	FREQ	MAG
8.000E 01	1.54002E 00	8.962E 01	1.55486E 00	1.004E 02	2.43963E 00	1.125E 02	1.06082E 00
8.036E 01	1.53885E 00	9.003E 01	1.55985E 00	1.009E 02	1.38988E 00	1.130E 02	1.06186E 00
8.073E 01	1.53774E 00	9.044E 01	1.56563E 00	1.013E 02	1.02433E 00	1.135E 02	1.06260E 00
8.110E 01	1.53670E 00	9.085E 01	1.57233E 00	1.018E 02	9.00377E-01	1.140E 02	1.06306E 00
8.147E 01	1.53573E 00	9.126E 01	1.58011E 00	1.022E 02	8.70761E-01	1.145E 02	1.06327E 00
8.184E 01	1.53485E 00	9.168E 01	1.58915E 00	1.027E 02	8.76410E-01	1.151E 02	1.06325E 00
8.221E 01	1.53405E 00	9.210E 01	1.59967E 00	1.032E 02	8.93983E-01	1.156E 02	1.06302E 00
8.258E 01	1.53336E 00	9.251E 01	1.61197E 00	1.036E 02	9.14457E-01	1.161E 02	1.06260E 00
8.296E 01	1.53276E 00	9.294E 01	1.62639E 00	1.041E 02	9.34405E-01	1.166E 02	1.06199E 00
8.334E 01	1.53229E 00	9.336E 01	1.64342E 00	1.046E 02	9.52633E-01	1.172E 02	1.06123E 00
8.372E 01	1.53194E 00	9.378E 01	1.66365E 00	1.051E 02	9.68835E-01	1.177E 02	1.06031E 00
8.410E 01	1.53173E 00	9.421E 01	1.68791E 00	1.055E 02	9.83041E-01	1.182E 02	1.05926E 00
8.448E 01	1.53167E 00	9.464E 01	1.71730E 00	1.060E 02	9.95413E-01	1.188E 02	1.05807E 00
8.487E 01	1.53178E 00	9.507E 01	1.75337E 00	1.065E 02	1.00615E 00	1.193E 02	1.05677E 00
8.525E 01	1.53207E 00	9.550E 01	1.79838E 00	1.070E 02	1.01546E 00	1.199E 02	1.05535E 00
8.564E 01	1.53255E 00	9.594E 01	1.85566E 00	1.075E 02	1.02351E 00	1.204E 02	1.05384E 00
8.603E 01	1.53327E 00	9.637E 01	1.93044E 00	1.080E 02	1.03045E 00	1.209E 02	1.05222E 00
8.642E 01	1.53423E 00	9.681E 01	2.03136E 00	1.085E 02	1.03644E 00	1.215E 02	1.05052E 00
8.682E 01	1.53546E 00	9.725E 01	2.17367E 00	1.089E 02	1.04158E 00	1.220E 02	1.04873E 00
8.721E 01	1.53699E 00	9.770E 01	2.38712E 00	1.094E 02	1.04598E 00	1.226E 02	1.04686E 00
8.761E 01	1.53885E 00	9.814E 01	2.73791E 00	1.099E 02	1.04972E 00	1.232E 02	1.04492E 00
8.801E 01	1.54109E 00	9.859E 01	3.40782E 00	1.104E 02	1.05287E 00	1.237E 02	1.04291E 00
8.841E 01	1.54376E 00	9.904E 01	5.13606E 00	1.109E 02	1.05551E 00	1.243E 02	1.04084E 00
8.881E 01	1.54690E 00	9.949E 01	1.79854E 01	1.115E 02	1.05768E 00	1.249E 02	1.03870E 00
8.921E 01	1.55058E 00	9.994E 01	7.05039E 00	1.120E 02	1.05944E 00	1.254E 02	1.03651E 00

GREATEST VALUE = 1.79854E 01   LOWEST VALUE = 0.0   INTERVAL = 1.56395E-01



FREQ	PHA	FREQ	PHA	FREQ	PHA	FREQ	PHA
8.000E 01	-9.93251E 01	8.962E 01	-9.50891E 01	1.004E 02	1.44277E 02	1.125E 02	-1.06240E 02
8.036E 01	-9.92117E 01	9.003E 01	-9.47981E 01	1.009E 02	1.62369E 02	1.130E 02	-1.05920E 02
8.073E 01	-9.90957E 01	9.044E 01	-9.44873E 01	1.013E 02	-1.78023E 02	1.135E 02	-1.05621E 02
8.110E 01	-9.89773E 01	9.085E 01	-9.41544E 01	1.018E 02	-1.60860E 02	1.140E 02	-1.05342E 02
8.147E 01	-9.88560E 01	9.126E 01	-9.37964E 01	1.022E 02	-1.47925E 02	1.145E 02	-1.05081E 02
8.184E 01	-9.87319E 01	9.168E 01	-9.34097E 01	1.027E 02	-1.38692E 02	1.151E 02	-1.04834E 02
8.221E 01	-9.86044E 01	9.210E 01	-9.29903E 01	1.032E 02	-1.32083E 02	1.156E 02	-1.04602E 02
8.258E 01	-9.84737E 01	9.251E 01	-9.25332E 01	1.036E 02	-1.27227E 02	1.161E 02	-1.04383E 02
8.296E 01	-9.83393E 01	9.294E 01	-9.20324E 01	1.041E 02	-1.23554E 02	1.166E 02	-1.04175E 02
8.334E 01	-9.82010E 01	9.336E 01	-9.14806E 01	1.046E 02	-1.20697E 02	1.172E 02	-1.03978E 02
8.372E 01	-9.80586E 01	9.378E 01	-9.08687E 01	1.051E 02	-1.18418E 02	1.177E 02	-1.03790E 02
8.410E 01	-9.79117E 01	9.421E 01	-9.01854E 01	1.055E 02	-1.16562E 02	1.182E 02	-1.03611E 02
8.448E 01	-9.77599E 01	9.464E 01	-8.94169E 01	1.060E 02	-1.15021E 02	1.188E 02	-1.03439E 02
8.487E 01	-9.76028E 01	9.507E 01	-8.85444E 01	1.065E 02	-1.13722E 02	1.193E 02	-1.03276E 02
8.525E 01	-9.74401E 01	9.550E 01	-8.75446E 01	1.070E 02	-1.12612E 02	1.199E 02	-1.03119E 02
8.564E 01	-9.72713E 01	9.594E 01	-8.63862E 01	1.075E 02	-1.11651E 02	1.204E 02	-1.02968E 02
8.603E 01	-9.70957E 01	9.637E 01	-8.50276E 01	1.080E 02	-1.10811E 02	1.209E 02	-1.02823E 02
8.642E 01	-9.69129E 01	9.681E 01	-8.34102E 01	1.085E 02	-1.10070E 02	1.215E 02	-1.02684E 02
8.682E 01	-9.67223E 01	9.725E 01	-8.14533E 01	1.089E 02	-1.09411E 02	1.220E 02	-1.02550E 02
8.721E 01	-9.65228E 01	9.770E 01	-7.90394E 01	1.094E 02	-1.08820E 02	1.226E 02	-1.02420E 02
8.761E 01	-9.63139E 01	9.814E 01	-7.59932E 01	1.099E 02	-1.08287E 02	1.232E 02	-1.02295E 02
8.801E 01	-9.60946E 01	9.859E 01	-7.20476E 01	1.104E 02	-1.07804E 02	1.237E 02	-1.02174E 02
8.841E 01	-9.58638E 01	9.904E 01	-6.67705E 01	1.109E 02	-1.07362E 02	1.243E 02	-1.02056E 02
8.881E 01	-9.56202E 01	9.949E 01	-5.92705E 01	1.115E 02	-1.06958E 02	1.249E 02	-1.01942E 02
8.921E 01	-9.53625E 01	9.994E 01	1.30155E 02	1.120E 02	-1.06585E 02	1.254E 02	-1.01832E 02

GREATEST VALUE = 1.62369E 02   LOWEST VALUE = -1.78023E 02   INTERVAL = 2.95953E 00



PLOT(TY=SE/FR=80/TO=126/EL=C3)

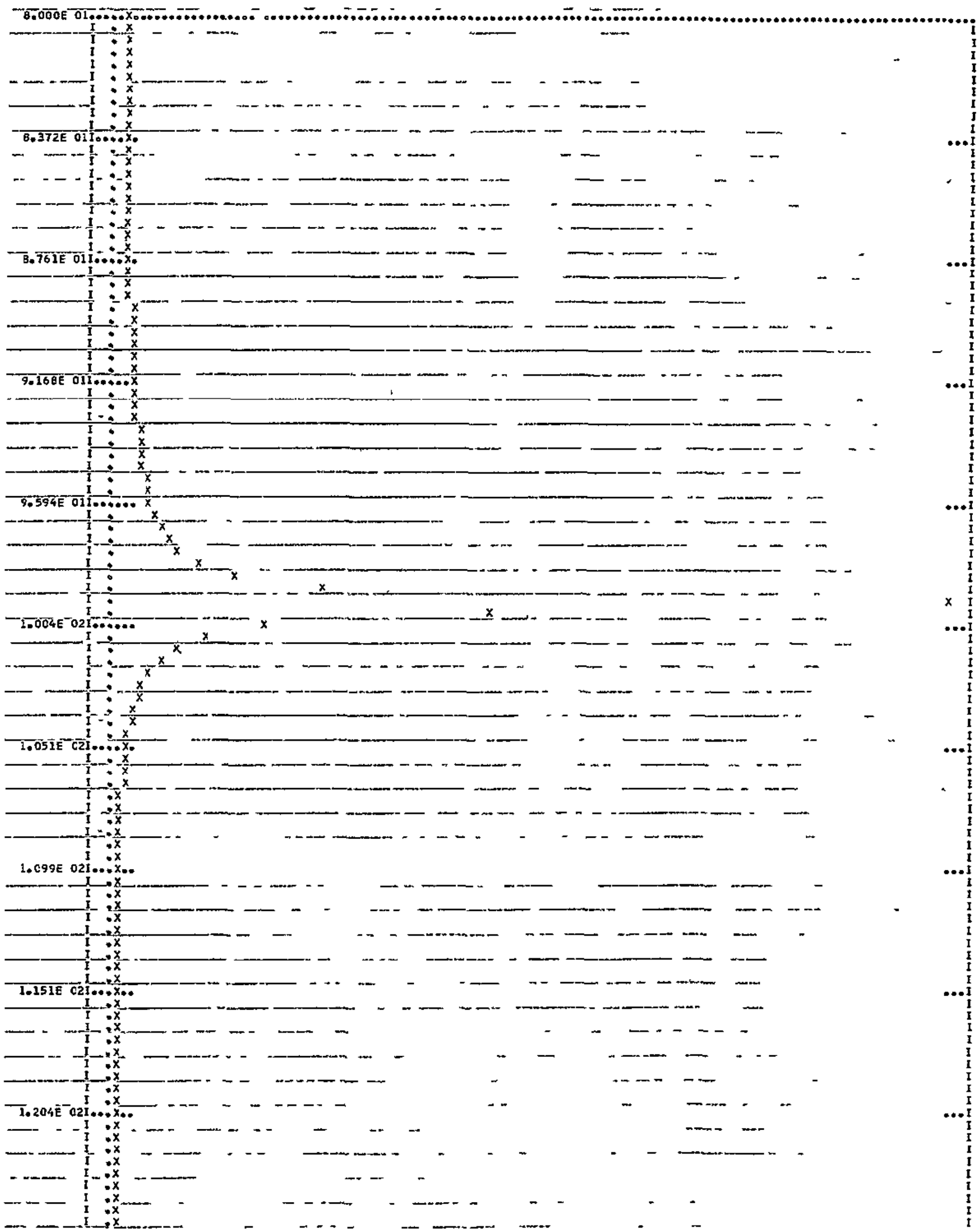
SENSITIVITY WITH RESPECT TO C3

THE TRANSFORM OF THE RESPONSE IS ---

EXP. OF S	NUMER. COEFFS.	DENOM. COEFFS.
0	0.0	3.05790E 03
1	1.01850E 01	1.57855E 02
2	4.40036E-01	4.51192E-01
3	2.48142E-03	-1.00514E-03
4	4.76209E-06	-2.98241E-06
5	5.40885E-09	-3.59256E-09
6	6.81754E-12	-1.04976E-11

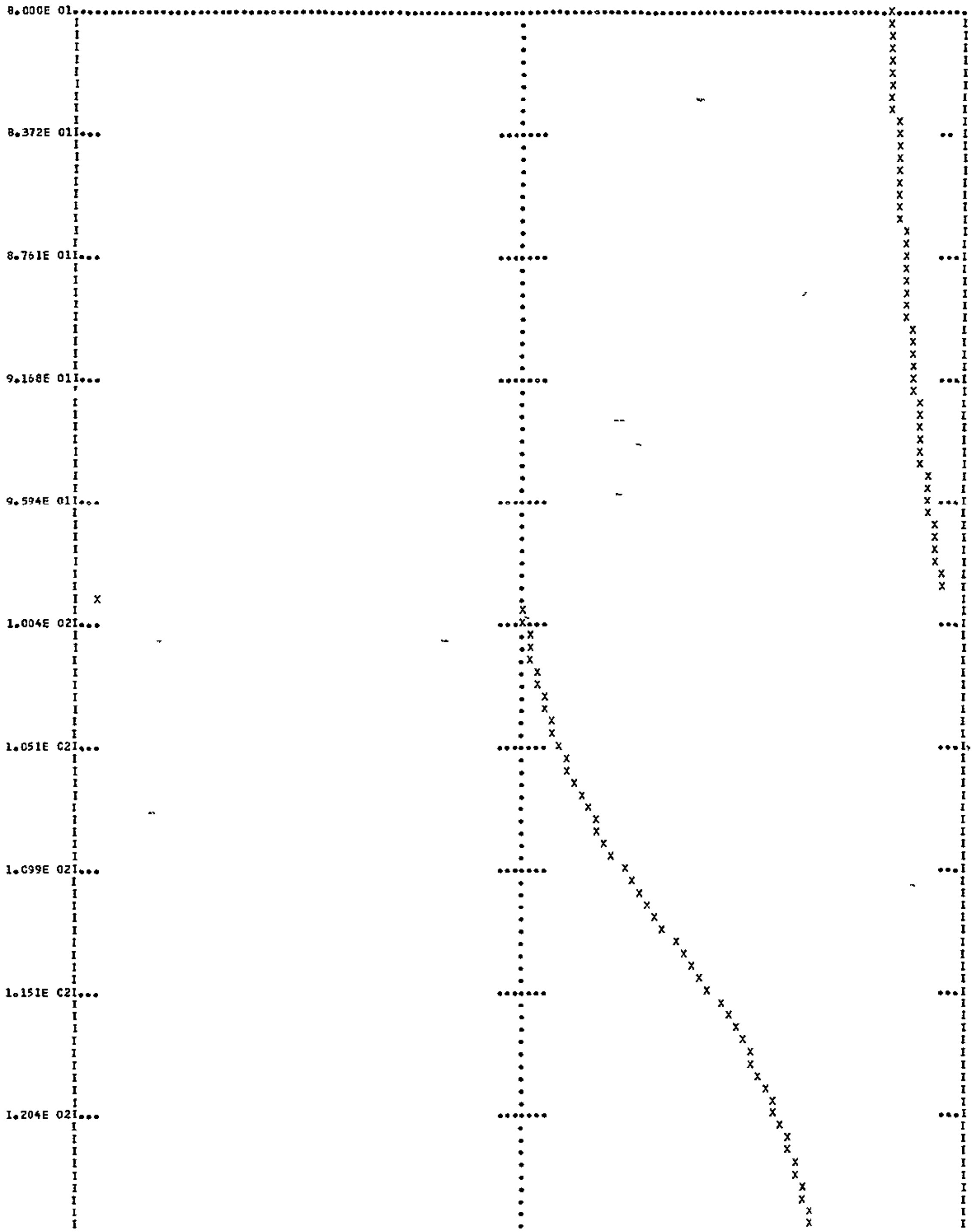
FREQ	MAC	FREQ	MAG	FREQ	MAG	FREQ	MAG
8.000E 01	9.93167E-01	8.962E 01	1.44792E 00	1.004E 02	1.15754E 01	1.125E 02	3.84091E-01
8.036E 01	1.00243E 00	9.003E 01	1.48684E 00	1.009E 02	7.05730E 00	1.130E 02	3.70437E-01
8.073E 01	1.01205E 00	9.044E 01	1.52939E 00	1.013E 02	4.99485E 00	1.135E 02	3.59235E-01
8.110E 01	1.02207E 00	9.085E 01	1.57612E 00	1.018E 02	3.81481E 00	1.140E 02	3.50224E-01
8.147E 01	1.03252E 00	9.126E 01	1.62768E 00	1.022E 02	3.05132E 00	1.145E 02	3.43154E-01
8.184E 01	1.04341E 00	9.168E 01	1.68485E 00	1.027E 02	2.51739E 00	1.151E 02	3.37800E-01
8.221E 01	1.05479E 00	9.210E 01	1.74865E 00	1.032E 02	2.12368E 00	1.156E 02	3.33950E-01
8.258E 01	1.06669E 00	9.251E 01	1.82026E 00	1.036E 02	1.82166E 00	1.161E 02	3.31406E-01
8.296E 01	1.07915E 00	9.294E 01	1.90125E 00	1.041E 02	1.58307E 00	1.166E 02	3.29995E-01
8.334E 01	1.09221E 00	9.336E 01	1.99256E 00	1.046E 02	1.39028E 00	1.172E 02	3.29552E-01
8.372E 01	1.10592E 00	9.378E 01	2.09980E 00	1.051E 02	1.23159E 00	1.177E 02	3.29936E-01
8.410E 01	1.12034E 00	9.421E 01	2.22333E 00	1.055E 02	1.09904E 00	1.182E 02	3.31020E-01
8.448E 01	1.13552E 00	9.464E 01	2.36873E 00	1.060E 02	9.87030E-01	1.188E 02	3.32689E-01
8.487E 01	1.15153E 00	9.507E 01	2.54243E 00	1.065E 02	8.91470E-01	1.193E 02	3.34848E-01
8.525E 01	1.16844E 00	9.550E 01	2.75355E 00	1.070E 02	8.09341E-01	1.199E 02	3.37412E-01
8.564E 01	1.18634E 00	9.594E 01	3.01558E 00	1.075E 02	7.38331E-01	1.204E 02	3.40307E-01
8.603E 01	1.20532E 00	9.637E 01	3.34939E 00	1.080E 02	6.76691E-01	1.209E 02	3.43469E-01
8.642E 01	1.22548E 00	9.681E 01	3.78919E 00	1.085E 02	6.23037E-01	1.215E 02	3.46845E-01
8.682E 01	1.24695E 00	9.725E 01	4.39469E 00	1.089E 02	5.76277E-01	1.220E 02	3.50391E-01
8.721E 01	1.26985E 00	9.770E 01	5.28114E 00	1.094E 02	5.35528E-01	1.226E 02	3.54067E-01
8.761E 01	1.29434E 00	9.814E 01	6.70283E 00	1.099E 02	5.00068E-01	1.232E 02	3.57839E-01
8.801E 01	1.32060E 00	9.859E 01	9.35157E 00	1.104E 02	4.69304E-01	1.237E 02	3.61681E-01
8.841E 01	1.34883E 00	9.904E 01	1.60225E 01	1.109E 02	4.42729E-01	1.243E 02	3.65569E-01
8.881E 01	1.37928E 00	9.949E 01	6.46326E 01	1.115E 02	4.19915E-01	1.249E 02	3.69483E-01
8.921E 01	1.41220E 00	9.994E 01	2.93580E 01	1.120E 02	4.00479E-01	1.254E 02	3.73409E-01

GREATEST VALUE = 6.46326E 01 LOWEST VALUE = 0.0 INTERVAL = 5.62022E-01



FREQ	PHA	FREQ	PHA	FREQ	PHA	FREQ	PHA
8.000E 01	1.56058E 02	8.962E 01	1.64618E 02	1.004E 02	1.67151E 00	1.125E 02	6.15355E 01
8.036E 01	1.56336E 02	9.003E 01	1.65063E 02	1.009E 02	2.84100E 00	1.130E 02	6.53606E 01
8.073E 01	1.56618E 02	9.044E 01	1.65519E 02	1.013E 02	4.05537E 00	1.135E 02	6.91708E 01
8.110E 01	1.56904E 02	9.085E 01	1.65988E 02	1.018E 02	5.32990E 00	1.140E 02	7.29337E 01
8.147E 01	1.57194E 02	9.126E 01	1.66471E 02	1.022E 02	6.67316E 00	1.145E 02	7.66230E 01
8.184E 01	1.57488E 02	9.168E 01	1.66967E 02	1.027E 02	8.09246E 00	1.151E 02	8.02135E 01
8.221E 01	1.57787E 02	9.210E 01	1.67478E 02	1.032E 02	9.59361E 00	1.156E 02	8.36835E 01
8.258E 01	1.58090E 02	9.251E 01	1.68005E 02	1.036E 02	1.11846E 01	1.161E 02	8.70195E 01
8.296E 01	1.58398E 02	9.294E 01	1.68549E 02	1.041E 02	1.28723E 01	1.166E 02	9.02080E 01
8.334E 01	1.58710E 02	9.336E 01	1.69111E 02	1.046E 02	1.46637E 01	1.172E 02	9.32433E 01
8.372E 01	1.59028E 02	9.378E 01	1.69691E 02	1.051E 02	1.65667E 01	1.177E 02	9.61214E 01
8.410E 01	1.59352E 02	9.421E 01	1.70292E 02	1.055E 02	1.85893E 01	1.182E 02	9.88437E 01
8.448E 01	1.59681E 02	9.464E 01	1.70914E 02	1.060E 02	2.07388E 01	1.188E 02	1.01412E 02
8.487E 01	1.60016E 02	9.507E 01	1.71559E 02	1.065E 02	2.30235E 01	1.193E 02	1.03832E 02
8.525E 01	1.60357E 02	9.550E 01	1.72228E 02	1.070E 02	2.54497E 01	1.199E 02	1.06109E 02
8.564E 01	1.60705E 02	9.594E 01	1.72924E 02	1.075E 02	2.80246E 01	1.204E 02	1.08251E 02
8.603E 01	1.61059E 02	9.637E 01	1.73648E 02	1.080E 02	3.07523E 01	1.209E 02	1.10264E 02
8.642E 01	1.61421E 02	9.681E 01	1.74402E 02	1.085E 02	3.36355E 01	1.215E 02	1.12157E 02
8.682E 01	1.61790E 02	9.725E 01	1.75189E 02	1.089E 02	3.66746E 01	1.220E 02	1.13938E 02
8.721E 01	1.62167E 02	9.770E 01	1.76012E 02	1.094E 02	3.98667E 01	1.226E 02	1.15614E 02
8.761E 01	1.62552E 02	9.814E 01	1.76876E 02	1.099E 02	4.32047E 01	1.232E 02	1.17192E 02
8.801E 01	1.62946E 02	9.859E 01	1.77786E 02	1.104E 02	4.66776E 01	1.237E 02	1.18679E 02
8.841E 01	1.63349E 02	9.904E 01	1.78766E 02	1.109E 02	5.02698E 01	1.243E 02	1.20083E 02
8.881E 01	1.63762E 02	9.949E 01	-1.79928E 02	1.115E 02	5.39604E 01	1.249E 02	1.21408E 02
8.921E 01	1.64185E 02	9.994E 01	4.69364E-01	1.120E 02	5.77256E 01	1.254E 02	1.22662E 02

GREATEST VALUE = 1.78766E 02 LOWEST VALUE = -1.79928E 02 INTERVAL = 3.11908E 00



PLOT(TY=SE/FR=80/TC=126/EL=L1)

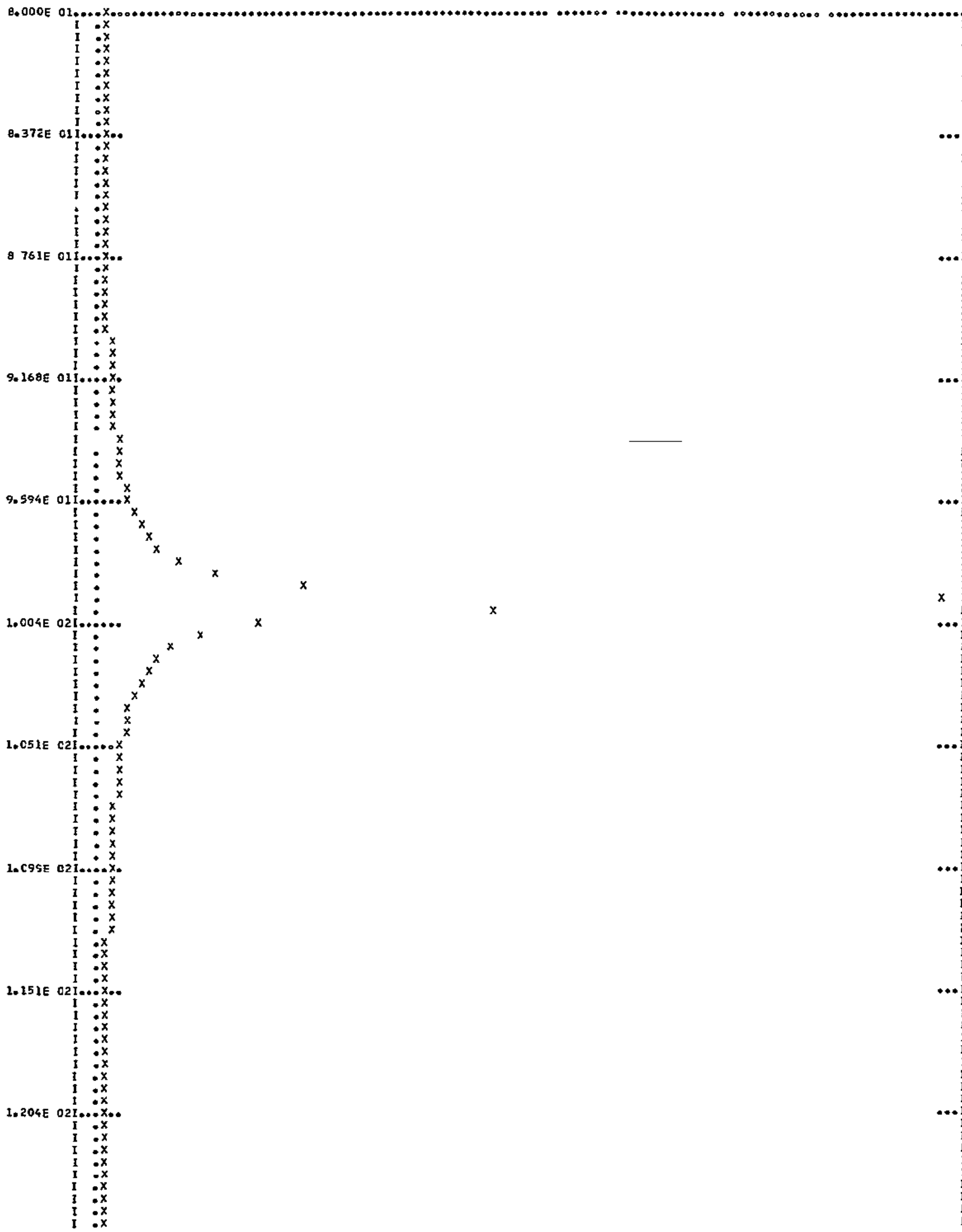
SENSITIVITY WITH RESPECT TO L1

THE TRANSFORM OF THE RESPONSE IS ---

EXP. OF S	NUMER. COEFFS.	DENOM. COEFFS.
0	0.0	3.05790E 03
1	0.0	1.57855E 02
2	9.59837E-03	4.51192E-01
3	4.03572E-04	-1.00514E-03
4	1.11326E-06	-2.98241E-06
5	-3.59256E-09	-3.59256E-09
6	-1.04976E-11	-1.04976E-11

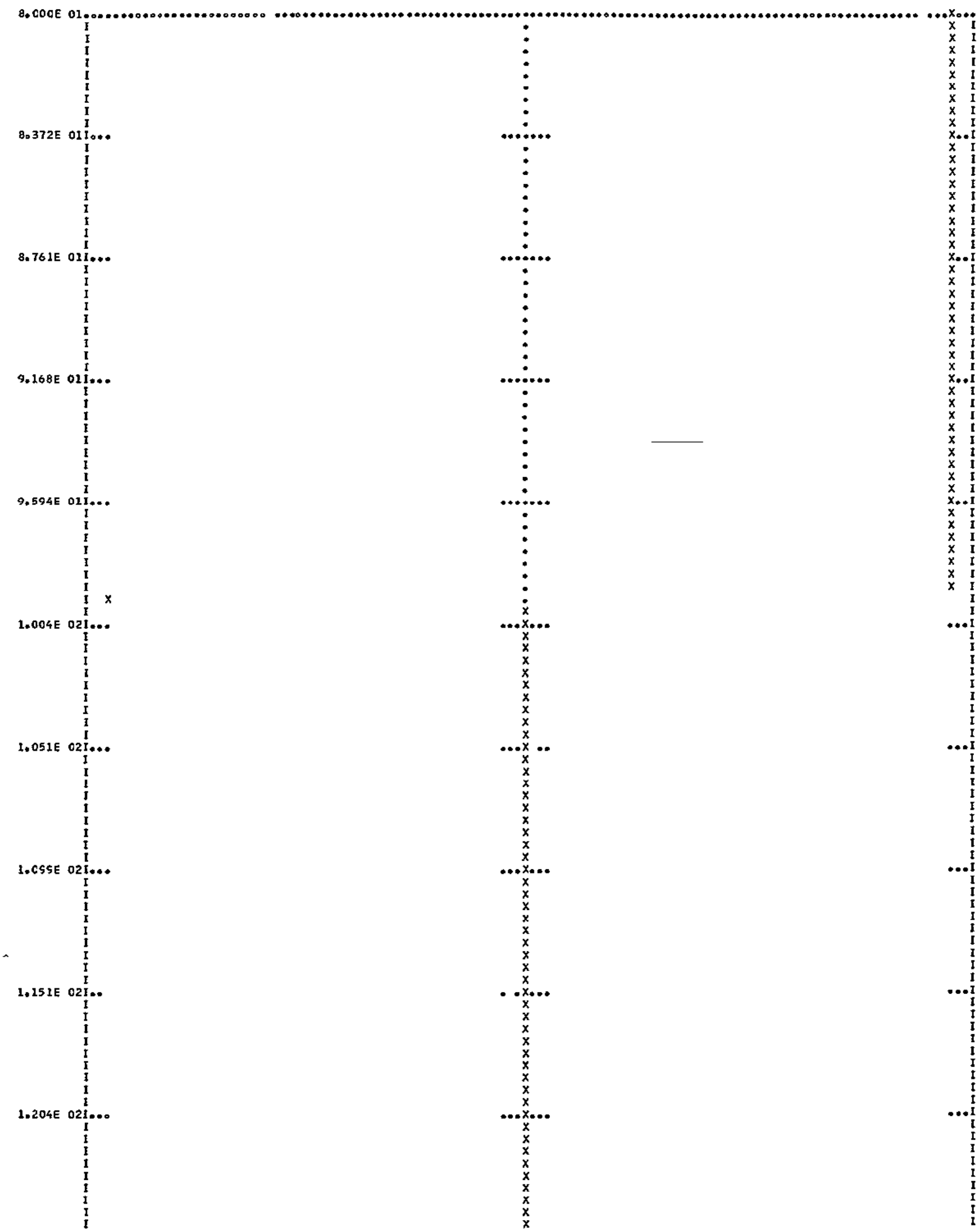
FREQ	MAG	FREQ	MAG	FREQ	MAG	FREQ	MAG
8.000E 01	1.80688E 00	8.962E 01	4.22268E 00	1.004E 02	6.57436E 01	1.125E 02	4.63346E 00
8.036E 01	1.85412E 00	9.003E 01	4.43279E 00	1.009E 02	4.14178E 01	1.130E 02	4.48566E 00
8.073E 01	1.90343E 00	9.044E 01	4.66267E 00	1.013E 02	3.03057E 01	1.135E 02	4.34820E 00
8.110E 01	1.95495E 00	9.085E 01	4.91523E 00	1.018E 02	2.39413E 01	1.140E 02	4.22009E 00
8.147E 01	2.00883E 00	9.126E 01	5.19404E 00	1.022E 02	1.98173E 01	1.145E 02	4.10035E 00
8.184E 01	2.06523E 00	9.168E 01	5.50332E 00	1.027E 02	1.69275E 01	1.151E 02	3.98822E 00
8.221E 01	2.12433E 00	9.210E 01	5.84846E 00	1.032E 02	1.47909E 01	1.156E 02	3.88302E 00
8.258E 01	2.18633E 00	9.251E 01	6.23596E 00	1.036E 02	1.31463E 01	1.161E 02	3.78408E 00
8.296E 01	2.25144E 00	9.294E 01	6.67418E 00	1.041E 02	1.18414E 01	1.166E 02	3.69091E 00
8.334E 01	2.31991E 00	9.336E 01	7.17365E 00	1.046E 02	1.07812E 01	1.172E 02	3.60298E 00
8.372E 01	2.39199E 00	9.378E 01	7.74836E 00	1.051E 02	9.90262E 00	1.177E 02	3.51992E 00
8.410E 01	2.46796E 00	9.421E 01	8.41649E 00	1.055E 02	9.16272E 00	1.182E 02	3.44129E 00
8.448E 01	2.54817E 00	9.464E 01	9.20265E 00	1.060E 02	8.53110E 00	1.188E 02	3.36676E 00
8.487E 01	2.63296E 00	9.507E 01	1.01415E 01	1.065E 02	7.98563E 00	1.193E 02	3.29603E 00
8.525E 01	2.72272E 00	9.550E 01	1.12821E 01	1.070E 02	7.50985E 00	1.199E 02	3.22882E 00
8.564E 01	2.81793E 00	9.594E 01	1.26971E 01	1.075E 02	7.09113E 00	1.204E 02	3.16485E 00
8.603E 01	2.91907E 00	9.637E 01	1.44988E 01	1.080E 02	6.71988E 00	1.209E 02	3.10392E 00
8.642E 01	3.02672E 00	9.681E 01	1.68714E 01	1.085E 02	6.38844E 00	1.215E 02	3.04583E 00
8.682E 01	3.14153E 00	9.725E 01	2.01362E 01	1.089E 02	6.09078E 00	1.220E 02	2.99036E 00
8.721E 01	3.26423E 00	9.770E 01	2.49135E 01	1.094E 02	5.82199E 00	1.226E 02	2.93736E 00
8.761E 01	3.39564E 00	9.814E 01	3.25717E 01	1.099E 02	5.57805E 00	1.232E 02	2.88666E 00
8.801E 01	3.53671E 00	9.859E 01	4.68339E 01	1.104E 02	5.35569E 00	1.237E 02	2.83811E 00
8.841E 01	3.68856E 00	9.904E 01	8.27423E 01	1.109E 02	5.15216E 00	1.243E 02	2.79160E 00
8.881E 01	3.85248E 00	9.949E 01	3.44344E 02	1.115E 02	4.96520E 00	1.249E 02	2.74698E 00
8.921E 01	4.02993E 00	9.994E 01	1.61451E 02	1.120E 02	4.79281E 00	1.254E 02	2.70415E 00

GREATEST VALUE = 3.44344E 02   LOWEST VALUE = 0.0   INTERVAL = 2.99430E 00



FREQ	PHA	FREQ	PHA	FREQ	PHA	FREQ	PHA
8.000E 01	1.79806E 02	8.962E 01	1.79856E 02	1.004E 02	-1.95429E-01	1.125E 02	-8.99090E-02
8.036E 01	1.79808E 02	9.003E 01	1.79858E 02	1.009E 02	-1.63555E-01	1.130E 02	-8.87521E-02
8.073E 01	1.79811E 02	9.044E 01	1.79860E 02	1.013E 02	-1.48256E-01	1.135E 02	-8.75768E-02
8.110E 01	1.79813E 02	9.085E 01	1.79862E 02	1.018E 02	-1.38999E-01	1.140E 02	-8.64330E-02
8.147E 01	1.79815E 02	9.126E 01	1.79863E 02	1.022E 02	-1.32524E-01	1.145E 02	-8.53314E-02
8.184E 01	1.79817E 02	9.168E 01	1.79865E 02	1.027E 02	-1.27623E-01	1.151E 02	-8.42231E-02
8.221E 01	1.79819E 02	9.210E 01	1.79867E 02	1.032E 02	-1.23769E-01	1.156E 02	-8.31641E-02
8.258E 01	1.79821E 02	9.251E 01	1.79870E 02	1.036E 02	-1.20480E-01	1.161E 02	-8.20624E-02
8.296E 01	1.79823E 02	9.294E 01	1.79872E 02	1.041E 02	-1.17607E-01	1.166E 02	-8.10410E-02
8.334E 01	1.79825E 02	9.336E 01	1.79874E 02	1.046E 02	-1.15117E-01	1.172E 02	-8.00000E-02
8.372E 01	1.79827E 02	9.378E 01	1.79876E 02	1.051E 02	-1.12834E-01	1.177E 02	-7.89902E-02
8.410E 01	1.79829E 02	9.421E 01	1.79878E 02	1.055E 02	-1.10650E-01	1.182E 02	-7.79770E-02
8.448E 01	1.79831E 02	9.464E 01	1.79881E 02	1.060E 02	-1.08804E-01	1.188E 02	-7.70124E-02
8.487E 01	1.79833E 02	9.507E 01	1.79883E 02	1.065E 02	-1.06949E-01	1.193E 02	-7.60295E-02
8.525E 01	1.79835E 02	9.550E 01	1.79886E 02	1.070E 02	-1.05274E-01	1.199E 02	-7.50936E-02
8.564E 01	1.79837E 02	9.594E 01	1.79890E 02	1.075E 02	-1.03586E-01	1.204E 02	-7.41185E-02
8.603E 01	1.79839E 02	9.637E 01	1.79893E 02	1.080E 02	-1.02031E-01	1.209E 02	-7.32120E-02
8.642E 01	1.79841E 02	9.681E 01	1.79898E 02	1.085E 02	-1.00549E-01	1.215E 02	-7.23273E-02
8.682E 01	1.79842E 02	9.725E 01	1.79903E 02	1.089E 02	-9.90711E-02	1.220E 02	-7.14133E-02
8.721E 01	1.79844E 02	9.770E 01	1.79910E 02	1.094E 02	-9.75906E-02	1.226E 02	-7.05210E-02
8.761E 01	1.79846E 02	9.814E 01	1.79921E 02	1.099E 02	-9.62740E-02	1.232E 02	-6.96591E-02
8.801E 01	1.79848E 02	9.859E 01	1.79941E 02	1.104E 02	-9.49544E-02	1.237E 02	-6.87708E-02
8.841E 01	1.79850E 02	9.904E 01	1.79987E 02	1.109E 02	-9.36335E-02	1.243E 02	-6.79095E-02
8.881E 01	1.79852E 02	9.949E 01	-1.79686E 02	1.115E 02	-9.23357E-02	1.249E 02	-6.70989E-02
8.921E 01	1.79854E 02	9.994E 01	-3.16724E-01	1.120E 02	-9.11362E-02	1.254E 02	-6.62280E-02

GREATEST VALUE = 1.75987E 02 LOWEST VALUE = -1.75686E 02 INTERVAL = 3.12759E 00



PLOT(TY=SE/FR=80/TO=126/EL=R3)

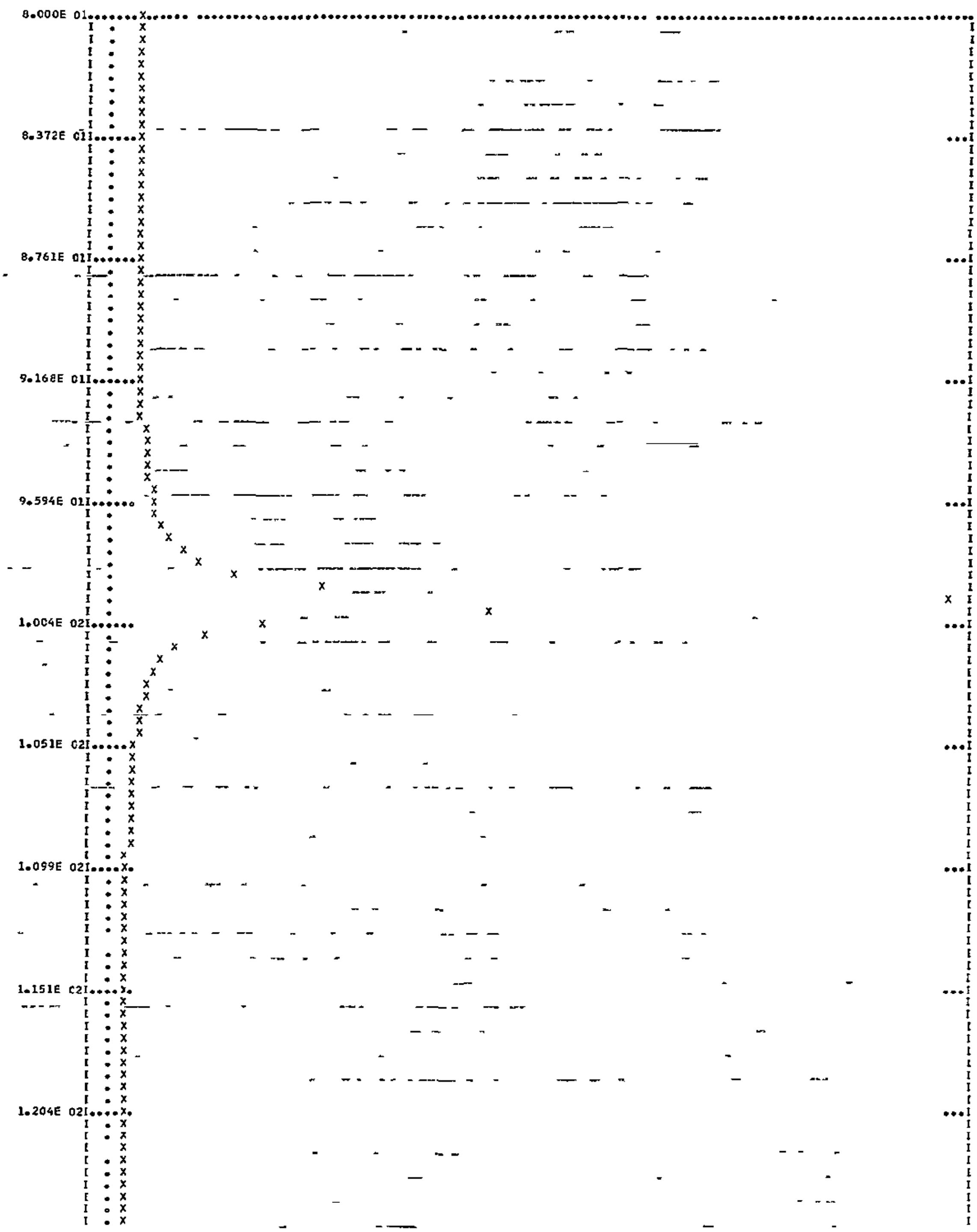
SENSITIVITY WITH RESPECT TO R3

THE TRANSFORM OF THE RESPONSE IS ---

EXP. OF S	NUMER. COEFFS.	DENOM. COEFFS.
0	4.28994E 01	3.05790E 03
1	1.21965E 02	1.57855E 02
2	3.17453E-01	4.51192E-01
3	3.41535E-04	-1.00514E-03
4	8.26703E-07	-2.98241E-06
5	2.52646E-10	-3.59256E-09
6	0.0	-1.04976E-11

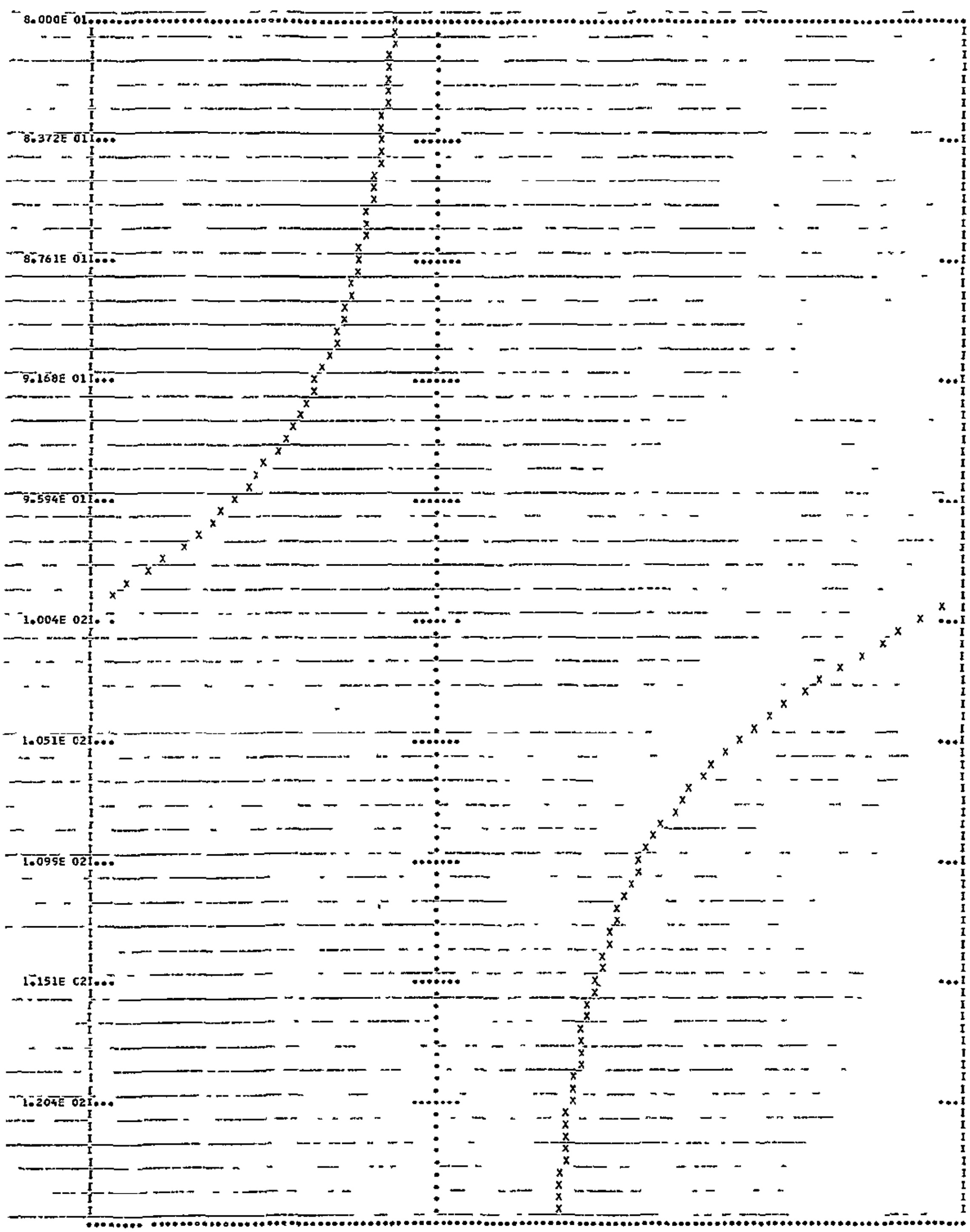
FREQ	MAG	FREQ	MAG	FREQ	MAG	FREQ	MAG
8.000E 01	2.35142E-01	8.962E 01	2.27433E-01	1.004E 02	1.17984E 00	1.125E 02	1.28336E-01
8.036E 01	2.34143E-01	9.003E 01	2.28767E-01	1.009E 02	7.31661E-01	1.130E 02	1.26757E-01
8.073E 01	2.33175E-01	9.044E 01	2.30404E-01	1.013E 02	5.30090E-01	1.135E 02	1.25270E-01
8.110E 01	2.32237E-01	9.085E 01	2.32391E-01	1.018E 02	4.17057E-01	1.140E 02	1.23862E-01
8.147E 01	2.31331E-01	9.126E 01	2.34789E-01	1.022E 02	3.45718E-01	1.145E 02	1.22523E-01
8.184E 01	2.30461E-01	9.168E 01	2.37670E-01	1.027E 02	2.97246E-01	1.151E 02	1.21243E-01
8.221E 01	2.29628E-01	9.210E 01	2.41128E-01	1.032E 02	2.62628E-01	1.156E 02	1.20017E-01
8.258E 01	2.28835E-01	9.251E 01	2.45276E-01	1.036E 02	2.36961E-01	1.161E 02	1.18837E-01
8.296E 01	2.28084E-01	9.294E 01	2.50262E-01	1.041E 02	2.17382E-01	1.166E 02	1.17699E-01
8.334E 01	2.27379E-01	9.336E 01	2.56275E-01	1.046E 02	2.02105E-01	1.172E 02	1.16597E-01
8.372E 01	2.26724E-01	9.378E 01	2.63563E-01	1.051E 02	1.89947E-01	1.177E 02	1.15528E-01
8.410E 01	2.26123E-01	9.421E 01	2.72458E-01	1.055E 02	1.80107E-01	1.182E 02	1.14488E-01
8.448E 01	2.25579E-01	9.464E 01	2.83410E-01	1.060E 02	1.72021E-01	1.188E 02	1.13475E-01
8.487E 01	2.25098E-01	9.507E 01	2.97054E-01	1.065E 02	1.65285E-01	1.193E 02	1.12485E-01
8.525E 01	2.24687E-01	9.550E 01	3.14295E-01	1.070E 02	1.59602E-01	1.199E 02	1.11518E-01
8.564E 01	2.24350E-01	9.594E 01	3.36482E-01	1.075E 02	1.54748E-01	1.204E 02	1.10570E-01
8.603E 01	2.24096E-01	9.637E 01	3.65701E-01	1.080E 02	1.50558E-01	1.209E 02	1.09641E-01
8.642E 01	2.23933E-01	9.681E 01	4.05383E-01	1.085E 02	1.46901E-01	1.215E 02	1.08729E-01
8.682E 01	2.23870E-01	9.725E 01	4.61521E-01	1.089E 02	1.43679E-01	1.220E 02	1.07832E-01
8.721E 01	2.23919E-01	9.770E 01	5.45696E-01	1.094E 02	1.40812E-01	1.226E 02	1.06949E-01
8.761E 01	2.24092E-01	9.814E 01	6.83450E-01	1.099E 02	1.38239E-01	1.232E 02	1.06080E-01
8.801E 01	2.24404E-01	9.859E 01	9.44290E-01	1.104E 02	1.35909E-01	1.237E 02	1.05223E-01
8.841E 01	2.24872E-01	9.904E 01	1.60900E 00	1.109E 02	1.33783E-01	1.243E 02	1.04379E-01
8.881E 01	2.25516E-01	9.949E 01	6.48621E 00	1.115E 02	1.31829E-01	1.249E 02	1.03545E-01
8.921E 01	2.26361E-01	9.994E 01	2.96041E 00	1.120E 02	1.30020E-01	1.254E 02	1.02721E-01

GREATEST VALUE = 6.48621E 00 LOWEST VALUE = 0.0 INTERVAL = 5.64018E-02



FREQ	PHA	FREQ	PHA	FREQ	PHA	FREQ	PHA
8.000E 01	-8.76958E 00	8.962E 01	-1.98211E 01	1.004E 02	1.03004E 02	1.125E 02	3.81297E 01
8.036E 01	-9.00115E 00	9.003E 01	-2.06726E 01	1.009E 02	9.87059E 01	1.130E 02	3.72148E 01
8.073E 01	-9.24275E 00	9.044E 01	-2.15820E 01	1.013E 02	9.43829E 01	1.135E 02	3.63599E 01
8.110E 01	-9.49504E 00	9.085E 01	-2.25546E 01	1.018E 02	9.00972E 01	1.140E 02	3.55607E 01
8.147E 01	-9.75857E 00	9.126E 01	-2.35961E 01	1.022E 02	8.58995E 01	1.145E 02	3.48122E 01
8.184E 01	-1.00340E 01	9.168E 01	-2.47129E 01	1.027E 02	8.18335E 01	1.151E 02	3.41107E 01
8.221E 01	-1.03221E 01	9.210E 01	-2.59124E 01	1.032E 02	7.79355E 01	1.156E 02	3.34525E 01
8.258E 01	-1.06238E 01	9.251E 01	-2.72020E 01	1.036E 02	7.42290E 01	1.161E 02	3.28342E 01
8.296E 01	-1.09397E 01	9.294E 01	-2.85906E 01	1.041E 02	7.07314E 01	1.166E 02	3.22530E 01
8.334E 01	-1.12710E 01	9.336E 01	-3.00872E 01	1.046E 02	6.74517E 01	1.172E 02	3.17059E 01
8.372E 01	-1.16184E 01	9.378E 01	-3.17028E 01	1.051E 02	6.43907E 01	1.177E 02	3.11908E 01
8.410E 01	-1.19833E 01	9.421E 01	-3.34477E 01	1.055E 02	6.15445E 01	1.182E 02	3.07051E 01
8.448E 01	-1.23667E 01	9.464E 01	-3.53334E 01	1.060E 02	5.89052E 01	1.188E 02	3.02470E 01
8.487E 01	-1.27699E 01	9.507E 01	-3.73732E 01	1.065E 02	5.64624E 01	1.193E 02	2.98144E 01
8.525E 01	-1.31942E 01	9.550E 01	-3.95795E 01	1.070E 02	5.42041E 01	1.199E 02	2.94058E 01
8.564E 01	-1.36413E 01	9.594E 01	-4.19652E 01	1.075E 02	5.21171E 01	1.204E 02	2.90194E 01
8.603E 01	-1.41128E 01	9.637E 01	-4.45419E 01	1.080E 02	5.01887E 01	1.209E 02	2.86541E 01
8.642E 01	-1.46105E 01	9.681E 01	-4.73214E 01	1.085E 02	4.84066E 01	1.215E 02	2.83083E 01
8.682E 01	-1.51366E 01	9.725E 01	-5.03114E 01	1.089E 02	4.67587E 01	1.220E 02	2.79808E 01
8.721E 01	-1.56928E 01	9.770E 01	-5.35174E 01	1.094E 02	4.52336E 01	1.226E 02	2.76707E 01
8.761E 01	-1.62820E 01	9.814E 01	-5.69377E 01	1.099E 02	4.38207E 01	1.232E 02	2.73767E 01
8.801E 01	-1.69064E 01	9.859E 01	-6.05619E 01	1.104E 02	4.25105E 01	1.237E 02	2.70980E 01
8.841E 01	-1.75693E 01	9.904E 01	-6.43584E 01	1.109E 02	4.12939E 01	1.243E 02	2.68337E 01
8.881E 01	-1.82736E 01	9.949E 01	-6.80513E 01	1.115E 02	4.01631E 01	1.249E 02	2.65829E 01
8.921E 01	-1.90228E 01	9.994E 01	1.07154E 02	1.120E 02	3.91104E 01	1.254E 02	2.63450E 01

GREATEST VALUE = 1.07154E 02 LOWEST VALUE = -6.80513E 01 INTERVAL = 1.52352E 00



END

```
****NASAP****

   NETWORK ANALYSIS ANC SYSTEMS APPLICATICN PRCGRAM

THIS VERSION WAS DEVELOPED AT UCLA ENGR. CEPT.


TREE   NASAP PROBLEM TELEMETER
J1(1-2)=1.0
CJ1(1-2)=100MF
        $$$$ UNITS ARE MF
RE1(1-2)=20K
        $$$$ UNITS ARE K
CJ2(2-3)=2MF
        $$$$ UNITS ARE MF
RJ2(2-3)=1C0
RE3(1-3)=220
LE1(4-1)=C.25
CJ3(2-4)=2.0MF
        $$$$ UNITS ARE MF
CJ4(3-1)=47MF
        $$$$ UNITS ARE MF
CJ5(3-4)=10MF
        $$$$ UNITS ARE MF
DJ2/IRJ2(4-3)=C.3
END
```

ELEMENT NUMBER	ELEMENT NAME	DEPENDENCY (IF ANY)	ORIGIN NODE	TARGET NODE	VALUE	TAG	GENER
1	J1		1	2	1.00000000E 00	0	1
2	CJ1		1	2	9.99999320E-05	0	0
3	RE1		1	2	2.00000000E 04	1	0
4	CJ2		2	3	1.99999886E-06	0	0
5	RJ2		2	3	1.00000000E 02	0	0
6	RE3		1	3	2.20000000E 02	1	0
7	LE1		4	1	2.49999881E-01	1	0
8	CJ3		2	4	1.99999886E-C6	0	0
9	CJ4		3	1	4.69999650E-C5	0	0
10	CJ5		3	4	9.99999429E-06	0	0
11	DJ2	IRJ2	4	3	2.99999893E-01	0	1

2	3	304	6
3	6	1536	2096
4	-7	2C48	1280

WORSTCASE

ELEMENT NAME	TOLERANCE
CJ1	0.1000
RE1	0.1000
CJ2	0.1000
RJ2	0.1000
RE3	0.1000
LE1	0.1000
CJ3	0.1000
CJ4	0.1000
CJ5	0.1000
DJ2	0.1000

FLOWGRAPH

FROM	TO	S	VALUE
13	2	1	9.99999320E-05
3	14	0	2.00000000E 04
15	4	1	1.99999886E-06
16	5	0	9.99999791E-03
6	17	0	2.20000000E 02
7	18	1	2.49999881E-01
19	8	1	1.99999886E-06
20	9	1	4.69999650E-05
21	10	1	9.99999429E-06
5	11	0	2.99999893E-01
4	3	0	1.00000000E 00
14	15	0	-1.00000000E 00
5	3	0	1.00000000E 00
14	16	0	-1.00000000E 00
8	3	0	1.00000000E 00
14	19	0	-1.00000000E 00
1	3	0	-1.00000000E 00
14	12	0	-1.00000000E 00
2	3	0	-1.00000000E 00
14	13	0	1.00000000E 00
9	6	0	1.00000000E 00
17	20	0	-1.00000000E 00
10	6	0	1.00000000E 00
17	21	0	-1.00000000E 00
4	6	0	-1.00000000E 00
17	15	0	1.00000000E 00
5	6	0	-1.00000000E 00
17	16	0	1.00000000E 00
11	6	0	-1.00000000E 00
17	22	0	-1.00000000E 00
8	7	0	1.00000000E 00
18	19	0	-1.00000000E 00
10	7	0	1.00000000E 00
18	21	0	-1.00000000E 00
11	7	0	-1.00000000E 00
18	22	0	-1.00000000E 00

ILE1/IJ1

NTIMES= 1

THE UNKNOWN TRANSMITTANCE GOES FROM NODE 7 TO 1

24588	6001
1180224	288001
3065304	748004
3098088	756003
786816	192002
3065304	748004
3098088	756003
3100136	1780003
2360448	576002
2608376	1660003
870824	1236002
2559200	1648002
1034744	1276003
247928	1084001
2313450	1589001
84138	1045000
248058	1085001
198752	1072000
49176	12001
245880	60001
245880	60001
2278618	557002
81960	20000
2311402	565001
540936	132001
541066	133001
2229312	544001
163920	40001
196704	48000

```
NO. OF FIRST ORDER LOOPS= 29
                    1
                    2
                    3
                    4
                    5
                    6
                    7
                    8
                    9
                   10
                   11
                   12
                   13
                   14
                   15
                   16
                   17
                   18
                   19
                   20
                   21
                   22
                   23
                   24
                   25
                   26
                   27
                   28
                   29
```

```
TRANSFER  FUNCTION


                                          2
 9.881E 04   + 4.282E 01S  - 1.000E 00S  + 0.0          + 0.0

--------------------------------------------------------------------

                                          2               3              4
 2.948E 09   + 1.509E 08S  + 3.994E 05S  + 3.650E 02S  + 1.000E 00S

THE FUNCTION FACTOR =   -8.780E 03
```

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2.03860E 02	1.04354E 01	2.76207E-02	2.66273E-05	6.91544E-08	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	6.00000E 01	2.59596E-02	-6.07199E-04	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

SENSITIVITY TO CJ1

$$\frac{0.0 - 1.141E\ 13S - 4.330E\ 10S^{2} + 6.947E\ 07S^{3} + 2.766E\ 05S^{4} + 2.541E\ 02S^{5} + 1.000E\ 00S^{6}}{-\ 2.913E\ 14 - 1.504E\ 13S - 4.298E\ 10S^{2} + 9.575E\ 07S^{3} + 2.841E\ 05S^{4} + 3.422E\ 02S^{5} + 1.000E\ 00S^{6}}$$

THE FUNCTION FACTOR = 9.671E-01

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.22316E 04	6.31421E 02	1.80477E 00	-4.02057E-03	-1.19296E-05	-1.43701E-08	-4.19904E-11	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	4.63198E 02	1.75831E 00	-2.82103E-03	-1.12337E-05	-1.03200E-08	-4.06092E-11	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

SENSITIVITY TO RE1

$$\frac{1.154E\ 08 + 3.882E\ 05S + 2.969E\ 02S^{2} + 1.000E\ 00S^{3} + 0.0}{2.948E\ 09 + 1.509E\ 08S + 3.994E\ 05S^{2} + 3.850E\ 02S^{3} + 1.000E\ 00S^{4}}$$

THE FUNCTION FACTOR = -4.836E-01

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2.03860E 02	1.04354E 01	2.76207E-02	2.66273E-05	6.91544E-08	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	-3.86000E 00	-1.29800E-02	-5.92998E-06	-3.34399E-08	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

SENSITIVITY TO CJ2

$$\frac{0.0 + 4.905E\ 11S + 2.154E\ 10S^{2} + 2.266E\ 08S^{3} + 4.656E\ 05S^{4} + 5.740E\ 02S^{5} + 1.000E\ 00S^{6}}{-\ 2.913E\ 14 - 1.504E\ 13S - 4.298E\ 10S^{2} + 9.575E\ 07S^{3} + 2.841E\ 05S^{4} + 3.422E\ 02S^{5} + 1.000E\ 00S^{6}}$$

THE FUNCTION FACTOR = -1.178E-01

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.22316E 04	6.31421E 02	1.80477E 00	-4.02057E-03	-1.15296E-05	-1.43701E-08	-4.19904E-11	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	2.42627E 00	1.06530E-01	1.12107E-03	2.30314E-06	2.83914E-09	4.94632E-12	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

SENSITIVITY TO RJ2

$$\frac{3.543E\ 09 + 7.898E\ 09 S + 1.307E\ 08 S^{2} + 4.279E\ 05 S^{3} + 3.403E\ 02 S^{4} + 1.000E\ 00 S^{5} + 0.0}{-\ 2.913E\ 14 - 1.504E\ 13 S - 4.298E\ 10 S^{2} + 9.575E\ 07 S^{3} + 2.841E\ 05 S^{4} + 3.422E\ 02 S^{5} + 1.000E\ 00 S^{6}}$$

THE FUNCTION FACTOR = 4.033E 02

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.22316E 04	6.31421E 02	1.80477E 00	-4.02057E-03	-1.19296E-05	-1.43701E-08	-4.19904E-11	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	-6.00000E 01	-1.33759E 02	-2.21403E 00	-7.24708E-03	-5.76366E-06	-1.69352E-08	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

SENSITIVITY TO RE3

$$\frac{1.698E\ 11 + 4.828E\ 11S + 1.257E\ 09S^{2} + 1.352E\ 06S^{3} + 3.272E\ 03S^{4} + 1.000E\ 00S^{5} + 0.0}{-\ 2.913E\ 14 - 1.504E\ 13S - 4.298E\ 10S^{2} + 9.575E\ 07S^{3} + 2.841E\ 05S^{4} + 3.422E\ 02S^{5} + 1.000E\ 00S^{6}}$$

THE FUNCTION FACTOR = -2.407E 01

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.22316E 04	6.31421E 02	1.80477E 00	-4.02057E-03	-1.19296E-05	-1.43701E-08	-4.19904E-11	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	1.71602E 02	4.87941E 02	1.26981E 00	1.36613E-03	3.30681E-06	1.01059E-09	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

SENSITIVITY TC LE1

$$\frac{0.0 + 0.0 - 9.144E\,08S^{2} - 3.844E\,07S^{3} - 1.060E\,05S^{4} + 3.422E\,02S^{5} + 1.000E\,00S^{6}}{-2.913E\,14 - 1.504E\,13S - 4.298E\,10S^{2} + 9.575E\,07S^{3} + 2.841E\,05S^{4} + 3.422E\,02S^{5} + 1.000E\,00S^{6}}$$

THE FUNCTICN FACTOR = 1.000E 00

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.22316E 04	6.31421E 02	1.80477E 00	-4.02057E-03	-1.19296E-05	-1.43701E-08	-4.19904E-11	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	3.83959E-02	1.61428E-03	4.45302E-06	-1.43701E-08	-4.19904E-11	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

SENSITIVITY TO CJ3

$$\frac{0.0 + 1.494E\ 12S + 6.454E\ 10S^{2} + 3.640E\ 08S^{3} + 6.985E\ 05S^{4} + 7.934E\ 02S^{5} + 1.000E\ 00S^{6}}{-\ 2.913E\ 14 - 1.504E\ 13S - 4.298E\ 10S^{2} + 9.575E\ 07S^{3} + 2.841E\ 05S^{4} + 3.422E\ 02S^{5} + 1.000E\ 00S^{6}}$$

THE FUNCTION FACTOR = -6.494E-01

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.22316E 04	6.31421E 02	1.80477E 00	-4.02057E-03	-1.19296E-05	-1.43701E-08	-4.19904E-11	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	4.07400E 01	1.76014E 00	9.92565E-03	1.90483E-05	2.16353E-08	2.72701E-11	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

SENSITIVITY TO CJ4

$$\frac{0.0 \quad + 1.698E\ 11S \ + 4.828E\ 11S^{2} + 1.257E\ 09S^{3} + 1.352E\ 06S^{4} + 3.272E\ 03S^{5} + 1.000E\ 00S^{6}}{- 2.913E\ 14 \quad - 1.504E\ 13S \ - 4.298E\ 10S^{2} + 9.575E\ 07S^{3} + 2.841E\ 05S^{4} + 3.422E\ 02S^{5} + 1.000E\ 00S^{6}}$$

THE FUNCTION FACTOR = 2.489E-01

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.22316E 04	6.31421E 02	1.80477E 00	-4.02057E-03	-1.19296E-05	-1.43701E-08	-4.19904E-11	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	-1.77417E 00	-5.04530E 00	-1.31298E-02	-1.41258E-05	-3.41923E-08	-1.04494E-11	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0							

SENSITIVITY TO CJ5

$$\frac{0.0 \quad - 5.021E\ 12S - 2.136E\ 11S^{2} - 6.539E\ 08S^{3} - 7.576E\ 05S^{4} - 1.021E\ 03S^{5} + 1.000E\ 00S^{6}}{- 2.913E\ 14 \quad - 1.504E\ 13S - 4.298E\ 10S^{2} + 9.575E\ 07S^{3} + 2.841E\ 05S^{4} + 3.422E\ 02S^{5} + 1.000E\ 00S^{6}}$$

THE FUNCTION FACTOR = 5.513E-01

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.22316E 04	6.31421E 02	1.80477E 00	-4.02057E-03	-1.19296E-05	-1.43701E-08	-4.19904E-11	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.16230E 02	4.94397E 00	1.51372E-02	1.75366E-05	2.36339E-08	-2.31482E-11	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

SENSITIVITY TO DJ2

$$\frac{2.926E\ 11 + 1.599E\ 10S + 1.877E\ 08S^{2} + 4.503E\ 05S^{3} + 4.790E\ 02S^{4} + 1.000E\ 00S^{5} + 0.0}{-\ 2.913E\ 14 - 1.504E\ 13S - 4.298E\ 10S^{2} + 9.575E\ 07S^{3} + 2.841E\ 05S^{4} + 3.422E\ 02S^{5} + 1.000E\ 00S^{6}}$$

THE FUNCTION FACTOR = 9.923E 02

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.22316E 04	6.31421E 02	1.80477E 00	-4.02057E-03	-1.19296E-05	-1.43701E-08	-4.19904E-11	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	-1.21920E 04	-6.66412E 02	-7.81940E 00	-1.87628E-02	-1.99590E-05	-4.16667E-08	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

SQUARE OF WORST CASE TOLERANCE

$3.082E\ 28 + 3.406E\ 27S + 2.323E\ 26S^{2} + 3.244E\ 24S^{3} + 3.197E\ 22S^{4} + 1.376E\ 20S^{5} + 3.456E\ 17S^{6} + 7.276E\ 14S^{7}$

$+ 1.070E\ 12S^{8} + 9.990E\ 08S^{9} + 1.166E\ 06S^{10} + 5.955E\ 02S^{11} + 1.000E\ 00S^{12}$

$8.485E\ 28 + 8.761E\ 27S + 2.512E\ 26S^{2} + 1.237E\ 24S^{3} - 1.198E\ 21S^{4} - 1.697E\ 19S^{5} - 2.613E\ 16S^{6} - 5.087E\ 12S^{7}$

$+ 6.029E\ 10S^{8} + 3.860E\ 08S^{9} + 6.853E\ 05S^{10} + 6.844E\ 02S^{11} + 1.000E\ 00S^{12}$

THE FUNCTION FACTOR = 2.737E-02

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.49612E 08	1.54466E 07	4.42843E 05	2.18078E 03	-2.11199E 00	-2.99292E-02	-4.60699E-05	-8.96879E-09	1.06302E-10	6.80511E-13
1.20836E-15	1.20681E-18	1.76320E-21	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	1.48731E 06	1.64337E 05	1.12118E 04	1.56522E 02	1.54292E 00	6.64206E-03	1.66755E-05	3.51099E-08	5.16418E-11
4.82084E-14	5.62418E-17	2.87348E-20	4.82545E-23	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

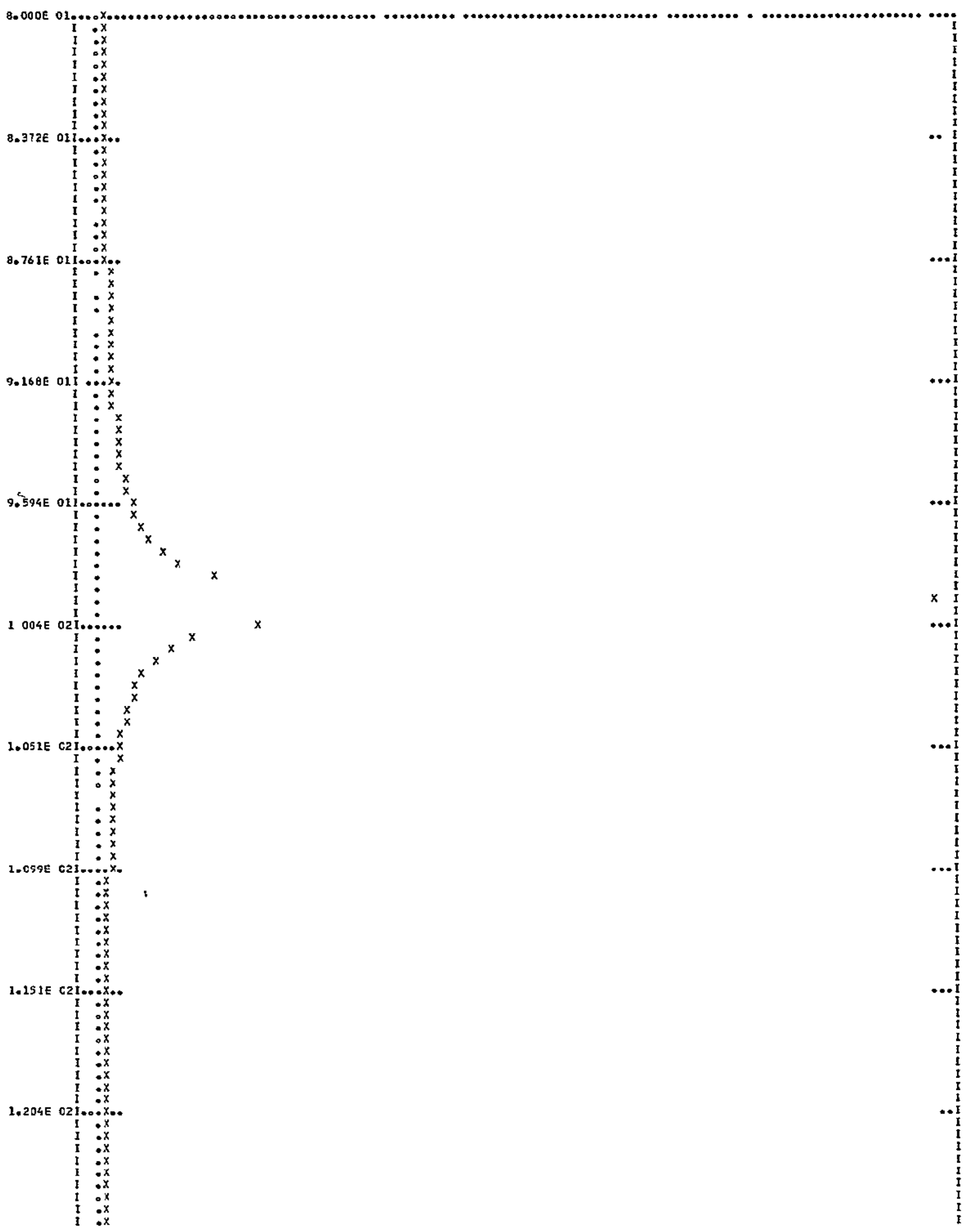
PLOT(TY=FR/FR=80/TO=126)

THE TRANSFORM OF THE RESPONSE IS ---

EXP. OF S	NUMER. COEFFS.	DENOM. COEFFS.
0	-6.00000E 01	2.03860E 02
1	-2.59996E-02	1.04354E 01
2	6.07199E-04	2.76207E-02
3	0.0	2.66273E-05
4	0.0	6.91544E-08

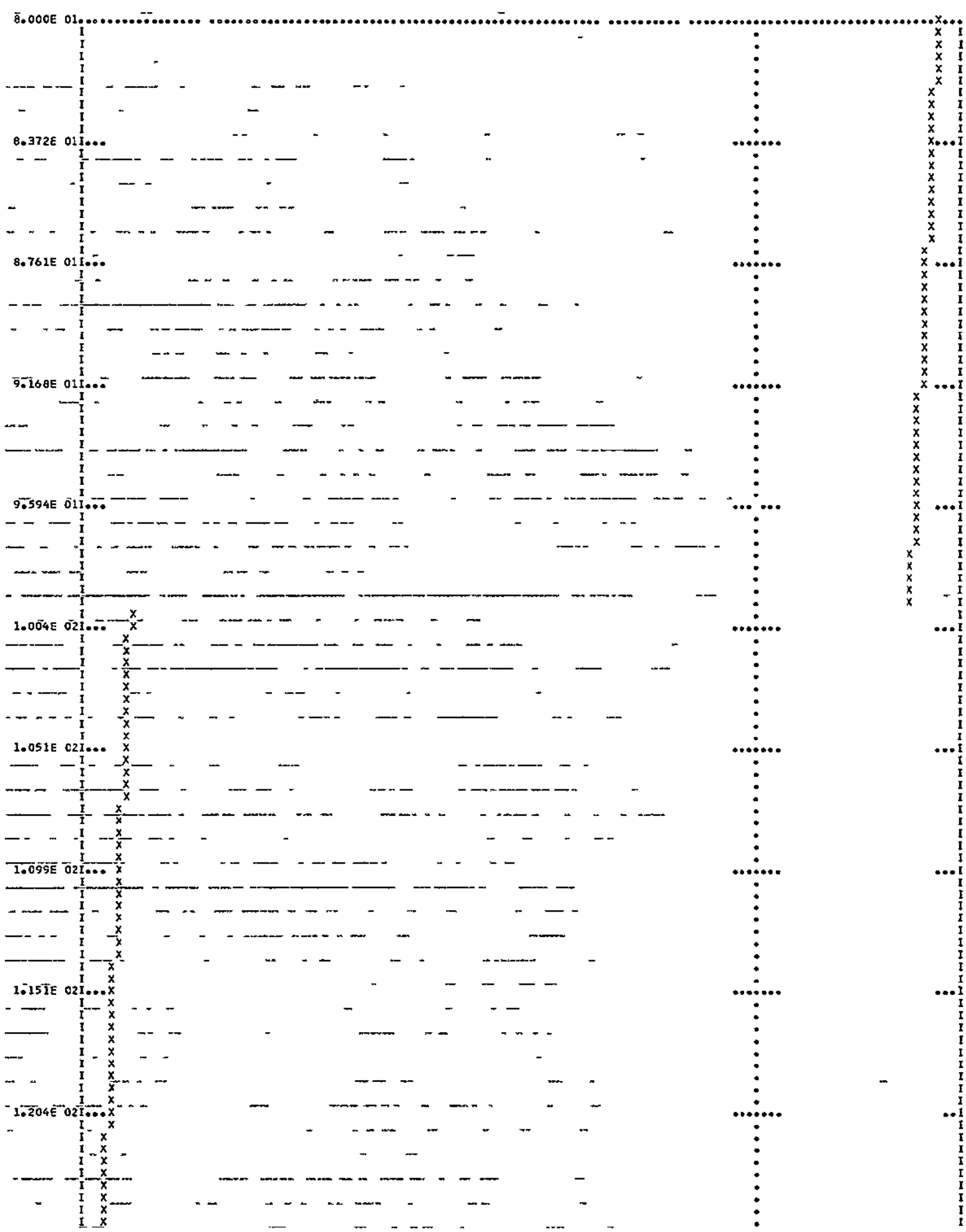
FREQ	MAG	FREQ	MAG	FREQ	MAG	FREQ	MAG
8.000E 01	7.11005E-02	8.962E 01	1.29373E-01	1.004E 02	1.57508E 00	1.125E 02	8.70464E-02
8.036E 01	7.22266E-02	9.003E 01	1.34469E-01	1.009E 02	9.82515E-01	1.130E 02	8.34607E-02
8.073E 01	7.34032E-02	9.044E 01	1.40046E-01	1.013E 02	7.11886E-01	1.135E 02	8.01263E-02
8.110E 01	7.46340E-02	9.085E 01	1.46176E-01	1.018E 02	5.56901E-01	1.140E 02	7.70192E-02
8.147E 01	7.59224E-02	9.126E 01	1.52945E-01	1.022E 02	4.56486E-01	1.145E 02	7.41163E-02
8.184E 01	7.72722E-02	9.168E 01	1.60456E-01	1.027E 02	3.86130E-01	1.151E 02	7.13983E-02
8.221E 01	7.86883E-02	9.210E 01	1.68840E-01	1.032E 02	3.34116E-01	1.156E 02	6.88488E-02
8.258E 01	8.01747E-02	9.251E 01	1.78256E-01	1.036E 02	2.94081E-01	1.161E 02	6.64518E-02
8.296E 01	8.17375E-02	9.294E 01	1.88906E-01	1.041E 02	2.62323E-01	1.166E 02	6.41952E-02
8.334E 01	8.33818E-02	9.336E 01	2.01048E-01	1.046E 02	2.36522E-01	1.172E 02	6.20662E-02
8.372E 01	8.51145E-02	9.378E 01	2.15022E-01	1.051E 02	2.15142E-01	1.177E 02	6.00553E-02
8.410E 01	8.69422E-02	9.421E 01	2.31270E-01	1.055E 02	1.97140E-01	1.182E 02	5.81523E-02
8.448E 01	8.88730E-02	9.464E 01	2.50392E-01	1.060E 02	1.81775E-01	1.188E 02	5.63493E-02
8.487E 01	9.09156E-02	9.507E 01	2.73230E-01	1.065E 02	1.68507E-01	1.193E 02	5.46385E-02
8.525E 01	9.30794E-02	9.550E 01	3.00980E-01	1.070E 02	1.56937E-01	1.199E 02	5.30134E-02
8.564E 01	9.53760E-02	9.594E 01	3.35411E-01	1.075E 02	1.46755E-01	1.204E 02	5.14670E-02
8.603E 01	9.78175E-02	9.637E 01	3.79254E-01	1.080E 02	1.37729E-01	1.209E 02	4.99946E-02
8.642E 01	1.00417E-01	9.681E 01	4.36996E-01	1.085E 02	1.29673E-01	1.215E 02	4.85912E-02
8.682E 01	1.03192E-01	9.725E 01	5.16454E-01	1.089E 02	1.22439E-01	1.220E 02	4.72514E-02
8.721E 01	1.06158E-01	9.770E 01	6.32732E-01	1.094E 02	1.15907E-01	1.226E 02	4.59717E-02
8.761E 01	1.09337E-01	9.814E 01	8.19125E-01	1.099E 02	1.09981E-01	1.232E 02	4.47480E-02
8.801E 01	1.12751E-01	9.859E 01	1.16621E 00	1.104E 02	1.04580E-01	1.237E 02	4.35766E-02
8.841E 01	1.16428E-01	9.904E 01	2.03995E 00	1.109E 02	9.96376E-02	1.243E 02	4.24546E-02
8.881E 01	1.20399E-01	9.949E 01	8.39652E 00	1.115E 02	9.50984E-02	1.249E 02	4.13787E-02
8.921E 01	1.24700E-01	9.994E 01	3.90782E 00	1.120E 02	9.09138E-02	1.254E 02	4.03461E-02

GREATEST VALUE = 8.39652E 00   LOWEST VALUE = 0.0   INTERVAL = 7.30132E-02



FREQ	PHA	FREQ	PHA	FREQ	PHA	FREQ	PHA
8.000E 01	4.18002E 01	8.962E 01	3.83331E 01	1.004E 02	-1.45088E 02	1.125E 02	-1.48157E 02
8.036E 01	4.16591E 01	9.003E 01	3.81974E 01	1.009E 02	-1.45184E 02	1.130E 02	-1.48279E 02
8.073E 01	4.15181E 01	9.044E 01	3.80619E 01	1.013E 02	-1.45298E 02	1.135E 02	-1.48401E 02
8.110E 01	4.13773E 01	9.085E 01	3.79268E 01	1.018E 02	-1.45419E 02	1.140E 02	-1.48522E 02
8.147E 01	4.12367E 01	9.126E 01	3.77918E 01	1.022E 02	-1.45542E 02	1.145E 02	-1.48644E 02
8.184E 01	4.10964E 01	9.168E 01	3.76572E 01	1.027E 02	-1.45666E 02	1.151E 02	-1.48764E 02
8.221E 01	4.09562E 01	9.210E 01	3.75229E 01	1.032E 02	-1.45791E 02	1.156E 02	-1.48885E 02
8.258E 01	4.08163E 01	9.251E 01	3.73889E 01	1.036E 02	-1.45917E 02	1.161E 02	-1.49005E 02
8.296E 01	4.06764E 01	9.294E 01	3.72552E 01	1.041E 02	-1.46043E 02	1.166E 02	-1.49125E 02
8.334E 01	4.05368E 01	9.336E 01	3.71219E 01	1.046E 02	-1.46169E 02	1.172E 02	-1.49245E 02
8.372E 01	4.03975E 01	9.378E 01	3.69887E 01	1.051E 02	-1.46295E 02	1.177E 02	-1.49364E 02
8.410E 01	4.02583E 01	9.421E 01	3.68561E 01	1.055E 02	-1.46420E 02	1.182E 02	-1.49483E 02
8.448E 01	4.01193E 01	9.464E 01	3.67239E 01	1.060E 02	-1.46546E 02	1.188E 02	-1.49602E 02
8.487E 01	3.99805E 01	9.507E 01	3.65921E 01	1.065E 02	-1.46671E 02	1.193E 02	-1.49720E 02
8.525E 01	3.98420E 01	9.550E 01	3.64608E 01	1.070E 02	-1.46796E 02	1.199E 02	-1.49838E 02
8.564E 01	3.97037E 01	9.594E 01	3.63303E 01	1.075E 02	-1.46921E 02	1.204E 02	-1.49956E 02
8.603E 01	3.95656E 01	9.637E 01	3.62006E 01	1.080E 02	-1.47046E 02	1.209E 02	-1.50073E 02
8.642E 01	3.94277E 01	9.681E 01	3.60717E 01	1.085E 02	-1.47171E 02	1.215E 02	-1.50191E 02
8.682E 01	3.92900E 01	9.725E 01	3.59444E 01	1.089E 02	-1.47295E 02	1.220E 02	-1.50307E 02
8.721E 01	3.91526E 01	9.770E 01	3.58196E 01	1.094E 02	-1.47419E 02	1.226E 02	-1.50424E 02
8.761E 01	3.90154E 01	9.814E 01	3.56987E 01	1.099E 02	-1.47543E 02	1.232E 02	-1.50540E 02
8.801E 01	3.88785E 01	9.859E 01	3.55870E 01	1.104E 02	-1.47666E 02	1.237E 02	-1.50656E 02
8.841E 01	3.87418E 01	9.904E 01	3.55055E 01	1.109E 02	-1.47789E 02	1.243E 02	-1.50771E 02
8.881E 01	3.86054E 01	9.949E 01	3.57307E 01	1.115E 02	-1.47912E 02	1.249E 02	-1.50887E 02
8.921E 01	3.84691E 01	9.994E 01	-1.45088E 02	1.120E 02	-1.48035E 02	1.254E 02	-1.51002E 02

GREATEST VALUE = 4.18002E 01 LOWEST VALUE = -1.51002E 02 INTERVAL = 1.67654E 00



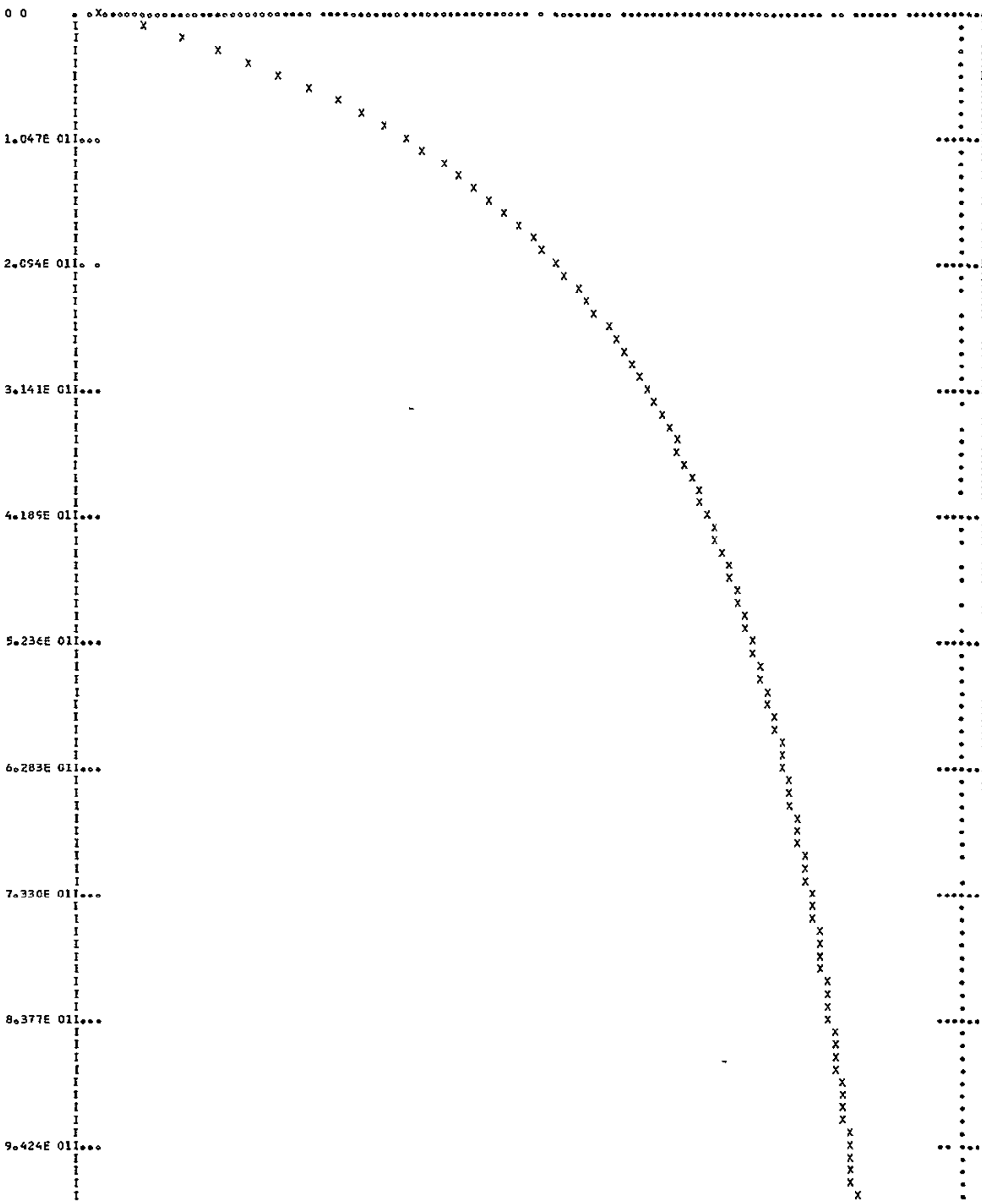
```
PLOT(TY=RE/FR=0/TC=100)

IHC210I IBCOM - PROGRAM INTERRUPT- IMPRECISE    OLD PSW IS  FF45004002234AE6

TRACEBACK FOLLOWS-   ROUTINE    ISN   REG. 14    REG. 15    REG.  0    REG.  1

                     PLOT  0135       4221C49A   0023401C   00000000   0021C9A4

                     MAIN             00019472   0121C518   00000015   0023BFF8

ENTRY POINT=  0121C518

STANDARD FIXUP TAKEN , EXECUTION CONTINUING
```

THE TRANSFORM OF THE RESPONSE IS ---

EXP. OF S	NUMER. COEFFS.	DENOM. COEFFS.
0	-6.00000E 01	2.03860E 02
1	-2.59996E-02	1.04354E 01
2	6.07199E-04	2.76207E-02
3	0.0	2.66273E-05
4	0.0	6.91544E-08

FREQ	MAG	FREQ	MAG	FREQ	MAG	FREQ	MAG
0.0	-2.94319E-01	2.408E 01	-1.27807E-01	4.817E 01	-7.73230E-02	7.225E 01	-5.27031E-02
1.047E 00	-2.79430E-01	2.513E 01	-1.24523E-01	4.922E 01	-7.59075E-02	7.330E 01	-5.19132E-02
2.094E 00	-2.65909E-01	2.618E 01	-1.21384E-01	5.026E 01	-7.45330E-02	7.435E 01	-5.11403E-02
3.141E 00	-2.53575E-01	2.723E 01	-1.18281E-01	5.131E 01	-7.31978E-02	7.539E 01	-5.03838E-02
4.189E 00	-2.42278E-01	2.827E 01	-1.15506E-01	5.236E 01	-7.19001E-02	7.644E 01	-4.96432E-02
5.236E 00	-2.31892E-01	2.932E 01	-1.12749E-01	5.340E 01	-7.06385E-02	7.749E 01	-4.89181E-02
6.283E 00	-2.22310E-01	3.037E 01	-1.10104E-01	5.445E 01	-6.94113E-02	7.853E 01	-4.82078E-02
7.330E 00	-2.13443E-01	3.141E 01	-1.07565E-01	5.550E 01	-6.82172E-02	7.958E 01	-4.75120E-02
8.377E 00	-2.05214E-01	3.246E 01	-1.05124E-01	5.654E 01	-6.70549E-02	8.063E 01	-4.68303E-02
9.424E 00	-1.97555E-01	3.351E 01	-1.02777E-01	5.759E 01	-6.59230E-02	8.168E 01	-4.61621E-02
1.047E 01	-1.90409E-01	3.456E 01	-1.00518E-01	5.864E 01	-6.48205E-02	8.272E 01	-4.55071E-02
1.152E 01	-1.83726E-01	3.560E 01	-9.83412E-02	5.969E 01	-6.37460E-02	8.377E 01	-4.48649E-02
1.257E 01	-1.77462E-01	3.665E 01	-9.62433E-02	6.073E 01	-6.26986E-02	8.482E 01	-4.42352E-02
1.361E 01	-1.71578E-01	3.770E 01	-9.42196E-02	6.178E 01	-6.16773E-02	8.586E 01	-4.36174E-02
1.466E 01	-1.66042E-01	3.874E 01	-9.22663E-02	6.283E 01	-6.06809E-02	8.691E 01	-4.30114E-02
1.571E 01	-1.60823E-01	3.979E 01	-9.03797E-02	6.387E 01	-5.97088E-02	8.796E 01	-4.24168E-02
1.675E 01	-1.55894E-01	4.084E 01	-8.85563E-02	6.492E 01	-5.87599E-02	8.901E 01	-4.18332E-02
1.780E 01	-1.51231E-01	4.189E 01	-8.67930E-02	6.597E 01	-5.78334E-02	9.005E 01	-4.12604E-02
1.885E 01	-1.46814E-01	4.293E 01	-8.50870E-02	6.702E 01	-5.69285E-02	9.110E 01	-4.06980E-02
1.990E 01	-1.42623E-01	4.398E 01	-8.34353E-02	6.806E 01	-5.60445E-02	9.215E 01	-4.01458E-02
2.094E 01	-1.38642E-01	4.503E 01	-8.18354E-02	6.911E 01	-5.51806E-02	9.319E 01	-3.96035E-02
2.199E 01	-1.34855E-01	4.607E 01	-8.02849E-02	7.016E 01	-5.43362E-02	9.424E 01	-3.90708E-02
2.304E 01	-1.31247E-01	4.712E 01	-7.87815E-02	7.120E 01	-5.35106E-02	9.529E 01	-3.85475E-02

GREATEST VALUE = 0.0   LOWEST VALUE = -2.94319E-01   INTERVAL = 2.55930E-03



*** SVOBODA POLYNOMIAL ROOTFINDER ***

H.F.OKRENT
JAN. 1969

POLYNOMIAL OF DEGREE 4 --

0.2038599396E 03 X^{0} 0.1043535995E 02 X^{1} 0.2762065083E-01 X^{2} 0.2662734187E-04 X^{3} 0.6915439599E-07 X^{4}

ACCURACY REQUESTED -- 6 SIGNIFICANT FIGURES

REAL PART	IMAGINARY PART	EXPONENT
-3.644140000	0.0	2
0.000080000	6.260100000	2
0.000080000	-6.260100000	2
-2.064200000	0.0	1

*** SVOBODA POLYNOMIAL ROOTFINDER ***

H.F.OKRENT
JAN. 1969

POLYNOMIAL OF DEGREE 2 --

0.5999995422E 02 X^{0} 0.2599960566E-01 X^{1} -0.6071985699E-03 X^{2}

ACCURACY REQUESTED -- 6 SIGNIFICANT FIGURES

REAL PART	IMAGINARY PART	EXPONENT
-2.936660000	0.0	2
3.364850000	0.0	2

SENSITIVITY OF POLES TO CJ1

POLES REAL	POLES IMAG	SENSITIVITY REAL	SENSITIVITY IMAG
-3.6441E 02	0.0	6.8343E 01	0.0
8.0000E-03	6.2601E 02	-5.9182E-03	-2.0911E 02
8.0000E-03	-6.2601E 02	-5.9182E-03	1.7692E 02
-2.0642E 01	0.0	1.5917E 01	0.0

SENSITIVITY OF POLES TO RE1

POLES REAL	POLES IMAG	SENSITIVITY REAL	SENSITIVITY IMAG
-3.6441E 02	0.0	9.2617E-02	0.0
8.0000E-03	6.2601E 02	-1.9173E 01	-2.4059E 02
8.0000E-03	-6.2601E 02	-1.9173E 01	1.4154E 02
-2.0642E 01	0.0	3.8550E-01	0.0

SENSITIVITY OF POLES TO CJ2

POLES REAL	POLES IMAG	SENSITIVITY REAL	SENSITIVITY IMAG
-3.6441E 02	0.0	1.8731E 01	0.0
8.0000E-03	6.2601E 02	-3.1010E-05	-8.3186E 00
8.0000E-03	-6.2601E 02	-3.1010E-05	1.7944E 01
-2.0642E 01	0.0	2.5619E-02	0.0

SENSITIVITY OF POLES TO RJ2

POLES REAL	POLES IMAG	SENSITIVITY REAL	SENSITIVITY IMAG
-3.6441E 02	0.0	-2.9972E 02	0.0
8.0000E-03	6.2601E 02	-1.9445E 01	-7.4025E 01
8.0000E-03	-6.2601E 02	-1.9445E 01	-1.6092E 02
-2.0642E 01	0.0	-3.2968E 00	0.0

SENSITIVITY OF POLES TO RE3

POLES REAL	POLES IMAG	SENSITIVITY REAL	SENSITIVITY IMAG
-3.6441E 02	0.0	6.2044E 01	0.0
8.0000E-03	6.2601E 02	-2.8046E-01	-2.1803E 02
8.0000E-03	-6.2601E 02	-2.8046E-01	1.8059E 02
-2.0642E 01	0.0	1.6971E 01	0.0

SENSITIVITY OF POLES TO LE1

POLES REAL	POLES IMAG	SENSITIVITY REAL	SENSITIVITY IMAG
-3.6441E 02	0.0	1.2784E 00	0.0
8.0000E-03	6.2601E 02	-3.9258E-09	-1.2070E 02
8.0000E-03	-6.2601E 02	-3.9258E-09	7.1477E 01
-2.0642E 01	0.0	-5.4710E-03	0.0

SENSITIVITY OF POLES TO CJ3

POLES REAL	POLES IMAG	SENSITIVITY REAL	SENSITIVITY IMAG
-3.6441E 02	0.0	1.8455E 00	0.0
8.0000E-03	6.2601E 02	-1.1837E-04	-2.6657E 01
8.0000E-03	-6.2601E 02	-1.1837E-04	1.7046E 01
-2.0642E 01	0.0	3.1672E-01	0.0

SENSITIVITY OF POLES TO CJ4

POLES REAL	POLES IMAG	SENSITIVITY REAL	SENSITIVITY IMAG
-3.6441E 02	0.0	2.3379E 02	0.0
8.0000E-03	6.2601E 02	-1.5934E-03	-1.4712E 02
8.0000E-03	-6.2601E 02	-1.5934E-03	2.5368E 02
-2.0642E 01	0.0	3.6222E 00	0.0

SENSITIVITY OF POLES TO CJ5

POLES REAL	POLES IMAG	SENSITIVITY REAL	SENSITIVITY IMAG
-3.6441E 02	0.0	4.0428E 01	0.0
8.0000E-03	6.2601E 02	-3.3902E-04	-1.1410E 02
8.0000E-03	-6.2601E 02	-3.3902E-04	8.8946E 01
-2.0642E 01	0.0	7.6547E-01	0.0

SENSITIVITY OF POLES TO DJ2

POLES REAL	POLES IMAG	SENSITIVITY REAL	SENSITIVITY IMAG
-3.6441E 02	0.0	-4.2723E 01	0.0
8.0000E-03	6.2601E 02	-6.4276E-02	-7.5835E 00
8.0000E-03	-6.2601E 02	-6.4276E-02	-1.7599E 01
-2.0642E 01	0.0	2.9086E 00	0.0

PLOT(TY=SE/FR=80/TO=126/EL=CJ1)

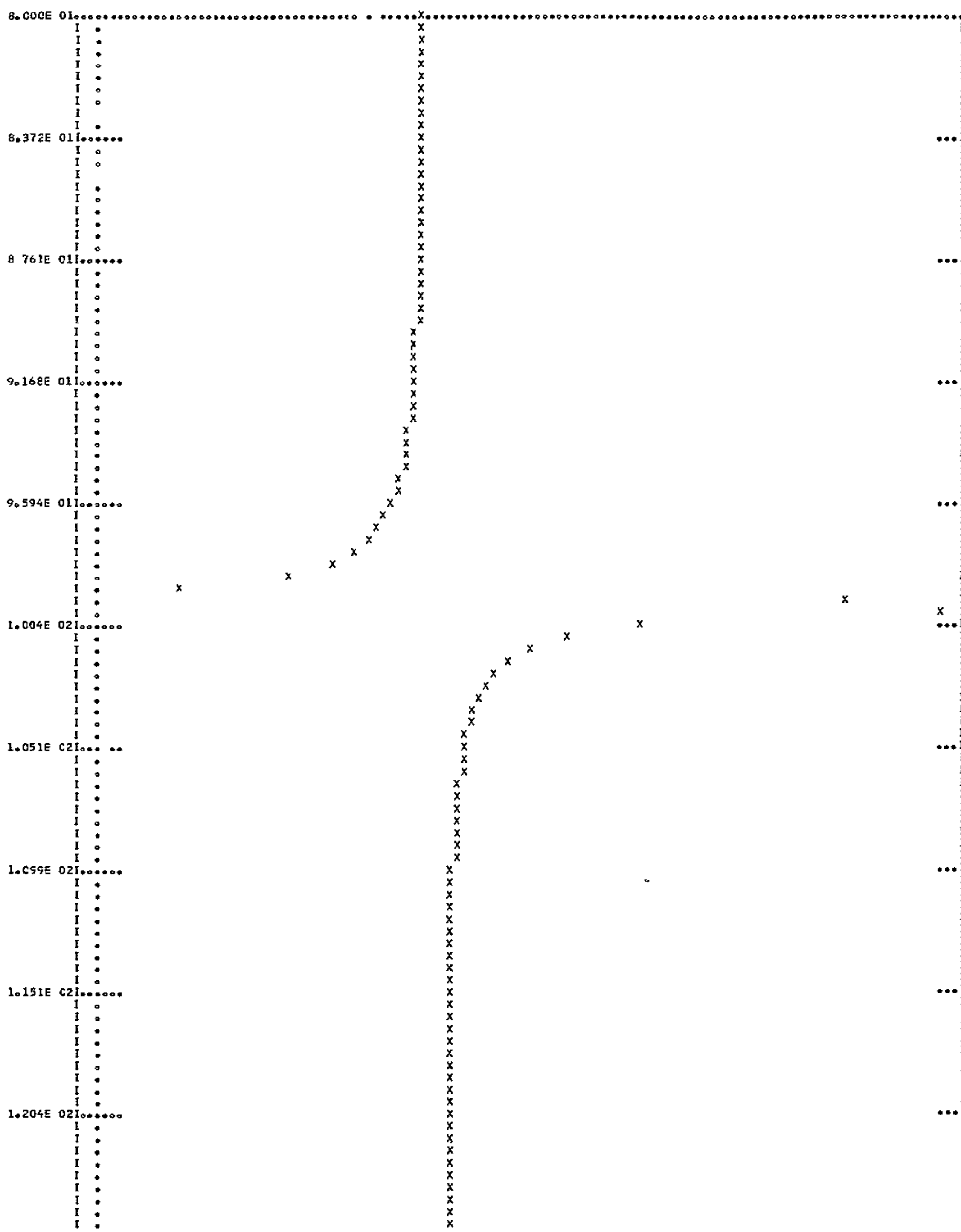
SENSITIVITY WITH RESPECT TO CJ1

THE TRANSFORM OF THE RESPONSE IS ---

EXP. OF S	NUMER. COEFFS.	DENOM. COEFFS.
0	0.0	1.22316E 04
1	4.63198E 02	6.31421E 02
2	1.75831E 00	1.80477E 00
3	-2.82103E-03	-4.02057E-03
4	-1.12337E-05	-1.19296E-05
5	-1.03200E-08	-1.43701E-08
6	-4.06092E-11	-4.19904E-11

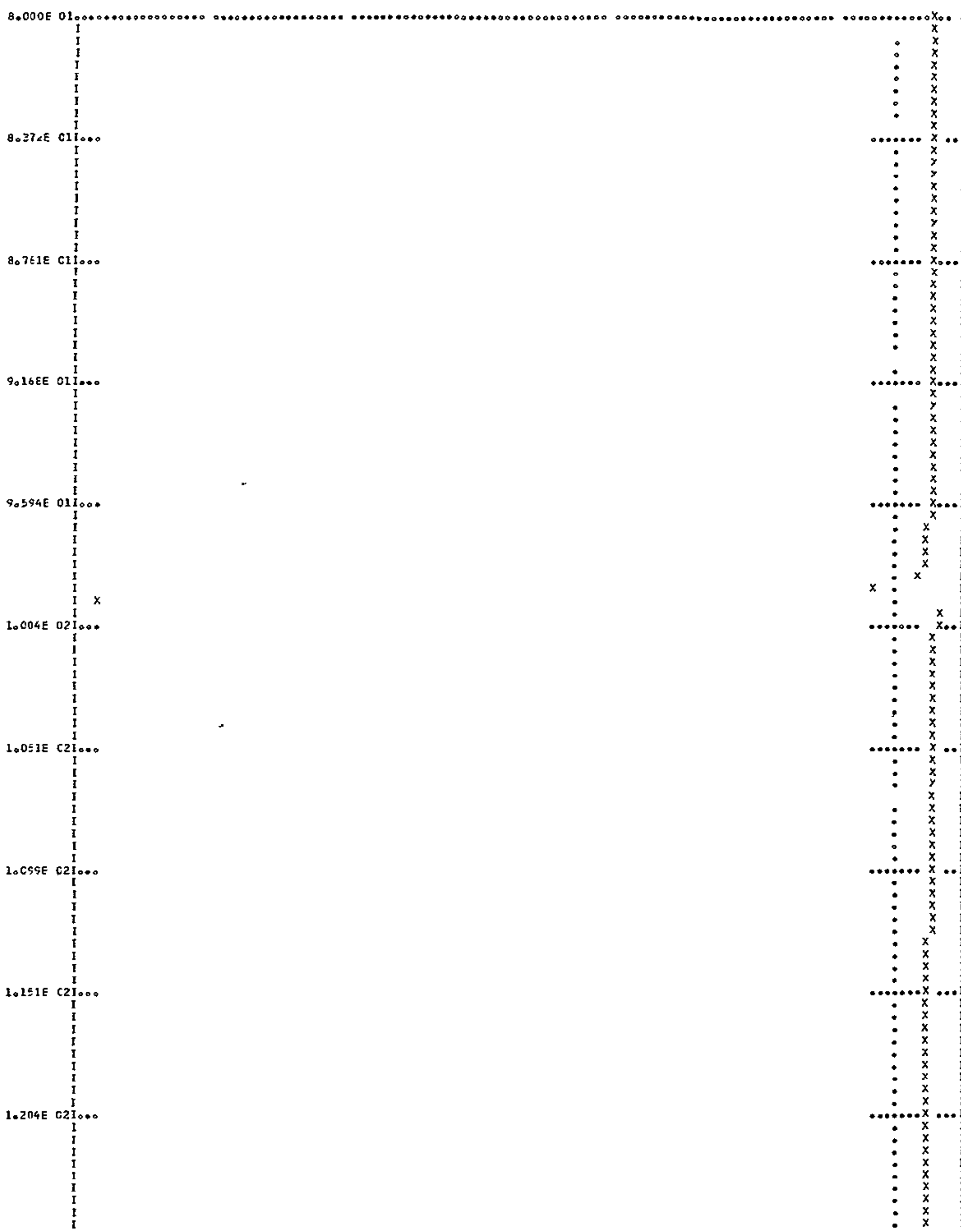
FREQ	MAC	FREQ	MAG	FREQ	MAG	FREQ	MAG
8.000E 01	8.85258E-01	8.962E 01	8.73057E-01	1.004E 02	1.48177E 00	1.125E 02	9.63559E-01
8.036E 01	8.85208E-01	9.003E 01	8.71578E-01	1.009E 02	1.27297E 00	1.130E 02	9.62545E-01
8.073E 01	8.85140E-01	9.044E 01	8.69929E-01	1.013E 02	1.17780E 00	1.135E 02	9.61616E-01
8.110E 01	8.85052E-01	9.085E 01	8.68083E-01	1.018E 02	1.12339E 00	1.140E 02	9.60767E-01
8.147E 01	8.84942E-01	9.126E 01	8.66012E-01	1.022E 02	1.08827E 00	1.145E 02	9.59990E-01
8.184E 01	8.84809E-01	9.168E 01	8.63677E-01	1.027E 02	1.06373E 00	1.151E 02	9.59274E-01
8.221E 01	8.84652E-01	9.210E 01	8.61033E-01	1.032E 02	1.04567E 00	1.156E 02	9.58618E-01
8.258E 01	8.84469E-01	9.251E 01	8.58025E-01	1.036E 02	1.03183E 00	1.161E 02	9.58015E-01
8.296E 01	8.84259E-01	9.294E 01	8.54579E-01	1.041E 02	1.02090E 00	1.166E 02	9.57460E-01
8.334E 01	8.84017E-01	9.336E 01	8.50608E-01	1.046E 02	1.01208E 00	1.172E 02	9.56947E-01
8.372E 01	8.83744E-01	9.378E 01	8.45989E-01	1.051E 02	1.00481E 00	1.177E 02	9.56474E-01
8.410E 01	8.83436E-01	9.421E 01	8.40568E-01	1.055E 02	9.98736E-01	1.182E 02	9.56039E-01
8.448E 01	8.83089E-01	9.464E 01	8.34134E-01	1.060E 02	9.93588E-01	1.188E 02	9.55638E-01
8.487E 01	8.82703E-01	9.507E 01	8.26388E-01	1.065E 02	9.89181E-01	1.193E 02	9.55268E-01
8.525E 01	8.82273E-01	9.550E 01	8.16912E-01	1.070E 02	9.85367E-01	1.199E 02	9.54927E-01
8.564E 01	8.81795E-01	9.594E 01	8.05080E-01	1.075E 02	9.82040E-01	1.204E 02	9.54612E-01
8.603E 01	8.81264E-01	9.637E 01	7.89928E-01	1.080E 02	9.79121E-01	1.209E 02	9.54321E-01
8.642E 01	8.80677E-01	9.681E 01	7.69890E-01	1.085E 02	9.76539E-01	1.215E 02	9.54054E-01
8.682E 01	8.80027E-01	9.725E 01	7.42201E-01	1.089E 02	9.74252E-01	1.220E 02	9.53807E-01
8.721E 01	8.79308E-01	9.770E 01	7.01561E-01	1.094E 02	9.72207E-01	1.226E 02	9.53580E-01
8.761E 01	8.78512E-01	9.814E 01	6.36254E-01	1.099E 02	9.70378E-01	1.232E 02	9.53373E-01
8.801E 01	8.77633E-01	9.859E 01	5.14571E-01	1.104E 02	9.68731E-01	1.237E 02	9.53182E-01
8.841E 01	8.76659E-01	9.904E 01	2.10826E-01	1.109E 02	9.67241E-01	1.243E 02	9.53008E-01
8.881E 01	8.75581E-01	9.949E 01	2.04210E 00	1.115E 02	9.65896E-01	1.249E 02	9.52849E-01
8.921E 01	8.74385E-01	9.994E 01	2.30479E 00	1.120E 02	9.64673E-01	1.254E 02	9.52702E-01

GREATEST VALUE = 2.3047°E 00 LOWEST VALUE = 0.0 INTERVAL = 2.00416E-02



FREQ	PHA	FREQ	PHA	FREQ	PHA	FREQ	PHA
8.000E 01	7.61846E 00	8.962E 01	7.02242E 00	1.004E 02	7.88587E 00	1.125E 02	6.25863E 00
8.036E 01	7.59053E 00	9.003E 01	6.99440E 00	1.009E 02	7.54165E 00	1.130E 02	6.23376E 00
8.073E 01	7.57457E 00	9.044E 01	6.96564E 00	1.013E 02	7.33122E 00	1.135E 02	6.20915E 00
8.110E 01	7.55245E 00	9.085E 01	6.93598E 00	1.018E 02	7.18594E 00	1.140E 02	6.18487E 00
8.147E 01	7.53027E 00	9.126E 01	6.90545E 00	1.022E 02	7.07748E 00	1.145E 02	6.16081E 00
8.184E 01	7.50796E 00	9.168E 01	6.87375E 00	1.027E 02	6.99101E 00	1.151E 02	6.13702E 00
8.221E 01	7.48555E 00	9.210E 01	6.84071E 00	1.032E 02	6.92006E 00	1.156E 02	6.11348E 00
8.258E 01	7.46300E 00	9.251E 01	6.80609E 00	1.036E 02	6.85921E 00	1.161E 02	6.09006E 00
8.296E 01	7.44034E 00	9.294E 01	6.76947E 00	1.041E 02	6.80586E 00	1.166E 02	6.06690E 00
8.334E 01	7.41753E 00	9.336E 01	6.73057E 00	1.046E 02	6.75791E 00	1.172E 02	6.04386E 00
8.372E 01	7.39460E 00	9.378E 01	6.68859E 00	1.051E 02	6.71432E 00	1.177E 02	6.02094E 00
8.410E 01	7.37152E 00	9.421E 01	6.64292E 00	1.055E 02	6.67391E 00	1.182E 02	5.99826E 00
8.448E 01	7.34824E 00	9.464E 01	6.59231E 00	1.060E 02	6.63639E 00	1.188E 02	5.97569E 00
8.487E 01	7.32483E 00	9.507E 01	6.53541E 00	1.065E 02	6.60091E 00	1.193E 02	5.95327E 00
8.525E 01	7.30120E 00	9.550E 01	6.46988E 00	1.070E 02	6.56714E 00	1.199E 02	5.93094E 00
8.564E 01	7.27738E 00	9.594E 01	6.39190E 00	1.075E 02	6.53476E 00	1.204E 02	5.90874E 00
8.603E 01	7.25329E 00	9.637E 01	6.29553E 00	1.080E 02	6.50382E 00	1.209E 02	5.88666E 00
8.642E 01	7.22900E 00	9.681E 01	6.17085E 00	1.085E 02	6.47383E 00	1.215E 02	5.86469E 00
8.682E 01	7.20445E 00	9.725E 01	5.99727E 00	1.089E 02	6.44474E 00	1.220E 02	5.84279E 00
8.721E 01	7.17960E 00	9.770E 01	5.73106E 00	1.094E 02	6.41643E 00	1.226E 02	5.82102E 00
8.761E 01	7.15441E 00	9.814E 01	5.24777E 00	1.099E 02	6.38887E 00	1.232E 02	5.79938E 00
8.801E 01	7.12895E 00	9.859E 01	4.04860E 00	1.104E 02	6.36198E 00	1.237E 02	5.77775E 00
8.841E 01	7.10301E 00	9.904E 01	-5.07980E 00	1.109E 02	6.33544E 00	1.243E 02	5.75629E 00
8.881E 01	7.07664E 00	9.949E 01	-1.67772E 02	1.115E 02	6.30950E 00	1.249E 02	5.73489E 00
8.921E 01	7.04985E 00	9.994E 01	8.54908E 00	1.120E 02	6.28390E 00	1.254E 02	5.71350E 00

GREATEST VALUE = 8.54908E 00 LOWEST VALUE = -1.67772E 02 INTERVAL = 1.53323E 00



PLOT(TY=SE/FR=80/TC=126/EL=DJ2)

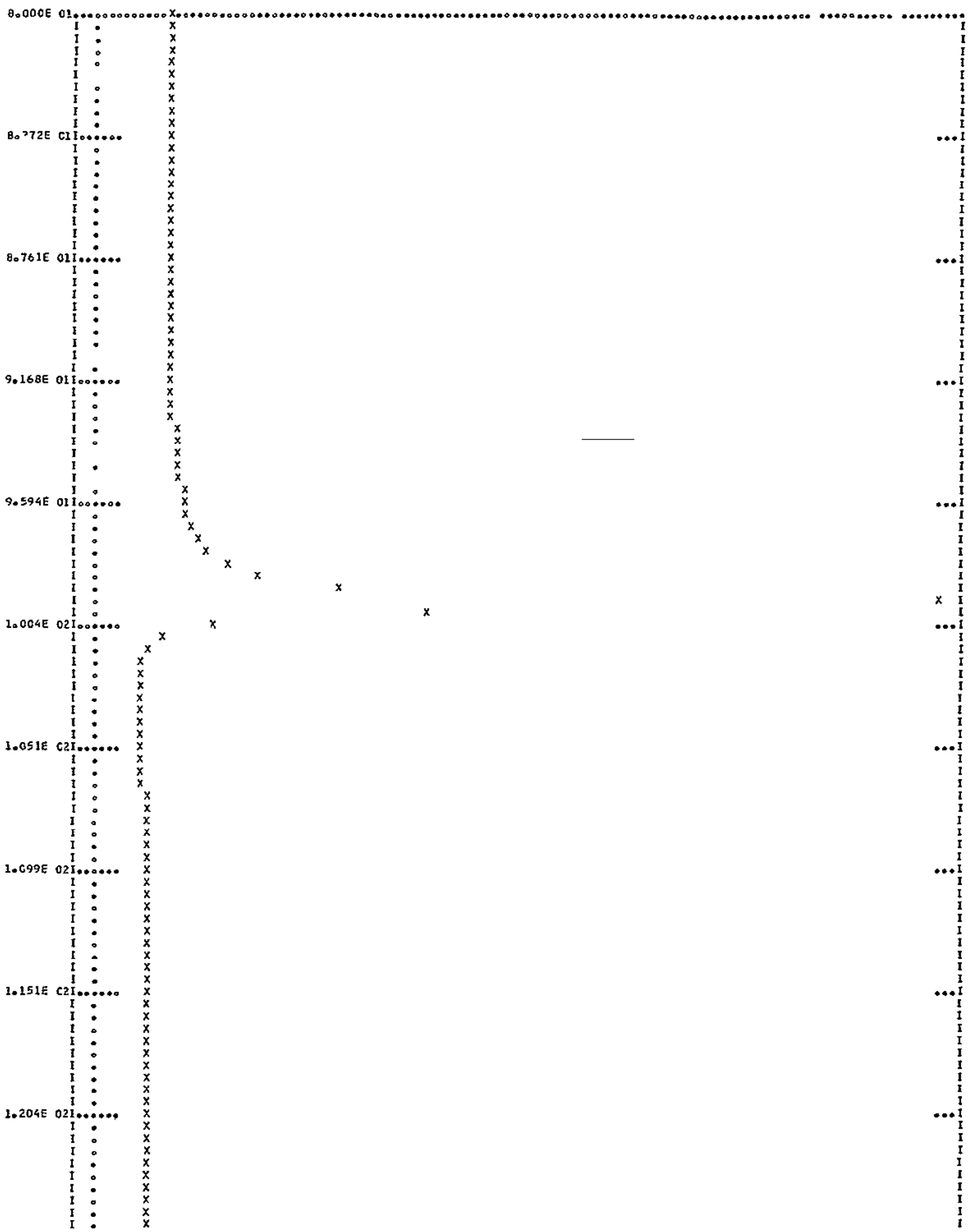
SENSITIVITY WITH RESPECT TO DJ2

THE TRANSFORM OF THE RESPONSE IS ---

EXP. OF S	NUMER. COEFFS.	DENOM. COEFFS.
0	-1.21920E 04	1.22316E 04
1	-6.66412E 02	6.31421E 02
2	-7.81940E 00	1.80477E 00
3	-1.87628E-02	-4.02057E-03
4	-1.99590E-05	-1.19296E-05
5	-4.16667E-08	-1.43701E-08
6	0.0	-4.19904E-11

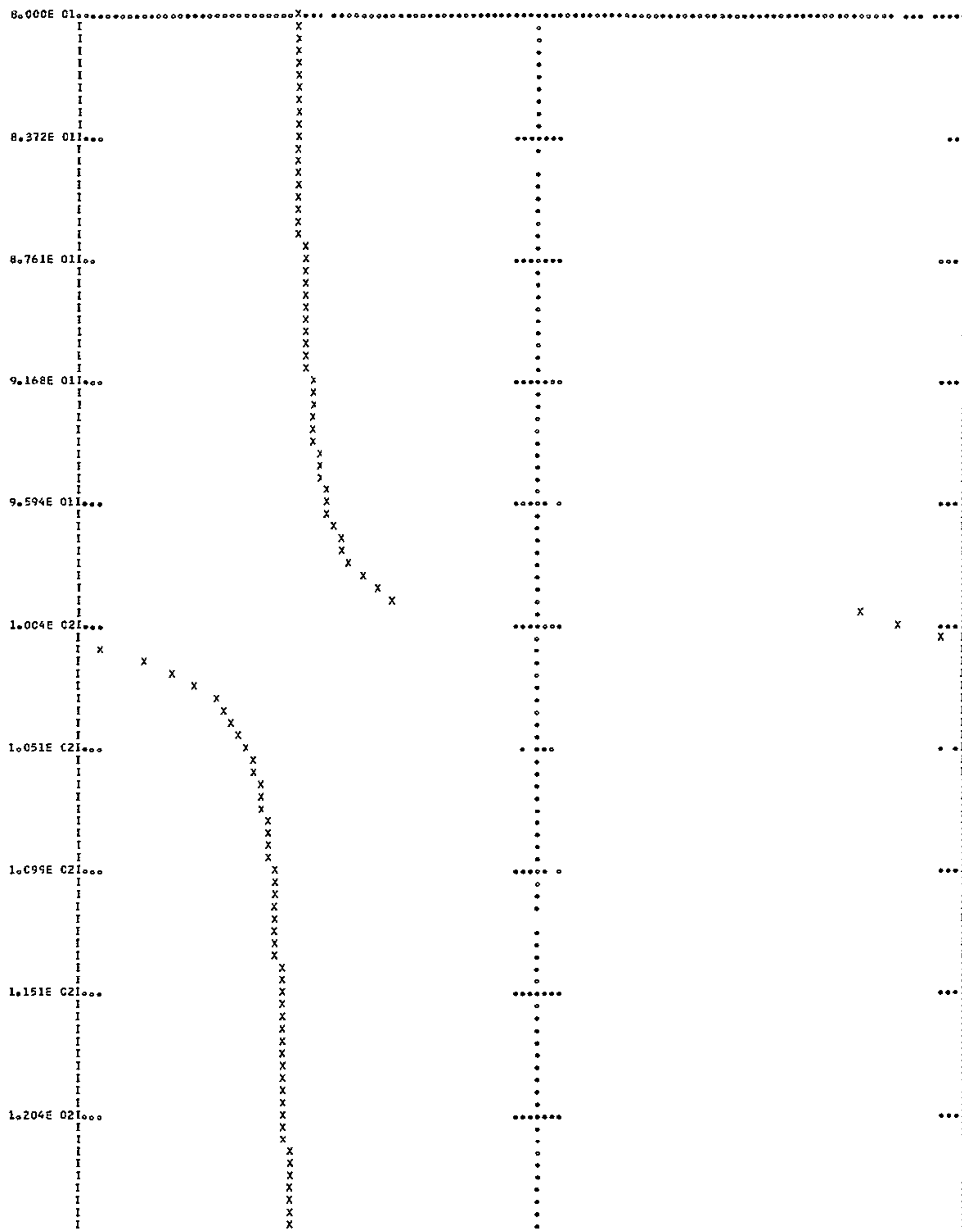
FREQ	MAG	FREQ	MAG	FREQ	MAG	FREQ	MAG
8.000E 01	1.54004E 00	8.962E 01	1.55487E 00	1.004E 02	2.44042E 00	1.125E 02	1.06081E 00
8.036E 01	1.53886E 00	9.003E 01	1.55985E 00	1.009E 02	1.39009E 00	1.130E 02	1.06185E 00
8.073E 01	1.53775E 00	9.044E 01	1.56564E 00	1.013E 02	1.02435E 00	1.135E 02	1.06259E 00
8.110E 01	1.53671E 00	9.085E 01	1.57234E 00	1.018E 02	9.00334E-01	1.140E 02	1.06305E 00
8.147E 01	1.53574E 00	9.126E 01	1.58012E 00	1.022E 02	8.70700E-01	1.145E 02	1.06326E 00
8.184E 01	1.53485E 00	9.168E 01	1.58916E 00	1.027E 02	8.76356E-01	1.151E 02	1.06324E 00
8.221E 01	1.53406E 00	9.210E 01	1.59968E 00	1.032E 02	8.93934E-01	1.156E 02	1.06301E 00
8.258E 01	1.53336E 00	9.251E 01	1.61198E 00	1.036E 02	9.14408E-01	1.161E 02	1.06259E 00
8.296E 01	1.53277E 00	9.294E 01	1.62640E 00	1.041E 02	9.34367E-01	1.166E 02	1.06199E 00
8.334E 01	1.53230E 00	9.336E 01	1.64343E 00	1.046E 02	9.52597E-01	1.172E 02	1.06122E 00
8.372E 01	1.53195E 00	9.378E 01	1.66366E 00	1.051E 02	9.68804E-01	1.177E 02	1.06030E 00
8.410E 01	1.53174E 00	9.421E 01	1.68792E 00	1.055E 02	9.83009E-01	1.182E 02	1.05925E 00
8.448E 01	1.53168E 00	9.464E 01	1.71730E 00	1.060E 02	9.95384E-01	1.188E 02	1.05807E 00
8.487E 01	1.53178E 00	9.507E 01	1.75336E 00	1.065E 02	1.00613E 00	1.193E 02	1.05676E 00
8.525E 01	1.53207E 00	9.550E 01	1.79838E 00	1.070E 02	1.01544E 00	1.199E 02	1.05535E 00
8.564E 01	1.53256E 00	9.594E 01	1.85566E 00	1.075E 02	1.02349E 00	1.204E 02	1.05383E 00
8.603E 01	1.53328E 00	9.637E 01	1.93041E 00	1.080E 02	1.03043E 00	1.209E 02	1.05221E 00
8.642E 01	1.53424E 00	9.681E 01	2.03133E 00	1.085E 02	1.03642E 00	1.215E 02	1.05051E 00
8.682E 01	1.53546E 00	9.725E 01	2.17364E 00	1.089E 02	1.04157E 00	1.220E 02	1.04872E 00
8.721E 01	1.53699E 00	9.770E 01	2.38702E 00	1.094E 02	1.04596E 00	1.226E 02	1.04685E 00
8.761E 01	1.53886E 00	9.814E 01	2.73773E 00	1.099E 02	1.04970E 00	1.232E 02	1.04491E 00
8.801E 01	1.54110E 00	9.859E 01	3.40739E 00	1.104E 02	1.05286E 00	1.237E 02	1.04290E 00
8.841E 01	1.54377E 00	9.904E 01	5.13446E 00	1.109E 02	1.05550E 00	1.243E 02	1.04083E 00
8.881E 01	1.54691E 00	9.949E 01	1.79582E 01	1.115E 02	1.05767E 00	1.249E 02	1.03870E 00
8.921E 01	1.55058E 00	9.994E 01	7.05597E 00	1.120E 02	1.05943E 00	1.254E 02	1.03650E 00

GREATEST VALUE = 1.795E2E 01 LOWEST VALUE = 0.0 INTERVAL = 1.56158E-01



FREQ	PHA	FREQ	PHA	FREQ	PHA	FREQ	PHA
8.000E 01	-9.93250E 01	8.962E 01	-9.50890E 01	1.004E 02	1.44258E 02	1.125E 02	-1.06240E 02
8.036E 01	-9.92116E 01	9.003E 01	-9.47980E 01	1.009E 02	1.62350E 02	1.130E 02	-1.05920E 02
8.073E 01	-9.90957E 01	9.044E 01	-9.44873E 01	1.013E 02	-1.78038E 02	1.135E 02	-1.05621E 02
8.110E 01	-9.89772E 01	9.085E 01	-9.41544E 01	1.018E 02	-1.60871E 02	1.140E 02	-1.05342E 02
8.147E 01	-9.88560E 01	9.126E 01	-9.37964E 01	1.022E 02	-1.47933E 02	1.145E 02	-1.05081E 02
8.184E 01	-9.87318E 01	9.168E 01	-9.34097E 01	1.027E 02	-1.38698E 02	1.151E 02	-1.04835E 02
8.221E 01	-9.86044E 01	9.210E 01	-9.29903E 01	1.032E 02	-1.32086E 02	1.156E 02	-1.04603E 02
8.258E 01	-9.84736E 01	9.251E 01	-9.25332E 01	1.036E 02	-1.27230E 02	1.161E 02	-1.04383E 02
8.296E 01	-9.83393E 01	9.294E 01	-9.20324E 01	1.041E 02	-1.23556E 02	1.166E 02	-1.04175E 02
8.334E 01	-9.82010E 01	9.336E 01	-9.14805E 01	1.046E 02	-1.20699E 02	1.172E 02	-1.03978E 02
8.372E 01	-9.80585E 01	9.378E 01	-9.08686E 01	1.051E 02	-1.18420E 02	1.177E 02	-1.03790E 02
8.410E 01	-9.79116E 01	9.421E 01	-9.01855E 01	1.055E 02	-1.16563E 02	1.182E 02	-1.03611E 02
8.448E 01	-9.77598E 01	9.464E 01	-8.94168E 01	1.060E 02	-1.15022E 02	1.188E 02	-1.03439E 02
8.487E 01	-9.76028E 01	9.507E 01	-8.85444E 01	1.065E 02	-1.13723E 02	1.193E 02	-1.03276E 02
8.525E 01	-9.74400E 01	9.550E 01	-8.75446E 01	1.070E 02	-1.12612E 02	1.199E 02	-1.03119E 02
8.564E 01	-9.72712E 01	9.594E 01	-8.63862E 01	1.075E 02	-1.11651E 02	1.204E 02	-1.02968E 02
8.603E 01	-9.70957E 01	9.637E 01	-8.50276E 01	1.080E 02	-1.10811E 02	1.209E 02	-1.02823E 02
8.642E 01	-9.69129E 01	9.681E 01	-8.34102E 01	1.085E 02	-1.10070E 02	1.215E 02	-1.02684E 02
8.682E 01	-9.67222E 01	9.725E 01	-8.14536E 01	1.089E 02	-1.09411E 02	1.220E 02	-1.02550E 02
8.721E 01	-9.65228E 01	9.770E 01	-7.90393F 01	1.094E 02	-1.08821E 02	1.226E 02	-1.02420E 02
8.761E 01	-9.63139E 01	9.814E 01	-7.59931E 01	1.099E 02	-1.08288E 02	1.232E 02	-1.02295E 02
8.801E 01	-9.60945E 01	9.859E 01	-7.20461E 01	1.104E 02	-1.07804E 02	1.237E 02	-1.02174E 02
8.841E 01	-9.58637E 01	9.904E 01	-6.67669E 01	1.109E 02	-1.07363E 02	1.243E 02	-1.02056E 02
8.881E 01	-9.56201E 01	9.949E 01	-5.92356E 01	1.115E 02	-1.06958E 02	1.249E 02	-1.01943E 02
8.921E 01	-9.53624E 01	9.994E 01	1.30125E 02	1.120E 02	-1.06585E 02	1.254E 02	-1.01832E 02

GREATEST VALUE = 1.62350E 02 LOWEST VALUE = -1.78038E 02 INTERVAL = 2.5599CE 00



PLOT(TY=SE/FR=80/TL=126/EL=CJ3)

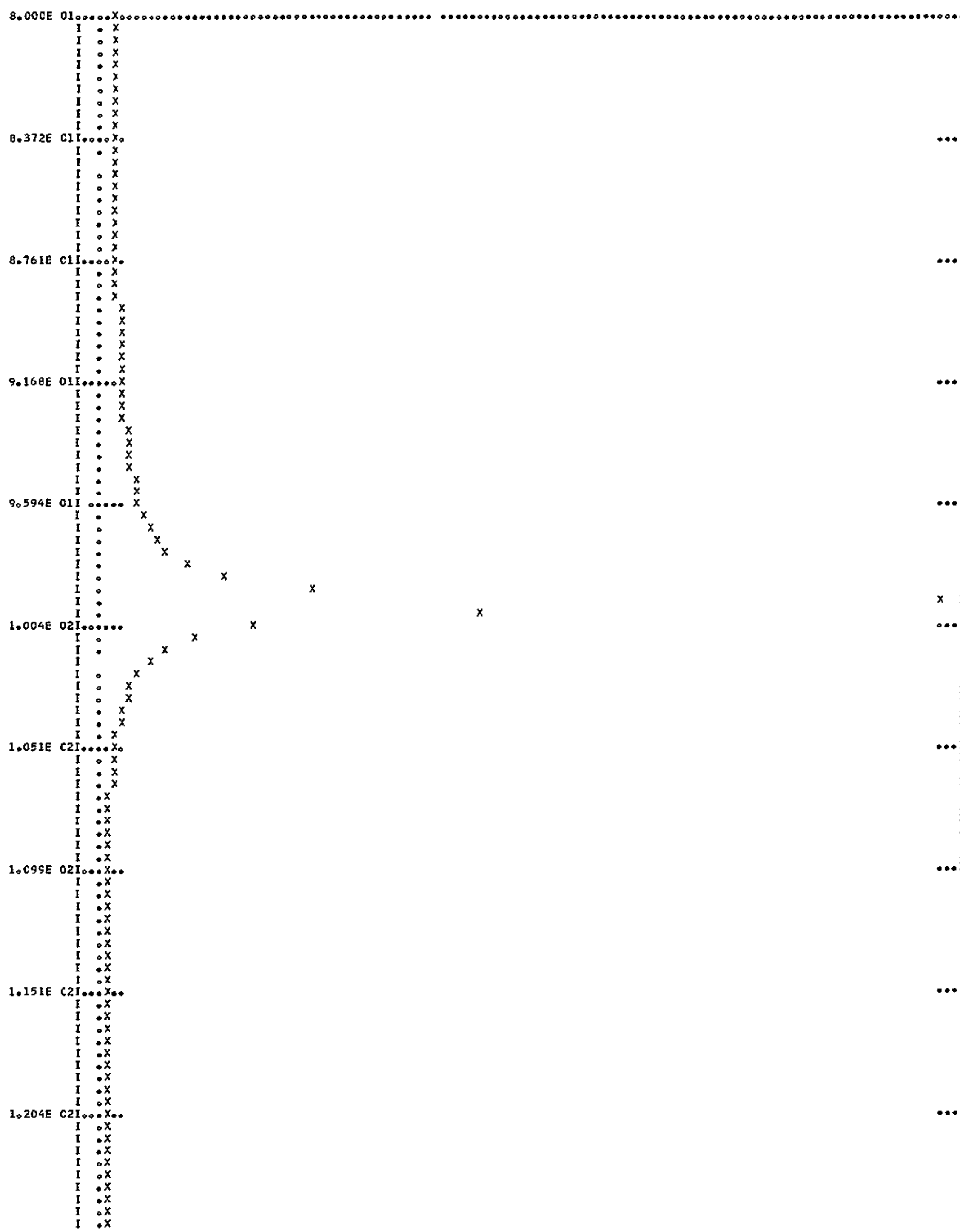
SENSITIVITY WITH RESPECT TO CJ3

THE TRANSFORM OF THE RESPONSE IS ---

EXP. OF S	NUMER. COEFFS.	DENOM. COEFFS.
0	0.0	1.22316E 04
1	4.07400E 01	6.31421E 02
2	1.76014E 00	1.80477E 00
3	9.92565E-03	-4.02057E-03
4	1.90483E-05	-1.19296E-05
5	2.16353E-08	-1.43701E-08
6	2.72701E-11	-4.19904E-11

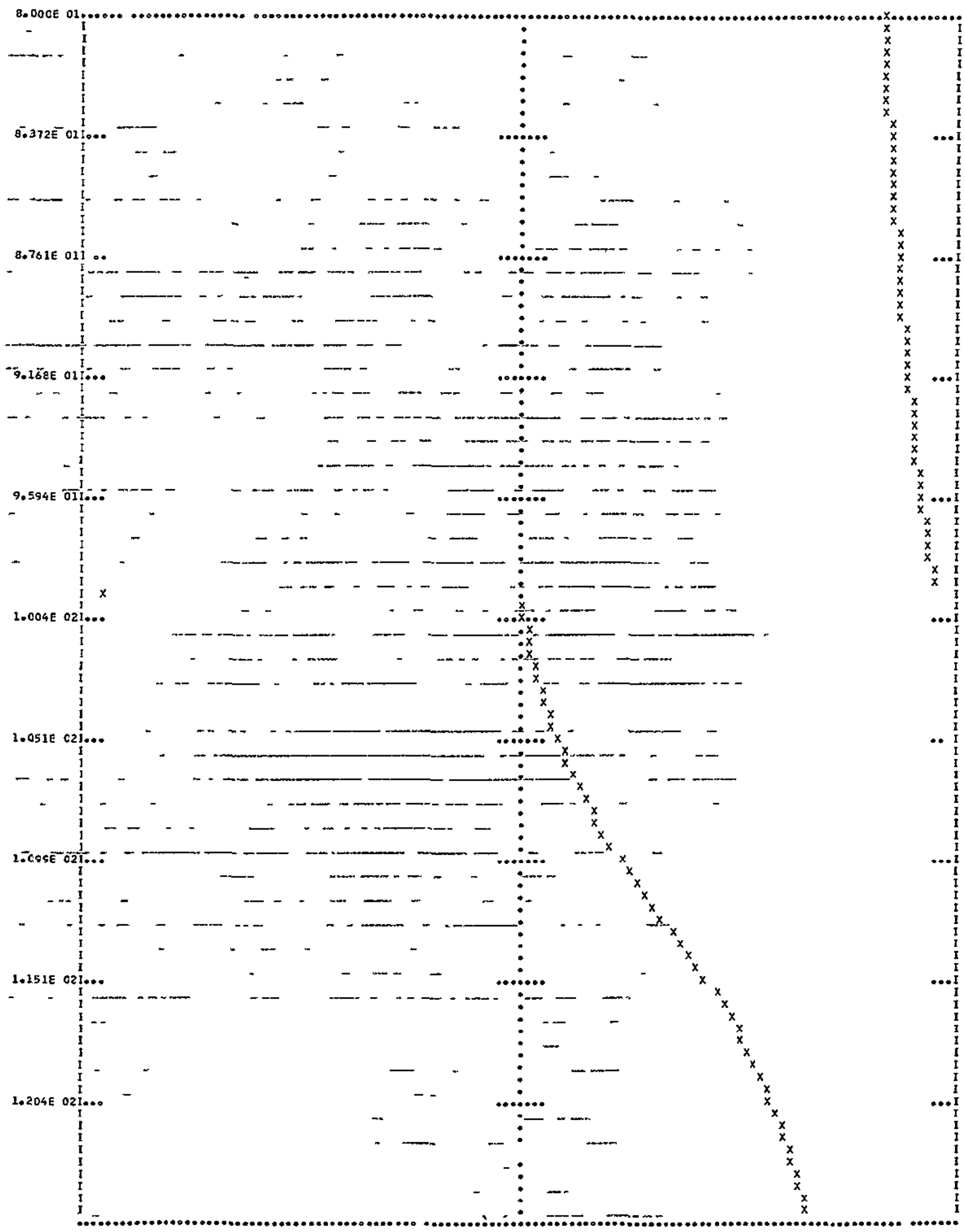
FREQ	MAG	FREQ	MAG	FREQ	MAG	FREQ	MAG
8.000E 01	9.93158E-01	8.962E 01	1.44789E 00	1.004E 02	1.15788E 01	1.125E 02	3.84085E-01
8.036E 01	1.00242E 00	9.003E 01	1.48681E 00	1.009E 02	7.05857E 00	1.130E 02	3.70435E-01
8.073E 01	1.01204E 00	9.044E 01	1.52936E 00	1.013E 02	4.99556E 00	1.135E 02	3.59231E-01
8.110E 01	1.02206E 00	9.085E 01	1.57608E 00	1.018E 02	3.81518E 00	1.140E 02	3.50220E-01
8.147E 01	1.03251E 00	9.126E 01	1.62764E 00	1.022E 02	3.05155E 00	1.145E 02	3.43151E-01
8.184E 01	1.04340E 00	9.168E 01	1.68481E 00	1.027E 02	2.51755E 00	1.151E 02	3.37796E-01
8.221E 01	1.05478E 00	9.210E 01	1.74860E 00	1.032E 02	2.12381E 00	1.156E 02	3.33946E-01
8.258E 01	1.06668E 00	9.251E 01	1.82021E 00	1.036E 02	1.82174E 00	1.161E 02	3.31402E-01
8.296E 01	1.07914E 00	9.294E 01	1.90118E 00	1.041E 02	1.58314E 00	1.166E 02	3.29991E-01
8.334E 01	1.09220E 00	9.336E 01	1.99349E 00	1.046E 02	1.39033E 00	1.172E 02	3.29549E-01
8.372E 01	1.10591E 00	9.378E 01	2.09972E 00	1.051E 02	1.23163E 00	1.177E 02	3.29933E-01
8.410E 01	1.12033E 00	9.421E 01	2.22324E 00	1.055E 02	1.09906E 00	1.182E 02	3.31017E-01
8.448E 01	1.13551E 00	9.464E 01	2.36863E 00	1.060E 02	9.87051E-01	1.188E 02	3.32688E-01
8.487E 01	1.15152E 00	9.507E 01	2.54231E 00	1.065E 02	8.91485E-01	1.193E 02	3.34846E-01
8.525E 01	1.16843E 00	9.550E 01	2.75340E 00	1.070E 02	8.09354E-01	1.199E 02	3.37409E-01
8.564E 01	1.18633E 00	9.594E 01	3.01540E 00	1.075E 02	7.38338E-01	1.204E 02	3.40305E-01
8.603E 01	1.20530E 00	9.637E 01	3.34913E 00	1.080E 02	6.76694E-01	1.209E 02	3.43468E-01
8.642E 01	1.22546E 00	9.681E 01	3.78889E 00	1.085E 02	6.23040E-01	1.215E 02	3.46844E-01
8.682E 01	1.24693E 00	9.725E 01	4.39427E 00	1.089E 02	5.76280E-01	1.220E 02	3.50391E-01
8.721E 01	1.26983E 00	9.770E 01	5.28052E 00	1.094E 02	5.35528E-01	1.226E 02	3.54067E-01
8.761E 01	1.29432E 00	9.814E 01	6.70184E 00	1.099E 02	5.00069E-01	1.232E 02	3.57838E-01
8.801E 01	1.32058E 00	9.859E 01	9.34957E 00	1.104E 02	4.69302E-01	1.237E 02	3.61681E-01
8.841E 01	1.34881E 00	9.904E 01	1.60163E 01	1.109E 02	4.42725E-01	1.243E 02	3.65569E-01
8.881E 01	1.37925E 00	9.949E 01	6.45291E 01	1.115E 02	4.19912E-01	1.249E 02	3.69484E-01
8.921E 01	1.41217E 00	9.994E 01	2.93790E 01	1.120E 02	4.00476E-01	1.254E 02	3.73408E-01

GREATEST VALUE = 6.45291E 01 LOWEST VALUE = 0.0 INTERVAL = 5.61123E-01



FREQ	PHA	FREQ	PHA	FREQ	PHA	FREQ	PHA
8.000E 01	1.56058E 02	8.962E 01	1.64618E 02	1.004E 02	1.66334E 00	1.125E 02	6.15347E 01
8.036E 01	1.56336E 02	9.003E 01	1.65063E 02	1.009E 02	2.83499E 00	1.130E 02	6.53602E 01
8.073E 01	1.56618E 02	9.044E 01	1.65519E 02	1.013E 02	4.05106E 00	1.135E 02	6.91704E 01
8.110E 01	1.56904E 02	9.085E 01	1.65988E 02	1.018E 02	5.32634E 00	1.140E 02	7.29336E 01
8.147E 01	1.57194E 02	9.126E 01	1.66471E 02	1.022E 02	6.67005E 00	1.145E 02	7.66228E 01
8.184E 01	1.57488E 02	9.168E 01	1.66967E 02	1.027E 02	8.08932E 00	1.151E 02	8.02134E 01
8.221E 01	1.57786E 02	9.210E 01	1.67478E 02	1.032E 02	9.59104E 00	1.156E 02	8.36836E 01
8.258E 01	1.58090E 02	9.251E 01	1.68005E 02	1.036E 02	1.11822E 01	1.161E 02	8.70195E 01
8.296E 01	1.58397E 02	9.294E 01	1.68549E 02	1.041E 02	1.28701E 01	1.166E 02	9.02080E 01
8.334E 01	1.58710E 02	9.336E 01	1.69111E 02	1.046E 02	1.46615E 01	1.172E 02	9.32435E 01
8.372E 01	1.59028E 02	9.378E 01	1.69691E 02	1.051E 02	1.65647E 01	1.177E 02	9.61215E 01
8.410E 01	1.59352E 02	9.421E 01	1.70292E 02	1.055E 02	1.85872E 01	1.182E 02	9.88440E 01
8.448E 01	1.59681E 02	9.464E 01	1.70914E 02	1.060E 02	2.07370E 01	1.188E 02	1.01413E 02
8.487E 01	1.60016E 02	9.507E 01	1.71559E 02	1.065E 02	2.30218E 01	1.193E 02	1.03832E 02
8.525E 01	1.60357E 02	9.550E 01	1.72229E 02	1.070E 02	2.54481E 01	1.199E 02	1.06109E 02
8.564E 01	1.60705E 02	9.594E 01	1.72925E 02	1.075E 02	2.80229E 01	1.204E 02	1.08251E 02
8.603E 01	1.61059E 02	9.637E 01	1.73649E 02	1.080E 02	3.07506E 01	1.209E 02	1.10265E 02
8.642E 01	1.61421E 02	9.681E 01	1.74404E 02	1.085E 02	3.36340E 01	1.215E 02	1.12158E 02
8.682E 01	1.61790E 02	9.725E 01	1.75191E 02	1.089E 02	3.66732E 01	1.220E 02	1.13939E 02
8.721E 01	1.62167E 02	9.770E 01	1.76015E 02	1.094E 02	3.98653E 01	1.226E 02	1.15614E 02
8.761E 01	1.62552E 02	9.814E 01	1.76879E 02	1.099E 02	4.32036E 01	1.232E 02	1.17192E 02
8.801E 01	1.62946E 02	9.859E 01	1.77791E 02	1.104E 02	4.66766E 01	1.237E 02	1.18680E 02
8.841E 01	1.63349E 02	9.904E 01	1.78775E 02	1.109E 02	5.02689E 01	1.243E 02	1.20083E 02
8.881E 01	1.63762E 02	9.949E 01	-1.79886E 02	1.115E 02	5.39597E 01	1.249E 02	1.21409E 02
8.921E 01	1.64185E 02	9.994E 01	4.48695E-01	1.120E 02	5.77251E 01	1.254E 02	1.22662E 02

GREATEST VALUE = 1.78775E 02   LOWEST VALUE = -1.79886E 02   INTERVAL = 3.11879E 00



PLOT(TY=SE/FR=80/TO=126/EL=LE1)

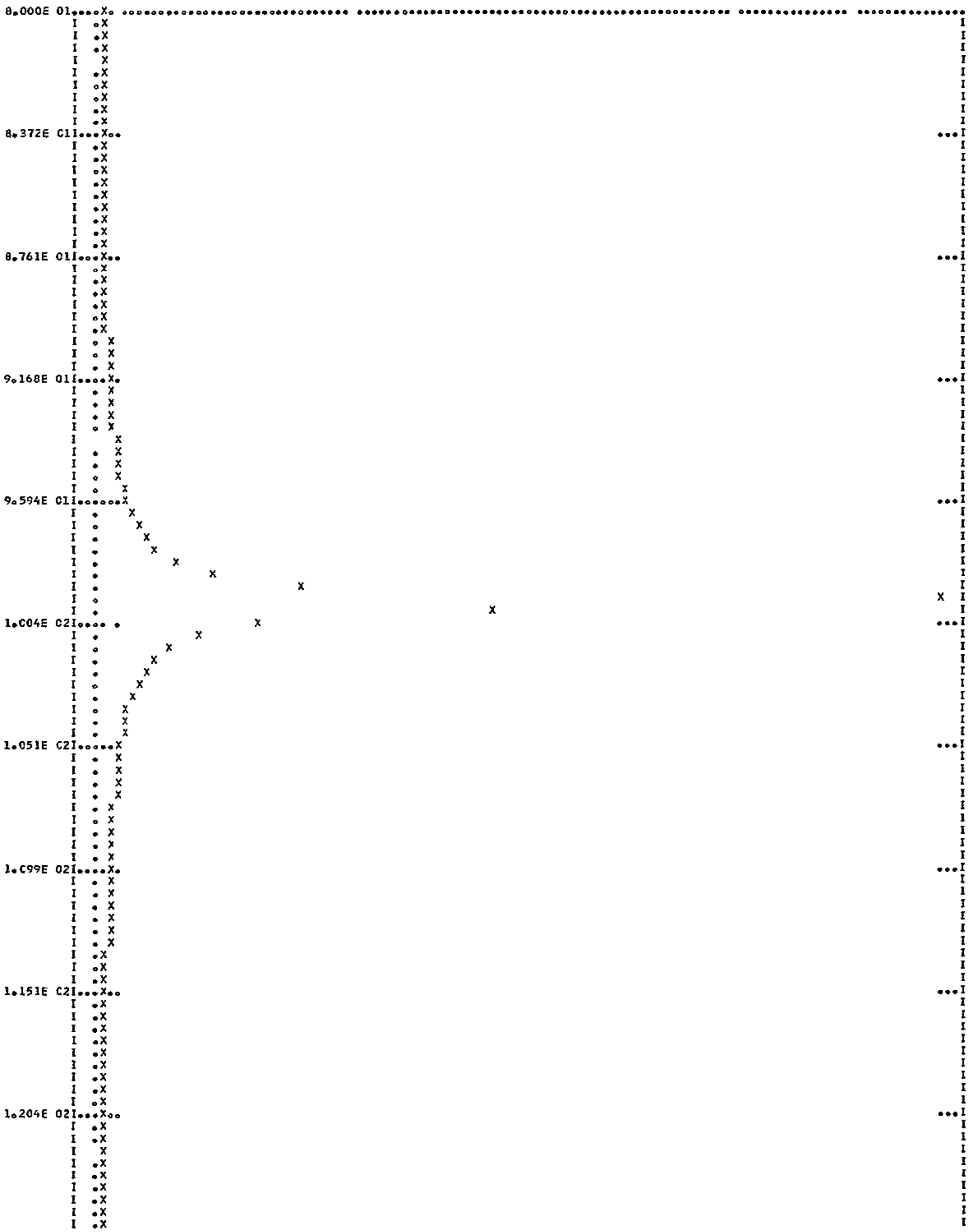
SENSITIVITY WITH RESPECT TO LE1

THE TRANSFORM OF THE RESPONSE IS ---

EXP. OF S	NUMER. COEFFS.	DENOM. COEFFS.
0	0.0	1.22316E 04
1	0.0	6.31421E 02
2	3.83959E-02	1.80477E 00
3	1.61428E-03	-4.02057E-03
4	4.45302E-06	-1.19296E-05
5	-1.43701E-08	-1.43701E-08
6	-4.19904E-11	-4.19904E-11

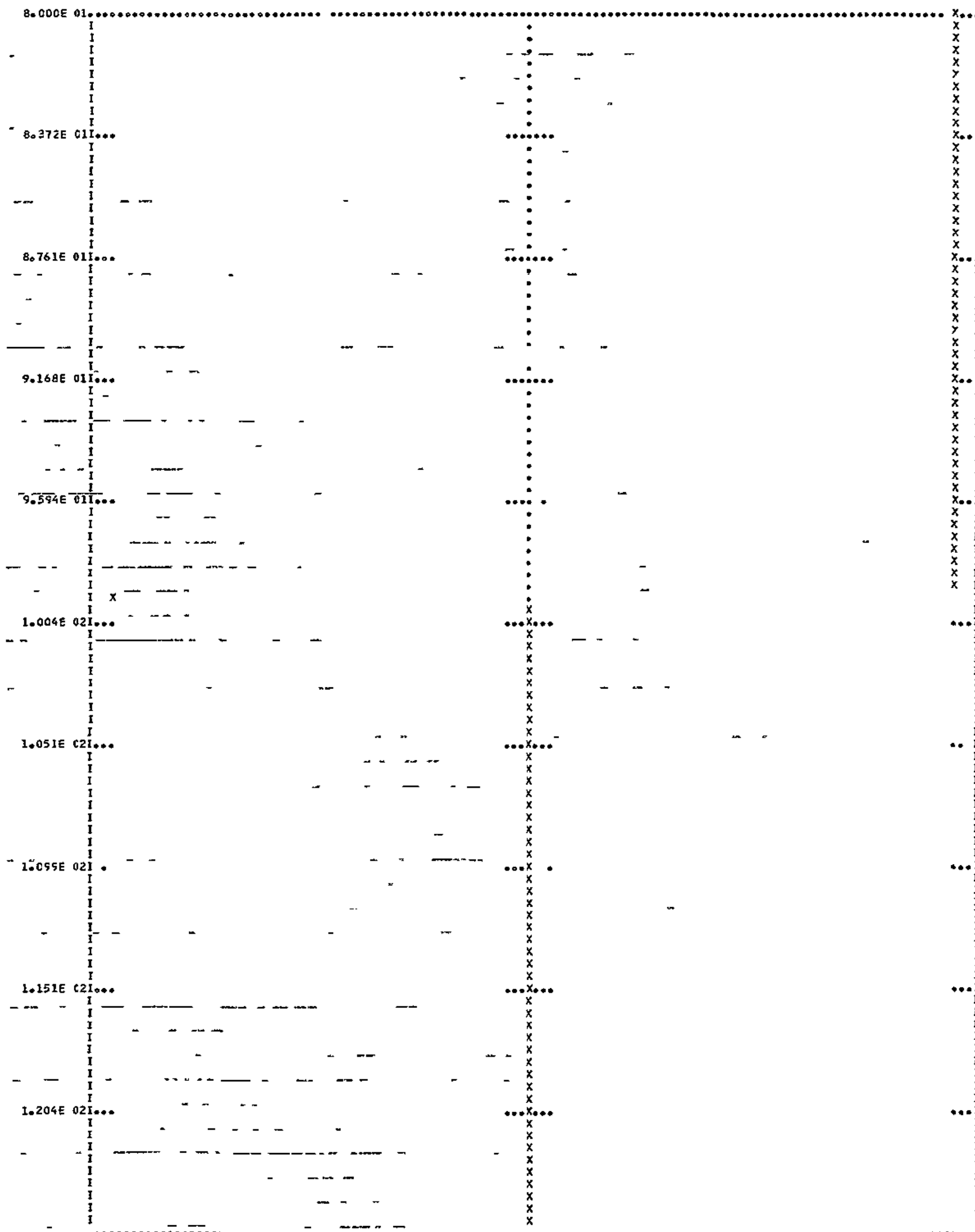
FREQ	MAG	FREQ	MAG	FREQ	MAG	FREQ	MAG
8.000E 01	1.80686E 00	8.962E 01	4.22258E 00	1.004E 02	6.57633E 01	1.125E 02	4.63352E 00
8.036E 01	1.85409E 00	9.003E 01	4.43268E 00	1.009E 02	4.14254E 01	1.130E 02	4.48574E 00
8.073E 01	1.90340E 00	9.044E 01	4.66255E 00	1.013E 02	3.03101E 01	1.135E 02	4.34826E 00
8.110E 01	1.95492E 00	9.085E 01	4.91510E 00	1.018E 02	2.39437E 01	1.140E 02	4.22014E 00
8.147E 01	2.00880E 00	9.126E 01	5.19390E 00	1.022E 02	1.98190E 01	1.145E 02	4.10041E 00
8.184E 01	2.06520E 00	9.168E 01	5.50316E 00	1.027E 02	1.69287E 01	1.151E 02	3.98827E 00
8.221E 01	2.12430E 00	9.210E 01	5.84827E 00	1.032E 02	1.47919E 01	1.156E 02	3.88307E 00
8.258E 01	2.18629E 00	9.251E 01	6.23576E 00	1.036E 02	1.31469E 01	1.161E 02	3.78412E 00
8.296E 01	2.25141E 00	9.294E 01	6.67394E 00	1.041E 02	1.18420E 01	1.166E 02	3.69095E 00
8.334E 01	2.31987E 00	9.336E 01	7.17338E 00	1.046E 02	1.07817E 01	1.172E 02	3.60302E 00
8.372E 01	2.39195E 00	9.378E 01	7.74807E 00	1.051E 02	9.90303E 00	1.177E 02	3.51995E 00
8.410E 01	2.46792E 00	9.421E 01	8.41613E 00	1.055E 02	9.16303E 00	1.182E 02	3.44131E 00
8.448E 01	2.54813E 00	9.464E 01	9.20223E 00	1.060E 02	8.53138E 00	1.188E 02	3.36679E 00
8.487E 01	2.63291E 00	9.507E 01	1.01410E 01	1.065E 02	7.98588E 00	1.193E 02	3.29606E 00
8.525E 01	2.72267E 00	9.550E 01	1.12814E 01	1.070E 02	7.51008E 00	1.199E 02	3.22885E 00
8.564E 01	2.81787E 00	9.594E 01	1.26963E 01	1.075E 02	7.09132E 00	1.204E 02	3.16487E 00
8.603E 01	2.91902E 00	9.637E 01	1.44976E 01	1.080E 02	6.72003E 00	1.209E 02	3.10394E 00
8.642E 01	3.02666E 00	9.681E 01	1.68700E 01	1.085E 02	6.38859E 00	1.215E 02	3.04585E 00
8.682E 01	3.14147E 00	9.725E 01	2.01343E 01	1.089E 02	6.09093E 00	1.220E 02	2.99038E 00
8.721E 01	3.26416E 00	9.770E 01	2.49106E 01	1.094E 02	5.82210E 00	1.226E 02	2.93738E 00
8.761E 01	3.39557E 00	9.814E 01	3.25669E 01	1.099E 02	5.57817E 00	1.232E 02	2.88668E 00
8.801E 01	3.53663E 00	9.859E 01	4.68240E 01	1.104E 02	5.35580E 00	1.237E 02	2.83813E 00
8.841E 01	3.68848E 00	9.904E 01	8.27105E 01	1.109E 02	5.15226E 00	1.243E 02	2.79161E 00
8.881E 01	3.85240E 00	9.949E 01	3.43794E 02	1.115E 02	4.96529E 00	1.249E 02	2.74700E 00
8.921E 01	4.02984E 00	9.994E 01	1.61567E 02	1.120E 02	4.79290E 00	1.254E 02	2.70416E 00

GREATEST VALUE = 3.43754E 02 LOWEST VALUE = 0.0 INTERVAL = 2.58951E 00



FREQ	PHA	FREQ	PHA	FREQ	PHA	FREQ	PHA
8.000E 01	1.79807E 02	8.962E 01	1.79856E 02	1.004E 02	-2.02724E-01	1.125E 02	-9.03177E-02
8.036E 01	1.79809E 02	9.003E 01	1.79858E 02	1.009E 02	-1.68681E-01	1.130E 02	-8.90664E-02
8.073E 01	1.79811E 02	9.044E 01	1.79860E 02	1.013E 02	-1.51651E-01	1.135E 02	-8.79326E-02
8.110E 01	1.79813E 02	9.085E 01	1.79862E 02	1.018E 02	-1.41631E-01	1.140E 02	-8.67897E-02
8.147E 01	1.79815E 02	9.126E 01	1.79864E 02	1.022E 02	-1.34740E-01	1.145E 02	-8.56667E-02
8.184E 01	1.79817E 02	9.168E 01	1.79866E 02	1.027E 02	-1.29776E-01	1.151E 02	-8.45574E-02
8.221E 01	1.79819E 02	9.210E 01	1.79868E 02	1.032E 02	-1.25441E-01	1.156E 02	-8.34391E-02
8.258E 01	1.79821E 02	9.251E 01	1.79870E 02	1.036E 02	-1.21969E-01	1.161E 02	-8.23924E-02
8.296E 01	1.79823E 02	9.294E 01	1.79873E 02	1.041E 02	-1.18838E-01	1.166E 02	-8.13072E-02
8.334E 01	1.79825E 02	9.336E 01	1.79875E 02	1.046E 02	-1.16317E-01	1.172E 02	-8.02982E-02
8.372E 01	1.79827E 02	9.378E 01	1.79877E 02	1.051E 02	-1.13863E-01	1.177E 02	-7.92832E-02
8.410E 01	1.79829E 02	9.421E 01	1.79880E 02	1.055E 02	-1.11750E-01	1.182E 02	-7.82904E-02
8.448E 01	1.79831E 02	9.464E 01	1.79882E 02	1.060E 02	-1.09711E-01	1.188E 02	-7.72662E-02
8.487E 01	1.79833E 02	9.507E 01	1.79885E 02	1.065E 02	-1.07730E-01	1.193E 02	-7.63029E-02
8.525E 01	1.79835E 02	9.550E 01	1.79888E 02	1.070E 02	-1.05946E-01	1.199E 02	-7.53452E-02
8.564E 01	1.79837E 02	9.594E 01	1.79891E 02	1.075E 02	-1.04348E-01	1.204E 02	-7.44328E-02
8.603E 01	1.79839E 02	9.637E 01	1.79895E 02	1.080E 02	-1.02695E-01	1.209E 02	-7.35039E-02
8.642E 01	1.79841E 02	9.681E 01	1.79900E 02	1.085E 02	-1.01147E-01	1.215E 02	-7.25805E-02
8.682E 01	1.79843E 02	9.725E 01	1.79906E 02	1.089E 02	-9.96634E-02	1.220E 02	-7.16563E-02
8.721E 01	1.79845E 02	9.770E 01	1.79914E 02	1.094E 02	-9.82193E-02	1.226E 02	-7.07533E-02
8.761E 01	1.79847E 02	9.814E 01	1.79925E 02	1.099E 02	-9.67665E-02	1.232E 02	-6.98642E-02
8.801E 01	1.79849E 02	9.859E 01	1.79947E 02	1.104E 02	-9.53704E-02	1.237E 02	-6.90120E-02
8.841E 01	1.79851E 02	9.904E 01	1.79997E 02	1.109E 02	-9.40755E-02	1.243E 02	-6.81387E-02
8.881E 01	1.79853E 02	9.949E 01	-1.79644E 02	1.115E 02	-9.27687E-02	1.249E 02	-6.72821E-02
8.921E 01	1.79855E 02	9.994E 01	-3.36544E-01	1.120E 02	-9.14843E-02	1.254E 02	-6.64654E-02

GREATEST VALUE = 1.79957E 02   LOWEST VALUE = -1.79644E 02   INTERVAL = 3.12731E 00



PLOT(TY=SE/FR=80/TC=126/EL=RE3)

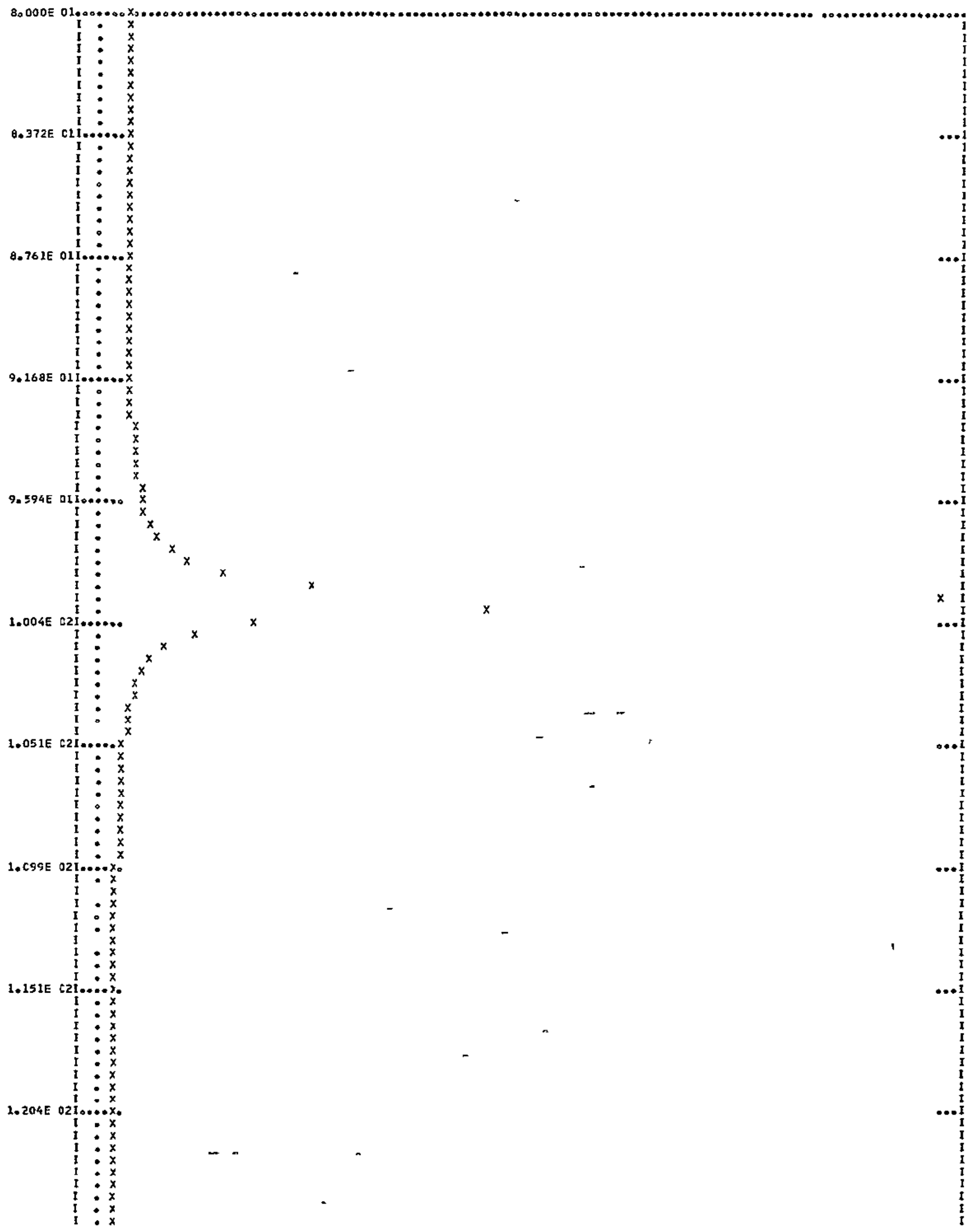
SENSITIVITY WITH RESPECT TO RE3

THE TRANSFORM OF THE RESPONSE IS ---

EXP. OF S	NUMER. COEFFS.	DENOM. COEFFS.
0	1.71602E 02	1.22316E 04
1	4.87941E 02	6.31421E 02
2	1.26981E 00	1.80477E 00
3	1.36613E-03	-4.02057E-03
4	3.30681E-06	-1.19296E-05
5	1.01059E-09	-1.43701E-08
6	0.0	-4.19904E-11

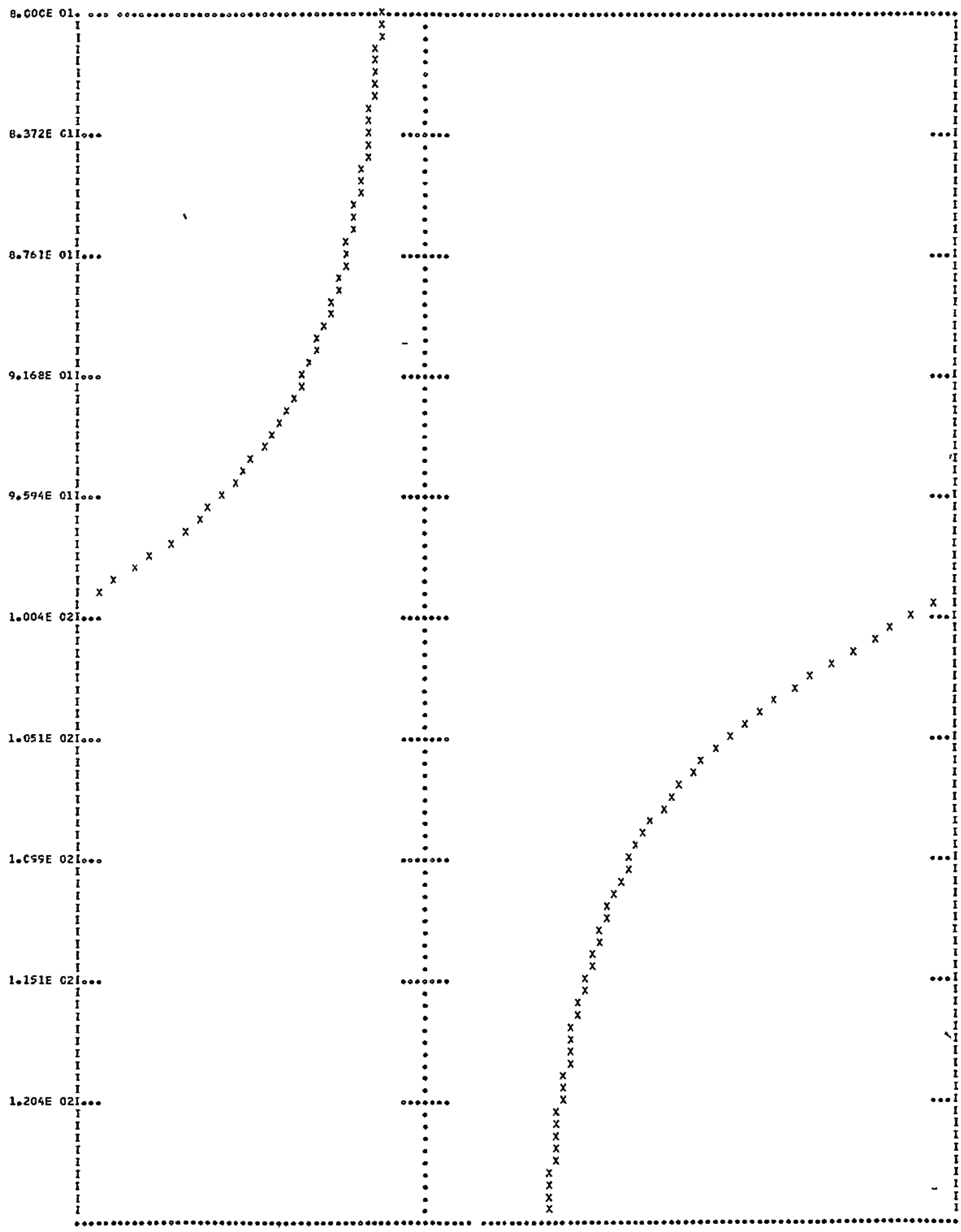
FREQ	MAG	FREQ	MAG	FREQ	MAG	FREQ	MAG
8.000E 01	2.35141E-01	8.962E 01	2.27432E-01	1.004E 02	1.18025E 00	1.125E 02	1.28338E-01
8.036E 01	2.34143E-01	9.003E 01	2.28766E-01	1.009E 02	7.31829E-01	1.130E 02	1.26760E-01
8.073E 01	2.33174E-01	9.044E 01	2.30402E-01	1.013E 02	5.30190E-01	1.135E 02	1.25272E-01
8.110E 01	2.32236E-01	9.085E 01	2.32389E-01	1.018E 02	4.17117E-01	1.140E 02	1.23864E-01
8.147E 01	2.31330E-01	9.126E 01	2.34787E-01	1.022E 02	3.45760E-01	1.145E 02	1.22525E-01
8.184E 01	2.30460E-01	9.168E 01	2.37668E-01	1.027E 02	2.97278E-01	1.151E 02	1.21246E-01
8.221E 01	2.29627E-01	9.210E 01	2.41126E-01	1.032E 02	2.62654E-01	1.156E 02	1.20019E-01
8.258E 01	2.28834E-01	9.251E 01	2.45274E-01	1.036E 02	2.36980E-01	1.161E 02	1.18839E-01
8.296E 01	2.28083E-01	9.294E 01	2.50259E-01	1.041E 02	2.17400E-01	1.166E 02	1.17701E-01
8.334E 01	2.27378E-01	9.336E 01	2.56271E-01	1.046E 02	2.02120E-01	1.172E 02	1.16598E-01
8.372E 01	2.26723E-01	9.378E 01	2.63559E-01	1.051E 02	1.89960E-01	1.177E 02	1.15529E-01
8.410E 01	2.26122E-01	9.421E 01	2.72454E-01	1.055E 02	1.80117E-01	1.182E 02	1.14489E-01
8.448E 01	2.25578E-01	9.464E 01	2.83405E-01	1.060E 02	1.72030E-01	1.188E 02	1.13476E-01
8.487E 01	2.25097E-01	9.507E 01	2.97048E-01	1.065E 02	1.65293E-01	1.193E 02	1.12487E-01
8.525E 01	2.24686E-01	9.550E 01	3.14288E-01	1.070E 02	1.59609E-01	1.199E 02	1.11520E-01
8.564E 01	2.24349E-01	9.594E 01	3.36473E-01	1.075E 02	1.54755E-01	1.204E 02	1.10572E-01
8.603E 01	2.24095E-01	9.637E 01	3.65685E-01	1.080E 02	1.50563E-01	1.209E 02	1.09642E-01
8.642E 01	2.23932E-01	9.681E 01	4.05366E-01	1.085E 02	1.46907E-01	1.215E 02	1.08730E-01
8.682E 01	2.23869E-01	9.725E 01	4.61497E-01	1.089E 02	1.43684E-01	1.220E 02	1.07833E-01
8.721E 01	2.23918E-01	9.770E 01	5.45654E-01	1.094E 02	1.40817E-01	1.226E 02	1.06950E-01
8.761E 01	2.24091E-01	9.814E 01	6.83379E-01	1.099E 02	1.38243E-01	1.232E 02	1.06081E-01
8.801E 01	2.24403E-01	9.859E 01	9.44129E-01	1.104E 02	1.35913E-01	1.237E 02	1.05225E-01
8.841E 01	2.24871E-01	9.904E 01	1.60845E 00	1.109E 02	1.33786E-01	1.243E 02	1.04380E-01
8.881E 01	2.25515E-01	9.949E 01	6.47613E 00	1.115E 02	1.31832E-01	1.249E 02	1.03546E-01
8.921E 01	2.26359E-01	9.994E 01	2.96266E 00	1.120E 02	1.30023E-01	1.254E 02	1.02722E-01

GREATEST VALUE = 6.47613E 00 LOWEST VALUE = 0.0 INTERVAL = 5.63142E-02



FREQ	PHA	FREQ	PHA	FREQ	PHA	FREQ	PHA
8.000E 01	-8.76978E 00	8.962E 01	-1.98212E 01	1.004E 02	1.02997E 02	1.125E 02	3.81304E 01
8.036E 01	-9.00136E 00	9.003E 01	-2.06726E 01	1.009E 02	9.87016E 01	1.130E 02	3.72155E 01
8.073E 01	-9.24296E 00	9.044E 01	-2.15820E 01	1.013E 02	9.43805E 01	1.135E 02	3.63607E 01
8.110E 01	-9.49521E 00	9.085E 01	-2.25545E 01	1.018E 02	9.00957E 01	1.140E 02	3.55613E 01
8.147E 01	-9.75875E 00	9.126E 01	-2.35961E 01	1.022E 02	8.58987E 01	1.145E 02	3.48129E 01
8.184E 01	-1.00342E 01	9.168E 01	-2.47129E 01	1.027E 02	8.18328E 01	1.151E 02	3.41113E 01
8.221E 01	-1.03224E 01	9.210E 01	-2.59123E 01	1.032E 02	7.79353E 01	1.156E 02	3.34532E 01
8.258E 01	-1.06240E 01	9.251E 01	-2.72018E 01	1.036E 02	7.42291E 01	1.161E 02	3.28348E 01
8.296E 01	-1.09399E 01	9.294E 01	-2.85904E 01	1.041E 02	7.07317E 01	1.166E 02	3.22536E 01
8.334E 01	-1.12712E 01	9.336E 01	-3.00870E 01	1.046E 02	6.74521E 01	1.172E 02	3.17064E 01
8.372E 01	-1.16187E 01	9.378E 01	-3.17024E 01	1.051E 02	6.43913E 01	1.177E 02	3.11913E 01
8.410E 01	-1.19835E 01	9.421E 01	-3.34471E 01	1.055E 02	6.15450E 01	1.182E 02	3.07056E 01
8.448E 01	-1.23669E 01	9.464E 01	-3.53330E 01	1.060E 02	5.89060E 01	1.188E 02	3.02475E 01
8.487E 01	-1.27700E 01	9.507E 01	-3.73725E 01	1.065E 02	5.64631E 01	1.193E 02	2.98149E 01
8.525E 01	-1.31944E 01	9.550E 01	-3.95785E 01	1.070E 02	5.42049E 01	1.199E 02	2.94063E 01
8.564E 01	-1.36415E 01	9.594E 01	-4.19641E 01	1.075E 02	5.21178E 01	1.204E 02	2.90199E 01
8.603E 01	-1.41131E 01	9.637E 01	-4.45408E 01	1.080E 02	5.01895E 01	1.209E 02	2.86545E 01
8.642E 01	-1.46107E 01	9.681E 01	-4.73197E 01	1.085E 02	4.84074E 01	1.215E 02	2.83087E 01
8.682E 01	-1.51367E 01	9.725E 01	-5.03094E 01	1.089E 02	4.67594E 01	1.220E 02	2.79812E 01
8.721E 01	-1.56931E 01	9.770E 01	-5.35143E 01	1.094E 02	4.52343E 01	1.226E 02	2.76711E 01
8.761E 01	-1.62822E 01	9.814E 01	-5.69340E 01	1.099E 02	4.38215E 01	1.232E 02	2.73771E 01
8.801E 01	-1.69065E 01	9.859E 01	-6.05557E 01	1.104E 02	4.25113E 01	1.237E 02	2.70984E 01
8.841E 01	-1.75694E 01	9.904E 01	-6.43487E 01	1.109E 02	4.12947E 01	1.243E 02	2.68341E 01
8.881E 01	-1.82737E 01	9.949E 01	-6.80084E 01	1.115E 02	4.01638E 01	1.249E 02	2.65834E 01
8.921E 01	-1.90229E 01	9.994E 01	1.07134E 02	1.120E 02	3.91111E 01	1.254E 02	2.63454E 01

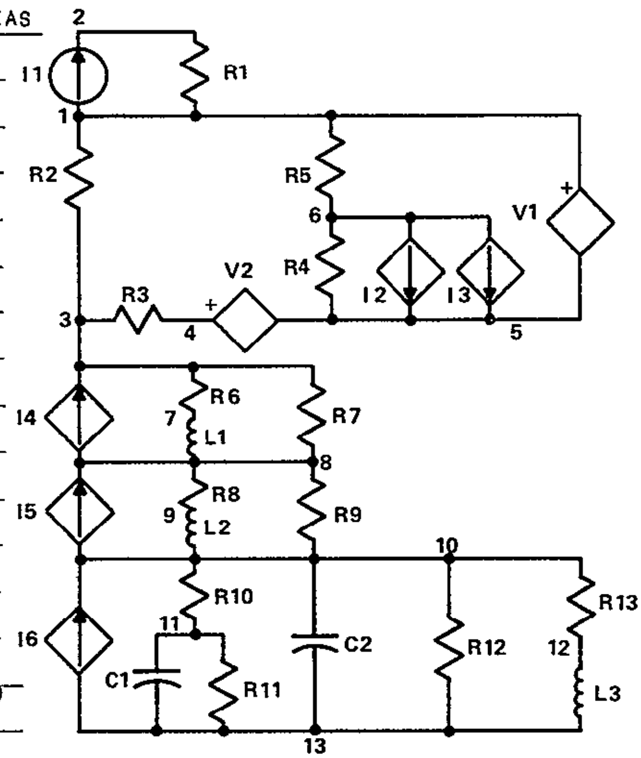
GREATEST VALUE = 1.07124E 02 LOWEST VALUE = -6.80084E 01 INTERVAL = 1.52258E 00



END

Example 2: Illustration of the (TYPE=REAL) Option (Taken from the Report, "Designed Manual for Computer-Aided Design of Communication Circuits," written by K. N. Haag, E. W. Weber, Illinois Institute of Technology, Chicago, Illinois.

```
NASAP TEMPERATURE STABILITY FIXED BIAS
V1 5 1 20 VR1
V2 5 4 1 VR9
R1 2 1 1
R2 1 3 510K
R3 3 4 1K
R4 6 5 40K
R5 1 6 5K
R6 3 7 .5
R7 3 8 10K
R8 8 9 .25
R9 8 10 10K
R10 10 11 1
R11 11 13 1.923M
R12 10 13 -.8037
R13 10 12 .0090349
C1 11 13 .02UF
C2 10 13 .007783
L1 7 8 .02
L2 9 10 -.002
L3 12 13 .019154
I1 1 2 1
I2 6 5 1 VR7
I3 6 5 1 IR10
I4 8 3 50 IR3
I5 10 8 1 VR1
I6 13 10 50 VR1
ØUTPUT
IR5/II1
PLØT(TYPE=REAL/FRØM=-5/TØ=95/STEP=1)
RØØTS
END
```



```
****NASAP****

    NETWORK ANALYSIS AND SYSTEMS APPLICATICN PRCGRAM

THIS VERSION WAS DEVELCPED AT UCLA ENGR. CEPT.


NASAP TEMPERATURE STABILITY FIXEC BIAS
V1 5 1 2C VR1
V2 5 4 1 VR9
R1 2 1 1
R2 1 3 510K
R3 3 4 1K
R4 6 5 40K
R5 1 6 5K
R6 3 7 .5
R7 3 8 10K
R8 8 9 .25
R9 8 10 10K
R10 1C 11 1
R11 11 13 1.923M
R12 10 13 -.8037
R13 1C 12 .CC90349
C1 11 13 .02UF
C2 10 13 .007783
L1 7 8 .02
L2 9 10 -.C02
L3 12 13 .019154
I1 1 2 1
I2 6 5 1 VR7
I3 6 5 1 IR10
I4 8 3 50 IR3
I5 10 8 1 VR1
I6 13 10 50 VR1
OUTPLT
```

ELEMENT NUMBER	ELEMENT NAME	DEPENDENCY (IF ANY)	ORIGIN NODE	TARGET NODE	VALUE	TAG	GENER
1	V1	VR1	5	1	2.00000000E 01	1	1
2	V2	VR9	5	4	1.00000000E 00	1	1
3	R1		2	1	1.00000000E 00	1	0
4	R2		1	3	5.10000000E 05	1	0
5	R3		3	4	1.00000000E 03	0	0
6	R4		6	5	4.00000000E 04	0	0
7	R5		1	6	5.00000000E 03	1	0
8	R6		3	7	4.99999821E-01	0	0
9	R7		3	8	1.00000000E 04	1	0
10	R8		8	9	2.49999881E-01	0	0
11	R9		8	10	1.00000000E 04	1	0
12	R10		10	11	1.00000000E 00	1	0
13	R11		11	13	1.92299800E 06	0	0
14	R12		10	13	-8.03699672E-01	1	0
15	R13		10	12	9.03489068E-03	1	0
16	C1		11	13	1.99959839E-08	0	0
17	C2		10	13	7.78299198E-03	0	0
18	L1		7	8	1.99999958E-02	1	0
19	L2		9	10	-1.99999986E-03	1	0
20	L3		12	13	1.91539861E-02	0	0
21	I1		1	2	1.00000000E 00	0	1
22	I2	VR7	6	5	1.00000000E 00	0	1
23	I3	IR10	6	5	1.00000000E 00	0	1
24	I4	IR3	8	3	5.00000000E 01	0	1
25	I5	VR1	10	8	1.00000000E 00	0	1
26	I6	VR1	13	10	5.00000000E 01	0	1

2	-3	0	2097152
3	4	32	0
4	2	0	32
5	-1	0	12583008
6	7	12582976	0
7	-18	0	256
8	9	16777216	256
9	-19	0	1024
10	11	33554432	1024
11	12	73728	0
12	15	1048576	0
13	14	67108864	1253376

FLOWGRAPH

FROM	TO	S	VALUE
29	27	0	2.00000000E 01
37	28	0	1.00000000E 00
3	29	0	1.00000000E 00
4	30	0	5.10000000E 05
31	5	0	9.99999931E-04
32	6	0	2.49999866E-05
7	33	0	5.00000000E 03
34	8	0	2.00000000E 00
9	35	0	1.00000000E 04
36	10	0	4.00000191E 00
11	37	0	1.00000000E 04
12	38	0	1.00000000E 00
39	13	0	5.20021331E-07
14	40	0	-8.03699672E-01
15	41	0	9.03489068E-03
42	16	1	1.99999839E-08
43	17	1	7.78299198E-03
18	44	1	1.99999958E-02
19	45	1	-1.99999986E-03
46	20	-1	5.22084503E 01
35	22	0	1.00000000E 00
12	23	0	1.00000000E 00
5	24	0	5.00000000E 01
29	25	0	1.00000000E 00
29	26	0	5.00000000E 01
21	3	0	1.00000000E 00
29	47	0	1.00000000E 00
5	4	0	1.00000000E 00
30	31	0	-1.00000000E 00
5	2	0	-1.00000000E 00
28	31	0	-1.00000000E 00
5	1	0	1.00000000E 00
27	31	0	1.00000000E 00
6	1	0	1.00000000E 00
27	32	0	1.00000000E 00
22	1	0	1.00000000E 00
27	48	0	-1.00000000E 00
23	1	0	1.00000000E 00
27	49	0	-1.00000000E 00
6	7	0	1.00000000E 00
33	32	0	-1.00000000E 00
22	7	0	1.00000000E 00
33	48	0	1.00000000E 00
23	7	0	1.00000000E 00
33	49	0	1.00000000E 00
8	18	0	1.00000000E 00
44	34	0	-1.00000000E 00
24	9	0	1.00000000E 00
35	50	0	1.00000000E 00
8	9	0	-1.00000000E 00
35	34	0	1.00000000E 00
10	19	0	1.00000000E 00
45	36	0	-1.00000000E 00
25	11	0	1.00000000E 00
37	51	0	1.00000000E 00
10	11	0	-1.00000000E 00
37	36	0	1.00000000E 00
13	12	0	1.00000000E 00
38	39	0	-1.00000000E 00
16	12	0	1.00000000E 00
38	42	0	-1.00000000E 00
20	15	0	1.00000000E 00
41	46	0	-1.00000000E 00
26	14	0	1.00000000E 00
40	52	0	1.00000000E 00
13	14	0	-1.00000000E 00
40	39	0	1.00000000E 00
16	14	0	-1.00000000E 00
40	42	0	1.00000000E 00
17	14	0	-1.00000000E 00
40	43	0	1.00000000E 00
20	14	0	-1.00000000E 00
40	46	0	1.00000000E 00