28

664574

DP-894

AEC RESEARCH AND DEVELOPMENT REPORT

LASS, AN IBM 704 PROGRAM FOR CALCULATING SYSTEM STABILITY

S. M. KATZ AND D. S. ST. JOHN

DU PONT
REG. U. S. PAT. OFF.

Savannah River Laboratory

Aiken, South Carolina

Printed in USA. Price $2.50
Available from the Clearinghouse for Federal Scientific
and Technical Information, National Bureau of Standards,
U. S. Department of Commerce, Springfield, Virginia

664574

DP-894

Mathematics and Computers
(TID-4500, 31st Ed.)

LASS, AN IBM 704 PROGRAM FOR CALCULATING SYSTEM STABILITY

by

Sidney M. Katz and Daniel S. St. John
Theoretical Physics Division

July 1964

E. I. DU PONT DE NEMOURS & COMPANY
SAVANNAH RIVER LABORATORY
AIKEN, SOUTH CAROLINA

*CONTRACT AT(07-2)-1 WITH THE
UNITED STATES ATOMIC ENERGY COMMISSION*

ABSTRACT

The LASS code is an IBM 704 program to compute the stability and transient response of dynamic systems. Up to 50 system equations (in up to 50 system variables) are written in linearized form (for example, by assuming small perturbations about a steady-state operating condition). After taking Laplace transforms, a system of linear, algebraic equations is obtained in which the coefficients are polynomials in the Laplace transform variable, s.

The LASS code accepts the matrix of these coefficient polynomials, reduces it to a single transfer function (ratio of polynomials in s), and calculates and plots the stability characteristics, frequency response, and transient response of the system.

A sample problem from a nuclear reactor system has been used to illustrate the use of the code. The system contained nine equations and system variables, three parameters (power level and two reactivity coefficients), and one input excitation. The 130-page output contained the transient response of two of the system variables to impulse, step, and ramp inputs for each of five power levels. The total running time on the IBM 704 was 40 minutes.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LASS, AN IBM 704 PROGRAM FOR CALCULATING SYSTEM STABILITY

INTRODUCTION

LASS (Linear Analysis of System Stability) is an IBM 704 computer program for the analysis of the behavior of linear, or linearizable, dynamic systems by means of transfer function procedures. Although written primarily for use in the study of the stability and kinetic behavior of nuclear reactors, LASS is applicable to any dynamic system that can be described by a set of linear, or nearly linear, ordinary differential equations, subject to the limitations on size to be described later.

There are many pertinent references relating to the use of transfer functions in the design and analysis of linear feedback systems(1-8), and the application of these methods to the analysis of nuclear reactors has been investigated extensively (some indication of the extent can be obtained from References 9 and 10); however, there has been little implementation of these techniques for digital computers.

Only one general-purpose program for a complete analysis has been reported(11,12). This program was originally written in machine language for the IBM 704 at the Aircraft Nuclear Propulsion Department of the General Electric Corporation Atomic Products Division(11). It was later rewritten in ALTAC-TAC for the Philco 2000 at the Knolls Atomic Power Laboratory(12) where it was given the name GALS (Generalized Analysis of Linear Systems). GALS is a four-part program that calculates

- Transfer function of the over-all system from the transfer functions of the individual components
- Stability analysis (root loci data)
- Frequency response data (Bode and Nyquist plots)
- Transient response to a step function input

Other digital computer programs are more limited in scope. The IBM 704 program ANC001(13), written in FORTRAN by Jerald Dick of the Argonne National Laboratory, calculates the transfer function of a given system and obtains the frequency response for given input frequencies. Like GALS, this program requires that the system be described by block diagram; there is a further restriction on the type of feedback loops that can be considered. Another FORTRAN program, ARL(14), from Lawrence Radiation Laboratory, provides a root locus plot when the system transfer function and various values of gain are given. Although written in FORTRAN for the IBM 7090, ARL calls upon several machine language (FAP) subroutines. At the General Electric Co., Cincinnati, Ohio, there are two Transfer Function Programs, Mark I and Mark II(15), which obtain the zeros of a matrix with polynomial elements for application to the stability analysis of control systems. The calculations of the roots of a matrix with polynomial elements (in a more restricted form, the determination of the eigenvalues of a characteristic matrix) has application to many problems other than stability analysis and there are a number of programs to achieve this calculation. One in particular is the IBM 704 program NA-299 prepared at North American Aviation(16). This program expands a matrix of polynomial elements into a polynomial equation and obtains the roots of the equation; however, it was written for use with a special monitor system and is not compatible with most IBM 704 operations. Also, multiprogram packages are used to provide a complete system analysis. Reference 17 describes very briefly work along these lines at two English locations. At the College of Aeronautics, Cranfield, five Autocode programs are available for the Ferranti Pegasus computer. These programs are used to teach control system design and apparently require very special restrictions on the nature of the systems considered. Further details of these programs have not been available at the Savannah River Laboratory (SRL). The other location is the UKAEA installation at Winfrith, England, where a number of programs(18-20) provide a four-part analysis similar to that from GALS. There is also provision to handle nonlinear systems by considering small perturbations about steady-state conditions. However, the Winfrith package involves three different computers: the IBM 7090, the Ferranti Mercury, and the ESIAC* special analog computer.

Through the courtesy of the General Electric Co., a copy of the object program deck for the IBM 704 version of GALS was obtained for use at Savannah River. When this program was applied to the analysis of nuclear reactor systems, it was found that nuclear reactors are of a higher degree of complexity than the standard feedback control systems for which GALS was written.

GALS is applicable to a system in which a single input excites only one output variable, where stability is affected by a single gain parameter, and where the over-all system can be described by a block diagram.

---

*Electro-Scientific Industries Algebraic Computer, Electro-Scientific Industries, LTD., Portland, Oregon.

Nuclear reactors, on the other hand, are multivariable systems with a number of possible input excitations, and the stability behavior frequently depends on a number of parametric quantities (for example, the various types of temperature coefficients of reactivity). Furthermore, even after linearization the system equations produce an extremely complicated block diagram which requires a large amount of hand calculation to reduce it to a form suitable for GALS. Consequently, the analysis of a reactor system with GALS is a tedious and awkward process involving a number of separate machine passes, plus some hand calculation between passes.

It was these inherent limitations in GALS that led to the LASS program. The objective of LASS was to provide a complete system analysis, similar to that of GALS, that was directly applicable to the more complex systems typified by nuclear reactors. In broad concept, LASS is a direct extension of GALS; in specific details, however, the two follow divergent paths.

SUMMARY

The LASS code is an IBM 704 program to compute the stability and transient response of dynamic systems. Up to 50 system equations (in up to 50 system variables) are written in linearized form (for example, by assuming small perturbations about a steady-state operating condition). After taking Laplace transforms, a system of linear, algebraic equations is obtained in which the coefficients are polynomials in the Laplace transform variable, s.

The LASS code accepts the matrix of these coefficient polynomials, reduces it to a single transfer function (ratio of polynomials in s), and calculates and plots the stability characteristics, frequency response, and transient response of the system.

The LASS code contains six KEY parts, each called by setting a KEY digit to 1. These KEY parts and their functions are:

- KEY 1 calls a subroutine, CALMAT, which must be coded specially for each dynamic system. This subroutine can be used to read data, calculate steady-state values, and calculate the desired matrix elements. If the matrix elements are known, KEY 1 is not used.
- KEY 2 reduces the matrix to a single characteristic equation. This is a polynomial in the Laplace variable up to 49th order. The coefficients are functions of up to four system parameters (such as a reactivity

coefficient or controller gain) that can be varied in later studies.

- KEY 3 plots the regions of stability for two of the system parameters for particular values of the other two.

- KEY 4 calculates the numerators of the transfer function from any of up to four excitation variables to as many as ten of the system variables for which the response is desired, and for particular values of the system parameters.

- KEY 5 computes the frequency response and prepares log-amplitude (Bode) and polar (Nyquist) plots.

- KEY 6 computes and plots the transient response to impulse, step, and ramp excitations.

A sample problem, taken from a nuclear reactor system, has been used to illustrate the use of the code. The system contained nine equations and system variables, three parameters (power level and two reactivity coefficients), and one input excitation. The 130-page output contained the transient response of two of the system variables to impulse, step, and ramp inputs for each of five power levels. The total running time on the IBM 704 was 40 minutes.

DISCUSSION

1. The LASS Code

1.1 Statement of the Problem

The linearized analysis of system stability (LASS) code is designed to compute the response of dynamic systems. The time-dependent integral and differential equations describing the behavior of a physical system can be reduced to a linear set of simultaneous algebraic equations by assuming small perturbations about a steady-state condition and taking Laplace transforms. The coefficients in these equations are functions of the Laplace transform variable, s. The LASS code accepts the matrix of these coefficients in which the matrix elements are polynomials in s, reduces the matrix to a characteristic equation and a number of transfer functions, and computes and plots the stability, frequency response, and transient response characteristics of the system.

We consider a physical system such as a nuclear reactor or an electronic circuit for which a set of integral and differential equations can be written to describe the time-dependent behavior. This set of equations must then be transformed into a set of linear ordinary equations. If the system contains partial differential equations, for example, these can be converted into ordinary differential equations (or the equivalent) by (a) expanding in spatial harmonic modes for which the time-dependence of the spatial harmonics satisfies a set of ordinary differential equations, (b) using difference techniques, such as solving for the temperature at a finite number of points inside a fuel element, to give a set of coupled ordinary differential equations, or (c) deriving a transcendental transfer function for that component of the system and fitting this transcendental function to a transfer function in the form of a ratio of polynomials over the frequency range of interest. Nonlinear systems can be linearized by considering only small perturbations about a steady-state operating condition. The Laplace transforms of the set of linear ordinary differential equations gives a linear set of algebraic equations in which the variables are the time-dependent variables of the physical system (or perturbations in these variables about the steady state) and the coefficients are polynomials in the Laplace transform variable, s, with numerical coefficients related to the physical constants and steady-state operating values of the system.

The LASS code accepts the matrix of coefficients of the algebraic equations and reduces this matrix to a transfer function in the form of a ratio of polynomials, P(s)/Q(s), where P and Q are polynomials in s up to 49th order. This matrix reduction corresponds to the usual block diagram reduction for obtaining the transfer function of a complicated system.

One of the valuable features of LASS is that it is possible to "save out" up to four parameters so that in the final transfer function, each coefficient of P and Q is the sum of up to 16 terms, a constant term and a constant times each of the 15 possible cross-products that are first order in each of the four parameters. These four parameters are the ones which can be varied in studying the stability properties of the system and correspond to four feedback loops with variable gain. These parameters might be such things as the power level and reactivity coefficients in a nuclear reactor or the controller settings in a system with automatic control. The value of this feature is that the majority of the computing time is spent reducing the matrix to the transfer function. Once this has been done the stability of the system for a range of settings of each of the four parameters can be computed very quickly. Excitations in up to four variables can be introduced simultaneously. For example, the transient response of the reactor power level due to changes in a pressure, a flow rate, a temperature, and a control rod setting can be calculated in one pass. For each output variable for which the transient response is desired, a separate complete computation of the numerator of the transfer function is required. Up to ten such variables can be specified and calculated in one pass.

1.2 Method of Solution

The N linear algebraic equations can be represented in matrix form as:

$$[A + \sum_{p} \alpha_p B_p]\overline{X} = V\overline{Y} \qquad (1)$$

$\overline{X}$ is an N^{th} order column vector of the Laplace transforms of the variables of the system, with element $\overline{X}_j$.

$\overline{Y}$ is an $NVAR^{th}$ order column vector of the excitation variables, with elements $\overline{Y}_q$. There can be up to 4 such excitation variables.

V is an N by NVAR matrix with constant elements V_{iq}. One advantage of LASS is that a given excitation variable can appear in any number of the original equations, and hence can act on any number of points in the system.

A and B_p are N by N matrices with elements A_{ij} and B_{pij}.

The α_p's are the parameters (up to 4) that can be varied in the subsequent stability analysis.

$$A_{ij} = \sum_{r=0}^{M} a_{ijr} s^r$$

$$B_{pij} = \sum_{r=0}^{M} b_{pijr} s^r \qquad (2)$$

M is the highest power of s that can appear in any of the matrix elements and must be no greater than four. If it is desired to introduce transfer functions into the matrix with higher powers of s, it is necessary to introduce dummy variables to meet this requirement.

The solution is obtained from the equation:

$$\overline{X}_j = \frac{\left| A + \sum_p \alpha_p B_p \right|_j}{\left| A + \sum_p \alpha_p B_p \right|} \qquad (3)$$

The denominator is the determinant of the matrix

$$\left| A + \sum_p \alpha_p B_p \right|$$

and the numerator is the determinant of the matrix obtained by replacing the j^{th} column of

$$\left| A + \sum_p \alpha_p B_p \right|$$

with the column vector $V\overline{Y}$.

Equation 3 can be written as follows:

$$\overline{X}_j = \frac{\sum_{q=1}^{NVAR} \overline{Y}_q \sum_{g=1}^{2^{NPAR}} P_{qg} \prod_{p=1}^{NPAR} \alpha_p^{m_p}}{\sum_{g=1}^{2^{NPAR}} Q_g \prod_{p=1}^{NPAR} \alpha_p^{m_p}} \qquad (4)$$

$P_{qg}(s)$ and $Q_g(s)$ are both polynomial functions of s.

$\overline{Y}_q(s)$ is the Laplace transform of the q^{th} excitation variable (a constant for an impulse, a constant times 1/s for a step input, or a constant times $1/s^2$ for a "ramp" input).

NPAR is the number (no greater than four) of parameters α_p to be varied in the later stability studies.

m_p is zero or unity for each parameter α_p.

g is related to m_p by

$$g = 1 + \sum_{p=1}^{NPAR} m_p 2^{p-1} \tag{5}$$

Thus there are up to 16 values of g corresponding to the 16 possible cross-products of the α_p that are zero or first order in each.

Equation 4 is solved in the LASS code by the evaluation of K N x N determinants, where:

$$K = \prod_{j=1}^{M} (M_j + 1)$$

M_j is the highest order of s appearing in the j^{th} column of the matrix $(A + \Sigma\ \alpha_p B_p)$. Since K can become quite large, appreciable savings in machine time may be effected by simple rearrangements of the matrix elements to reduce this product as discussed in Section 2.3.

1.3 Sections of the LASS Code

The LASS code contains six sections, any one of which may be called by setting a KEY digit on the first input card to 1. The functions are: KEY 1 prepares input data for later sections; KEY 2 computes the characteristic equation of the system (denominator of the transfer function); KEY 3 computes and plots the regions of stability and instability; KEY 4 computes the transfer function from any of the excitation variables to any of the system variables; KEY 5 computes and plots the response of any variable to a sinusoidal excitation of an excitation variable in log-amplitude and phase (Bode) and polar (Nyquist) forms; and KEY 6 computes and plots the transient response to impulse, step, or ramp (amplitude increasing linearly in time) inputs to the exciting variables.

To accomplish these objectives LASS is made up of six KEY and two subsidiary parts, as follows:

The Prolog. This section sets up branching switches to provide correct routing through the program and writes the title page for the output.

KEY 1. This part calls a subroutine, CALMAT, that is completely at the disposal of the user. If the matrix elements are known, these are entered on the input cards, and only a dummy CALMAT is used. However, for a system that will be studied in detail, it is often easier to prepare a special CALMAT subroutine in FORTRAN. This code can read data, calculate steady-state values of the system parameters, and calculate the matrix elements from these values. A dummy CALMAT is shown in the FORTRAN listing, of the complete code in Appendix I. A special CALMAT for a sample problem is shown in Section 3.4.

KEY 2. This section reduces the matrix to obtain the characteristic equation of the system (denominator of the transfer function) as a polynomial in s. This polynomial is a function of the system parameters (up to four) that can be varied in the later stability studies.

KEY 3. This section plots the regions of stability of the system. The region of stability is plotted as a function of two of the system parameters for specified values of the other two. (In this linear analysis, the system stability depends only on the denominator of the transfer function from KEY 2, and not on the excitation variables or the numerator of the transfer function.)

Pre-4. This section sets up proper constants in preparation for the later sections of the program.

KEY 4. This section calculates the numerator of the transfer function as a polynomial in s for the desired system variables, $\overline{X}_j$, and the exciting variables, $\overline{Y}_q$, for one set of values of the system parameters. KEYs 4, 5, and 6 are repeated in sequence for each set of parameters desired.

KEY 5. This section computes, tabulates, and plots the steady-state response of the variables $\overline{X}_j$ to sinusoidal excitations for a set of values of the system parameters. Log-amplitude and phase (Bode) and polar (Nyquist) plots are prepared.

KEY 6. This section computes, tabulates, and plots the transient response of the variables $\overline{X}_j$, to impulse, step function, and/or ramp inputs for each of the excitation variables.

2. Operation of LASS

2.1 Limitations

The maximum numbers of system components that can be handled by LASS may be summarized as follows:

- N, the number of dependent variables and equations, 50.
- M, the highest order of s in any element of the original matrix, 4.
- NMAX, 1 plus the order of s in the numerator or denominator of the transfer function, 50.
- NPAR, the number of parameters that can be varied in the stability study, 4.
- NVAR, the number of excitation functions, 4.
- NIV, the number of variables for which the response is calculated, 10.

In addition, each parameter that can be varied in the stability analysis, α_p, must appear in only one column of the matrix. In other words, the elements of the matrix B_p must be zero everywhere except in one column, NC. When the system is described by a block diagram, this means that each parameter must appear in only one block, and that the input to this block must be a single dependent variable. If the input is actually the sum of several dependent variables, this sum must be defined as a new dummy variable.

2.2 Operation of the LASS Code

The complete LASS program is so large that it was necessary to break it into three chained sections, LASS 1, 2, and 3, for use with a 32K IBM 704, with at least four tape units.

The first section, LASS 1, includes KEY 1 of the program. If KEY 1 is used, a special CALMAT subroutine must be coded for each new problem. LASS 2 and LASS 3 are permanent and need no further handling. Therefore, LASS 2 and LASS 3 have been placed on one tape unit, Logical Unit No. 2, while LASS 1 is on a separate tape, Logical Unit No. 1. At SRL, these programs have been loaded onto tape by using the NU LUCY loader(21) (see Appendix II).

To operate the program, the tape reel containing LASS 1 is placed on Logical Unit 1, the tape with LASS 2 and LASS 3 on Unit 2, a scratch tape on Unit 3, and a tape for BCD output on Unit 6. The data cards are placed in the card reader and the load tape button is pressed.

In operation, as each section is completed, the core data needed for the next section are read out onto Tape 3 and the next section is read in by a load tape sequence. At the end of LASS 3, LASS 1 is returned to the core so that more than one problem can be processed in sequence.

Quite often a given system may be processed at several different times, using different parts of the program each time. For example, the first pass may use KEYs 2 and 3 to obtain the stability plot, while at a later time KEYs 4, 5, and 6 may be used to obtain frequency and transient responses for various values of the system parameters. Since the output from one part may be required in a later part, the program was coded so that all output that might be used later is punched out on cards, thereby eliminating hand punching. Because several problems may be run in sequence, the punched card output deck may have cards for several problems. To facilitate separating these cards, a special card, in which the word LASS is spelled out with the punches, is produced at the beginning of each problem.

At the end of each problem, a single end-of-problem card (not shown on the input sheets) is required. This card is blank except for a 9 punch in column 1. LASS 1 starts over and calls for card read.

Finally, at the end of all problems in a sequence, three or four blank cards should be added. The first blank card indicates there are no more problems and calls for end-of-file and rewind of output Tape 6. The rest of the blank cards make it possible for all data cards to be read without operator attention.

2.3 Input Data

For the complete LASS program, eight different data sheets are required in addition to any special data needed for the CALMAT subroutine to calculate matrix elements in KEY 1. The flow of data through the program and the relationships between the parts of the program and the data sheets are summarized in Table I. The data sheets are shown in Figures 1 through 8.

Figure 1 (input data sheet for LASS, page 1) represents the KEY card and three title cards called for by the prolog.

48%

TABLE I

Data Flow for LASS

Key	Figure	Function	Input	From	Output	To
Prolog	1	Sets switches Writes title page	KEY card, title cards	Cards	Title page	Tape 6
1	*	Calculates matrix elements	System data	Cards	A,G,V matrices	KEYs 2,4 Punched cards Tape 6
2	2,3,4	Reduces matrix to denominator of transfer function	A,G matrices	KEY 1 or cards 10X	Coefficient of characteristic equation	KEYs 3,5,6 Punched cards Tape 6
3	5,6	Calculates and plots stability regions	Coefficient of characteristic equation, and range of parameters	Cards 30X and KEY 2 or cards 20X	Stability plots	Tape 6
Pre-4	7	Sets for KEYs 4,5,6	List of response variables, and values of system parameters	Card 401 Card 402	None	-
4	2,3,4	Numerator of transfer function	A,G,V matrices	KEY 1 or cards 10X	Coefficients of numerator polynomial	KEYs 5,6 Punched cards Tape 6
5	7,8,5	Response to sine input	Frequency values, transfer function numerator Transfer function denominator	Cards Figure 8; KEY 4 or cards Figure 7; KEY 2 or cards 20X	Tables, Bode and Nyquist plots	Tape 6
6	8	Response to impulse, step, or ramp input	Time values, excitation magnitudes, transfer function	Cards 6 Figure 8; KEYs 2,4 or cards 20X and Figure 7	Transient response, tables, and plots	Tape 6

---

*Requires special CALMAT subroutine (see Section 3.4)

NOTE: For one problem, KEYs 4, 5, and 6 can be repeated for as many values of the system parameters (Card 402) as desired for the same set of response variables (Card 401)

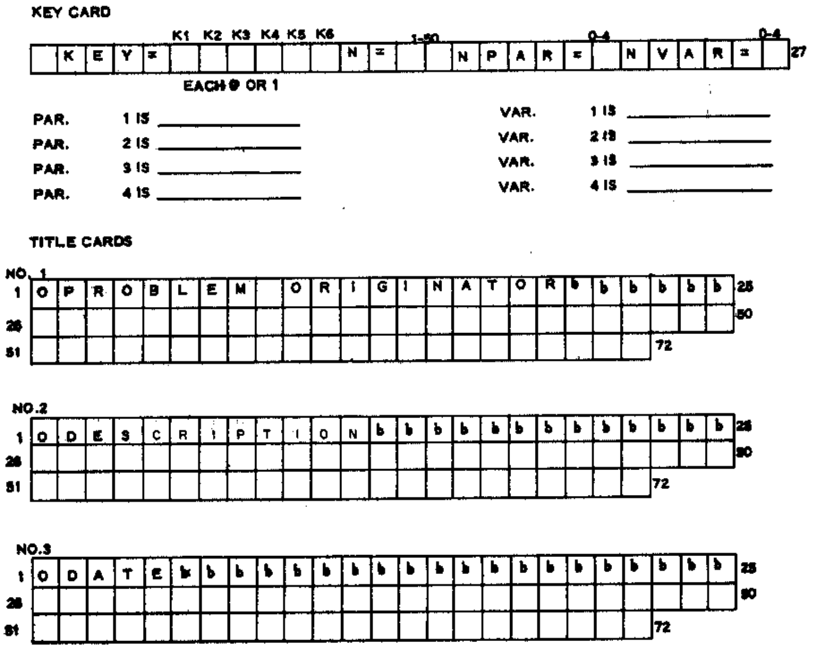


FIG. 1 INPUT DATA SHEET FOR LASS – PAGE 1

On the KEY card are:

- A 6-digit KEY word that designates which parts of LASS are to be used. Each digit must be zero or 1, with a 1 for digit K signifying that KEY K is to be used.
- A 2-digit number, N, giving the number of dependent variables or equations in the system (up to 50). This number is used explicitly only when KEYs 2 and 4 are called upon; however, it must always be non-zero (a zero or blank in this position signifies the end of all problems and sets up end of file and rewind on Tape 6 in preparation for off-line printing).
- A 1-digit number, NPAR (no greater than 4), giving the number of parameters to be varied during the stability analysis.
- A 1-digit number, NVAR (no greater than 4), giving the number of excitation variables for transient and frequency response calculations.

The information on the three title cards is placed on the output tape just as it appears on the cards. Anything at all may be written on these three cards, except that a zero must appear in position 1 for each card. The sheet shown in Figure 1, as used at SRL, provides for entering the name of the problem originator, a short description of the problem, and the date.

Figures 2, 3, and 4 represent the input data for KEYs 2 and 4 (unless provided by KEY 1), the elements of the matrices, A, B_p, and V.

Figure 2 (input data for LASS, page 2) shows the format for the A matrix data cards. As shown earlier, each matrix element is a polynomial function in the Laplace transform variable, s, and may be written as

$$A_{ij} = \sum_{r=0}^{M} a_{ijr}s^r$$

Thus, A actually represents a three-dimensional array of elements. In two dimensions, i and j, the array size is N, the number of dependent variables; this number has already been made available to the program. In the third dimension, the size is M+1, where M is the maximum power of s in any of the elements of the array.

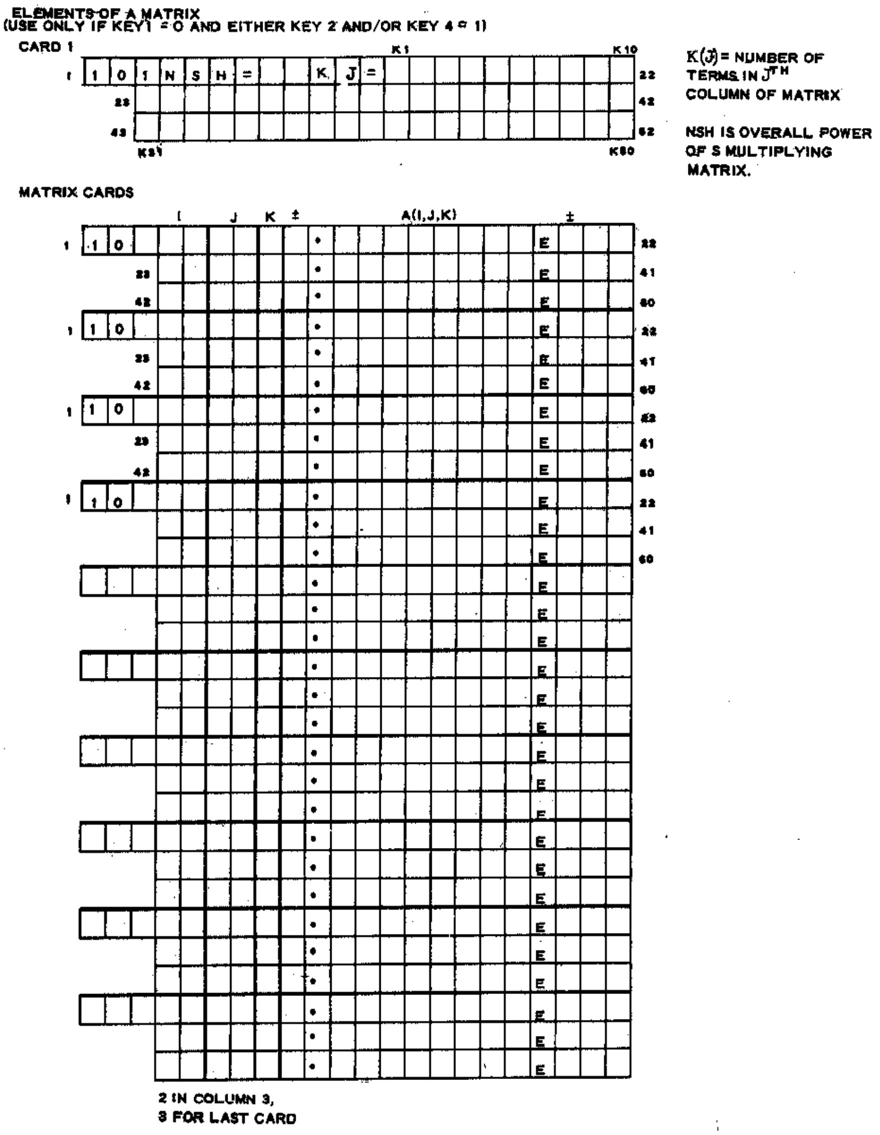


FIG. 2 INPUT DATA SHEET FOR LASS – PAGE 2

Considerable savings in machine time can be attained by specifying the size of each column of the matrix rather than that of the over-all matrix alone. If the maximum power of s in column j is M(j), then the size of this column is K(j) = M(j)+1. These single digit values of K(j) are punched in card 1 of page 2 of the data sheets (Figure 2) beginning in column 13. Each matrix column must be represented so that N digits must be punched. As stated earlier, M must be less than or equal to 4, so that each K(j) must be less than or equal to 5.

Usually, the term labeled NSH on card 1 will be left blank. In special cases it is possible to achieve a saving in machine time by using this term. These cases arise when all the polynomial functions A_{ij} in one column or row begin with some non-zero power of s. That is, if a column or row can be divided by a power of s, the matrix system can be expressed as this power of s times a reduced matrix.

When this procedure is applied column-wise, the value of K(j) for this column can be decreased by R_j, the lowest power of s appearing in the j^{th} column. The magnitude of NSH is the sum over all the columns of R_j. Since the time required to expand the matrix into the numerators and denominators of the transfer functions is directly proportional to the product of all the K(j), any decrease in these terms will speed up the calculations accordingly.

The application of this procedure to matrix rows will be beneficial only if there is a resultant decrease in one or more of the K(j) values. When carrying out the procedure row-wise, remember that the extraction of powers of s must extend over the entire row of the matrix system, including the corresponding terms in the V matrix. Since the V matrix is limited to constant terms only (that is, does not depend on s), this division can only be done on rows with V matrix elements zero.

If this reduction can be applied to both columns and rows, the columns should be reduced first, and then the rows, if still possible.

The sample problem in Section 3 illustrates this procedure.

The remaining cards in Figure 2 give the values of the matrix elements and their positions in the A matrix. The first three punches in each card are 102 except for the last card which is punched 103. The next two columns contain the index, I, of the matrix row, and the next two the index, J, of the matrix column. K is equal to r+1, that is, one greater than the power of s which A(I,J,K) multiplies. K must have a value from one to five.

Three elements are punched on each card and under no circumstances may any of the indices I, J, or K be left blank or given zero values. Therefore, if there are not enough terms to fill out the last card completely, previous terms should be repeated to provide a full card. Only non-zero values of A(I,J,K) need to be put on the cards, since the entire array is cleared before beginning the problem. The elements of the array may be punched in any order.

Figure 3 (input data sheet for LASS, page 3) shows the format for the data cards that provide the elements of the

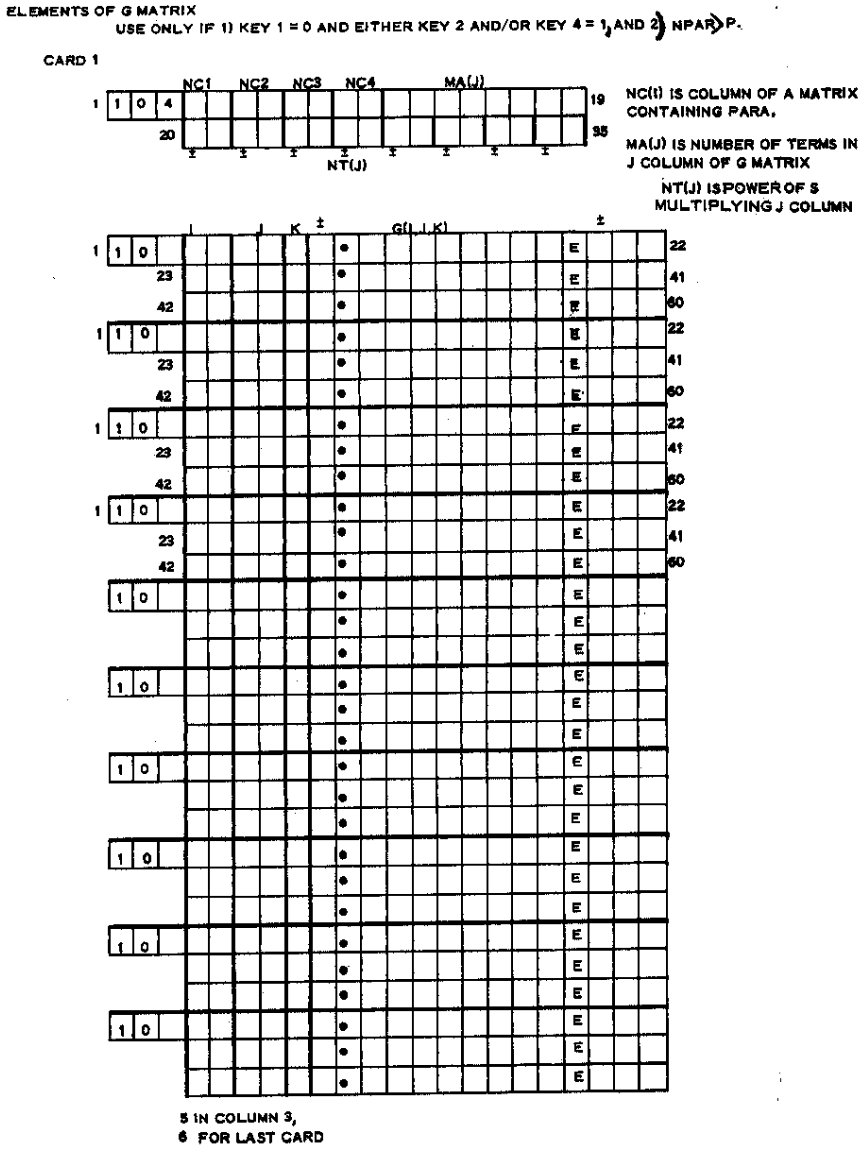


FIG. 3 INPUT DATA SHEET FOR LASS – PAGE 3

B_p matrices. This information is presented as one array, called the G matrix. This data sheet is used only if some system parameters are to be varied during the stability analysis of the transfer function generated in KEYs 2 and 4, and if these elements have not already been provided by the special CALMAT coding in KEY 1.

We recall that the matrix describing the characteristic equation is given by:

$$A + \sum_{p=1}^{NPAR} \alpha_p B_p \equiv G'$$

This is really a four-dimensional matrix. The first dimension is the number of the equation, i, describing the system. The second is the number of the system variable, j. The third is the power of s, indicated by k. The fourth indicates the parameter, α_p. This system is reduced to two three-dimensional matrices (A and B_p) by requiring that any one parameter can appear in only one column, so that the coefficients of all NPAR (up to 4) parameters appear in only NPAR columns. The G matrix is made up of these four columns of the G' matrix, including both the B_p elements and the corresponding elements of the A matrix.

The G matrix contains twice NPAR (up to eight) columns designated by J on page 3 of the input sheets (Figure 3). The first column (J = 1) contains the elements of the A matrix corresponding to the first parameter, and the second column (J = 2) contains the coefficients B_p of the first parameter. Similarly, J equal to 4, 6, and 8 contain coefficients of the second, third, and fourth parameters, while J equal to 3, 5, and 7 contain the corresponding elements of the A matrix. The terms of the A matrix must appear on input sheet 3 even though they may have also been entered (unnecessarily) on input sheet 2.

The I values of the G matrix are the same as the A matrix. The K values are one more than the power of s as in the A matrix.

The first card shown in Figure 3 contains the following:

- The number NC(p) of the column in the original matrix in which the parameter, α_p, appears. This provides the cross-identification of the J values in the original matrix and those in the G matrix.
- The maximum number of terms (one plus the highest power of s) in each column of the G matrix. There is room for eight one-digit numbers on the card. Since two (one for the A matrix and one for the B_p matrix) are required for each parameter. These quantities, called MA(J) in Figure 3, have the same purpose as the K(J) terms of the first card of Figure 2.
- The quantities NT(J) are related to the NSH term of Figure 2. On some occasions it may turn out that in the column-wise or row-wise division by a power of s, one of the parameters becomes divided by the power of s. This is allowed and is the only place in the original matrix where negative powers of s are allowed. To compensate for this, the corresponding NT(J) is set equal to this

power. There is room on the card for eight one-digit values together with the sign. The first of each pair corresponding to parameter, α_p, is zero while the second of each pair is zero or a negative number corresponding to a column in the B_p matrix. Zero values may be left blank. The sample problem in Section 3 illustrates this procedure.

The remaining cards in Figure 3 contain the values of the G matrix elements and their positions in the G array. I is the number of the equation in which the term appears, J goes from 1 to a maximum of 8 as explained above, and K is one greater than the power of s in the term. As with the A matrix, the elements may be written in any order and each card must contain three elements so that repetition of the last element may be required to fill out the last card. The first three columns of each card contain 105 except for the last card, which contains 106. These are used for identification and checking.

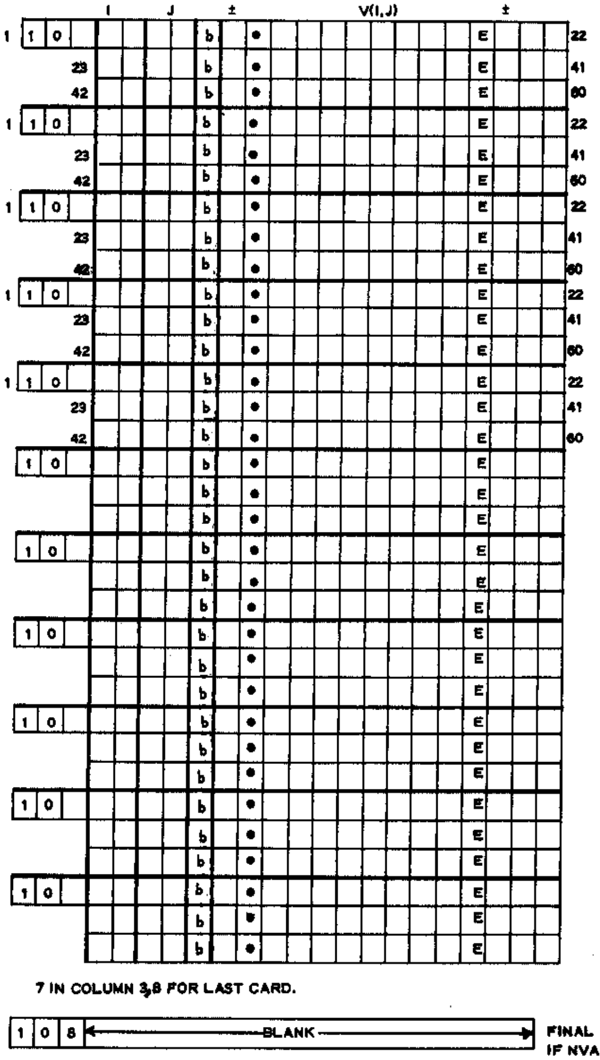
Figure 4 (input data for LASS, page 4) provides for the elements of the V matrix for the excitation variables needed for the numerator of the transfer function, KEY 4 for transient and frequency response calculations.

The elements of V are entered and indexed as they were in the A matrix, except that since all of the elements are constants (not functions of s) the column corresponding to K is left blank. (Requiring these elements to be constant means that excitations cannot be entered through integrating or differentiating circuits. Dummy variables can be introduced if needed to meet the requirement.) I is the number of the original equation, as in the A matrix. J is the number of the excitation variable (1 to 4) as identified by the variable number on page 1 (Figure 1). The first three digits on each card are used for identification purposes with 107 punched in each card except for the last card which carries 108. Each card must contain three elements with the last element repeated if necessary to fill out the last card.

If there are no excitation variables (NVAR = 0) there are no elements of the V matrix. In this case it is necessary to insert a card with 108 punched in the first three columns (and blank otherwise) for identification.

Figure 5 (input data sheet for LASS, page 5) provides for the introduction of a characteristic equation (denominator of the transfer function) that has been obtained from an earlier run or by other means, and for which stability or transient response calculations are desired.

ELEMENTS OF V MATRIX
(USE ONLY IF KEYI = 0 AND EITHER KEY2 AND/OR KEY4 = 1. IF NVAR =0, USE LAST CARD ONLY.)



7 IN COLUMN 3,8 FOR LAST CARD.

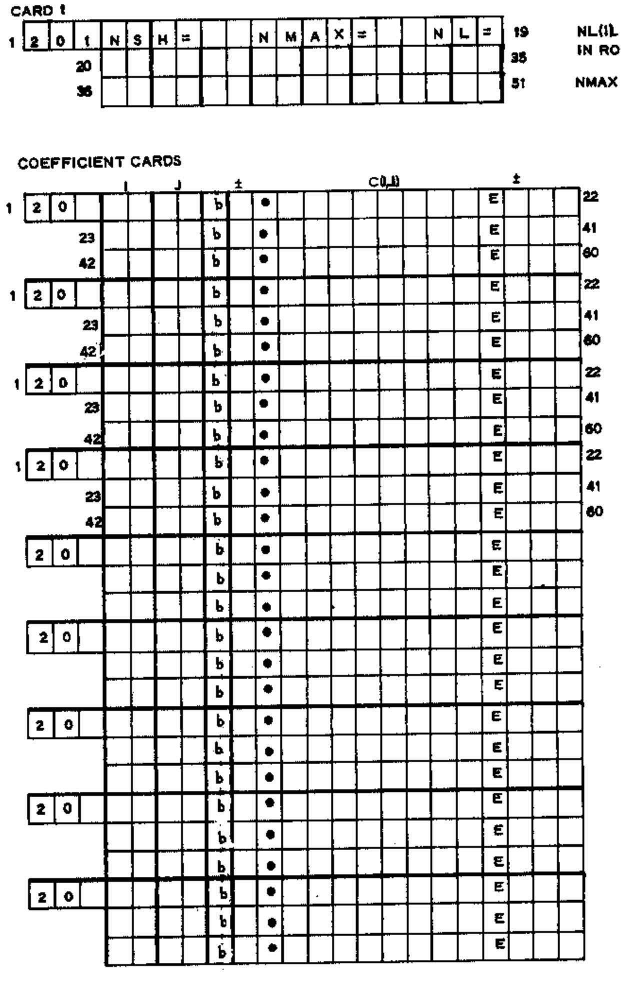
FINAL CARD, USE ONLY IF NVAR = 0

FIG. 4 INPUT DATA SHEET FOR LASS – PAGE 4

COEFFICIENTS OF CHARACTERISTIC EQUATION:

$$Q(S) = \sum_{J=1}^{NL} (C_{1,J} \pm C_{2,J}\,\alpha_1 \pm C_{3,J}\,\alpha_2 + \ldots\ldots C_{16,J}\,\alpha_1\alpha_2\alpha_3\alpha_4)\, S^{J-1}$$

(USE ONLY IF KEY2 = 0 AND EITHER KEY3 AND/OR KEY5 AND/OR KEY6 = 1.)



NL(I) IS NUMBER OF TERMS IN ROW I

NMAX IS MAXIMUM OF NL(I)

2 IN COLUMN 3, 3 FOR LAST CARD

FIG. 5 INPUT DATA SHEET FOR LASS – PAGE 5

This equation, Q(s) (Section 1.2, Equation 4), is a polynomial in s. Each of the coefficients in this polynomial can contain up to 16 terms corresponding to all of the possible combinations of the products of the zero and first powers of the four system parameters to be varied during the stability calculations. Thus, the coefficients C_{ij} form a matrix with the value of j one greater than the corresponding power of s.

NL(I) is the maximum value of j (one more than the highest power of s) in the i^{th} row of the C matrix. There is room on the card for 16 two-digit values of NL(I). Only non-zero values of NL need be provided.

NMAX is the maximum of the NL(I) values. NSH is not used, and can be left blank.

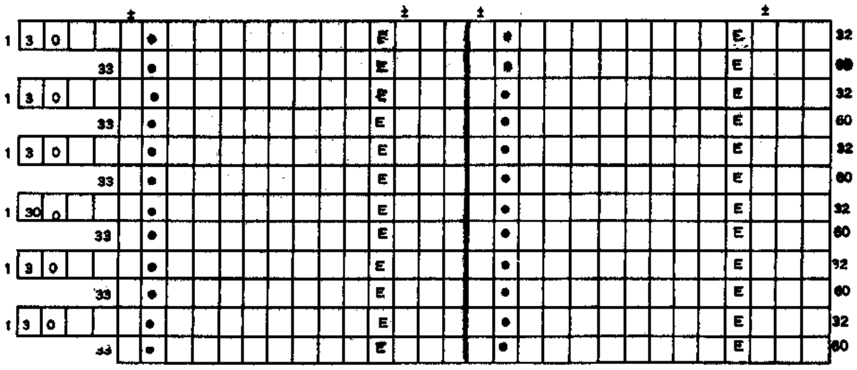
The remaining cards in Figure 5 are for inserting the elements C(I,J) as on Figure 4. The elements may be entered in any order and the last card must be filled even if it is necessary to repeat some of the elements. The first three punches contain 202 for all of the cards except for the last card which contains 203 for identification and checking purposes.

Figure 6 (input data sheet for LASS, page 6) contains the information required to plot the stability of the characteristic equation for ranges of the parameters. If the system has three or four parameters, two cards are required per plot, while for a one- or two-parameter system, only the second card is required.

For a three- or four-parameter system the two cards each contain two special identification digits. In the first card, column 3 contains a 4 for all plots except the last one and a 5 for the last plot desired in the sequence. Column 4 contains a 1 to obtain a plot of parameter 2 versus parameter 1, or a 2 for a plot of parameter 4 versus parameter 3. Columns 5 through 32 contain the values of the parameters 3 and 4 (or 1 and 2) at which the plot of parameter 2 versus parameter 1 (or parameter 4 versus parameter 3) is to be obtained.

The second card for each plot contains a 2 in column 3 and a 0 in column 4. Columns 5 through 32 contain the lower and upper values of the X coordinate (parameter 1 or parameter 3) and columns 33 through 60 contain the lower and upper values of the Y coordinate (parameter 2 or parameter 4).

Figure 7 (input data sheet for LASS, page 7) contains the information required to obtain frequency and transient response calculations and plots.



3 OR 4 PARAMETER SYSTEM: TWO CARDS PER PLOT

CARD 1: COLUMN 3 - 4 FOR ALL PLOTS EXCEPT LAST ONE
5 FOR LAST PLOT.

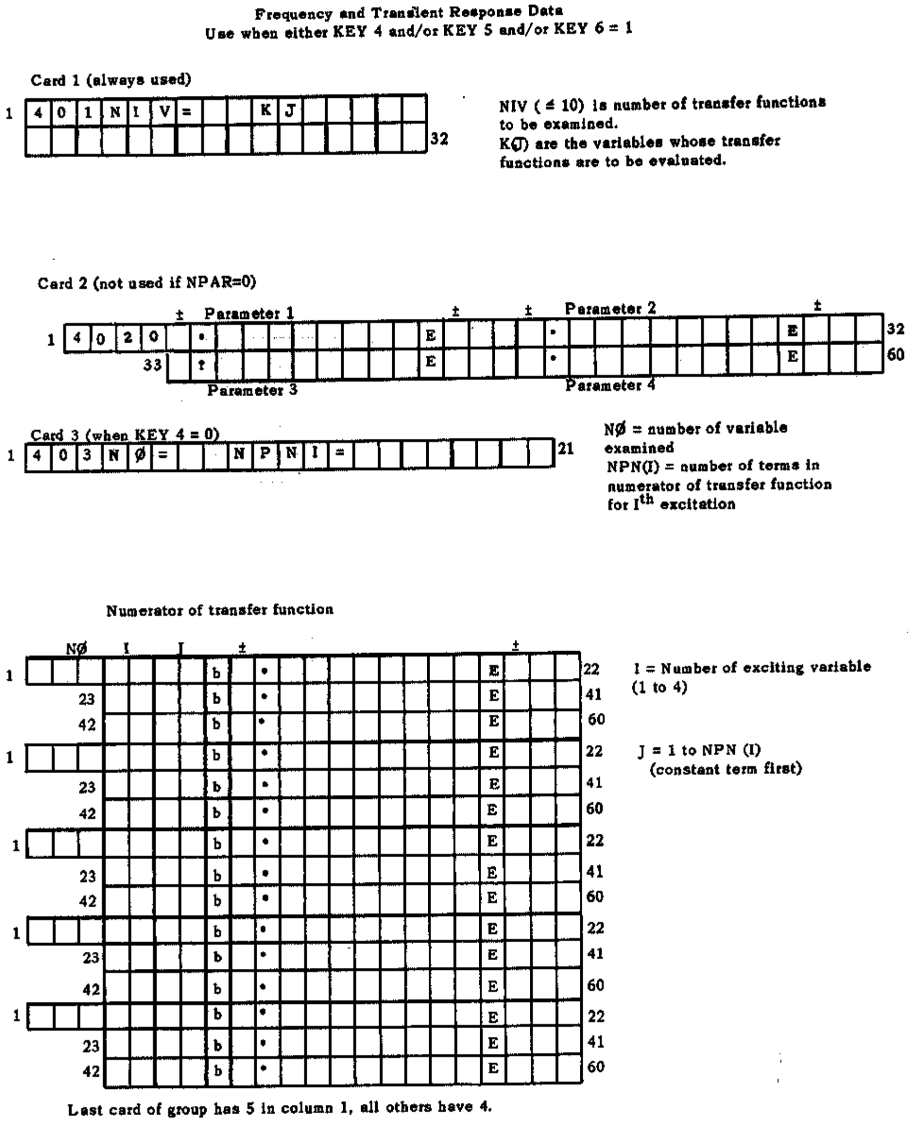
COLUMN 4 - 1 FOR PLOT OF PARAMETER 2 vs 1
2 FOR PLOT OF PARAMETER 4 vs 3
OR FOR STABILITY OF PARAMETER 3 ALONE.

COLUMNS 5 THROUGH 32 - VALUES OF PARAMETERS
3 AND 4 (1 AND 2) AT WHICH PLOT OF 2 vs 1
(4 vs 3) IS OBTAINED

CARD 2: COLUMN 3 - 2
COLUMN 4 -0
COLUMNS 5 THROUGH 32 - LOWER AND UPPER VALUES OF X CO-ORDINATE
COLUMNS 33 THROUGH 60 - LOWER AND UPPER VALUES OF Y
CO-ORDINATE

1 OR 2 PARAMETER SYSTEM: ONE CARD ONLY, SAME AS FOR CARD 2 ABOVE.

FIG. 6 INPUT DATA SHEET FOR LASS – PAGE 6



NIV (≤ 10) is number of transfer functions to be examined.
K(J) are the variables whose transfer functions are to be evaluated.

NØ = number of variable examined
NPN(I) = number of terms in numerator of transfer function for Ith excitation

I = Number of exciting variable (1 to 4)

J = 1 to NPN (I) (constant term first)

Last card of group has 5 in column 1, all others have 4.

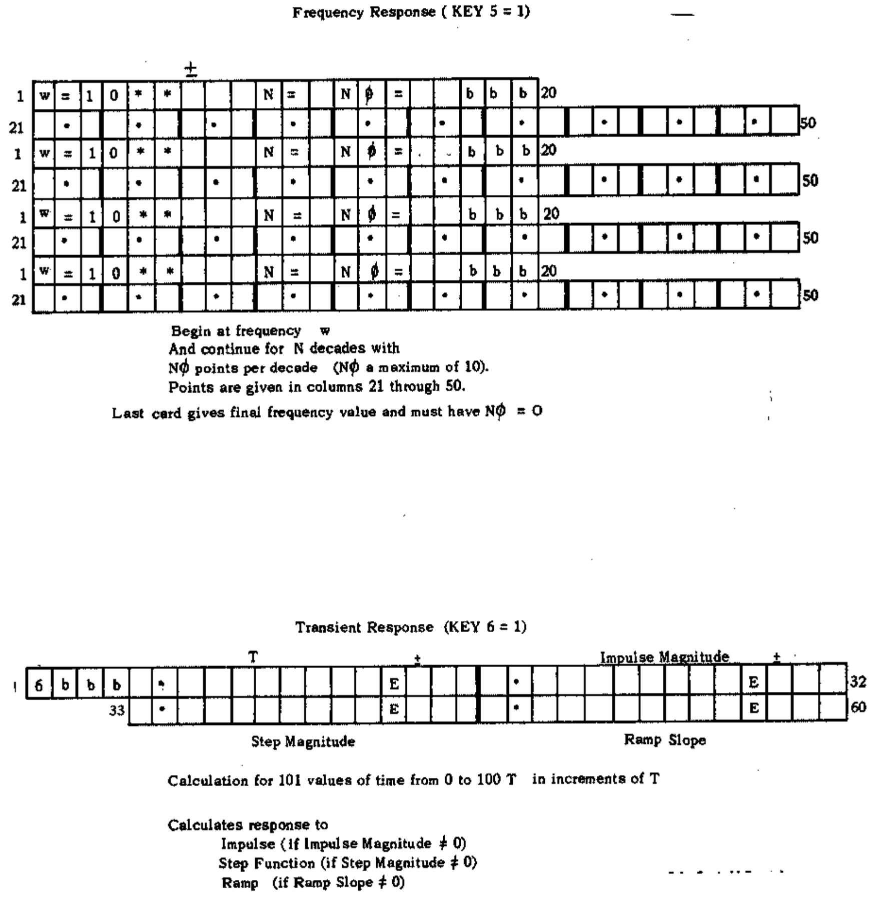
FIG. 7 INPUT DATA SHEET FOR LASS - PAGE 7

The first card contains the number NIV (no greater than 10) of system variables for which the frequency or transient response is desired and the two-digit identification K(I) of each variable (the value of J in the $\overline{X}_j$ vector). The response will be computed for each of these variables and for each of the NVAR excitation variables indicated on Figure 1 and Figure 4. Thus, if NIV is 6 and NVAR is 3 a total of 18 frequency or transient response plots will be obtained.

The second card lists the values of the parameters at which the frequency and transient response is to be calculated. A new card 1 must be employed with each set of values of the parameters. Card 2 is not required if NPAR is 0.

Card 3 and the remaining cards on page 7 are used if the numerator of the transfer function is not supplied by KEY 4 but has been obtained from an earlier run or other means. Card 3 has the number of the variable for which the numerator of the transfer function is being supplied and up to four values of NPN(I), the number of terms (one more than the highest order of s) in the numerator of the transfer function for the I^{th} excitation. The remaining cards are identified by a 4 in column 1 for all the cards in the group except for the last card which has a 5 in column 1 and by the two-digit number of the variable to be examined. The coefficients of the numerator of the transfer function are inserted as a two-dimensional array where I corresponds to the number of the excitation variable (1 to 4) and J corresponds to one more than the power of s (J goes from 1 to NPN(I)).

Figure 8 (input data sheet for LASS, page 8) contains the ranges of frequencies for the frequency response calculation and the magnitudes of the excitations for the transient response.



Begin at frequency w
And continue for N decades with
Nϕ points per decade (Nϕ a maximum of 10).
Points are given in columns 21 through 50.
Last card gives final frequency value and must have Nϕ = 0

Calculation for 101 values of time from 0 to 100 T in increments of T

Calculates response to
Impulse (if Impulse Magnitude ≠ 0)
Step Function (if Step Magnitude ≠ 0)
Ramp (if Ramp Slope ≠ 0)

FIG. 8 INPUT DATA SHEET FOR LASS – PAGE 8

The frequency response calculation begins at the power of 10 indicated at the beginning of the first card and continues to higher frequencies for N decades computed at NØ (a maximum of 10) points per decade. Columns 21 through 50 allow for ten two-digit numbers (from 1.0 to 9.9) corresponding to the points at which the frequency response will be calculated. Intermediate cards can be used as desired to change the spacing of points within a decade at which the calculations are carried out. The last card gives the highest frequency value and must have N equal to 0.

If it is desired for example, to have 20 points in the frequency decade from 10^3 to 10^4 radians/sec, two cards with $w = 10^3$, $N = 1$, and $NØ = 10$ can be used with the first ten values listed in columns 21 through 50 of the first card and the next ten values listed on the second card.

The transient response card at the bottom of Figure 8 contains the time increment for which the transient response will be calculated and plotted. The calculation will be carried out for a time from 0 to 100T. The remaining spaces on the card are for the magnitudes of the impulse, step, and ramp excitations as desired to initiate the transient response.

3. Sample Problem

3.1 Derivation of Matrices

As a sample problem we consider a portion of the stability analysis of the HWCTR$^{(26)}$. It was desired to obtain the transfer function from the D_2O outlet from the steam generator through the reactor and back to the D_2O inlet to the steam generator and also from the steam generator outlet to the reactor power to help in studying the coupling between the steam generator feedwater control loops and the reactor itself.

The coolant flow diagram is shown in Figure 9. The differential equations that approximate the transient response of the components of the system are as follows:

Differential Equations:

30%

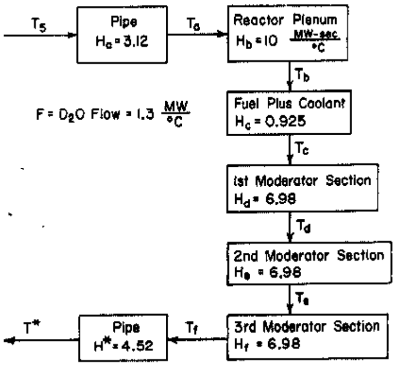


FIG. 9 SAMPLE PROBLEM – COOLANT FLOW DIAGRAM

$$
\begin{aligned}
H_a\dot{T}_a &= F(T_5 - T_a)\\
H_b\dot{T}_b &= F(T_a - T_b)\\
H_c\dot{T}_c &= F(T_b - T_c) + f\Psi\\
H_d\dot{T}_d &= F(T_c - T_d) + W_1(0.5 - f)\Psi\\
H_e\dot{T}_e &= F(T_d - T_e) + W_2(0.5 - f)\Psi\\
H_f\dot{T}_f &= F(T_e - T_f) + W_3(0.5 - f)\Psi\\
H^*\dot{T}^* &= F(T_f - T^*)\\
\Psi &= \Psi^o\phi\\
\overline{\Phi} &= G_o(s)\overline{k} \quad \text{(Laplace transformed)}\\
k &= \alpha_3(T_c - T_c^o) + \alpha_2 W_4(T_d - T_d^o)\\
&\quad + \alpha_2 W_5(T_e - T_e^o) + \alpha_2 W_6(T_f - T_f^o)
\end{aligned} \tag{7}
$$

In these equations the T's are the temperatures leaving the blocks indicated in Figure 9, the H's are the heat capacities of the components (in MW-sec/°C), F is the D_2O flow (in MW/°C), f is the fraction of the pile power transmitted from the fuel to coolant, and (0.5-f) is the fraction deposited directly in the moderator by gamma and neutron heating (f = 0.45 — the total heat deposited is only 0.5 of the pile power, Ψ, because only one of the two coolant loops is considered). Ψ is the pile power, Ψ^o is the steady-state operating power, and Φ is the flux level relative to the steady state. $G_o(s)$, the transfer function from reactivity to flux level, depends upon the delayed neutron fraction and is known from other work. The reactivity k depends upon the fuel and coolant temperature coefficient, α_3 and the moderator temperature coefficient α_2. The weight functions W_1, W_2, and W_3 correspond to the volume fractions of the three moderator sections; and W_4, W_5, and W_6 correspond to their reactivity statistical weights.

We linearize the equations by considering only small perturbations about the steady-state operating power and introduce a dummy variable, T_m, to meet the requirement that the parameter α_2 appears in only one column of the matrix. This gives the following linear algebraic equations in the Laplace transformed variables:

$$
\begin{aligned}
(F+H_a s)\overline{T}_a &= F\overline{T}_5 \\
- F\overline{T}_a + (F+H_b s)\overline{T}_b &= 0 \\
- F\overline{T}_b + (F+H_c s)\overline{T}_c - f\overline{\Psi} &= 0 \\
- F\overline{T}_c + (F+H_d s)\overline{T}_d - W_1(0.5-f)\overline{\Psi} &= 0 \\
- F\overline{T}_d + (F+H_e s)\overline{T}_e - W_2(0.5-f)\overline{\Psi} &= 0 \\
- F\overline{T}_e + (F+H_f s)\overline{T}_f - W_3(0.5-f)\overline{\Psi} &= 0 \\
- F\overline{T}_f + (F+H^* s)\overline{T}^* &= 0 \\
\overline{\Psi} - \Psi^o\overline{\Phi} &= 0 \\
(0.00875s+0.00739s^2)\overline{\Phi} + (-0.0943-1.45s-s^2)\overline{k} &= 0 \\
- \alpha_3\overline{T}_e - \alpha_2\overline{T}_m + \overline{k} &= 0 \\
\overline{T}_m - W_4\overline{T}_d - W_5\overline{T}_e - W_6\overline{T}_f &= 0
\end{aligned}
\qquad (8)
$$

Equations 8 are 11 linear algebraic equations in the 11 variables of the system. There is one excitation variable, T_5, and we wish to consider three parameters for stability studies, Ψ^o, α_2, and α_3. When we substitute numerical values for the other constants of the system, we obtain the matrix equation shown in Figure 10.

The bottom row of Figure 10 shows the values of K(J), one plus the maximum power of s in the j^{th} column. The product of these values, $2^7 \times 3^2 = 1152$, is the number of determinants to be evaluated by the code, and is a measure of the computing time required.

41%

$i \backslash j$	T_a 1	T_b 2	T_c 3	T_d 4	T_e 5	T_f 6	T^* 7	Ψ 8	Φ 9	k 10	T_m 11	Excitation Variable, T_s
1	$F+H_a s$ +1.3 +3.12s											F +1.3
2	-F	$F+H_b s$ +1.3 +10s										
3		-F -1.3	$F+H_c s$ +1.3 +0.925s					-f -0.45				
4			-F -1.3	$F+H_d s$ +1.3 +6.98s				$-W_1(0.5-f)$ -0.014				
5				-F -1.3	$F+H_e s$ -1.3 +6.98s			$-W_2(0.5-f)$ -0.022				
6					-F -1.3	$F+H_f s$ +1.3 +6.98s		$-W_3(0.5-f)$ -0.014				
7						-F -1.3	$F+H^*s$ +1.3 +4.52s					
8								1	$-\Psi^o$			
9									$A_{992}s$ $+A_{993}s^2+$ 0.00875s+ $0.00739s^2$	$A_{9,10,1}$ $+A_{9,10,2}s$ $+A_{9,10,3}s^2$ -0.0943 $-1.45s-s^2$		
10			$-\alpha_3+$ $2.65x10^{-8}$							1 1	$-\alpha_2$ $5x10^{-4}$	
11				$-W_4$ -0.224	$-W_5$ -0.552	$-W_6$ -0.224					1 1	
K(J)	2	2	2	2	2	2	2	1	3	3	1	

FIG. 10 SAMPLE PROBLEM – INPUT MATRIX

This time can be reduced by eliminating the variables $\overline{T}_a$ and $\overline{T}_f$ from Equations 8 to give Equations 8'.

$$
\begin{aligned}
(F+H_a s)(F+H_b s)\overline{T}_b &= F^2\overline{T}_s \\
-F\overline{T}_b + (F+H_c s)\overline{T}_c - f\overline{\Psi} &= 0 \\
-F\overline{T}_c + (F+H_d s)\overline{T}_d - W_1(0.5-f)\overline{\Psi} &= 0 \\
-F\overline{T}_d + (F+H_e s)\overline{T}_e - W_2(0.5-f)\overline{\Psi} &= 0 \\
-F\overline{T}_e + (F+H_f s)(F+H^* s)(1/f)\overline{T}^* - W_3(0.5-f)\overline{\Psi} &= 0 \qquad (8') \\
\overline{\Psi} - \Psi^o\overline{\Phi} &= 0 \\
(0.00875s+0.00739s^2)\overline{\Phi} + (-0.0943-1.45s-s^2)\overline{k} &= 0 \\
-\alpha_3\overline{T}_c - \alpha_2\overline{T}_m + \overline{k} &= 0 \\
\overline{T}_m - W_4\overline{T}_d - W_5\overline{T}_e - (W_6/F)(F+H^* s)\overline{T}^* &= 0
\end{aligned}
$$

The modified matrix is shown in Figure 11. The product of the K(J) has been further reduced by noting that column 7 (column 9 of Figure 10) contained no constant terms in the A matrix. Thus we can divide this column by s, reduce K(7) from 3 to 2, and make NSH equal to 1 on card 1 of page 2 of the input sheets. This can be done even though there was a constant term, -1 (times Ψ^o) in the B matrix in this column. This is handled by making the appropriate NT(J) equal to -1 in card 1 of page 3 of the input sheets.

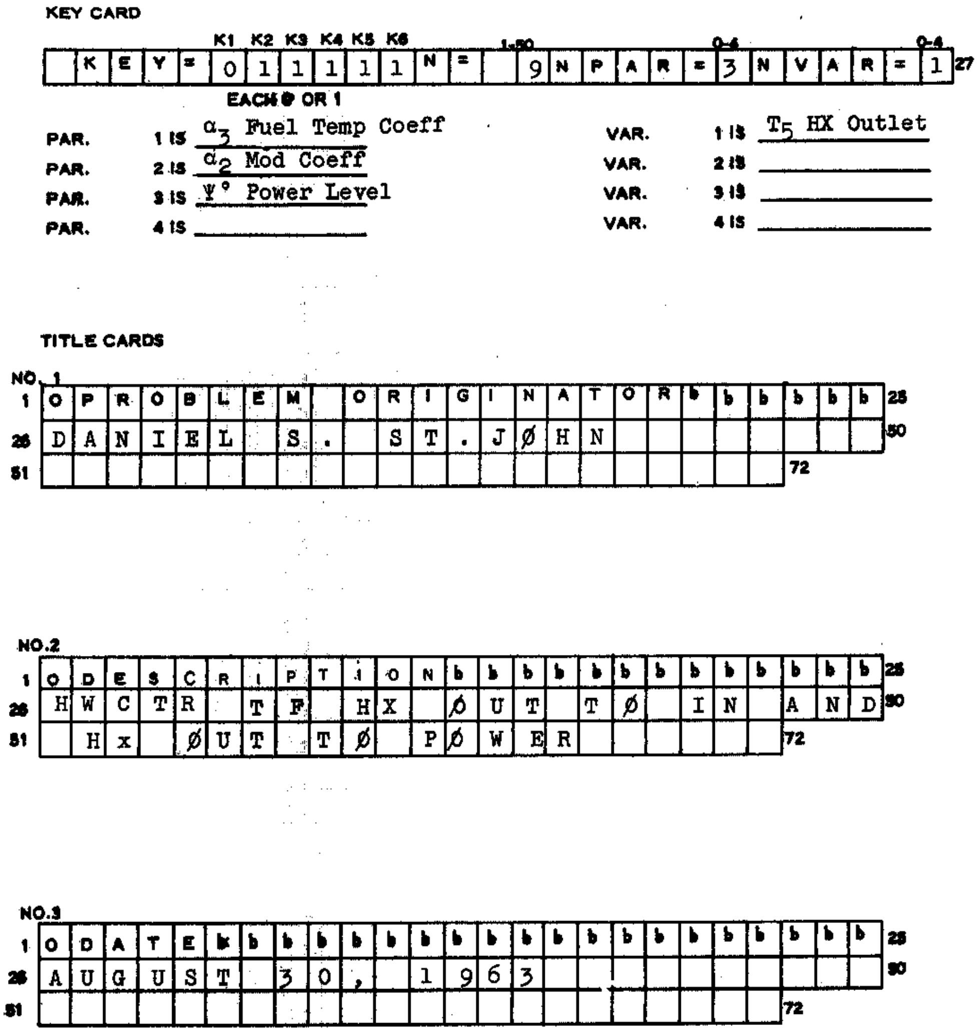
With these modifications, the product of K(J) is 2^4 x 3^3 = 432 or 3/8 of the original value of 1152. The computing time should be reduced by about this factor.

i \ J	T_b 1	T_c 2	T_d 3	T_e 4	T^* 5	Ψ 6	Φ 7	k 8	T_m 9	Excitation Variable, T_s
1	1.69 +17.056s $+31.2s^2$									1.69
2	-1.3	1.3 +0.925s				-0.45				
3		-1.3	1.3 +6.98s			-0.014				
4			-1.3	1.3 +6.98s		-0.022				
5				-1.3	1.3 +11.50s $+24.24s^2$	-0.014				
6						1	$\frac{-\Psi^o}{s}$			
7							0.00875 +0.00739s	-0.0943 -1.45s $-s^2$		
8		$-\alpha_3$						1	$-\alpha_2$	
9			-0.224	-0.552	-0.224 -0.7788s				1	
K(J)	3	2	2	2	3	1	2	3	1	

FIG. 11 SAMPLE PROBLEM – REDUCED MATRIX

3.2 Input Data

Figure 12 shows page 1 of the input data sheet for this sample problem. K_1 is 0, indicating that the matrix elements have been developed outside the code and no special programming is necessary to provide for their computation. K_2 through K_6 are 1, indicating a desire to carry through all of the parts of the program including computation of the denominator of the transfer function, its stability analysis with respect to the indicated parameters, the numerator of the



KEY CARD

					K1	K2	K3	K4	K5	K6			1-50							0-4						0-4	
	K	E	Y	=	0	1	1	1	1	1	N	=		9	N	P	A	R	=	3	N	V	A	R	=	1	27
					EACH 0 OR 1																						

PAR. 1 IS α_3 Fuel Temp Coeff

PAR. 2 IS α_2 Mod Coeff

PAR. 3 IS Ψ° Power Level

PAR. 4 IS ______

VAR. 1 IS T_5 HX Outlet

VAR. 2 IS ______

VAR. 3 IS ______

VAR. 4 IS ______

TITLE CARDS

NO. 1

1	O	P	R	O	B	L	E	M		O	R	I	G	I	N	A	T	O	R	b	b	b	b	b	b	25
26	D	A	N	I	E	L		S	.		S	T	.	J	Ø	H	N									50
51																						72				

NO. 2

1	O	D	E	S	C	R	I	P	T	I	O	N	b	b	b	b	b	b	b	b	b	b	b	b	b	25
26	H	W	C	T	R		T	F		H	X		Ø	U	T		T	Ø		I	N		A	N	D	50
51		H	x		Ø	U	T		T	Ø		P	Ø	W	E	R							72			

NO. 3

1	O	D	A	T	E	b	b	b	b	b	b	b	b	b	b	b	b	b	b	b	b	b	b	b	b	25
26	A	U	G	U	S	T		3	0	,		1	9	6	3											50
51																						72				

FIG. 12 SAMPLE INPUT DATA – PAGE 1

transfer function, and the frequency and transient response. The number of variables, N, is 9 (including the dummy variable, T_m). The number of parameters for stability studies, NPAR, is three (α_3, α_2, and Ψ^o), and the number, NVAR, of excitation variables is one (T_5).

Figure 13 shows page 2 of the input data. NSH is 1 because column 7 was divided by one power of s. The K(J) are entered as they appear at the bottom of Figure 11. The non-zero values of a_{ijk} are entered from Figure 11. The last value is repeated to fill out the last card at the bottom of page 2 of the input sheet. All terms in the A matrix (all terms in Figure 11 that do not multiply one of the three parameters α_3, α_2, or Ψ^o) are entered.

ELEMENTS OF A MATRIX
(USE ONLY IF KEY1 = 0 AND EITHER KEY 2 AND/OR KEY 4 = 1)

CARD 1

													K1									K10	
1	1	0	1	N	S	H	=		1	K	J	=	3	2	2	2	3	1	2	3	1		22
23																							42
43																							62
			K61																			K80	

K(J)= NUMBER OF TERMS IN J^{TH} COLUMN OF MATRIX

NSH IS OVERALL POWER OF S MULTIPLYING MATRIX.

MATRIX CARDS

		I	J	K	±	A(I,J,K)		±		
1	1 0 2	1	1	1	+	•169	E	+	01	22
23		1	1	2	+	•17056	E	+	02	41
42		1	1	3	+	•312	E	+	02	60
1	1 0 2	2	1	1	-	•13	E	+	01	22
23		2	2	1	+	•13	E	+	01	41
42		2	2	2	+	•925	E			60
1	1 0 2	2	6	1	-	•45	E			22
23		3	2	1	-	•13	E	+	01	41
42		3	3	1	+	•13	E	+	01	60
1	1 0 2	3	3	2	+	•698	E	+	01	22
		3	6	1	-	•014	E			41
		4	3	1	-	•13	E	+	01	60
	1 0 2	4	4	1	+	•13	E	+	01	
		4	4	2	+	•698	E	+	01	
		4	6	1	-	•022	E			
	1 0 2	5	4	1	-	•13	E	+	01	
		5	5	1	+	•13	E	+	01	
		5	5	2	+	•115	E	+	02	
	1 0 2	5	5	3	+	•2424	E	+	02	
		5	6	1	-	•014	E			
		6	6	1	+	•1	E	+	01	
	1 0 2	7	8	1	-	•0943	E			
		7	8	2	-	•145	E	+	01	
		7	8	3	-	•1	E	+	01	
	1 0 2	8	8	1	+	•1	E	+	01	
		9	3	1	-	•224	E			
		9	4	1	-	•552	E			
	1 0 3	9	5	1	-	•224	E			
		9	5	2	-	•7788	E			
		9	5	2	-	•7788	E			

2 IN COLUMN 3,
3 FOR LAST CARD

FIG. 13 SAMPLE INPUT DATA – PAGE 2

Figure 14 shows the input data sheet, page 3 for the elements of the G matrix.

As indicated on the first input sheet, parameter 1 is taken to be α_3 which appears in column 2 of Figure 10. Therefore NC1 is 2. Similarly parameter 2 is α_2 and NC2 is 9, and parameter 3 is Ψ^{o} making NC3 equal to 7.

ELEMENTS OF G MATRIX

USE ONLY IF 1) KEY 1 = 0 AND EITHER KEY 2 AND/OR KEY 4 = 1, AND 2) NPAR > P.

CARD 1

				NC1		NC2		NC3		NC4				MA(J)						
1	1	0	4		2		9		7			2	1	1	1	2	1			19
			20											-	1					35
				±		±		±		±		±		±		±		±		

NT(J)

NC(I) IS COLUMN OF A MATRIX CONTAINING PARA.

MA(J) IS NUMBER OF TERMS IN J COLUMN OF G MATRIX

NT(J) IS POWER OF S MULTIPLYING J COLUMN

				I			J	K	±				G(I,J,K)							±			
1	1	0	5		2		1	1	+	•	1	3							E	+	0	1	22
			23		2		1	2	+	•	9	2	5						E				41
			42		3		1	1	-	•	1	3							E	+	0	1	60
1	1	0	5		8		2	1	-	•	1								E	+	0	1	22
			23		9		3	1	+	•	1								E	+	0	1	41
			42		8		4	1	-	•	1								E	+	0	1	60
1	1	0	6		7		5	1	+	•	0	0	8	7	5				E				22
			23		7		5	2	+	•	0	0	7	3	9				E				41
			42		6		6	1	-	•	1								E	+	0	1	60
1	1	0								•									E				22
			23							•									E				41
			42							•									E				60
	1	0								•									E				
										•									E				
										•									E				
	1	0								•									E				
										•									E				
										•									E				
	1	0								•									E				
										•									E				
										•									E				
	1	0								•									E				
										•									E				
										•									E				
	1	0								•									E				
										•									E				
										•									E				
	1	0								•									E				
										•									E				
										•									E				

5 IN COLUMN 3,
6 FOR LAST CARD

FIG. 14 SAMPLE INPUT DATA – PAGE 3

The MA(j) are one greater than the highest power of s in each of the parameter columns. There are up to four pairs of one-digit numbers. The first digit refers to the terms that do not multiply the parameters (and are just the corresponding K(J) on input page 2), while the second refers to the terms containing the parameters (all are zero order in s in this example, so the second digit of each of the three pairs is 1). Since there is a term in 1/s multiplying the third parameter, Ψ^o, the sixth NT(J) is -1 (there is one pair of NT(J) for each parameter; the first of each pair is always zero, the second is a negative number equal to the negative power of s multiplying that parameter).

The elements in the columns containing parameters are entered on the rest of page 3. Here i has the value in Figure 11.

For the column containing the first parameter (α_3), j is 1 for terms not multiplying α_3, and j is 2 for terms containing α_3. Similarly j is 3 or 4 for the column containing parameter 2, and so on.

Figure 15 shows the input data sheet, page 4, for the element of the V matrix corresponding to the excitation variable T_5. As seen from Figure 11, T_5 appears in the first row so i = 1. There is only one such variable, so its j is 1. The value is repeated three times to fill out the card.

This problem does not use input data sheet, page 5 (Figure 5), since the characteristic equation (or denominator of the transfer function) is developed in earlier parts of the code rather than being read in as input at this point.

Figure 16 shows input data sheet, page 6, which shows the values and ranges of the parameters for which stability plots are desired. In this case, stability plots were desired for power levels of 5, 10, 20, 40, and 80 MW. The plots are of parameter 2 (α_2) vs parameter 1 (α_3) for values of α_2 from -1×10^{-3} to $+1 \times 10^{-3}$ and of α_3 from -1×10^{-4} to $+1 \times 10^{-4}$.

Two cards are required per plot, since this is a three-parameter system. In the first card, column four is 1 since we want a plot of parameter 2 is 1. The first value of Ψ^o (parameter 3) follows. Card 2 has the desired ranges of parameter 1 (α_3) (appears as X coordinate) and of parameter 2 (α_2) (Y coordinate).

Similar pairs are included for each value of Ψ^o.

ELEMENTS OF V MATRIX
(USE ONLY IF KEYI = 0 AND EITHER KEY2 AND/OR KEY4 = 1. IF NVAR =0, USE LAST CARD ONLY.)

				I		J			±						V(I,J)					±			
1	1	0	8		1		1	b	+	•	1	6	9						E	+	0	1	22
23					1		1	b	+	•	1	6	9						E	+	0	1	41
42					1		1	b	+	•	1	6	9						E	+	0	1	60
1	1	0						b		•									E				22
23								b		•									E				41
42								b		•									E				60
1	1	0						b		•									E				22
23								b		•									E				41
42								b		•									E				60
1	1	0						b		•									E				22
23								b		•									E				41
42								b		•									E				60
1	1	0						b		•									E				22
23								b		•									E				41
42								b		•									E				60
	1	0						b		•									E				
								b		•									E				
								b		•									E				
	1	0						b		•									E				
								b		•									E				
								b		•									E				
	1	0						b		•									E				
								b		•									E				
								b		•									E				
	1	0						b		•									E				
								b		•									E				
								b		•									E				
	1	0						b		•									E				
								b		•									E				
								b		•									E				
	1	0						b		•									E				
								b		•									E				
								b		•									E				

7 IN COLUMN 3,8 FOR LAST CARD.

1	0	8	← BLANK →	**FINAL CARD, USE ONLY IF NVAR = 0**

FIG. 15 SAMPLE INPUT DATA – PAGE 4

					±				±		±				±		
1	3	0	4	1	+	•	5	E	+	01		•		E			32
33						•		E				•		E			60
1	3	0	2	0	-	•	1	E	-	03	+	•	1	E	-	03	32
33					-	•	1	E	-	02	+	•	1	E	-	02	60
1	3	0	4	1	+	•	1	E	+	02		•		E			32
33						•		E				•		E			60
1	30	0	2	0	-	•	1	E	-	03	+	•	1	E	-	03	32
33					-	•	1	E	-	02	+	•	1	E	-	02	60
1	3	0	4	1	+	•	2	E	+	02		•		E			32
33						•		E				•		E			60
1	3	0	2	0	-	•	1	E	-	03	+	•	1	E	-	03	32
33					-	•	1	E	-	02	+	•	1	E	-	02	60

					±				±		±				±		
1	3	0	4	1	+	•	4	E	+	02		•		E			32
33						•		E				•		E			60
1	3	0	2	0	-	•	1	E	-	03	+	•	1	E	-	03	32
33					-	•	1	E	-	02	+	•	1	E	-	02	60
1	3	0	5	1	+	•	8	E	+	02		•		E			32
33						•		E				•		E			60
1	30	0	2	0	-	•	1	E	-	03	+	•	1	E	-	03	32
33					-	•	1	E	-	02	+	•	1	E	-	02	60
1	3	0				•		E				•		E			32
33						•		E				•		E			60
1	3	0				•		E				•		E			32
33						•		E				•		E			60

3 OR 4 PARAMETER SYSTEM: TWO CARDS PER PLOT

CARD 1: COLUMN 3 - 4 FOR ALL PLOTS EXCEPT LAST ONE
5 FOR LAST PLOT.

COLUMN 4 - 1 FOR PLOT OF PARAMETER 2 vs 1
2 FOR PLOT OF PARAMETER 4 vs 3
OR FOR STABILITY OF PARAMETER 3 ALONE.

COLUMNS 5 THROUGH 32 - VALUES OF PARAMETERS
3 AND 4 (1 AND 2) AT WHICH PLOT OF 2 vs 1
(4 vs 3) IS OBTAINED

CARD 2: COLUMN 3 - 2
COLUMN 4 -0
COLUMNS 5 THROUGH 32 - LOWER AND UPPER VALUES OF X CO-ORDINATE
COLUMNS 33 THROUGH 60 - LOWER AND UPPER VALUES OF Y CO-ORDINATE

1 OR 2 PARAMETER SYSTEM: ONE CARD ONLY, SAME AS FOR CARD 2 ABOVE.

Parameter 1 or 3 is X coordinate
Parameter 2 or 4 is Y coordinate

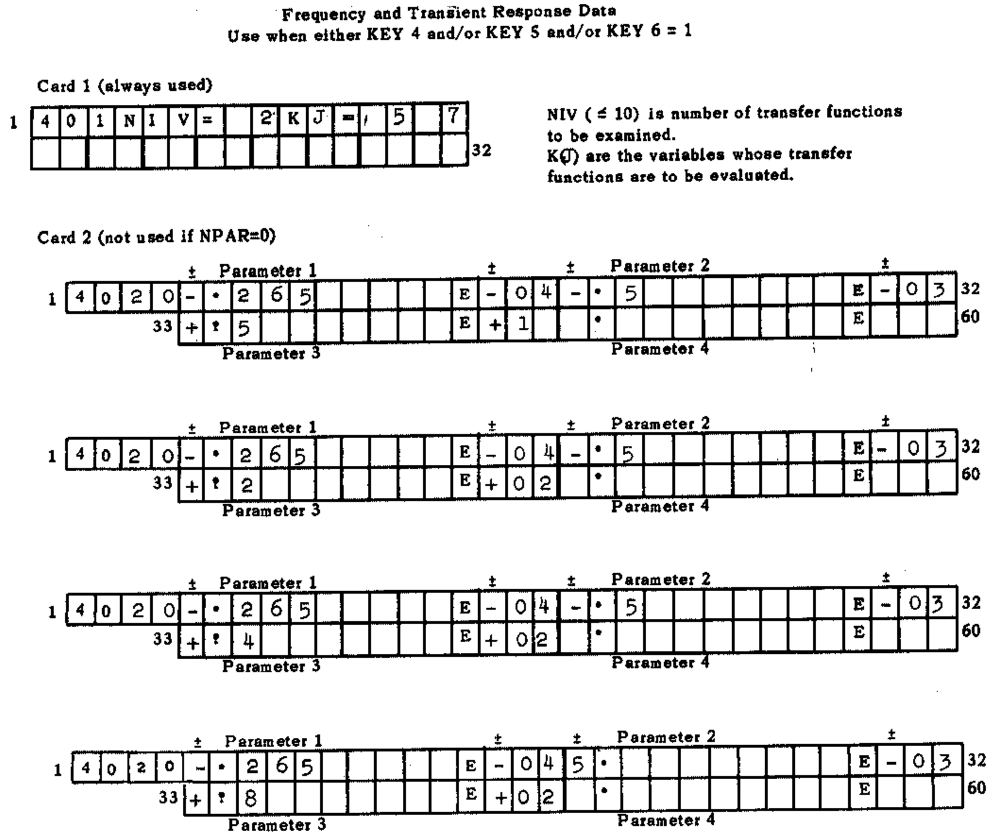
FIG. 16 SAMPLE INPUT DATA – PAGE 6

Figure 17 shows input data sheet, page 7, which specifies data for the frequency and transient response calculations. We wish the response of the heat exchanger inlet, T* (column 5 of Figure 11) and of the flux, Φ (column 7) to variations in the heat exchanger outlet, T_5, the excitation variable specified on data sheet page 4 (Figure 14).

NIV is 2, the number of variables (Φ, card T*) for which the response is to be calculated. The K(J) are 5 and 7 corresponding to the numbers of these variables in Figure 11.

Card 2 lists one set of values of the parameters. Since the response for power levels of 5, 10, 20, 40, and 80 MW are desired for a single set of values of parameters 1 and 2, card 2 of page 7 is repeated for each power level. Only one card 1 is required for the entire set of parameter values.

Card 3 and following cards are not required since the numerators of the transfer functions are calculated from data already in the problem. These cards are used to read in the numerator of a known transfer function.



Frequency and Transient Response Data
Use when either KEY 4 and/or KEY 5 and/or KEY 6 = 1

Card 1 (always used)

1 | 4 | 0 | 1 | N | I | V | = | | 2 | K | J | = | , | 5 | | 7 |
32

NIV (≤ 10) is number of transfer functions to be examined.
K(J) are the variables whose transfer functions are to be evaluated.

Card 2 (not used if NPAR=0)

Parameter 1: 1 | 4 | 0 | 2 | 0 | - | • | 2 | 6 | 5 | E | - | 0 | 4 | Parameter 2: - | • | 5 | E | - | 0 | 3 | 32
Parameter 3: 33 | + | • | 5 | E | + | 1 | Parameter 4: • | E | 60

Parameter 1: 1 | 4 | 0 | 2 | 0 | - | • | 2 | 6 | 5 | E | - | 0 | 4 | Parameter 2: - | • | 5 | E | - | 0 | 3 | 32
Parameter 3: 33 | + | • | 2 | E | + | 0 | 2 | Parameter 4: • | E | 60

Parameter 1: 1 | 4 | 0 | 2 | 0 | - | • | 2 | 6 | 5 | E | - | 0 | 4 | Parameter 2: - | • | 5 | E | - | 0 | 3 | 32
Parameter 3: 33 | + | • | 4 | E | + | 0 | 2 | Parameter 4: • | E | 60

Parameter 1: 1 | 4 | 0 | 2 | 0 | - | • | 2 | 6 | 5 | E | - | 0 | 4 | Parameter 2: 5 | • | E | - | 0 | 3 | 32
Parameter 3: 33 | + | • | 8 | E | + | 0 | 2 | Parameter 4: • | E | 60

FIG. 17 SAMPLE INPUT DATA – PAGE 7

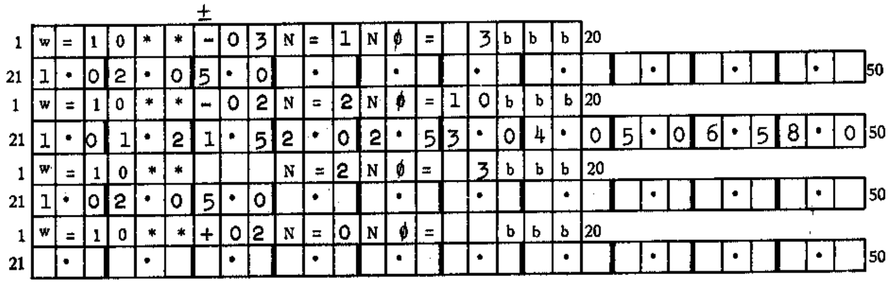
Figure 18 shows input data sheet, page 8, with data for the frequency range desired in the frequency response and the magnitudes of the perturbations in the exciting variables for the transient response.

The first cards show that points are to be calculated and plotted at only three points (1.0, 2.0, and 5.0) in the decade from 10^{-3} to 10^{-2} rad/sec. Points at 1.0, 1.2, 1.5, 2.0, 2.5, 3.0, 4.0, 5.0, 6.5, and 8.0 are obtained for the two decades, 10^{-2} to 1 rad/sec. Three points are obtained for frequencies 1 to 100, and the last card defines the upper frequency limit at 100 rad/sec.

The bottom card of Figure 18 shows the response will be calculated at intervals of 1 sec over the range from 0 to 100 sec for an impulse of 10 (°C - sec) a step of 1 (°C) and a ramp of 0.1 (°C/sec). These perturbations will be used for each of the excitation variables (only one in this example).

The cards on page 8 were punched and reproduced to provide a deck for each value of the power level. The final deck is in the sequence: card 1 from page 7, card 2 from

Frequency Response (KEY 5 = 1)

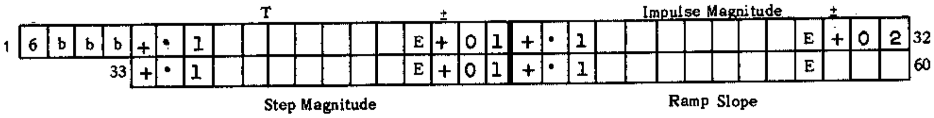


```
                  ±
 1  w=10**-03N=1Nϕ= 3bbb  20
21  1.02.05.0 .  .  .  .  .  .  .   50
 1  w=10**-02N=2Nϕ=10bbb  20
21  1.01.21.52.02.53.04.05.06.58.0  50
 1  w=10**   N=2Nϕ= 3bbb  20
21  1.02.05.0 .  .  .  .  .  .  .   50
 1  w=10**+02N=0Nϕ=  bbb  20
21   .  .  .  .  .  .  .  .  .  .   50
```

Begin at frequency w
And continue for N decades with
Nϕ points per decade (Nϕ a maximum of 10).
Points are given in columns 21 through 50.

(Last card gives final frequency value and must have Nϕ = 0)

Transient Response (KEY 6 = 1)



```
 1  6bbb+.1       E+01+.1       E+02  32
        33  +.1       E+01+.1       E     60
```

Calculation for 101 values of time from 0 to 100 T in increments of T

Calculates response to
Impulse (if Impulse Magnitude ≠ 0)
Step Function (if Step Magnitude ≠ 0)
Ramp (if Ramp Slope ≠ 0)

FIG. 18 SAMPLE INPUT DATA – PAGE 8

page 7, cards from page 8, another card 2 from page 7, another set from page 8, etc., until all desired sets of parameters from pages 7 have been used.

3.3 Sample Output

Some of the printouts for the sample problem are shown in Figures 19-29.

Figure 19 shows the title page and the matrix elements. The title page includes a list of the KEY sections of the code that are used in the problem. In the example, all parts are employed including KEY 1, the special CALMAT coding illustrated in Section 3.4.

The A, G, and V matrix elements are printed out, and can be used to check the input cards or the operation of any special CALMAT coding designed to compute these elements.

Figure 20 lists the coefficients of the characteristic equation (denominator of the transfer function).

Figure 21 shows the stability plot prepared by KEY 3 for a power level of 5 MW, as a function of the fuel and moderator temperature coefficients (parameters 1 and 2, respectively). Five such plots were called for by the cards from input sheet 6, one for each of five power levels (Figure 16). On the plots, an s indicates a stable region (all roots have negative real parts), a T indicates a threshold at least one root with a zero real part and the rest with negative real parts), and a blank indicates an unstable region (at least one root with a positive real part). If the entire region is stable, the printout contains a simple statement to that effect rather than a plot. If the stability is calculated with respect to only one parameter, the plot is replaced by a table of values of the parameter at which calculations were made and a statement STABLE, THRESHOLD, or UNSTABLE for each.

Figure 22 lists the coefficients of the transfer function for variable 5 (reactor flux level) excited by a change in T_5, the heat exchange outlet temperature. The transfer function is for a power level of 5 MW at the nominal values of the temperature coefficients, parameters 1 and 2.

The response results are shown in Figures 23-29. A total of ten such sets of output were called for by the cards on page 7 of the input sheets (Figure 17), corresponding to five power levels and two response variables (flux and heat exchanger inlet temperature).

Figure 23 lists the real and imaginary parts and absolute value (gain) of the transfer function together with the gain in decibels and the phase angle in degrees for each frequency point indicated by the cards from page 8 of the input sheet (Figure 18). Figure 24 is a log-amplitude and phase plot (Bode diagram) of the same data, and Figure 25 is a polar plot (Nyquist diagram).

Figure 26 lists the singularities of the transfer function required in the calculation of the transient response. As a check, the coefficients of the transfer function are reconstituted from these roots for comparison with the original transfer function listed in Figure 22. This check is desirable since the root-locating procedure may fail if roots are too close together or too close to zero. A considerable effort was made to incorporate the best available routine, as discussed in Section 4.3.12.

Figure 27 shows the response to a pulse input. The approximate response equation in the form of a sum of terms corresponding to the singularities of the transfer function is listed together with the response calculated at 100 equally spaced time points and a plot of the results.

Figures 28 and 29 show similar results for step and ramp excitations.

The magnitudes of the excitations and the time for which the response is desired were entered on the last card of input sheet 8 (Figure 18).

This sample problem produced a total of 131 pages of output and required 40 minutes on the IBM 704.

```
                    LINEAR ANALYSIS OF SYSTEM STABILITY          PAGE 1


PROBLEM.ORIGINATOR       DAN ST. JOHN BY BBO TO TEST USE OF CALMAT SUBR.

DESCRIPTION              HWCTR TF HX OUT TO IN AND HX OUT TO POWER

DATE                     APRIL 7, 1964


OBTAIN THE FOLLOWING

     1. CALCULATE MATRIX ELEMENTS

     2. DENOMINATOR OF TRANSFER FUNCTION(PARAMETRIC)

     3. STABILITY LIMITS

     4. NUMERATOR OF TRANSFER FUNCTION

     5. FREQUENCY RESPONSE

     6. TRANSIENT RESPONSE


                                   A(I,J,1) ELEMENTS        PAGE 2


            1                 2                 3                 4                 5                 6
1    0.16899999E 01    0.                0.                0.                0.                0.
2   -0.13000000E 01    0.13000000E 01    0.                0.                0.               -0.45000000E-00
3    0.               -0.13000000E 01    0.13000000E 01    0.                0.               -0.13999999E-01
4    0.                0.               -0.13000000E 01    0.13000000E 01    0.               -0.22000000E-01
5    0.                0.                0.               -0.13000000E 01    0.13000000E 01   -0.13999999E-01
6    0.                0.                0.                0.                0.                0.09999999E 01
7    0.                0.                0.                0.                0.                0.
8    0.                0.                0.                0.                0.                0.
9    0.                0.               -0.22400000E-00   -0.55199999E 00   -0.22400000E-00    0.
```

```
                                 A(I,J,1) ELEMENTS        PAGE 3

           7                   8                   9
1     0.                  0.                  0.
2     0.                  0.                  0.
3     0.                  0.                  0.
4     0.                  0.                  0.
5     0.                  0.                  0.
6     0.                  0.                  0.
7     0.                 -0.94300000E-01      0.
8     0.                  0.09999999E 01      0.
9     0.                  0.                  0.

                                 A(I,J,2) ELEMENTS        PAGE 4

           1                   2                   3                   4                   5                   7
1     0.17055999E 02      0.                  0.                  0.                  0.                  0.
2     0.                  0.92500000E 00      0.                  0.                  0.                  0.
3     0.                  0.                  0.69799999E 01      0.                  0.                  0.
4     0.                  0.                  0.                  0.69799999E 01      0.                  0.
5     0.                  0.                  0.                  0.                  0.11499999E 02      0.
6     0.                  0.                  0.                  0.                  0.                  0.
7     0.                  0.                  0.                  0.                  0.                  0.
8     0.                  0.                  0.                  0.                  0.                  0.
9     0.                  0.                  0.                  0.                 -0.77883076E 00      0.

                                 A(I,J,2) ELEMENTS        PAGE 5

           8
1     0.
2     0.
3     0.
4     0.
5     0.
6     0.
7    -0.14500000E 01
8     0.
9     0.

                                 A(I,J,3) ELEMENTS        PAGE 6

           1                   5                   C
1     0.31199999E 02      0.                  0.
2     0.                  0.                  0.
3     0.                  0.                  0.
4     0.                  0.                  0.
5     0.                  0.24268922E 02      0.
6     0.                  0.                  0.
7     0.                  0.                 -0.09999999E 01
8     0.                  0.                  0.
9     0.                  0.                  0.

                                 G(I,J,1) ELEMENTS        PAGE 7

           1                   2                   3                   4                   5                   6
1     0.                  0.                  0.                  0.                  0.                  0.
2     0.13000000E 01      0.                  0.                  0.                  0.                  0.
3    -0.13000000E 01      0.                  0.                  0.                  0.                  0.
4     0.                  0.                  0.                  0.                  0.                  0.
5     0.                  0.                  0.                  0.                  0.                  0.
6     0.                  0.                  0.                  0.                  0.                 -0.09999999E 01
7     0.                  0.                  0.                  0.                  0.87499999E-02      0.
8     0.                 -0.09999999E 01      0.                 -0.09999999E 01      0.                  0.
9     0.                  0.                  0.09999999E 01      0.                  0.                  0.

                                 G(I,J,2) ELEMENTS        PAGE 8

           1                   5
1     0.                  0.
2     0.92500000E 00      0.
3     0.                  0.
4     0.                  0.
5     0.                  0.
6     0.                  0.
7     0.                  0.73899999E-02
8     0.                  0.
9     0.                  0.

                                 V(I,J) ELEMENTS          PAGE 9

           1
1     0.16899999E 01
2     0.
3     0.
4     0.
5     0.
6     0.
7     0.
8     0.
9     0.
```

FIG. 19 SAMPLE OUTPUT – MATRIX ELEMENTS

```
                     CHARACTERISTIC EQUATION                   PAGE10

                 NON PARAMETRIC           PAR 1                  PAR 2               PAR 1   X   PAR 2

                      0.                    0.                     0.                     0.
S** 1                 0.42234576E-01        0.                     0.                     0.
S** 2                 0.13191139E 01        0.                     0.                     0.
S** 3                 0.17121394E 02        0.                     0.                     0.
S** 4                 0.11956541E 03        0.                     0.                     0.
S** 5                 0.48529533E 03        0.                     0.                     0.
S** 6                 0.11521674E 04        0.                     0.                     0.
S** 7                 0.15263589E 04        0.                     0.                     0.
S** 8                 0.10042755E 04        0.                     0.                     0.
S** 9                 0.25217503E 03        0.                     0.                     0.

                                                               PAGE11

                   PAR 3               PAR 1   X   PAR 3         PAR 2   X   PAR 3       PAR 1 X PAR 2 X PAR 3

                      0.                   -0.15755817E-00        -0.16953539E-00          0.
S** 1                 0.                   -0.70985284E 01        -0.58548409E 01          0.
S** 2                 0.                   -0.13007011E 03        -0.74945955E 02          0.
S** 3                 0.                   -0.12736886E 04        -0.47024556E 03          0.
S** 4                 0.                   -0.72861716E 04        -0.15911573E 04          0.
S** 5                 0.                   -0.24762598E 05        -0.29343263E 04          0.
S** 6                 0.                   -0.48072105E 05        -0.27553929E 04          0.
S** 7                 0.                   -0.47196284E 05        -0.10698209E 04          0.
S** 8                 0.                   -0.16600777E 05        -0.90032129E 02          0.
S** 9                 0.                    0.                     0.                      0.
```

FIG. 20 SAMPLE OUTPUT – CHARACTERISTIC EQUATION

```
STABILITY REGIONS FOR PARAMETER 1 AND PARAMETER 2          PAGE 12

                         PARAMETER 3=  5.0000000E 00

                              RANGE EXAMINED FOR PARAMETER 1 IS -9.9999999E-05 TO  9.9999999E-05

                              RANGE EXAMINED FOR PARAMETER 2 IS -9.9999999E-04 TO  9.9999999E-04

                  +---------+---------+---------+---------+---------+
             10. +                                                   +
                 I                                                   I
                 I                                                   I
                 I                                                   I
                 I                                                   I
              6. +                                                   +
                 I                                                   I
PARAMETER        I                                                   I
    2            I                                                   I
                 I                                                   I
              2. +                                                   +
                 I                                                   I
  TIMES          IS S S S S S S S                                    I
 1C**   4        IS S S S S S S S S S S S S S S S S S                I
                 IS S S S S S S S S S S S S S S S S S S S S S S S S SI
             -2. +S S S S S S S S S S S S S S S S S S S S S S S S S S+
                 IS S S S S S S S S S S S S S S S S S S S S S S S S SI
                 IS S S S S S S S S S S S S S S S S S S S S S S S S SI
                 IS S S S S S S S S S S S S S S S S S S S S S S S S SI
                 IS S S S S S S S S S S S S S S S S S S S S S S S S SI
             -6. +S S S S S S S S S S S S S S S S S S S S S S S S S S+
                 IS S S S S S S S S S S S S S S S S S S S S S S S S SI
                 IS S S S S S S S S S S S S S S S S S S S S S S S S SI
                 IS S S S S S S S S S S S S S S S S S S S S S S S S SI
                 IS S S S S S S S S S S S S S S S S S S S S S S S S SI
            -10. +S S S S S S S S S S S S S S S S S S S S S S S S S S+
                  +---------+---------+---------+---------+---------+

                -10.      -6.       -2.        2.        6.        10.
                                PARAMETER 1 TIMES 10**  5
```

FIG. 21 SAMPLE OUTPUT – STABILITY PLOT

```
                 TRANSFER FUNCTION FOR VARIABLE NUMBER  5           PAGE 17

PARAMETER 1=  -2.6500000E-05            PARAMETER 2=  -4.9999999E-04           PARAMETER 3=   5.0000000E 00

                   DENOMINATOR                              NUMERATOR FOR EXCITING VARIABLE NUMBER
                                            1
                 4.4471493E-04        -1.6142704E-05
   S**  1        5.7812233E-02         4.1959165E-02
   S**  2        1.5237131E 00         3.5062075E-02
   S**  3        1.8465772E 01        -5.7627547E-04
   S**  4        1.2450872E 02        -1.9855681E-04
   S**  5        4.9591219E 02         0.
   S**  6        1.1654254E 03         0.
   S**  7        1.5352869E 03         0.
   S**  8        1.0067002E 03         0.
   S**  9        2.5217503E 02         0.
```

FIG. 22 SAMPLE OUTPUT – TRANSFER FUNCTION

```
      FREQUENCY RESPONSE FOR VARIABLE NUMBER  5             PAGE 18
           EXCITED BY FUNCTION NUMBER 1
      PARAMETER 1=  -2.6500000E-05           PARAMETER 2=  -4.9999999E-04           PARAMETER 3=   5.0000000E 00

FREQUENCY              X                 Y                GAIN               GAIN             PHASE ANGLE
 RAD/SEC             NUMBER            NUMBER             NUMBER             DECIBELS           DEGREES
  9.9999999E-04    -2.3752997E-02     9.7772532E-02     1.0061646E-01     -1.9946618E 01     1.0365498E 02
  2.0000000E-03     1.2388210E-02     1.8806117E-01     1.8846876E-01     -1.4495212E 01     8.6231182E 01
  4.9999999E-03     2.1498829F-01     3.6426622E-01     4.2297735E-01     -7.4736577E 00     5.9451087E 01
  9.9999999E-03     5.7346674E-01     3.3591129E-01     6.6460551E-01     -3.5487213E 00     3.0359873E 01
  1.2000000E-02     6.7019242F-01     2.6272558E-01     7.1984901E-01     -2.8551715E 00     2.1405931E 01
  1.5000000E-02     7.6382685E-01     1.3245086E-01     7.7522556F-01     -2.2114381E 00     9.8375108E 00
  2.0000000E-02     8.1823583E-01    -8.7529631E-02     8.2290418E-01     -1.6930146E 00     3.5389408E 02
  2.4999999E-02     7.9217427E-01    -2.8043150E-01     8.4034629E-01     -1.5100343E 00     3.4050592E 02
  2.9999999E-02     7.1871402E-01    -4.3789361E-01     8.4160600E-01     -1.4978236E 00     3.2864715E 02
  3.9999999E-02     4.9911994E-01    -6.4840717E-01     8.1826172E-01     -1.7421532E 00     3.0758775E 02
  4.9999999E-02     2.4773800F-01    -7.3550067E-01     7.7610266E-01     -2.2016163E 00     2.8861502E 02
  6.4999998E-02    -9.0643036E-02    -6.8946056E-01     6.9539343F-01     -3.1553883E 00     2.6251030E 02
  7.9999999E-02    -3.1728915E-01    -5.1653228E-01     6.0619963E-01     -4.3476364E 00     2.3843893E 02
  9.9999999E-02    -4.2771655E-01    -2.3639246E-01     4.8869503E-01     -6.2192411E 00     2.0892876E 02
  1.2000000E-01    -3.8301992E-01    -1.3824577E-02     3.8326933E-01     -8.3299181E 00     1.8206711E 02
  1.5000000E-01    -2.1413356E-01     1.4290844E-01     2.5744126E-01     -1.1786436E 01     1.4628165E 02
  2.0000000E-01    -1.5569367F-02     1.2658515E-01     1.2753903E-01     -1.7887137E 01     9.7011890E 01
  2.5000000E-01     3.3456601E-02     5.3661817E-02     6.3237131E-02     -2.3980556E 01     5.8057647E 01
  3.0000000E-01     2.8800144E-02     1.4543337E-02     3.2263864E-02     -2.9825671E 01     2.6792609E 01
  4.0000000E-01     8.8721613E-03    -3.2290069E-03     9.4414837E-03     -4.0499108E 01     3.4000112E 02
  4.9999999E-01     1.9232335E-03    -2.5816135E-03     3.2192477E-03     -4.9044911E 01     3.0668513E 02
  6.4999999E-01     1.8882104E-05    -8.1095111E-04     8.1117030F-04     -6.1817751E 01     2.7133383E 02
  7.9999999E-01    -1.0186227F-04    -2.3221756E-04     2.5357625E-04     -7.1917825E 01     2.4631533E 02
  1.0000000E 00    -5.0756661F-05    -4.6040778E-05     6.8527307E-05     -8.3282723E 01     2.2221080E 02
  2.0000000E 00    -8.4271301E-07     2.4296279E-07     8.7703828E-07     -1.2113962E 02     1.6391719E 02
  5.0000000E 00    -9.6164857E-10     1.7104682E-09     1.9622613E-09     -1.7414486E 02     1.1934529E 02
  1.0000000E 01    -4.7049586E-12     2.1208189E-11     2.1723810E-11     -2.1326127E 02     1.0250828E 02
  2.0000000E 01    -2.8755079E-14     3.5434528E-13     3.5551010E-13     -2.4898295E 02     9.4639380E 01
  5.0000000E 01    -6.5033447E-17     2.6989576E-15     2.6997410E-15     -2.9137354E 02     9.1380320E 01
  1.0000000E 02    -8.9760312E-19     8.0141041E-17     8.0146067E-17     -3.2192235E 02     9.0641704E 01
```

FIG. 23 SAMPLE OUTPUT – FREQUENCY RESPONSE

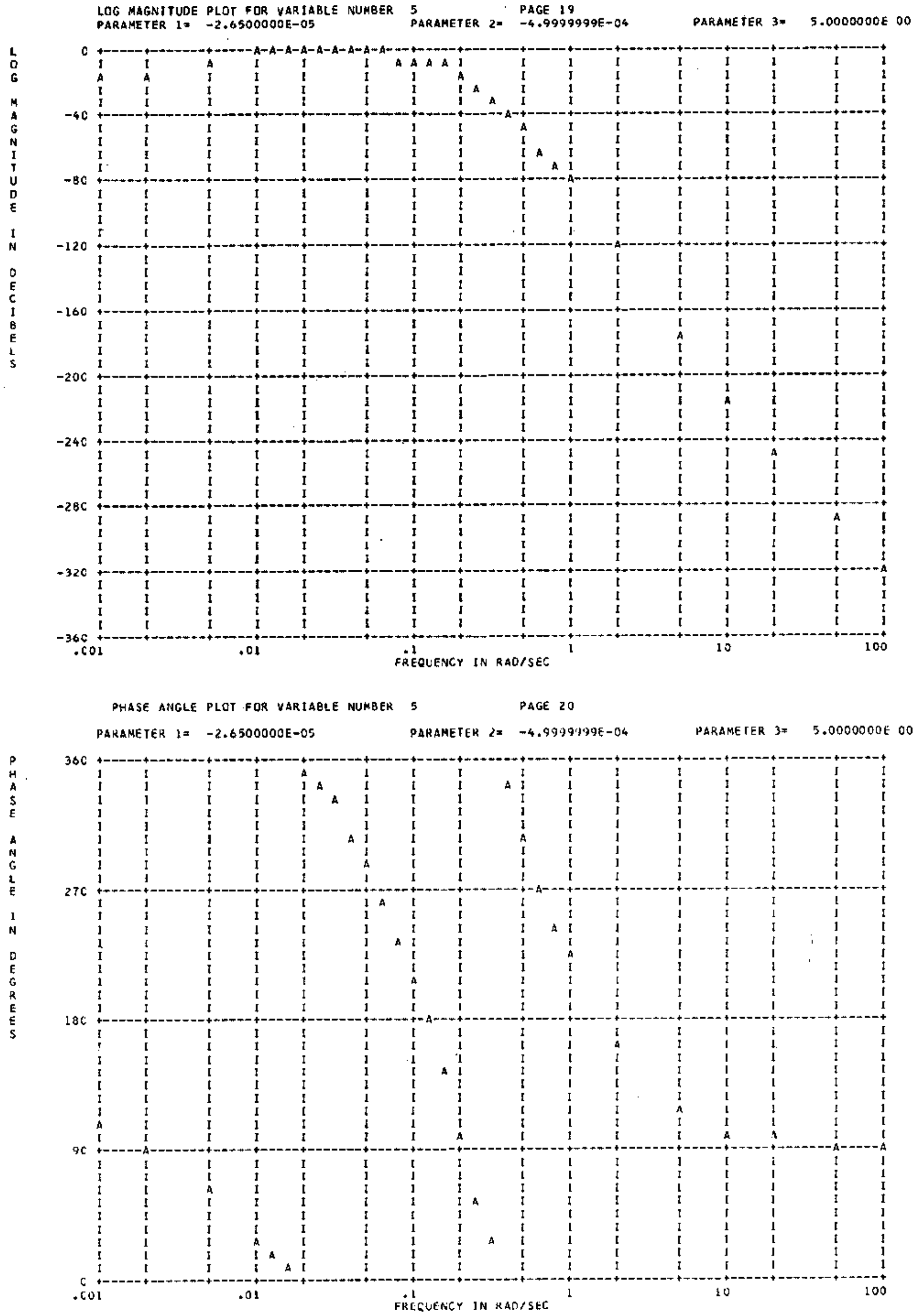


FIG. 24 SAMPLE OUTPUT - PHASE AND LOG-AMPLITUDE PLOT

```
        POLAR PLOT FOR VARIABLE NUMBER  5                    PAGE 21
     PARAMETER 1=  -2.6500000E-05           PARAMETER 2=  -4.9999999E-04          PARAMETER 3=   5.0000000E 00
```

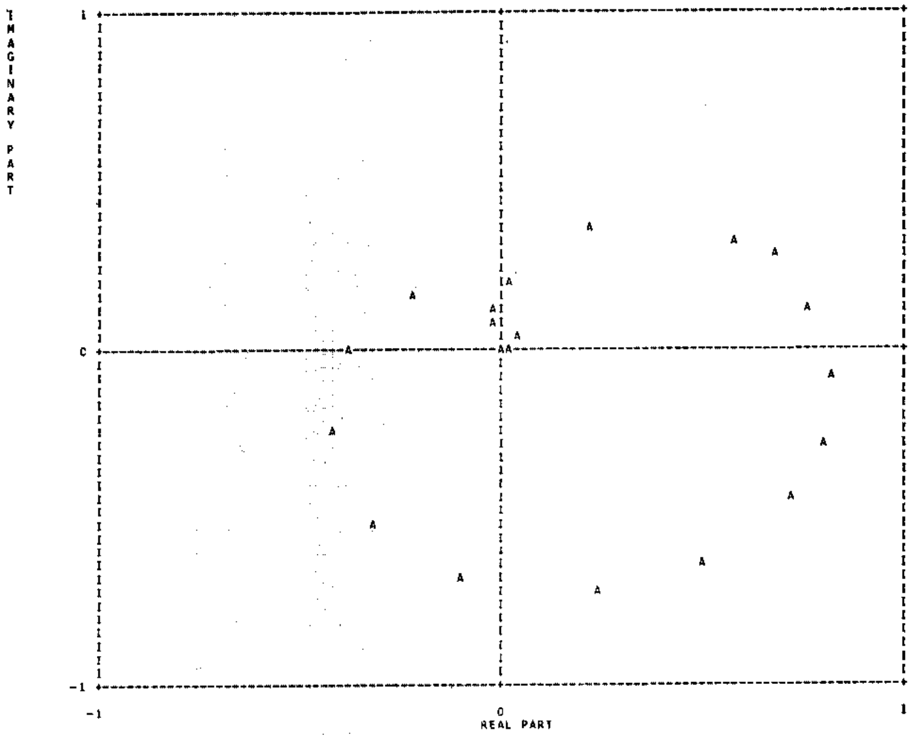


FIG. 25 SAMPLE OUTPUT – POLAR PLOT

```
       SINGULARITIES OF THE TRANSFER FUNCTION                 PAGE 22
                      FOR
     PARAMETER 1=  -2.6500000E-05          PARAMETER 2=  -4.9999999E-04           PARAMETER 3=   5.0000000E 00

       SINGULARITIES                                COEFFICIENTS OF EQUATION (CONSTANT TERM FIRST)
     REAL PART        IMAGINARY PART                                 ORIGINAL        RE-CONSTITUTED
-1.0052390E-02         0.                                        4.4471493E-04      4.4471558E-04
-1.3000041E-01         0.                                        5.7812233E-02      5.7812309E-02
-2.2990023E-01         0.                                        1.5237131E 00      1.5237144E 00
-2.8762012E-01         0.                                        1.8465772E 01      1.8465781E 01
-4.1666328E-01         0.                                        1.2450872E 02      1.2450875E 02
-1.1878991E 00         0.                                        4.9591219E 02      4.9591219E 02
-1.4059312E 00         0.                                        1.1654254E 03      1.1654253E 03
-1.6200123E-01         5.5527926E-02                             1.5352869E 03      1.5352868E 03
-1.6200123E-01        -5.5527926E-02                             1.0067002E 03      1.0067001E 03
                                                                 2.5217503E 02      2.5217503E 02
```

FIG. 26 SAMPLE OUTPUT – SINGULARITIES

```
          TRANSIENT RESPONSE FOR VARIABLE NUMBER  5                 PAGE 23
    EXCITED BY PULSE OF MAGNITUDE      10.0000000 IN FUNCTION NUMBER 1      (REFER TO PRINT-OUT PAGE 22)
          PARAMETER 1=  -2.6500000E-05                PARAMETER 2=  -4.9999999E-04                PARAMETER 3=   5.0000000E 00

            F(T) APPROXIMATELY GIVEN BY                              T               F(T)               T               F(T)
-1.345E-01        EXP(-1.005E-02 X TIME)      +    I       0.                   7.05470E-03       5.10000E 01      -1.23232E-02
 6.443E 01        EXP(-1.300E-01 X TIME)      +    I       1.00000E 00          6.39582E-03       5.20000E 01      -1.82104E-02
-1.508E 02        EXP(-2.290E-01 X TIME)      +    I       2.00000E 00          5.27167E-03       5.30000E 01      -2.35364E-02
 5.877E 01        EXP(-2.876E-01 X TIME)      +    I       3.00000E 00          4.71592E-03       5.40000E 01      -2.83363E-02
-3.098E 00        EXP(-4.167E-01 X TIME)      +    I       4.00000E 00          5.31369E-03       5.50000E 01      -3.26450E-02
-3.897E-05        EXP(-1.188E 00 X TIME)      +    I       5.00000E 00          7.87564E-03       5.60000E 01      -3.64970E-02
 5.631E-04        EXP(-1.406E 00 X TIME)      +    1       6.00000E 00          1.32208E-02       5.70000E 01      -3.93263E-02
 1.069E 02        EXP(-1.620E-01 X TIME)      X    I       7.00000E 00          2.19876E-02       5.80000E 01      -4.27633E-02
   COS( 5.553E-02 X TIME +  1.278E 00)        +    1       8.00000E 00          3.45051E-02       5.90000E 01      -4.56455E-02
                                                           9.00000E 00          5.07562E-02       6.00000E 01      -4.79967E-02
                                                           1.00000E 01          7.04038E-02       6.10000E 01      -5.00471E-02
                                                           1.10000E 01          9.28401E-02       6.20000E 01      -5.18235E-02
                                                           1.20000E 01          1.17281E-01       6.30000E 01      -5.33506E-02
                                                           1.30000E 01          1.42840E-01       6.40000E 01      -5.46519E-02
                                                           1.40000E 01          1.68605E-01       6.50000E 01      -5.57488E-02
                                                           1.50000E 01          1.93708E-01       6.60000E 01      -5.66613E-02
                                                           1.60000E 01          2.17367E-01       6.70000E 01      -5.74080E-02
                                                           1.70000E 01          2.38920E-01       6.80000E 01      -5.80056E-02
                                                           1.80000E 01          2.57839E-01       6.90000E 01      -5.84696E-02
                                                           1.90000E 01          2.73741E-01       7.00000E 01      -5.88143E-02
                                                           2.00000E 01          2.86301E-01       7.10000E 01      -5.90526E-02
                                                           2.10000E 01          2.95643E-01       7.20000E 01      -5.91961E-02
                                                           2.20000E 01          3.01523E-01       7.30000E 01      -5.92554E-02
                                                           2.30000E 01          3.04117E-01       7.40000E 01      -5.92402E-02
                                                           2.40000E 01          3.03595E-01       7.50000E 01      -5.91591E-02
                                                           2.50000E 01          3.00190E-01       7.60000E 01      -5.90200E-02
                                                           2.60000E 01          2.94176E-01       7.70000E 01      -5.88299E-02
                                                           2.70000E 01          2.85855E-01       7.80000E 01      -5.85950E-02
                                                           2.80000E 01          2.75545E-01       7.90000E 01      -5.83211E-02
                                                           2.90000E 01          2.63566E-01       8.00000E 01      -5.80131E-02
                                                           3.00000E 01          2.50232E-01       8.10000E 01      -5.76756E-02
                                                           3.10000E 01          2.35842E-01       8.20000E 01      -5.73126E-02
                                                           3.20000E 01          2.20678E-01       8.30000E 01      -5.69276E-02
                                                           3.30000E 01          2.05000E-01       8.40000E 01      -5.65239E-02
                                                           3.40000E 01          1.89040E-01       8.50000E 01      -5.61043E-02
                                                           3.50000E 01          1.73006E-01       8.60000E 01      -5.56712E-02
                                                           3.60000E 01          1.57078E-01       8.70000E 01      -5.52269E-02
                                                           3.70000E 01          1.41412E-01       8.80000E 01      -5.47734E-02
                                                           3.80000E 01          1.26136E-01       8.90000E 01      -5.43123E-02
                                                           3.90000E 01          1.11358E-01       9.00000E 01      -5.38452E-02
                                                           4.00000E 01          9.71610E-02       9.10000E 01      -5.33734E-02
                                                           4.10000E 01          8.36097E-02       9.20000E 01      -5.28981E-02
                                                           4.20000E 01          7.07511E-02       9.30000E 01      -5.24203E-02
                                                           4.30000E 01          5.86164E-02       9.40000E 01      -5.19409E-02
                                                           4.40000E 01          4.72231E-02       9.50000E 01      -5.14608E-02
                                                           4.50000E 01          3.65770E-02       9.60000E 01      -5.09806E-02
                                                           4.60000E 01          2.66741E-02       9.70000E 01      -5.05009E-02
                                                           4.70000E 01          1.75023E-02       9.80000E 01      -5.00222E-02
                                                           4.80000E 01          9.04270E-03       9.90000E 01      -4.95450E-02
                                                           4.90000E 01          1.27118E-03       1.00000E 02      -4.90697E-02
                                                           5.00000E 01         -5.84041E-03
```

```
      PLCT OF TRANSIENT RESPONSE FOR VARIABLE NUMBER  5           PAGE 24
  EXCITED BY PULSE OF MAGNITUDE      10.0000000                            (REFER TO PRINT-OUT PAGE 22)
        PARAMETER 1=  -2.6500000E-05              PARAMETER 2=  -4.9999999E-04              PARAMETER 3=   5.0000000E 00
```

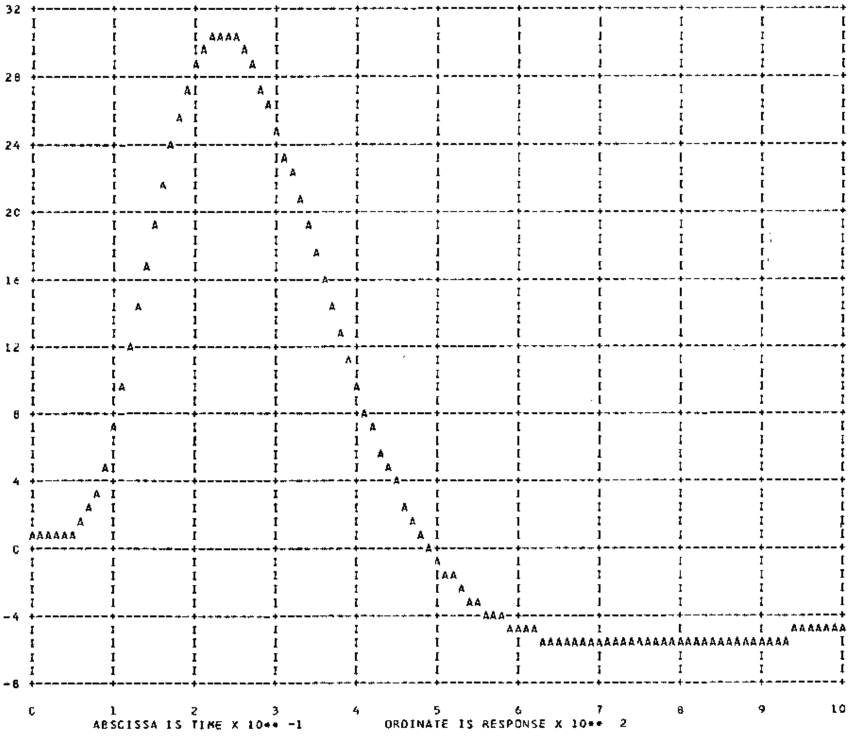


FIG. 27 SAMPLE OUTPUT – RESPONSE TO A PULSE INPUT

```
          TRANSIENT RESPONSE FOR VARIABLE NUMBER  5              PAGE 25
  EXCITED BY STEP  OF MAGNITUDE       1.0000000 IN FUNCTION NUMBER 1      (REFER TO PRINT-OUT PAGE 22)
          PARAMETER 1=  -2.6500000E-05              PARAMETER 2=  -4.9999999E-04              PARAMETER 3=    5.0000000E 00

          F(T) APPROXIMATELY GIVEN BY                             T                F(T)              T              F(T)
-3.630E-02                                       +   I     0.                -3.99151E-03      5.10000E 01     7.04476E-01
 1.338E 00        EXP(-1.005E-02 X TIME)         +   I     1.00000E 00       -3.28195E-03      5.20000E 01     7.02944E-01
-4.956E 01        EXP(-1.300E-01 X TIME)         +   I     2.00000E 00       -2.70241E-03      5.30000E 01     7.00852E-01
 6.558E 01        EXP(-2.299E-01 X TIME)         +   I     3.00000E 00       -2.20972E-03      5.40000E 01     6.98255E-01
-2.043E 01        EXP(-2.876E-01 X TIME)         +   I     4.00000E 00       -1.72091E-03      5.50000E 01     6.95202E-01
 7.436E-01        EXP(-4.167E-01 X TIME)         +   I     5.00000E 00       -1.08218E-03      5.60000E 01     6.91741E-01
 3.281E-06        EXP(-1.188E 00 X TIME)         +   I     6.00000E 00       -5.32866E-05      5.70000E 01     6.87916E-01
-4.005E-05        EXP(-1.406E 00 X TIME)         +   I     7.00000E 00        1.67644E-03      5.80000E 01     6.83769E-01
 6.243E 01        EXP(-1.620E-01 X TIME)         X   I     8.00000E 00        4.46934E-03      5.90000E 01     6.79335E-01
   COS( 5.553E-02 X TIME + -1.533E 00)           +   I     9.00000E 00        8.70234E-03      6.00000E 01     6.74650E-01
                                                           1.00000E 01        1.47339E-02      6.10000E 01     6.69746E-01
                                                           1.10000E 01        2.28761E-02      6.20000E 01     6.64650E-01
                                                           1.20000E 01        3.33690E-02      6.30000E 01     6.59389E-01
                                                           1.30000E 01        4.63693E-02      6.40000E 01     6.53988E-01
                                                           1.40000E 01        6.19438E-02      6.50000E 01     6.48466E-01
                                                           1.50000E 01        8.00683E-02      6.60000E 01     6.42844E-01
                                                           1.60000E 01        1.00637E-01      6.70000E 01     6.37139E-01
                                                           1.70000F 01        1.23471E-01      6.80000E 01     6.31367E-01
                                                           1.80000E 01        1.48333E-01      6.90000E 01     6.25543E-01
                                                           1.90000E 01        1.74938E-01      7.00000E 01     6.19677E-01
                                                           2.00000E 01        2.02972E-01      7.10000E 01     6.13783E-01
                                                           2.10000E 01        2.32102E-01      7.20000E 01     6.07870E-01
                                                           2.20000E 01        2.61988E-01      7.30000E 01     6.01947E-01
                                                           2.30000E 01        2.92297E-01      7.40000E 01     5.96021E-01
                                                           2.40000E 01        3.22707E-01      7.50000E 01     5.90101E-01
                                                           2.50000E 01        3.52919E-01      7.60000E 01     5.84192E-01
                                                           2.60000E 01        3.82658E-01      7.70000E 01     5.78299E-01
                                                           2.70000E 01        4.11678E-01      7.80000E 01     5.72427E-01
                                                           2.80000E 01        4.39763E-01      7.90000E 01     5.66581E-01
                                                           2.90000E 01        4.66731E-01      8.00000E 01     5.60764E-01
                                                           3.00000E 01        4.92431E-01      8.10000E 01     5.54979E-01
                                                           3.10000E 01        5.16742E-01      8.20000E 01     5.49230E-01
                                                           3.20000E 01        5.39574E-01      8.30000E 01     5.43518E-01
                                                           3.30000E 01        5.60861E-01      8.40000E 01     5.37845E-01
                                                           3.40000E 01        5.80564E-01      8.50000E 01     5.32213E-01
                                                           3.50000E 01        5.98666E-01      8.60000E 01     5.26624E-01
                                                           3.60000E 01        6.15169E-01      8.70000E 01     5.21079E-01
                                                           3.70000E 01        6.30091E-01      8.80000E 01     5.15579E-01
                                                           3.80000E 01        6.43464E-01      8.90000E 01     5.10125E-01
                                                           3.90000E 01        6.55335E-01      9.00000E 01     5.04717E-01
                                                           4.00000E 01        6.65755E-01      9.10000E 01     4.99356E-01
                                                           4.10000E 01        6.74788E-01      9.20000E 01     4.94042E-01
                                                           4.20000E 01        6.82501E-01      9.30000E 01     4.88777E-01
                                                           4.30000E 01        6.88963E-01      9.40000E 01     4.83558E-01
                                                           4.40000E 01        6.94249E-01      9.50000E 01     4.78388E-01
                                                           4.50000E 01        6.98432E-01      9.60000E 01     4.73266E-01
                                                           4.60000E 01        7.01589E-01      9.70000E 01     4.68192E-01
                                                           4.70000E 01        7.03792E-01      9.80000E 01     4.63166E-01
                                                           4.80000E 01        7.05113E-01      9.90000E 01     4.58188E-01
                                                           4.90000E 01        7.05623E-01      1.00000E 02     4.53257E-01
                                                           5.00000E 01        7.05389E-01
```

```
       PLOT OF TRANSIENT RESPONSE FOR VARIABLE NUMBER  5          PAGE 26
  EXCITED BY STEP  OF MAGNITUDE       1.0000000                        (REFER TO PRINT-OUT PAGE 22)
         PARAMETER 1=  -2.6500000E-05             PARAMETER 2=  -4.9999999E-04             PARAMETER 3=    5.0000000E 00
```

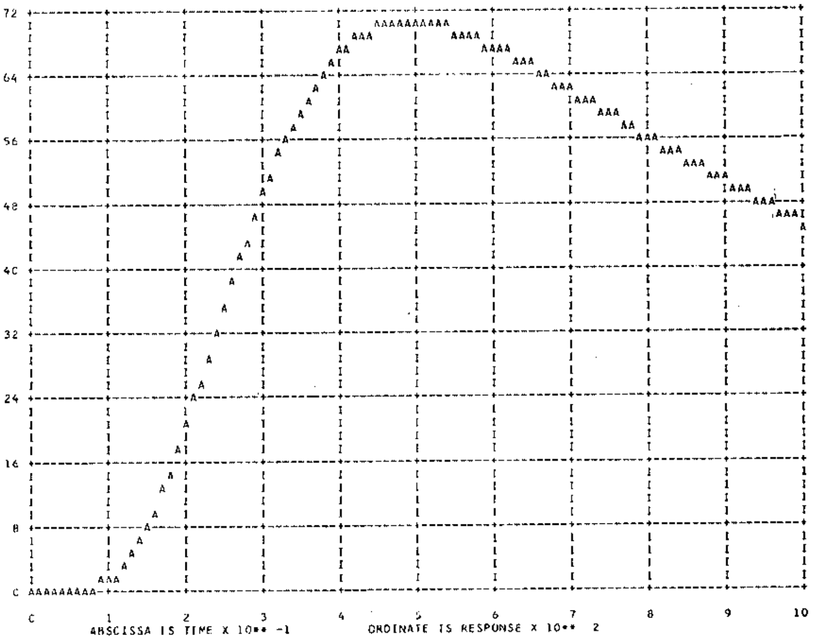


FIG. 28 SAMPLE OUTPUT – RESPONSE TO A STEP INPUT

```
          TRANSIENT RESPONSE FOR VARIABLE NUMBER  5              PAGE 27
   EXCITED BY RAMP  OF MAGNITUDE       0.1000000 IN FUNCTION NUMBER 1      (REFER TO PRINT-OUT PAGE 22)
          PARAMETER 1=  -2.6500000E-05              PARAMETER 2=  -4.9999999E-04              PARAMETER 3=   5.0000000E 00

          F(T) APPROXIMATELY GIVEN BY                              T                  F(T)              T                  F(T)
 9.907E 00                                         +    I    0.                   2.11740E-03     5.10000E 01       1.80070E 00
-3.630E-03              X  TIME                    +    I    1.00000E 00          1.75500E-03     5.70000E 01       1.87108E 00
-1.331E 01       EXP(-1.005E-02 X TIME)            +    I    2.00000E 00          1.45733E-03     5.30000E 01       1.94127E 00
 3.812E 01       EXP(-1.300E-01 X TIME)            +    I    3.00000E 00          1.21212E-03     5.40000E 01       2.01123E 00
-2.853E 01       EXP(-2.299E-01 X TIME)            +    I    4.00000E 00          1.01495E-03     5.50000E 01       2.08091E 00
 7.104E 00       EXP(-2.876E-01 X TIME)            +    I    5.00000E 00          8.73089E-04     5.60000E 01       2.15026E 00
-1.785E-01       EXP(-4.167E-01 X TIME)            +    I    6.00000E 00          8.11815E-04     5.70000E 01       2.21925E 00
-2.762E-07       EXP(-1.188E 00 X TIME)            +    I    7.00000E 00          8.85725E-04     5.80000E 01       2.28783E 00
 2.849E-06       EXP(-1.406E 00 X TIME)            +    I    8.00000E 00          1.18297E-03     5.90000E 01       2.35599E 00
 3.646E 01       EXP(-1.620E-01 X TIME)            X    I    9.00000E 00          1.82784E-03     6.00000E 01       2.42369E 00
   COS( 5.553E-02 X TIME + -4.344E 00)             +    I    1.00000E 01          2.98369E-03     6.10000E 01       2.49091E 00
                                                             1.10000E 01          4.84514E-03     6.20000E 01       2.55763E 00
                                                             1.20000E 01          7.63714E-03     6.30000E 01       2.62384E 00
                                                             1.30000E 01          1.16028E-02     6.40000E 01       2.68951E 00
                                                             1.40000E 01          1.69967E-02     6.50000E 01       2.75463E 00
                                                             1.50000E 01          2.40764E-02     6.60000E 01       2.81920E 00
                                                             1.60000E 01          3.30921E-02     6.70000E 01       2.88320E 00
                                                             1.70000E 01          4.42796E-02     6.80000E 01       2.94662E 00
                                                             1.80000E 01          5.78541E-02     6.90000E 01       3.00947E 00
                                                             1.90000E 01          7.40042E-02     7.00000E 01       3.07173E 00
                                                             2.00000E 01          9.28892E-02     7.10000E 01       3.13340E 00
                                                             2.10000E 01          1.14635E-01     7.20000E 01       3.19449E 00
                                                             2.20000E 01          1.39335E-01     7.30000E 01       3.25498E 00
                                                             2.30000E 01          1.67047E-01     7.40000E 01       3.31488E 00
                                                             2.40000E 01          1.97797E-01     7.50000E 01       3.37418E 00
                                                             2.50000E 01          2.31582E-01     7.60000E 01       3.43290E 00
                                                             2.60000E 01          2.68366E-01     7.70000E 01       3.49102E 00
                                                             2.70000E 01          3.08089E-01     7.80000E 01       3.54856E 00
                                                             2.80000E 01          3.50670E-01     7.90000E 01       3.60551E 00
                                                             2.90000E 01          3.96005E-01     8.00000E 01       3.66187E 00
                                                             3.00000E 01          4.43974E-01     8.10000E 01       3.71766E 00
                                                             3.10000E 01          4.94445E-01     8.20000E 01       3.77287E 00
                                                             3.20000E 01          5.47273E-01     8.30000E 01       3.82751E 00
                                                             3.30000E 01          6.02308E-01     8.40000E 01       3.88158E 00
                                                             3.40000E 01          6.59392E-01     8.50000E 01       3.93508E 00
                                                             3.50000E 01          7.18367E-01     8.60000E 01       3.98802E 00
                                                             3.60000E 01          7.79072E-01     8.70000E 01       4.04040E 00
                                                             3.70000E 01          8.41348E-01     8.80000E 01       4.09224E 00
                                                             3.80000E 01          9.05039E-01     8.90000E 01       4.14352E 00
                                                             3.90000E 01          9.69991E-01     9.00000E 01       4.19426E 00
                                                             4.00000E 01          1.03606E 00     9.10000E 01       4.24447E 00
                                                             4.10000E 01          1.10310E 00     9.20000E 01       4.29414E 00
                                                             4.20000E 01          1.17097E 00     9.30000E 01       4.34328E 00
                                                             4.30000E 01          1.23955E 00     9.40000E 01       4.39189E 00
                                                             4.40000E 01          1.30872E 00     9.50000E 01       4.43999E 00
                                                             4.50000E 01          1.37837E 00     9.60000E 01       4.48757E 00
                                                             4.60000E 01          1.44838E 00     9.70000E 01       4.53464E 00
                                                             4.70000E 01          1.51865E 00     9.80000E 01       4.58121E 00
                                                             4.80000E 01          1.58911E 00     9.90000E 01       4.62728E 00
                                                             4.90000E 01          1.65965E 00     1.00000E 02       4.67285E 00
                                                             5.00000E 01          1.73021E 00
```

```
    PLOT OF TRANSIENT RESPONSE FOR VARIABLE NUMBER  5          PAGE 28
EXCITED BY RAMP  OF MAGNITUDE       0.1000000                          (REFER TO PRINT-OUT PAGE 22)
       PARAMETER 1=  -2.6500000E-05                PARAMETER 2=  -4.9999999E-04                PARAMETER 3=   5.0000000E 00
```

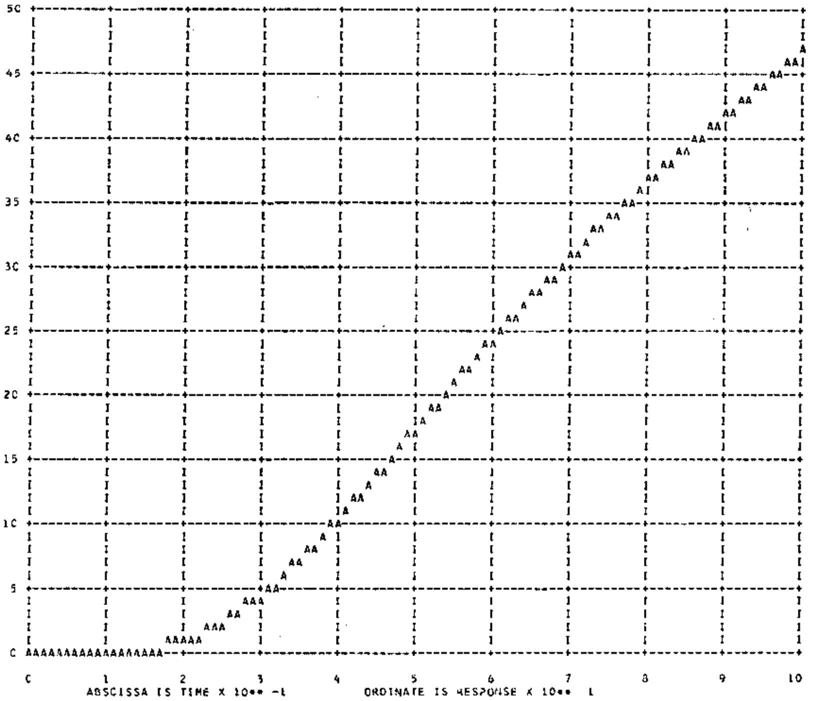


FIG. 29 SAMPLE OUTPUT – RESPONSE TO A RAMP INPUT

3.4 Sample CALMAT Coding for LASS KEY 1

In the preceding example, the matrix elements were computed (as listed in Figure 11) and entered on input sheets 2, 3, and 4. These elements would have to be recalculated for each change in constants describing the system.

KEY 1 of LASS provides for writing a special subroutine, CALMAT, to compute input data for data sheets 2, 3, and 4 from the original equations. (A dummy CALMAT is provided in the code in case KEY 1 is not used.)

In order to illustrate this procedure, a sample CALMAT subroutine was prepared to calculate the matrix elements for the sample problem from Equations 8'. To make the procedures more graphic, the matrix of Equations 8' is shown in Figure 30 (in the same format as used for the numerical values of the matrix elements in Figure 11).

50/d

j \ i	T_b 1	T_c 2	T_d 3	T_e 4	T^* 5	Ψ 6	ϕ 7	k 8	T_m 9	Excitation Variable, T_5
1	$F^2 + F(H_a+H_b)s + H_aH_bs^2$									F^2
2	$-F$	$F+H_cs$				$-f$				
3		$-F$	$F+H_ds$			$-W_1(0.5-f)$				
4			$-F$	$F+H_es$		$-W_2(0.5-f)$				
5				$-F$	$F + (H_f+H^*)s + \frac{H_fH^*}{F} s^2$	$-W_3(0.5-f)$				
6						1	$\frac{-\Psi^0}{s}$			
7							$0.00875 + 0.00739s$	$-0.0943 - 1.45s - s^2$		
8		$-\alpha_3$						1	$-\alpha_2$	
9			$-W_4$	$-W_5$	$-W_6 - \frac{W_6H^*}{F} s$				1	
K(J)	3	2	2	2	3	1	2	3	1	

FIG. 30 SAMPLE PROBLEM MATRIX

Figure 31 shows a FORTRAN listing of the sample CALMAT subroutine. The listing follows directly the development of data on input sheets 2, 3, and 4 (Figures 13, 14, and 15). A comparison shows that the same mnemonics are used in CALMAT as appear on the input sheets. This subroutine replaces the dummy shown in the listing of LASS in the Appendix.

```
C       BB OCCHIPINTI  COST CODE 8210
C       CALMAT SUBROUTINE  FOR SAMPLE PROBLEM   LASS DP   4-13-64
C
        SUBROUTINE CALMAT( N, NPAR, NVAR,NSH, IPAGE, NER)
C
C       DIMENSION AND COMMON STATEMENTS ARE UNCHANGED FROM MAIN PROGRAM
C
        DIMENSION  A(50,50,5),  B(50,50), C(201), K(50), NK(50),
       1 G(50,8,5), CHAR(16,50), NC(4), MA(8), NT(8), V(50,4)  .
        COMMON  A, B, C, K, NK, G, CHAR, NC, MA, NT, V
C
C       READ INPUT DATA CARDS FOR SUBROUTINE CALMAT
C
        READ 100, FLOW, HA, HB, HC, HD, HE, HF, HSTAR, FRAC, W1, W2,
       1  W3, W4, W5, W6
    100 FORMAT(7E10.3)
C
C       PRINT INPUT DATA FOR SUBROUTINE CALMAT
C
        PRINT 101, FLOW, HA, HB, HC, HD, HE, HF, HSTAR, FRAC, W1, W2,
       1 W3, W4, W5, W6
    101 FORMAT(5H FLOWE12.3, 2X,2HHAE12.3, 2X,2HHBE12.3,  2X, 2HHCE12.3,
       1 2X, 2HHDE12.3,  2X,  2HHEE12.3, 2X, 2HHFE12.3/6H HSTARE12.3, 2X,
       24HFRACE12.3,  2X,  2HW1E12.3, 2X, 2HW2E12.3,  2X,  2HW3E12.3, 2X,
       3 2HW4E12.3,  2HW5E12.3/    4H  W6F12.3)
C
C       BEGIN INPUT FOR  PAGE 2
C
        NSH=1
        K(1) =3
        K(2) = 2
        K(3) = 2
        K(4) = 2
        K(5) = 3
        K(6) = 1
        K(7) = 2
        K(8) = 3
        K(9) = 1
        A(1,1,1)  =  FLOW * FLOW
        A(1,1,2) = (HA+ HB) * FLOW
        A(1,1,3)  =  HA * HB
        A(2,1,1)  =  -FLOW
        A(2,2,1)  =  FLOW
        A(2,2,2)  =  HC
        A(2,6,1)  =  - FRAC
        A(3,2,1)  =  - FLOW
        A(3,3,1)  =  FLOW
        A(3,3,2)  =  HD
        A(3,6,1)  =  - W1 *( 0.5 - FRAC)
        A( 4,3,1) =   - FLOW
        A( 4,4,1) =     FLOW
        A( 4,4,2) =  HE
        A( 4,6,1) =  - W2 * (0.5 - FRAC)
        A( 5,4,1) = -FLOW
        A(5,5,1)  =  FLOW
        A(5,5,2)  =  HF + HSTAR
        A(5,5,3)  = (HF * HSTAR)/ FLOW
        A(5,6,1)  =  -W3 * (0.5 - FRAC)
        A(6,6,1)  =  1.0
        A(7,8,1) = -0.0943
        A(7,8,2) = -1.45
        A(7,8,3) = -1.0
        A(8,8,1) =  1.0
        A(9,3,1) =  -W4
        A(9,4,1) = - W5
        A(9,5,1) = - W6
        A(9,5,2) = (-W6 * HSTAR)/ FLOW
```

```
C
C       BEGIN  INPUT FOR PAGE 3 OF INPUT SHEETS
C
        NC(1) = 2
        NC(2)= 9
        NC(3)= 7
        MA(1) = 2
        MA(2) = 1
        MA(3) = 1
        MA(4) = 1
        MA(5) = 2
        MA(6) = 1
        NT(1)= 0
        NT(2)=0
        NT(3)= 0
        NT(4)= 0
        NT(5)= 0
        NT(6)= -1
        G(2,1,1) =  FLOW
        G(2,1,2) =  HC
        G(3,1,1) =  - FLOW
        G(8,2,1) =  - 1.0
        G(9,3,1) =  1.0
        G(8,4,1) =  -1.0
        G(7,5,1) = 0.00875
        G(7,5,2) = 0.00739
        G(6,6,1) = -1.0
C
C       PAGE 4  OF INPUT SHEET
C
        V(1,1) =  FLOW * FLOW
        RETURN
        END( 2, 0, 0, 0, 1)
```

FIG. 31 FORTRAN LISTING OF SAMPLE CALMAT

A READ statement is provided to read in the necessary data.

Page 1 of the input sheets is required as before (Figure 12) except that K1 is now unity instead of zero since KEY 1 of LASS is being used. Pages 2, 3, and 4 are no longer required. Pages 5 through 8 are used as before.

While this example is a rather trivial one in which it is probably easier to use the LASS input sheets directly, there are often cases in which the matrix elements involve the steady-state operating conditions in a more complicated way, or in which the steady-state operating conditions themselves must be calculated by a tedious iterative procedure from the values of the system parameters, such as the power level. Any coding required to carry out such calculations can be incorporated in the CALMAT.

4. Description of the LASS Code

4.1 Subroutines in LASS 1

4.1.1 Flowsheet for LASS 1

As explained in Section 2.2, the LASS code is in three sections for use with a 32K IBM 704 with at least four tape units.

The flowsheet for LASS 1 is shown in Figure 32.

This section requires the following subroutines:

EFM
PUN
TESKEY
CALMAT
ERR
WTAP 1
LTAP 2

A complete listing of LASS and its subroutines is shown in Appendix I.

4.1.2 Subroutine EFM

This is a floating point trap subroutine that sets underflow to zero and stops on overflow. It was obtained from the AEC Computing and Applied Mathematics Center, Institute of Mathematical Sciences, New York University, New York.

4.1.3 Subroutine PUN

This is a SAP-coded subroutine that punches out at the beginning of each problem a special card with LASS spelled out in punches on the card. This makes it easier to separate the cards punched out for each problem.

4.1.4 Subroutine TESKEY

The purpose of TESKEY is to test the consistency of data on the KEY card. The following tests are performed:

- All digits of six-digit KEY word must be zero or one.
- If processing through KEY 1 is called for, then processing must be called for KEY 1 only, or else also KEY 2 and/or KEY 4. That is, if the first digit of KEY is 1, acceptable KEY words are:

100000
11X0XX
10X1XX
11X1XX

where X's may be zero or 1.

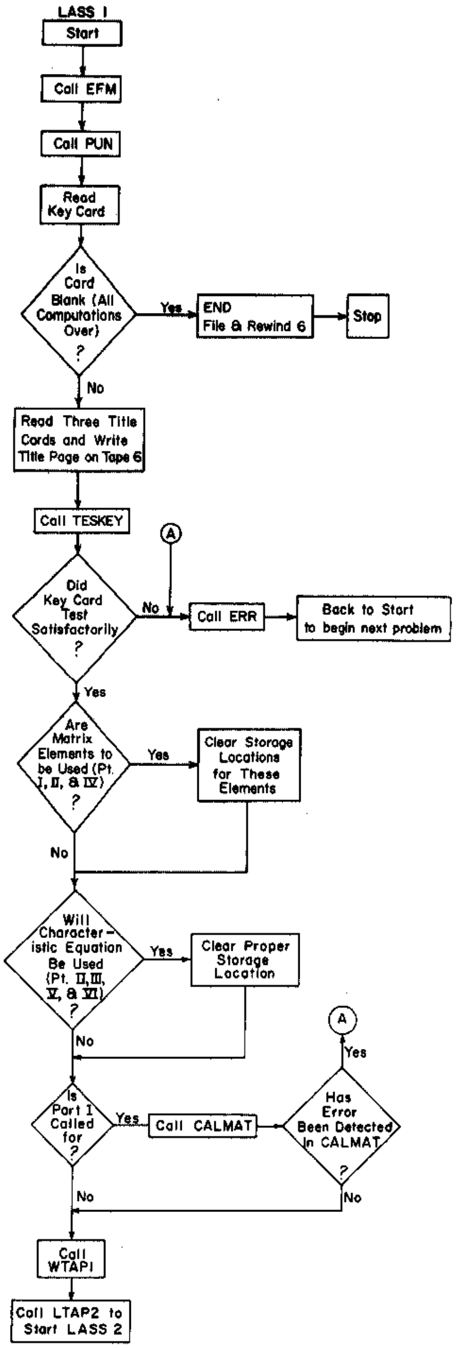


FIG. 32 FLOWSHEET FOR LASS 1

- If KEY 3 is to be processed, NPAR must be greater than zero.

- If KEY 4 is to be processed, NVAR must be greater than zero.

- $N \leqslant 50$.

- $0 \leqslant NPAR \leqslant 4$.

- $0 \leqslant NVAR \leqslant 4$.

In the evolution of LASS, there developed a number of vestigial remnants in the body that no longer serve a useful purpose, but have not been removed. TESKEY perhaps is the major example of such a remnant.

4.1.5 Subroutine CALMAT

CALMAT can be coded specially to assist in preparing input data, as illustrated in Section 3.4. It can be used for such things as linearizing the original equations, and calculating steady-state operating conditions, and calculating the required matrix elements. Even if KEY 1 is not used, some CALMAT subroutine must be present — otherwise machine comes up with missing subroutine stop.

The listing in Appendix I gives a dummy CALMAT subroutine that can be used for this purpose. In the event that a problem calling for KEY 1 is run with this dummy CALMAT inadvertently left on the tape, an error signal is generated and a diagnostic statement ("special CALMAT subroutine missing") is printed out.

The following source cards must be included in a FORTRAN-coded CALMAT subroutine:

```
SUBROUTINE CALMAT (N, NPAR, NVAR, NSH, IPAGE, NER)

DIMENSION  A(50,50,5),  B(50,50), C(201), K(50), NK(50),
           G(50,8,5), CHAR(16,50), NC(4), MA(8),
           NT(8), V(50,4)

COMMON  A, B, C, K, NK, G, CHAR, NC, MA, NT, V
```

If the particular version that is used requires reading in data from cards, then provision for testing these cards may be coded into the subroutine. It is also possible to provide for other tests, such as checking to see if various calculated quantities exceed required limits. If any of these fail, set NER = 11 and return to the main program for LASS 1. A diagnostic "Error in special CALMAT subroutine, please check" will be printed and LASS will go on to the next problem.

4.1.6 Subroutine ERR

This is a diagnostic program used in conjunction with various error tests throughout the program.

Each time data cards are read in, there is a test to see that proper cards have been provided (that is, cards are not out of place). This testing is provided to prevent machine stoppages that may arise when trying to process improper data; it is not completely foolproof — if a card with Hollerith data becomes misplaced and is read as ordinary fixed-point or floating point numbers, an illegal character stop will result.

A storage location designated by the mnemonic NER has set aside to indicate test failure. So long as there is a zero in this location, the data cards read in to that point are correct and in order. A non-zero value in this location signifies a bad data card; the magnitude of this quantity then enables ERR to choose which one of 19 diagnostic statements to print out.

When bad cards have been detected in a given problem, there is no need to continue on this calculation. ERR will pass the remaining data cards for that problem through the card reader to make ready for starting the next problem from the beginning of LASS 1.

4.1.7 Subroutine WTAP 1

This is a FORTRAN subroutine that writes scratch data, such as the A, G, and V matrix elements, on Tape 3 to preserve it from LASS 1 to LASS 2.

4.1.8 Subroutine LTAP 2

This is a SAP-coded program to simulate the load tape sequence to load from Logical Unit 2.

4.2 Subroutines in LASS 2

4.2.1 Flowsheet for LASS 2

The flowsheet for LASS 2 is shown in Figure 33. The following subroutines are called:

Subroutines in LASS 2 but not in LASS 1

RTAP 1
REMAT
LTAP 1
PUNMAT
PRIMAT
RECHAR
MESS 1 (calls POLYX that calls DETEV)
PUNCHA
PRICHA
STAB (calls ROUTH and HOLLER)
WTAP 2

Subroutines in LASS 2 and in LASS 1

EFM
ERR
LTAP 2

4.2.2 Subroutine RTAP 1

This reads in scratch data from Tape 3 that had been written at the end of LASS 1.

4.2.3 Subroutine REMAT

This reads elements of A, G, and V matrices (Figures and data sheets 2, 3, and 4) from cards when KEY 1 is not used to provide these elements and KEY 2 and/or KEY 4 are to be processed.

4.2.4 Subroutine LTAP 1

This is a SAP-coded program to simulate the load tape sequence to load from Logical Unit 1.

4.2.5 Subroutine PUNMAT

This subroutine follows CALMAT in LASS 1. After the elements of the A, G, and V matrices and other necessary terms have been calculated by CALMAT, they are then punched out on cards by PUNMAT in the proper form, as shown in Figures 2, 3, and 4, for processing later, if desired.

When these cards are hand punched, the element magnitudes are given an E14.8 format. The standard FORTRAN I/O programs place an extra zero in front of the decimal point when punching and printing. To suppress this extra character, punching format is 1PE14.7, which can be read by E14.8.

37%

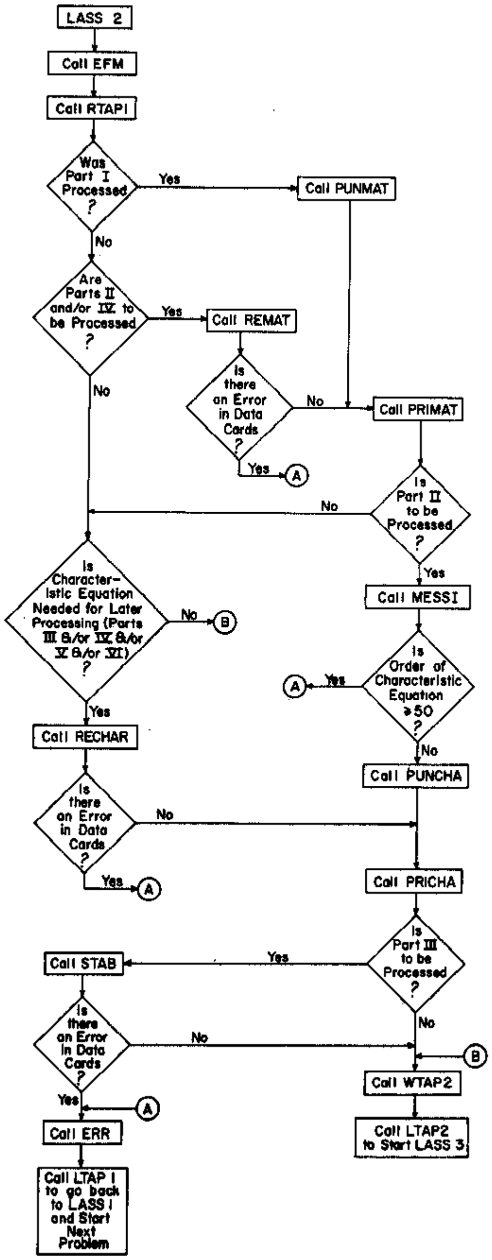


FIG. 33 FLOWSHEET FOR LASS 2

PUNMAT operates on each matrix in turn, punching out first the lead card, if such is required, of Figures 2 and 3. The matrix is then scanned for non-zero elements. (There is no need to put zero values on the cards.) When such an element is found, it is transferred to a punch buffer along with the indices (I,J,K) that identify the element. The buffer will hold three elements; when filled, one card is punched and the process is repeated until the entire matrix has been scanned.

Before the next matrix, the buffer is punched out one more time. This assures that all elements are on cards, even when the number of non-zero elements is not an integral multiple of three. On the other hand, when the number of non-zero elements is a multiple of three, the last two cards punched for each matrix are exact duplicates except for the integer in column one. This causes no difficulties on subsequent use.

One other consideration that was included in the logic of this subroutine involves matrices that have only one or two non-zero elements (such may often be the case in the V matrix and less often in the G matrix). To prevent difficulties in this event, the punch buffer is filled before scanning begins with non-zero values for the indices and zero for the elements. When non-zero elements are found in the scan, they are placed in the buffer from right to left, so that the non-zero value will over-ride the zero value to the left of it.

<u>Example</u>: V matrix with 1, 1 element equal to 1.0, all others zero.

<u>Original setup in buffer</u>

I	J	V	I	J	V	I	J	V
1	1	0.0	2	1	0.0	3	1	0

<u>Final setup in buffer and on card</u>

1	1	0.0	2	1	0.0	1	1	1.0

4.2.6 Subroutine PRIMAT

This writes on the output Tape 6 the elements of the A, G, and V matrices. This subroutine is called whether these matrices are provided by KEY 1 or read from cards.

It is obvious that elements should be made available for examination when they are calculated by CALMAT in KEY 1.

In the second case, when the matrices are read from cards, putting them on tape for printing serves two functions: (1) easy check of the input data at any subsequent time, and (2) an excellent identification of the problem.

4.2.7 Subroutine RECHAR

This reads coefficients of the characteristic equation (Figure 5) from cards when KEY 2 is not used to provide the coefficients and any subsequent part is to be processed.

4.2.8 Subroutine MESS 1

This is the basic subroutine for KEY 2, which expands the characteristic matrix

$$G' = A + \sum_{p=1}^{NPAR} \alpha_p B_p$$

into the characteristic equation

$$\sum_{g=1}^{2^{NPAR}} Q_g \sum_{p=1}^{NPAR} \alpha_p^{m_p}$$

where $m_p = 0$ or 1

$$g = 1 + \sum_{p=1}^{NPAR} m_p 2^{p-1} = 1,\ 2\ \ldots\ 2^{NPAR}$$

Q_g = polynomial in s of order NL_{g-1}

The elements of G' are loaded into the machine core in two arrays called A and G, as described in the section Input Data Formats. As shown there, parameter α_p appears only in column NC(p) of G' and the elements of this column are of the form $A_{i,NC(p)} + \alpha_p B_{i,NC(p),p}$.

In matrix G, the elements of $A_{i,NC(p)}$ are loaded into column 2p-1, while $B_{i,NC(p),p}$ goes into column 2p.

From the basic rules of matrix algebra, it can be shown that Q_g is the determinant of the matrix obtained by placing either column 2p-1 or column 2p of the G matrix into column NC(p) of A.

When m_p is 0, column 2p-1 is used; when m_p is 1, column 2p is used. For example, consider $g = 11 = 1 + 2 + 2^3$, which implies that $m = 0$, $m_2 = 1$, $m_3 = 0$, and $m_4 = 1$; hence Q_{11} is the polynomial coefficient of $\alpha_2 \alpha_4$ and is the determinant of the matrix obtained by placing

column 1 of G in column NC(1) of A
column 4 of G in column NC(2) of A
column 5 of G in column NC(3) of A
column 8 of G in column NC(4) of A

MESS 1 sets up the equivalent of a counter wheel to count through the sequencing of g from 1 to 2^{NPAR}. For each value of g, the proper columns from the G matrix are placed in the A matrix and the proper values of MA(J) are placed in the K(J) array. The subroutine POLYX is then called to expand the matrix into the polynomial Q_g. Following this, the coefficients of Q_g are placed in the proper positions of the C array (Figure 5), after correcting for non-zero values of NSH and NT(J) (Figures 2 and 3).

4.2.9 Subroutine POLYX

This subroutine takes the matrix A with polynomial elements $A_{ij} = \sum_{r=0}^{M} a_{ijr}s^r$ and expands into the polynomial $Q_g = C_{g,1} + C_{g,2}s + \ldots + \ldots C_{g,NL_g}s^{NL_{g-1}}$. It is based upon the method described by Ojalvo(16). Since the program described in that reference was coded in machine language as a self-contained program for use with a special executive program, no attempt was made to use any of Ojalvo's coding in POLYX. Rather, POLYX was coded completely in FORTRAN using the logic described in Reference 16. The following description, taken from the reference, provided the basis for the programming and coding.

"In the ensuing discussion "n" is the order of the determinant, "m" the degree of the polynomial elements, and "s" the variable. In the polynomial-element square matrix each polynomial column is composed of m+1 term columns, ranging from the m^{th} degree term column to the zero degree term column. In the determinant evaluation this polynomial-element square matrix is decomposed into square matrices of the same order, which we shall call "simple" matrices, each column of which consists of single-term elements of the same degree. This decomposition involves extracting only one of the m+1 term columns in each j^{th} polynomial column of the original matrix to form the corresponding j^{th} column of the simple matrix. $1 \leqslant j \leqslant n$. For brevity we shall hereafter refer to the determinant of a square matrix as merely a "determinant". A polynomial-element determinant, for n = 2,

$m = 1$, is decomposed into simple determinants as follows:

$$\begin{vmatrix} a_{11}s+b_{11} & a_{12}s+b_{12} \\ a_{21}s+b_{21} & a_{22}s+b_{22} \end{vmatrix} = \underset{(1)}{\begin{vmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{vmatrix}} + \underset{(2)}{\begin{vmatrix} a_{11}s & b_{12} \\ a_{21}s & b_{22} \end{vmatrix}} + \underset{(3)}{\begin{vmatrix} b_{11} & a_{12}s \\ b_{21} & a_{22}s \end{vmatrix}} + \underset{(4)}{\begin{vmatrix} a_{11}s & a_{12}s \\ a_{21}s & a_{22}s \end{vmatrix}}$$

The evaluation of (1) produces the s^o term, the sum of the evaluations of (2) and (3) produce the s^1 term, and finally the evaluation of (4) produces the s^2 term. The original determinant remains intact in the data region, whereas the simple determinants are stored in the temporary region where they are destroyed in the evaluation process.

The technique used to decompose the original polynomial-element determinant into simple determinants is roughly analogous to a set of n counting wheels, one for each polynomial column. But, contrary to place-value number systems, the left-most wheel or digit is the least significant. The degree of the term column to be extracted from the j^{th} polynomial column of the original determinant, in forming the corresponding column of the current simple determinant, is determined by the digit appearing on the j^{th} counting wheel. A "0" means that the s^o-term column is to be extracted, a "1" that the s^1-term column is to be extracted, a "2", the s^2-term column, etc. The degree of the term resulting from the evaluation of a particular simple determinant is determined by the sum of the digits in the n counting wheels $\sum_{j=1}^{n} W_j$. The counting wheels are simulated in the computer by n consecutive storage cells in memory, whose fixed-point numbers are scaled at the maximum scale. (There is practically no limit to the degree of the polynomial elements that may be handled by this technique.) The digit configurations, in the order in which they are formed by the counting wheels, and the degree of the terms resulting from the evaluation of the corresponding simple determinants are as follows:

$n = 2,\ m = 1$	$n = 3,\ m = 1$	$n = 4,\ m = 2$
00-s^o	000-s^o	0000-s^o
10-s^1	100-s^1	1000-s^1
01-s^1	010-s^1	2000-s^2
11-s^2	110-s^2	0100-s^1
	001-s^1	1100-s^2
	101-s^2	2100-s^3
	011-s^2	0200-s^2
	111-s^3	2222-s^3

In the initial portion of the program the j^{th} polynomial column is examined, starting with the m^{th} degree term column and progressing downward to the zero degree term column, $1 \leqslant j \leqslant n$. The degree of the first non-zero term column encountered will establish the upper degree limit (U_j) of the j^{th} polynomial column, $1 \leqslant j \leqslant n$. Similarly the lower degree limit (L_j) of the non-zero term columns is established by examining the j^{th} polynomial column in the reverse order, from the zero degree term column to the m^{th} degree term column, $1 \leqslant j \leqslant n$. In the following example "0" indicates a zero term, "X" indicates a non-zero term, and "U_j/L_j" indicates the U_j and L_j of the j^{th} polynomial column, $1 \leqslant j \leqslant n$. For $n = 3$, $m = 2$, we have:

$$\begin{array}{ccccccccc} & 2/1 & & & 1/0 & & & 0/0 & \\ \hline X & X & 0 & 0 & 0 & X & 0 & 0 & X \\ X & 0 & 0 & 0 & X & X & 0 & 0 & X \\ X & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

On examination, the U_j-values of the three polynomial columns are found to be 2, 1, and 0, respectively, and the L_j-values are 1, 0, and 0, respectively. Thus in order to avoid extracting zero-term columns in forming simple determinants (whose values are zero) the j^{th} counting wheel, W_j is restricted to values $L_j \leqslant W_j \leqslant U_j$, where $1 \leqslant j \leqslant n$. Therefore the digit configurations, in the order in which they are formed, and the degree of the terms resulting from the corresponding simple determinants, are as follows:

2/1	1/0	0/0
1	0	$0-s^1$
2	0	$0-s^2$
1	1	$0-s^2$
2	1	$0-s^3$

The degree of the resulting polynomial is obtained by adding the upper degree limits of the n polynomial columns $\sum_{j=1}^{n} U_j$. Thus, it is determined that the resulting polynomial will be of third degree, and hence four terms are printed. Note that because there is no simple determinant to produce a constant term, a zero is printed in its place. The number of digit configurations of the counting wheels, and therefore, the number of simple determinants to evaluate, depends on the n-factor continued product of the number of non-zero term columns in the j^{th} polynomial column, $1 \leqslant j \leqslant n$, or

$\sum_{j=1}^{n} [(U_j-L_j) + 1]$. In the determinant shown above, this continued product is 2 x 2 x 1 = 4. If W_j were not restricted it could assume values $0 \leqslant W_j \leqslant 2$, and 3^3 or 27 simple determinants, most of which are zero, would have been formed."

To the above description, one exception must be noted. This is that the counting wheels are restricted to values $0 \leqslant W_j \leqslant U_j$. Instead of a lower limit value for each counting wheel, as Ojalvo used, the term NSH was introduced to accomplish the same purpose.

Note also that the number of determinants to be evaluated is $\prod_{j=1}^{M} (W_j + 1)$. This part of the program is by far the slowest section and it is extremely advisable to make changes in the original equations before making up the input data cards, if such changes will give a smaller value for the number of determinants to be evaluated.

POLYX itself sets up the counter wheels and places the proper elements of the three-dimensional A in a two-dimensional determinant B, calls DETEV to evaluate the determinant.

Some further comments on this method of expanding the matrix are in order, for this is by no means the only approach possible. Probably the most obvious one is to triangularize the matrix by Gaussian elimination. We considered this as an approach and decided that the method actually chosen was probably better for the following reasons:

- Triangularization of a matrix with polynomial elements means that each step in the process increases the power of each polynomial, thereby requiring enormously large storage capacity or a great deal of input/output from core to tape and back again. In the 704, which does not have buffered I/O, this appeared to lead to even slower processing times.

- Triangularization introduces extraneous polynomial factors in both numerator and denominator of the transfer function. With round-off and truncation errors entering in, these extraneous factors are often difficult to detect. This is one big fault in GALS.

Another method that came to our attention shortly after we had settled on the procedure used is described in Reference 20. This should be looked into further and might be used in place of POLYX, if it appears to be good.

4.2.10 Subroutine DETEV

This determinant evaluation subroutine was coded at SRL. A number of programs from the SHARE Library were found to be unsatisfactory. Those programs in SHARE had been written in conjunction with matrix inversion routines and were not suitable for zero-valued determinants. In the expansion of the characteristic matrix by Ojalvo's method, a large number of zero determinants are formed, and DETEV was called to provide an effective method for detecting these determinants and setting them identically equal to zero.

DETEV operates by Gaussian elimination, setting each subdiagonal element equal to zero by subtracting from each row a proper multiple of another row. Each time a subtraction is performed, the difference is compared with the subtrahend. If less than 10^{-6} times the subtrahend, the difference is set to zero. Pivoting to place maximum elements on the diagonal was considered but not used, for there was not enough improvement in accuracy to justify the increased machine time.

The SRL DETEV and various SHARE programs were tested on a single feedback system of twelve equations. DETEV gave results that agree to six or seven digits with the known solution; the results from the other programs were quite inaccurate.

This part of LASS is by far the slowest part. For example, one problem that was solved involved a set of 33 equations with four parameters. Therefore, MESS 1 set up 16 matrices to be expanded. The original equations were written so as to require the evaluation of some 73,000 determinants for each matrix, that is $\prod_{j=1}^{N} (U_j + 1) = 73,000$ (see POLYX write up). By a simple rewriting of a few of the equations, actually a preliminary adjustment of the G' matrix, this number was reduced to 4,096. Even after this 18-fold decrease in time the expansion of each matrix took approximately 100 minutes. Another example of this rewriting is shown in the sample problem, Section 3.1.

DETEV was originally written in FORTRAN as listed in Appendix I. In order to speed up the processing, the machine language listing of the subroutine was used as the starting point for an optimized program. From this listing, various SAP instructions that were superfluous (mostly saving and restoring index registers) were removed and a machine language program was assembled. This gave a 15% improvement in processing time.

4.2.11 Subroutine PUNCHA

This punches the coefficients of the characteristic equation on cards after they have been calculated by MESS 1.

This subroutine follows very closely the procedure and logic used in PUNMAT (Section 4.2.5).

4.2.12 Subroutine PRICHA

This writes on Tape 6 the coefficients of the characteristic equation. As with PRIMAT, this is done whether coefficients are calculated by KEY 2 or read in from cards.

4.2.13 Subroutine STAB

This is the master program for KEY 3 of LASS, the analysis of stability. STAB performs the following functions:

- It sets up proper flow paths defining the flow of the calculation, depending on the number of parameters in the system under consideration.
- For a three- or four-parameter system, it obtains from the first data card of the pair (Figure 6) the required data.
- From the second data card (in 3rd of 4th space) or the single data card (with only one or two parameters), it obtains the boundaries of parametric space to be examined and sets up grid points in this space. In a one-parameter system or for parameter 3 of a three-parameter system, this means a one-dimensional grid of 26 points; otherwise there is a two-dimensional grid of 676 points.
- At each grid point, it evaluates the characteristic equation as a function of s only; that is, it inserts the proper values for the parameters and calculates $Q(s) = Q_1(s) + \alpha_j Q_2(s) + \alpha_2 Q_3(s) + \alpha_1 \alpha_2 Q_4(s) + \ldots$.
- It calls the ROUTH subroutine, which examines the stability of the system at this point in the parametric space.
- It prepares the plot of the stability behavior and writes it on Tape 6 for later printing.
- For a system of three or four parameters, this entire process can be repeated any number of times by adding two data cards for each such plot (Figure 6).

STAB calls the HOLLER and ROUTH subroutines.

4.2.14 Subroutine HOLLER

This subroutine was obtained from E. W. Klingenberg of General Electric Defense Systems Department, Santa Barbara, California. It provides a means of getting Hollerith information into the machine core without requiring it to be read from data cards. This Hollerith information is then available for printing or writing on tape with variable captions, that is, one format statement can be used to write several different items, the exact terms written depending on the results of machine operation.

HOLLER is another vestigial remnant. In later subroutines, Hollerith information was provided directly in FORTRAN coding without recourse to a subroutine.

4.2.15 Subroutine ROUTH

Given the coefficients of a polynomial of order NMAX 1, this subroutine examines the polynomial to determine whether the system is stable or not. The procedure used is the method of Routh(22), which determines whether all roots of the polynomial are to the left of the imaginary axis in the complex plane. The subroutine returns a key number, NCRIT, to the calling program to describe the system. When all roots are to the left of the imaginary axis (that is, have negative real parts), the system is stable and NCRIT is set equal to 1. For a single root at the origin (real and imaginary parts both equal to zero) or a pair of pure complex roots, the system is on the stability threshold and NCRIT is equal to 2. For multiple roots at the origin, or roots with positive real parts (to the right of the imaginary axis), the system is unstable and NCRIT is set equal to 3.

4.2.16 Subroutine WTAP 2

This writes scratch data on Tape 3 to preserve it from LASS 2 to LASS 3.

4.3 Subroutines for LASS 3

4.3.1 Flowsheet for LASS 3

The flowsheet for LASS 3 is shown in Figure 33. The following subroutines are called:

Subroutines in LASS 3 but not in LASS 1 or 2

RTAP 2
RENUM
MESS 2 (used POLYX and DETEV)
PUNNU
PRITRA (calls ALPH)
FREQ (calls Michigan Plot Subroutines)
ROOTER (calls ALPH, EVAL, and MULLER)
TRANS

Subroutines in LASS 3 and in LASS 1 or 2

EFM
ERR
LTAP 1

4.3.2 Subroutine RTAP 2

This reads in from Tape 3 the scratch data that had been written at the end of LASS 2.

4.3.3 Subroutine RENUM

This reads in coefficients of the numerator of the transfer function for variable $X_{K(I)}$ (see flowsheet for LASS 3, Figure 34) when KEY 4 is not used to provide these elements and KEY 5 and/or KEY 6 are to be processed.

4.3.4 Subroutine MESS 2

This is the basic subroutine for KEY 4, which obtains the numerators of the desired transfer functions. This program is very similar to MESS 1 (Section 4.2.8) with the following differences:

- At this stage of the program, the calculations are being carried out for specific values of the parameters, hence the numerators can no longer be considered as functions of α_p, as they were written earlier. Instead they may may be written as

$$\sum_{q=1}^{NVAR} P_q(s)\overline{Y}_q(s)$$

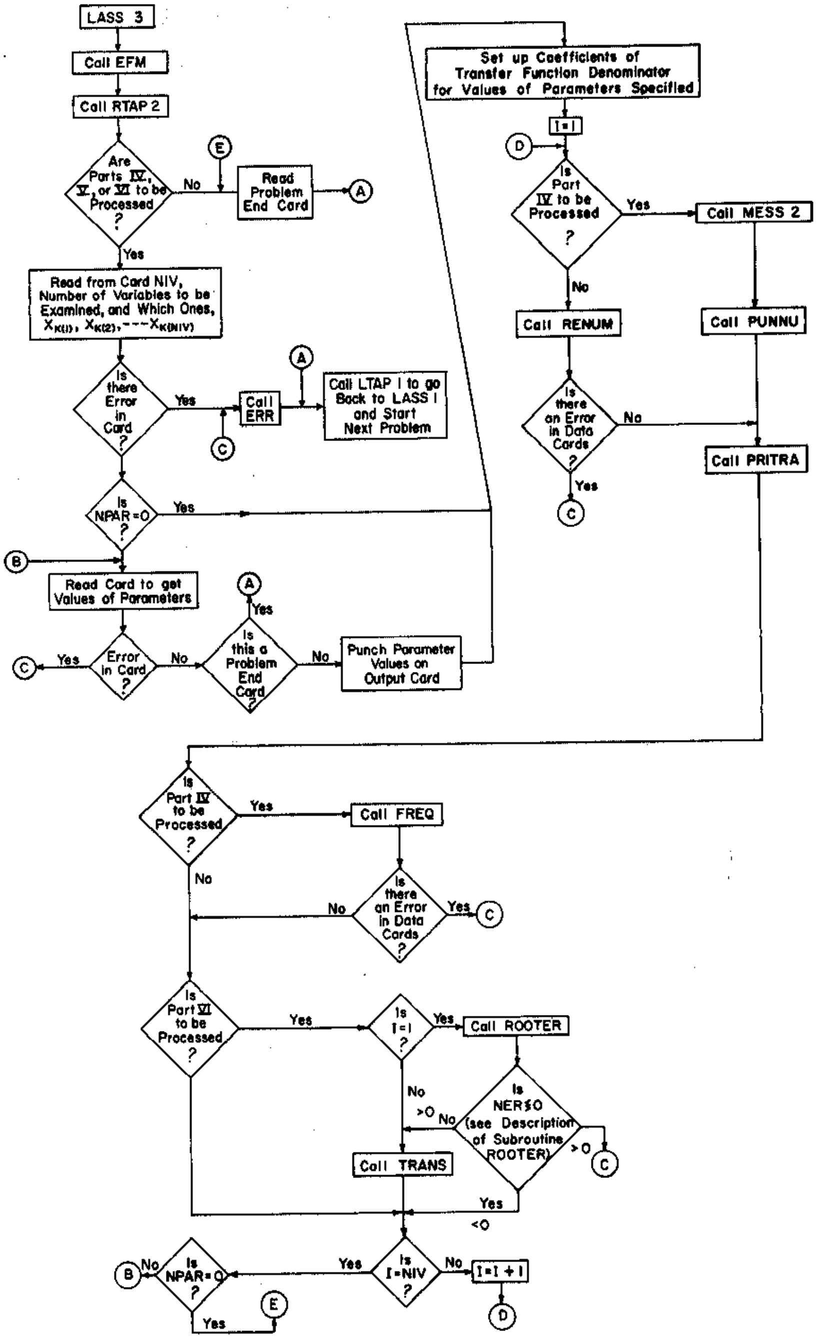


FIG. 34 FLOWSHEET FOR LASS 3

- MESS 2, then, first combines each pair of columns in the G matrix by evaluating $a_{1,2p-1,r} + \alpha_p b_{1,2p,r}$ and placing these terms in the NC(p) column of the A matrix.

- The next step is to replace column K(I) (the K(I) variable is under consideration) of the A matrix by column 1 of the V matrix and call POLYX (Section 4.2.9) to calculate the coefficients of $P_1(s)$. Repeat for column 2 of the V matrix to get $P_2(s)$ and continue up to $P_{NVAR}(s)$.

4.3.5 Subroutine PUNNU

This punches out on cards the coefficients of the transfer function numerator, for each variable examined, after the coefficients have been calculated by MESS 2. This subroutine follows very closely the procedure and logic used in PUNMAT (Section 4.2.5).

4.3.6 Subroutine PRITRA

This writes out on Tape 6 the coefficients of the transfer function numerator and denominator for each variable examined. As with PRIMAT and PRICHA, this is done whether coefficients are calculated by KEY 4 or read in from cards. This program calls subroutine ALPH.

4.3.7 Subroutine ALPH

This is a special program which writes out on tape a single line giving the values of the NPAR parameters. Since this same line is required in several subroutines of LASS 3, it was decided to call on one subroutine each time, thereby saving core storage.

4.3.8 Subroutine FREQ

This is the basic subroutine for KEY 5, the frequency response analysis. As background for this, we give the following brief review of the method.

Given the transfer function of a dynamic system as a function of the variable s, that is, T(s), then the steady-state response of the system to a sinusoidal input of frequency ω is the complex quantity

$$T(j\omega) = X(\omega) + jY(\omega) = A(\omega)e^{j\theta(\omega)}$$

where

$j = \sqrt{-1}$

$A(\omega)$ = ratio of the amplitudes of the system output to sinusoidal input

$\theta(\omega)$ = the phase angle between output and input

$X(\omega)$ and $Y(\omega)$ = coordinates of the cartesian representation of $T(j\omega)$

The principal function of FREQ, then is to perform the complex arithmetic to evaluate $T(j\omega)$ for the required values of ω. In addition there are some secondary functions. One such function is to read in from data cards (Figure 8) the frequency values to be used.

Here, as in MESS 2, the calculations are for one point in the parametric space. Furthermore, at each such point several system variables may be under examination. It is only necessary to read these frequencies once, since it is assumed that all system variables are to be examined at the same frequencies. For each point in the parametric space, then, there is a counter wheel set up to count the number of system variables examined. The first time through FREQ the counter, set at 1, will call for the reading of the frequency data. In subsequent passes through FREQ, this part is bypassed.

FREQ then calculates $T(j\omega)$ for each value of ω specified and obtains the results in terms of $X(\omega)$, $Y(\omega)$, $A(\omega)$, and $\theta(\omega)$. The amplitude ratio, $A(\omega)$, is calculated both as magnitude and in decibels ($20 \times \log_{10}$ magnitude), the phase angle, $\theta(\omega)$, is restricted to the region 0° to 360°. These calculations are carried out for each excitation in a multiple input system ($NVAR > 1$). When the calculations are completed, the program writes out the results on tape for printing in tabular form. Finally, FREQ sets up scale factors and coordinates preparatory to plotting the frequency response data. The plotting itself is done by the Michigan Plotting Subroutines(23), but FREQ must set up proper input data for these subroutines. Three plots are prepared:

- Log-amplitude of response versus frequency (semilogarithmic plot).
- Phase angle versus frequency (semilogarithmic plot). (These two semilogarithmic plots make up the so-called Bode plot.)

- Polar plot of $Y(\omega)$ versus $X(\omega)$ on linear scale (Nyquist plot).

FREQ calls ALPH, and the package making up the Michigan Plotting Subroutines (PLOT1, PLOT2, PLOT3, OMIT, FPLOT4).

4.3.9 Michigan Plotting Subroutines

Subroutines PLOT1, PLOT2, PLOT3, PLOT4, OMIT, FPLOT4, and PLTAPE make up the complete package of the University of Michigan Plotting Subroutine(23). This package is used as received from SHARE with no changes.

4.3.10 Subroutine ROOTER

This section of KEY 6 finds the positions of the poles of the transfer functions of the system variables for later use in obtaining the transient responses of these variables by performing the inverse transformation. The poles of the transfer functions are the roots of the denominator of the transfer function; since this polynomial is the same for all variables of the system, ROOTER need be called only once for each point in parametric space. Therefore, the same counter that was used in subroutine FREQ in determining whether input data cards are required is used here. ROOTER performs the following functions:

- Reads in data card (bottom of Figure 8) giving the time increment for transient response calculations and the magnitudes of the excitations. This, also, is required only once. As with all data cards, this card is tested for errors.

- Calls upon MULLER to calculate the roots of the transfer function denominator.

- Arranges the roots in ascending order, with the real roots placed ahead of the complex roots. At the same time, all roots are examined to see whether the real or imaginary parts are very small. The transient response is calculated over a time interval equal to 100 times the time increment, T. If the root is written as $\sigma+j\omega$ and either $100T\sigma < 0.001$ or $100T\omega < 0.001$ then σ or ω is set equal to 0, since there will be essentially no change in response due to this term over the time interval considered.

- Writes the results on output Tape 6.

- Does some testing of the roots to see that they appear correct (we have been unable to develop a root finder program that is absolutely 100% effective). The following tests are made:

  (a) Must be even number of complex roots.

  (b) Complex roots must appear as conjugate pairs.

  (c) Real roots not at the origin can have multiplicities no greater than two. This is really an arbitrary test, for there is no reason why inverse transforms with multiple real poles of multiplicity greater than two cannot be obtained. However, it was decided to apply this limitation, chiefly because MULLER gives very poor results for multiple roots.

  If any of these tests fail, the appropriate statement is put on the output tape. At the same time, failing these tests implies that there is no need to continue with KEY 6; so NER is set equal to -5 and the subroutine returns to LASS 3 which interprets a minus value for NER as instructions to bypass TRANS. ERR is not called and the calculation is continued.

  (d) Reconstitutes the polynomial function by multiplying out the roots obtained. That is, if the polynomial is $C_o + C_1 s + \ldots C_m s^m = Q(s)$ and the roots are r_1, r_2, ... r_m then $Q(s) = (s-r_1)(s-r_2) \ldots (s-r_m)$ and multiplying out these factors should give the coefficients C_j. In this test, the original and the reconstituted coefficients are put on the output tape to be printed out for visual comparison. There are no error stops and no diagnostics printed. If the coefficients agree to more than 4 or 5 digits, then it may be considered that MULLER has worked satisfactorily.

  (e) Evaluates derivatives of $Q(s)$ with respect to s at each root, so that the coefficients for the inverse transformation can be calculated in TRANS by partial fraction expansion.

ROOTER calls subroutines ALPH (Section 4.3.7), EVAL and MULLER.

4.3.11 Subroutine EVAL

This subroutine finds the value of a polynomial in s at a given value of s (real or complex).

4.3.12 Subroutine MULLER

This is a Polynomial Root Finder Subroutine from the SHARE Library(24). Originally written in FORTRAN for the IBM-7090, MULLER was readily adapted for the 704 machine. However, it was found that the original coding sometimes gave poor results. Therefore, some changes were made to improve the procedure. The modified MULLER has given consistently better results than several other root finder programs that have been tried. However, it is not completely foolproof; hence, the error tests in ROOTER.

4.3.13 Subroutine TRANS

This is the basic program for obtaining the transient response of each variable by inverse transformation of the transfer function times the transform of the excitation. The procedure used is the standard expansion about the singularities of the transfer function(25). This method was used in preference to integration in the complex plane along a line parallel to the imaginary axis primarily because it is immediately operable for all systems regardless of the positions of the singularities, hence can be used for obtaining the response of unstable systems.

The response to any one of three different types of excitation can be calculated: (1) an impulse, i.e. a delta function at time zero; (2) a step function at time zero; and (3) a ramp function, that is, a linearly increasing excitation starting at zero.

Which calculations are to be performed are defined by the input data card that was read in by ROOTER (bottom of Figure and data sheet 8). This card provides room for three excitation magnitudes. A non-zero value for the first one causes calculation of the response to an impulse; the second term sets up for the step function response, while the third provides for the ramp response. Conversely, a zero in any term will bypass the corresponding calculation.

Some further words about the impulse excitation are required. Strictly speaking, an impulse excitation is not physically realizable (certainly less so than a true step function); however, the calculated response to an impulse is useful in many cases, for it provides a good indication of the manner in which the system will recover from extraneous input spikes. On the other hand, for certain types of systems the response calculated by LASS will not be correct. These systems are those in which the numerator and denominator of the transfer function are the same order (this is frequently the case in the equations for nuclear reactors, where the neutron thermal lifetime is neglected relative to the time scale considered). For such a system, the response to an impulse is also an impulse at zero time plus other transient terms and the program does not give this impulse term.

TRANS first sets up 101 values of time from 0 to 100T (T is the time increment read in by ROOTER) at which the response is to be calculated.

For each type of excitation, the transfer function is then multiplied by the proper factor (magnitude alone for impulse, magnitude times 1/s for step function, and magnitude times $1/s^2$ for ramp function).

The inverse transform is then calculated by taking the partial fraction expansion of the transfer function about the poles.

$$F(s) = \sum_{m=1}^{NZER} \frac{A_m}{s^m} + \sum_{i=1}^{NRS} \frac{B_i}{s - \alpha_i} + \sum_{j=1}^{\frac{NRD}{2}} \left[\frac{C_j}{s - \beta_j} + \frac{D_j}{(s - \beta_j)^2}\right]$$

$$+ \sum_{k=1}^{\frac{NPI}{2}} \left[\frac{E_k e^{j\theta_k}}{s - j(\omega)_k} + \frac{E_k e^{-j\theta_k}}{s + j\omega_k}\right]$$

$$+ \sum_{\ell=1}^{\frac{NCS}{2}} \left[\frac{F_\ell^{j\phi_\ell}}{s - \gamma_\ell - j\eta_\ell} + \frac{F_\ell e^{-j\phi_\ell}}{s - \gamma_\ell + j\eta_\ell}\right]$$

where

NZER = multiplicity of roots at origin

NRS = number of single real roots not at origin

α_i = single real roots

NRD = number of double real roots not at origin

β_j = double real roots

NPI = number of pure imaginary roots

$j\omega_k$ = pure imaginary roots

NCS = number of complex roots

$\gamma_\ell \pm j\eta_\ell$ = the complex roots

Then, the transient response is given by

$$f(t) = \sum_{m=1}^{NZER} A_m \frac{t^{m-1}}{(m-1)!} + \sum_{i=1}^{NRS} \beta_i e^{\alpha_i t}$$

$$+ \sum_{j=1}^{\frac{NRD}{2}} (C_j + D_j t)e^{\beta_j t} + \sum_{k=1}^{\frac{NPI}{2}} E_k \sin(\omega_k t + \theta_k + 90^\circ)$$

$$+ \sum_{\ell=1}^{\frac{NCS}{2}} F_\ell e^{\gamma_\ell t} \sin(\eta_\ell t + \Phi_\ell + 90^\circ)$$

TRANS calculates the values of the coefficients (A_m, B_i, C_j, D_j, E_k, and F_ℓ) and the phase angles (Φ_k and Φ_ℓ). After these quantities have been evaluated, the magnitude of f(t) is obtained at each of the 101 values of time previously set up. Please note that the calculated value at t = 0 very often appears to be in error. For example, many times the correct value at zero time is zero, but the program by adding and subtracting terms may give a non-zero magnitude. When this happens, the result is several orders of magnitude less than at non-zero times and obviously differs from zero only because of round-off error.

After the response has been calculated at each time, the program then puts the data on the output tape for printing in tabular form. After this, TRANS sets up the data for plotting. As in FREQ, TRANS sets up scale factors and coordinates, then calls the Michigan Plotting Subroutine to make a single linear plot.

TRANS calls EVAL, ALPH, PLOT1, PLOT2, PLOT3, and FPLOT4 in addition to some of the standard library programs.

85%

APPENDIX I FORTRAN Listings

```
C                              LASS
C                 LINEAR ANALYSIS OF SYSTEM STABILITY
C     AUTHOR - S.M. KATZ    DATE - 1963
C
C                                LASS 1
C
C     LASS 1 CALLS THE FOLLOWING SUBROUTINES
C
C          EFM
C          PUN
C          TESKEY
C          CALMAT
C          ERR
C          WTAP1
C          LTAP2
C
C
      DIMENSION A(50,50,5),B(50,50),C(201),K(50),NK(50),G(50,8,5),
     1CHAR(16,50),NC(4),MA(8),NT(8),KEY(6),V(50,4),NL(16)
     2, CAN(4,50),D(2,50),E(200),R(50),RL(50),RU(50),S(50),JS1(50)
C
      COMMON A,B,C,K,NK,G,CHAR,NC,MA,NT,V
     1,CAN,D,E,R,RL,RU,S,JS1
C
    1 FORMAT(1H120X35HLINEAR ANALYSIS OF SYSTEM STABILITY10X4HPAGEI2///)
    2 FORMAT(72H
     1                 )
    3 FORMAT(5X6I1,2XI2,5XI1,5XI1)
    4 FORMAT(21H2OBTAIN THE FOLLOWING)
    5 FORMAT(1HOI5,27H. CALCULATE MATRIX ELEMENTS)
    6 FORMAT(1HOI5,46H. DENOMINATOR OF TRANSFER FUNCTION(PARAMETRIC))
    7 FORMAT(1HOI5,18H. STABILITY LIMITS)
    8 FORMAT(1HOI5,32H. NUMERATOR OF TRANSFER FUNCTION)
    9 FORMAT(1HOI5,20H. FREQUENCY RESPONSE)
   10 FORMAT(1HOI5,20H. TRANSIENT RESPONSE)
   11 FORMAT(////12H TO OPERATOR,20X,36HALL LASS PROBLEMS HAVE BEEN FINI
     1SHED/80H TAPE 6 HAS BEEN END FILED AND REWOUND. PLEASE UNLOAD TAPE
     2 6 AND PRINT OFF-LINE./48H CHECK FOR CARDS IN PUNCH AND REMOVE IF
     3PRESENT./////)
C
C     READ KEY AND TITLE CARDS, WRITE TITLE PAGE, TEST KEY WORDS
C
      CALL EFM
      REWIND 2
      REWIND 3
      CALL PUN
  100 READ 3,(KEY(I),I=1,6),N,NPAR,NVAR
      IF(N) 105,105,110
  105 END FILE 6
      REWIND 6
      PRINT 11
      STOP
  110 IPAGE=1
      WRITE OUTPUT TAPE 6,1, IPAGE
      PRINT 1, IPAGE
      READ 2
      WRITE OUTPUT TAPE 6,2
      READ 2
      WRITE OUTPUT TAPE 6,2
      PRINT 2
```

```
      READ 2
      WRITE OUTPUT TAPE 6,2
      CALL TESKEY(KEY,N,NPAR,NVAR,NER)
      IF(NER)180,120,180
  120 KEY1=KEY(1)+1
      KEY2=KEY(2)+1
      KEY3=KEY(3)+1
      KEY4=KEY(4)+1
      KEY5=KEY(5)+1
      KEY6=KEY(6)+1
      J=1
      WRITE OUTPUT TAPE 6,4
      GO TO (122,121),KEY1
  121 WRITE OUTPUT TAPE 6,5,J
      J=J+1
  122 GO TO (124,123),KEY2
  123 WRITE OUTPUT TAPE 6,6,J
      J=J+1
  124 GO TO (126,125),KEY3
  125 WRITE OUTPUT TAPE 6,7,J
      J=J+1
  126 GO TO (128,127),KEY 4
  127 WRITE OUTPUT TAPE 6,8,J
      J=J+1
  128 GO TO (130,129),KEY 5
  129 WRITE OUTPUT TAPE 6,9,J
      J=J+1
  130 GO TO (132,131),KEY 6
  131 WRITE OUTPUT TAPE 6,10,J
  132 IPAGE=IPAGE+1
C
C     IF A,G,V ARRAYS TO BE USED, CLEAR
C
      IF(KEY(1)+KEY(2)+KEY(4))150,150,136
  136 DO 145 I=1,N
      DO 140 J=1,N
      DO 140 M=1,5
  140 A(I,J,M)=0.0
      DO 142 L=1,4
  142 V(I,L)=0.0
      DO 145 L=1,8
      DO 145 M=1,5
  145 G(I,L,M)=0.0
C
C     IF CHAR ARRAY TO BE USED, CLEAR
C
  150 IF(KEY(2)+KEY(3)+KEY(5)+KEY(6))160,160,156
  156 L=2**NPAR
      DO 158 I=1,L
      DO 158 M=1,50
  158 CHAR(I,M)=0.0
C
C     KEY 1 BRANCH
C
  160 GO TO (182,170),KEY1
  170 CALL CALMAT (N,NPAR,NVAR,NSH,IPAGE,NER)
      IF (NER)180,182,180
  180 CALL ERR (NER)
      GO TO 100
  182 REWIND 3
```

```
      CALL WTAP1 (IPAGE,N,NPAK,NVAR,NER,KEY,NSH)
      REWIND 3
      REWIND 2
      CALL LTAP 2
      END(2,0,1,0,0)
```

```
C     EFM - SEE IBM REFERENCE MANUAL 704 DATA PROCESSING SYSTEM,
C               IBM FORM A22-6500-3, PAGE 23.
```

```
C     SUBROUTINE PUN      SAP CODED
       REL
       ORG 0
 ENT   SXD ST,1
       LXA A,1
       WPU
       CPY ST,1
       TIX *-1,1,1
       LXD ST,1
       TRA 1,4
 TAB   OCT 0,0,0,0,177770740017,77776077776
       OCT 177770377776,77777077777,170000177774
       OCT 17000017,170000074170,37777037777,170000036360
       OCT 77776077776,170000017740,74000074000,170000007700
       OCT 77777077777,170000003600,37777037777
 ST    PZE 0
 A     DEC 20
       END 0
```

```
C     TESKEY TO TEST VALIDITY OF KEY CARD
C
      SUBROUTINE TESKEY (KEY,N,NPAR,NVAR,NER)
      DIMENSION KEY(6)
      NER=0
      DO 10 I=1,6
      IF(KEY(I))101,10,5
    5 IF (KEY(I)-1)10,10,101
   10 CONTINUE
   15 IF (N-50)20,20,102
   20 IF (NPAR)103,30,25
   25 IF (NPAR-4)30,30,103
   30 IF(NVAR)104,40,35
   35 IF(NVAR-4)40,40,104
   40 IF(KEY(1))101,50,42
   42 J=0
      DO 45 I=2,6
   45 J=J+KEY(I)
      IF(J)101,50,46
   46 IF(KEY(2)+KEY(4))101,101,50
```

```
   50 IF (KEY(3))101,54,52
   52 IF (NPAR)103,105,54
   54 IF(KEY(4))101,58,56
   56 IF (NVAR)104,106,58
   58 RETURN
  101 NER=1
      GO TO 125
  102 NER=2
      GO TO 125
  103 NER=3
      GO TO 125
  104 NER=4
      GO TO 125
  105 NER=5
      GO TO 125
  106 NER=6
  125 RETURN
      END(2,2,1,2,0)
```

```
C     CALMAT, PERMANENT DUMMY
C
      SUBROUTINE CALMAT(N,NPAR,NVAR,NSH,IPAGE,NER)
      NER=7
      RETURN
      END(2,2,1,2,0)
```

```
C     ERR TO DESCRIBE NATURE OF ERROR IN PROBLEM         KATZ
C     THIS SUBROUTINE IS COMMON TO LASS 1, 2, AND 3
C
      SUBROUTINE ERR(NER)
      GO TO (51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69),
     1NER
    1 FORMAT (1H020X11HINVALID KEY)
    2 FORMAT (1H020X19HINVALID MATRIX SIZE)
    3 FORMAT (1H020X28HINVALID NUMBER OF PARAMETERS)
    4 FORMAT (1H020X27HINVALID NUMBER OF VARIABLES)
    5 FORMAT (1H020X26HKEY3 AND NPAR INCOMPATIBLE)
    6 FORMAT (1H020X26HKEY4 AND NVAR INCOMPATIBLE)
    7 FORMAT (1H020X33HSPECIAL CALMAT SUBROUTINE MISSING)
    8 FORMAT (1H020X20HERROR IN A,G,V CARDS)
    9 FORMAT (1H020X38HERROR IN CHARACTERISTIC EQUATION CARDS)
   10 FORMAT (1H020X42HERROR IN DATA CARDS FOR STABILITY ANALYSIS)
   11 FORMAT(1H020X48HERROR IN SPECIAL CALMAT SUBROUTINE, PLEASE CHECK)
   12 FORMAT(1H020X48HERROR IN DATA FOR NUMERATOR OF TRANSFER FUNCTION)
   13 FORMAT (1H020X48HPOWER OF CHARACTERISTIC EQUATION GREATER THAN 50)
   15 FORMAT(1H0,20X,39HIMPROPER NUMERATOR OF TRANSFER FUNCTION)
   16 FORMAT(1H0,20X,22HNO END-OF-PROBLEM CARD)
   17 FORMAT(1H020X,36HERROR IN DATA FOR FREQUENCY RESPONSE)
   18 FORMAT(1H020X,60HERROR IN FREQUENCY RESPONSE DATA (OVER 100 FREQUE
     1NCY VALUES))
   19 FORMAT(1H020X,32HERROR IN TRANSIENT RESPONSE DATA)
   40 FORMAT (I1)
```

```
 51 PRINT 1
    WRITE OUTPUT TAPE 6,1
    GO TO 100
 52 PRINT 2
    WRITE OUTPUT TAPE 6,2
    GO TO 100
 53 PRINT 3
    WRITE OUTPUT TAPE 6,3
    GO TO 100
 54 PRINT 4
    WRITE OUTPUT TAPE 6,4
    GO TO 100
 55 PRINT 5
    WRITE OUTPUT TAPE 6,5
    GO TO 100
 56 PRINT 6
    WRITE OUTPUT TAPE 6,6
    GO TO 100
 57 PRINT 7
    WRITE OUTPUT TAPE 6,7
    GO TO 100
 58 PRINT 8
    WRITE OUTPUT TAPE 6,8
    GO TO 100
 59 PRINT 9
    WRITE OUTPUT TAPE 6,9
    GO TO 100
 60 PRINT 10
    WRITE OUTPUT TAPE 6,10
    GO TO 100
 61 PRINT 11
    WRITE OUTPUT TAPE 6,11
    GO TO 100
 62 PRINT 12
    WRITE OUTPUT TAPE 6,12
    GO TO 100
 63 PRINT 13
    WRITE OUTPUT TAPE 6,13
    GO TO 100
 64 GO TO 62
 65 PRINT 15
    WRITE OUTPUT TAPE 6,15
    GO TO 100
 66 PRINT 16
    WRITE OUTPUT TAPE 6,16
    GO TO 100
 67 PRINT 17
    WRITE OUTPUT TAPE 6,17
    GO TO 100
 68 PRINT 18
    WRITE OUTPUT TAPE 6,18
    GO TO 100
 69 PRINT 19
    WRITE OUTPUT TAPE 6,19
100 READ 40, I
    IF (I-9)100,110,100
110 RETURN
    END(2,0,1,0,0)
```

```
C      WTAP 1, WRITE LASS1 DATA ON TAPE 3
C
       SUBROUTINE WTAP1(IPAGE,N,NPAR,NVAR,NER,KEY,NSH)
       DIMENSION A(19071),KEY(6)
       COMMON A
       WRITE TAPE 3,A,IPAGE,N,NPAR,NVAR,NER,KEY,NSH
       RETURN
       END(2,2,1,2,0)
```

```
C      SUBROUTINE LTAP2  LASS     CODED IN SAP
        REM LOAD PROGRAM FROM TAPE 2
        REM PROGRAM TO INITIATE TAPE READ
        REM
        ORG 0
        REL
 LTAP2  RTB 2
        CPY 0
        CPY 1
        TRA 0
        END 0
```

```
C                                     LASS 2
C
C     LASS 2 CALLS THE FOLLOWING SUBROUTINES
C          RTAP1
C          REMAT
C          ERR
C          LTAP1
C          PUNMAT
C          PRIMAT
C          RECHAR
C          MESSI
C          POLYX
C          DETEV
C          PUNCHA
C          PRICHA
C          STAB
C          HOLLER
C          ROUTH
C          WTAP2
C          LTAP2
C
C
C
      DIMENSION A(50,50,5),B(50,50),C(201),K(50),NK(50),G(50,8,5),
     1CHAR(16,50),NC(4),MA(8),NT(8),KEY(6),V(50,4),NL(16),
     2E(50)
C
      COMMON A,B,C,K,NK,G,CHAR,NC,MA,NT,V,E
C
      CALL EFM
      REWIND 1
      REWIND 3
      CALL RTAP1 (IPAGE,N,NPAR,NVAR,NER,KEY,NSH)
      REWIND 3
      IF(KEY(1)) 162,162,182
  162 IF(KEY(2)+KEY(4))165,190,165
  165 CALL REMAT (N,NPAR,NVAR,NSH,NER)
      IF (NER)180,185,180
  180 CALL ERR(NER)
      REWIND 1
      CALL LTAP1
  182 CALL PUNMAT(N,NPAR,NVAR,NSH)
  185 CALL PRIMAT (N,NPAR,NVAR,IPAGE)
      IF(KEY(2))190,190,195
  190 IF(KEY(3)+KEY(4)+KEY(5)+KEY(6))220,220,192
  192 CALL RECHAR (CHAR,NL,NMAX,NSH,NER)
      IF(NER)180,200,180
  195 CALL MESSI(N,NPAR,NSH,NL,NMAX)
      IF(NMAX-50)198,198,196
  196 NER=13
      GO TO 180
  198 CALL PUNCHA (CHAR,NL,NPAR,NSH,NMAX)
  200 CALL PRICHA(CHAR,NL,NPAR,NMAX,IPAGE)
      IF(KEY(3))220,220,210
  210 CALL STAB(NPAR,NMAX,IPAGE,NER)
       IF (NER) 180,220,180
  220 REWIND 3
      CALL WTAP2(IPAGE,N,NPAR,NVAR,NER,KEY,NSH,NMAX,NL)
      REWIND 3
      CALL LTAP 2
```

```
      END (2,0,1,0,1)



C     RTAP1, LOAD LASS1 DATA IN CORE
C
      SUBROUTINE RTAP1 (IPAGE,N,NPAR,NVAR,NER,KEY,NSH)
      DIMENSION A(19071),KEY(6)
      COMMON A
      READ TAPE 3,A,IPAGE,N,NPAR,NVAR,NER,KEY,NSH
      RETURN
      END(2,2,1,2,0)




C     REMAT, READ ELEMENTS OF A,G,V ARRAYS
C
      SUBROUTINE REMAT (N,NPAR,NVAR,NSH,NER)
      DIMENSION  A(50,50,5),B(50,50),C(201),K(50),NK(50),G(50,8,5),
     1CHAR (16,50),NC(4),MA(8),NT(8),V(50,4)
      COMMON A,B,C,K,NK,G,CHAR,NC,MA,NT,V
    1 FORMAT (I1,I2,4X,I2,3X,50I1)
    2 FORMAT (I1,I2,3(2I2,I1,E14.8))
    3 FORMAT (I1,5I2,8I1,8I2)
      READ 1,L1,L2,NSH,(K(I),I=1,N)
      IF(L1-1)105,110,105
  105 NER=8
  106 RETURN
  110 IF (L2-1)105,111,105
  111 READ 2,L1,L2,I1,J1,K1,(A(I1,J1,K1)),I2,J2,K2,(A(I2,J2,K2)),I3,J3,
     1K3,(A(I3,J3,K3))
      IF(L1-1)105,112,105
  112 IF(L2-2)105,111,115
  115 IF(L2-3)125,125,105
  125 IF(NPAR)130,150,130
  130 READ 3,L1,L2,(NC(I),I=1,4),(MA(J),J=1,8),(NT(M),M=1,8)
      IF(L1-1)105,132,105
  132 IF(L2-4)105,140,105
  140 READ 2,L1,L2,I1,J1,K1,(G(I1,J1,K1)),I2,J2,K2,(G(I2,J2,K2)),I3,J3,
     1K3,(G(I3,J3,K3))
      IF(L1-1)105,141,105
  141 IF(L2-5)105,140,142
  142 IF(L2-6)150,150,105
  150 IF(NVAR)152,165,152
  152 READ 2,L1,L2,I1,J1,K1,(V(I1,J1)),I2,J2,K2,(V(I2,J2)),I3,J3,K3,
     1(V(I3,J3))
      IF(L1-1)105,153,105
  153 IF(L2-7)105,152,155
  155 IF(L2-8)106,106,105
  165 READ 3,L1,L2
      IF(L1-1)105,166,105
  166 IF(L2-8)105,106,105
      END(2,2,1,2,0)
```

```
C      SUBROUTINE LTAP1  LASS      CODED IN SAP
        REM
        ORG 0
        REL
 LTAP1  RTB 1
        CPY 0
        CPY 1
        TRA 0
        END 0
```

```
C      PUNMAT,PUNCH ELEMENTS OF A,G,V ARRAYS                           KATZ
C
       SUBROUTINE PUNMAT (N,NPAR,NVAR,NSH)
       DIMENSION A(50,50,5),B(50,50),C(201),K(50),NK(50),G(50,8,5),
      1CHAR (16,50),NC(4),MA(8),NT(8),V(50,4)
       COMMON A,B,C,K,NK,G,CHAR,NC,MA,NT,V
     1 FORMAT (I1,I2,4X,I2,3X,50I1)
     2 FORMAT (I1,I2,3(2I2,I1,1PE14.7))
     3 FORMAT (I1,5I2,8I1,8I2)
       L1=1
       L2=1
       PUNCH 1,L1,L2,NSH,(K(J),J=1,N)
       L2=2
       II=7
       DO 320 I=1,9
       NK(I)=1
   320 C(I)=0.0
       NK(4)=2
       NK(7)=3
       DO 400 I=1,N
       DO 400 J=1,N
       MAX=K(J)
       DO 400 M=1,MAX
       IF(A(I,J,M))340,400,340
   340 NK(II)=I
       NK (II+1)=J
       NK (II+2)=M
       C(II)=A(I,J,M)
       IF(II-1)360,360,350
   350 II=II-3
       GO TO 400
   360 II=7
   370 PUNCH 2,L1,L2,(NK(JJ),JJ=1,3),C(1),(NK(MM),MM=4,6),C(4),
      1(NK(LL),LL=7,9),C(7)
   400 CONTINUE
       L2=3
   405 PUNCH 2,L1,L2,(NK(JJ),JJ=1,3),C(1),(NK(MM),MM=4,6),C(4),
      1(NK(LL),LL=7,9),C(7)
       IF (NPAR)410,510,410
   410 L2=4
       PUNCH 3,L1,L2,(NC(I),I=1,4),(MA(J),J=1,8),(NT(M),M=1,8)
       DO 415 I=1,9
       NK(I)=1
   415 C(I)=0.0
```

```
      NK(4)=2
      NK(7)=3
      NN=2*NPAR
      II=7
      L2=5
      DO 500 I=1,N
      DO 500 J=1,NN
      MAX=MA(J)
      DO 500 M=1,MAX
      IF(G(I,J,M))420,500,420
  420 NK(II)=I
      NK(II+1)=J
      NK(II+2)=M
      C(II)=G(I,J,M)
      IF(II-1)460,460,440
  440 II=II-3
      GO TO 500
  460 II=7
  470 PUNCH 2,L1,L2,(NK(JJ),JJ=1,3),C(1),(NK(MM),MM=4,6),C(4),
     1(NK(LL),LL=7,9),C(7)
  500 CONTINUE
      L2=6
  505 PUNCH 2,L1,L2,(NK(JJ),JJ=1,3),C(1),(NK(MM),MM=4,6),C(4),
     1(NK(LL),LL=7,9),C(7)
  510 IF (NVAR)515,610,515
  515 II=5
      DO 518 I=1,6
      NK(I)=1
  518 C(I)=0.0
      NK(3)=2
      NK(5)=3
      L2=7
      NK(7)=0
      DO 600 I=1,N
      DO 600 J=1,NVAR
      IF (V(I,J))520,600,520
  520 NK(II)=I
      NK(II+1)=J
      C(II)=V(I,J)
      IF(II-1)560,560,540
  540 II=II-2
      GO TO 600
  560 II=5
  570 PUNCH 2,L1,L2,NK(1),NK(2),NK(7),C(1),NK(3),NK(4),NK(7),C(3),NK(5),
     1NK(6),NK(7),C(5)
  600 CONTINUE
      L2=8
  605 PUNCH 2,L1,L2,NK(1),NK(2),NK(7),C(1),NK(3),NK(4),NK(7),C(3),NK(5),
     1NK(6),NK(7),C(5)
      RETURN
  610 L2=8
      PUNCH 3,L1,L2
      RETURN
      END(2,0,1,0,0)
```

```
C     PRIMAT, PRINT A,G,V ARRAYS
C
      SUBROUTINE PRIMAT (N,NPAR,NVAR,IPAGE)
      DIMENSION A(50,50,5),B(50,50),C(201),K(50),NK(50),G(50,8,5),
     1CHAR (16,50),NC(4),MA(8),NT(8),V(50,4)
      COMMON A,B,C,K,NK,G,CHAR, NC, MA, NT, V
    1 FORMAT(1H140X6HA(I,J,I1,10H) ELEMENTS8X4HPAGEI2///3X,6(10X,I2,7X))
    2 FORMAT (I3,6E19.8)
    3 FORMAT(1H140X6HG(I,J,I1,10H) ELEMENTS8X4HPAGEI2///3X,6(10X,I2,7X))
    4 FORMAT(1H142X15HV(I,J) ELEMENTS8X4HPAGEI2///3X,6(10X,I2,7X))
      MAX=1
      DO 90 I=1,N
   90 MAX=XMAXOF (MAX,K(I))
C
C     PRINT A ARRAY
C
      DO 200 L=1,MAX
  100 J=0
  110 M=1
  115 IF(J-N)120,130,130
  120 J=J+1
      IF(K(J)-L)115,150,150
  130 IF(M-1)200,200,140
  140 M=M-1
      GO TO 160
  150 NK(M)=J
      IF(M-6)155,160,160
  155 M=M+1
      GO TO 115
  160 WRITE OUTPUT TAPE 6,1,L,IPAGE,(NK(II),II=1,M)
      IPAGE=IPAGE+1
      DO 165 I=1,N
      DO 162 II=1,M
      JJ=NK(II)
  162 C(II)=A(I,JJ,L)
  165 WRITE OUTPUT TAPE 6,2,I,(C(II),II=1,M)
      GO TO 110
  200 CONTINUE
C
C     PRINT G ARRAY
C
      IF(NPAR)205,300,205
  205 NN=2*NPAR
      MAX=1
      DO 206 I=1,NN
  206 MAX=XMAXOF(MAX,MA(I))
      DO 295 L=1,MAX
      J=0
  210 M=1
  215 IF(J-NN)220,230,230
  220 J=J+1
      IF(MA(J)-L)215,250,250
  230 IF(M-1)295,295,240
  240 M=M-1
      GO TO 260
  250 NK(M)=J
      IF(M-6)255,260,260
  255 M=M+1
      GO TO 215
  260 WRITE OUTPUT TAPE 6,3,L, IPAGE,(NK(II),II=1,M)
```

```
      IPAGE=IPAGE+1
      DO 265 I=1,N
      DO 262 II=1,M
      JJ=NK(II)
  262 C(II)=G(I,JJ,L)
  265 WRITE OUTPUT TAPE 6,2,I,(C(II),II=1,M)
      GO TO 210
  295 CONTINUE
C
C     PRINT V ARRAY
C
  300 IF(NVAR)305,320,305
  305 WRITE OUTPUT TAPE 6,4,IPAGE,(J,J=1,NVAR)
      IPAGE=IPAGE+1
      DO 310 I=1,N
  310 WRITE OUTPUT TAPE 6,2,I,(V(I,J),J=1,NVAR)
  320 RETURN
      END(2,2,2,2,0)
```

```
C     RECHAR, READ COEFFICIENTS OF CHARACTERISTIC EQUATION
C
      SUBROUTINE RECHAR (CHAR,NL,NMAX,NSH,NER)
      DIMENSION CHAR(16,50),NL(16)
    1 FORMAT (I1,I2,4X,I2,5X,I2,3X,16I2)
    2 FORMAT (I1,I2,3(2I2,I1,E14.8))
      DO 10 I=1,16
   10 NL(I)=0
      READ 1,L1,L2,NSH,NMAX,(NL(I),I=1,16)
      IF(L1-2)105,110,105
  110 IF(L2-1)105,111,105
  105 NER=9
      RETURN
  111 READ 2,L1,L2,I1,J1,K1,(CHAR(I1,J1)),I2,J2,K2,(CHAR(I2,J2)),I3,J3,
     1K3,(CHAR(I3,J3))
      IF(L1-2)105,112,105
  112 IF(L2-2)105,111,115
  115 IF(L2-3)120,120,105
  120 RETURN
      END(2,2,1,2,0)
```

```
C     COLUMN  INTERCHANGE   IN  A  MATRIX TO  GET EFFCT  OF  PARAMETERS
C
C     FOLLOWING VARIABLES  NEED  TO  BE IN STORAGES
C
C     A(I,J,M)=MATRIX ELEMENTS FOR COLUMNS NOT CONTAINING  PARAMETERS
C
C     G(I,J,M)=MATRIX ELEMENTS FOR COLUMNS WITH PARAMETERS
C         J ODD IS NON  PARAMETRIC  PART
C         J EVEN  IS  PART  WITH PARAMETER
C
C     CHAR(I,J)=COEFFICIENTS OF  CHARACTERISTIC  EQUATION = 0.0
C
```

```
C     NPAR = NUMBER  OF PARAMETERS
C
C     NSH = OVERALL EXPONENT TO MULTIPLY  RESULT
C
C     NC(I) = COLUMN  CONTAINING  ITH  PARAMETER
C
C     MA(J) = NUMBER TERMS IN  J COLUMN  OF  G  MATRIX
C
C     K (J) = NUMBER TERMS IN  J COLUMN  OF  A  MATRIX
C
C     NT(J) = EXPONENT  FOR J COLUMN  OF  G  MATRIX
C
C
C     CALCULATED BY SUBROUTINE
C
C     CHAR (I,J) = COEFFICIENTS OF CHARACTERISTIC EQUATION
C
C     NL(I)=1+POWER OF ITH POLYNOMIAL
C
C
C
      SUBROUTINE MESSI(N,NPAR,NSH,NL,NMAX)
C
C
C
      DIMENSION A(50,50,5),B(50,50),C(201),K(50),NK(50),G(50,8,5),
     1CHAR(16,50),NC(4),MA(8),NT(8),KK(4),NL(16)
      COMMON  A,B,C,K,NK,G, CHAR,NC,MA,NT
      DO 2 I=1,16
    2 NL(I)=0
      IF(NPAR)95,95,5
    5 NMAX=0
      DO 10 I=1,NPAR
   10 KK(I)=1
   15 NS=NSH
      DO 50 I=1,NPAR
      NCOL=NC(I)
      NPOW=K(NCOL)
      DO 30 II=1,N
      DO 30 M=1,NPOW
   30 A(II,NCOL,M)=0.0
      J=2*(I-1)+KK(I)
      NS=NS+ NT(J)
      NPOW=MA(J)
      K(NCOL)=NPOW
      DO 40 II=1,N
      DO 40 M=1,NPOW
   40 A(II,NCOL,M)=G(II,J,M)
   50 CONTINUE
      CALL POLYX (N,L)
      IF (L)55,80,55
   55 ICH=1
      DO 60 I=1,NPAR
   60 ICH=ICH+(KK(I)-1)*2**(I-1)
      NL(ICH)=NS+L
      NMAX=XMAXOF(NMAX,NL(ICH))
      DO 70 I=1,L
      J=NS+I
   70 CHAR(ICH,J)=C(I)
   80 DO 90 I=1,NPAR
```

```
      IF(KK(I)-2)85,90,90
   85 KK(I)=2
      GO TO 15
   90 KK(I)=1
      RETURN
   95 CALL POLYX (N,L)
      NL(1)=NSH+L
      NMAX=NL(1)
      DO 100 I=1,L
      J=NSH+I
  100 CHAR (1,J)=C(I)
      RETURN
      END(2,2,1,2,0)
```

```
C     EXPAND MATRIX OF POLYNOMIAL ELEMENTS
C     SUBROUTINE  POLYX IS COMMON TO LASS 2 AND LASS 3
C
      SUBROUTINE POLYX (N,L)
C
      DIMENSION A(50,50,5),B(50,50),C(201),K(50),NK(50)
C
      COMMON A,B,C,K,NK
C
      DO15J=1,N
   15 NK(J)=1
      DO20L=1,201
   20 C(L)=0.0
   21 J=1
   25 M=NK(J)
      DO30I=1,N
   30 B(I,J)=A(I,J,M)
      IF(J-N)35,40,40
   35 J=J+1
      GO TO 25
   40 CONTINUE
      CALL DETEV (N,D)
   70 L=1-N
      DO50J=1,N
   50 L=L+NK(J)
      C(L)=C(L)+D
      DO60J=1,N
      IF(NK(J)-K(J))45,60,60
   45 NK(J)=NK(J)+1
      GO TO 21
   60 NK(J)=1
   61 IF(C(L))63,62,63
   62 L=L-1
      IF(L)61,63,61
   63 RETURN
      END (2,2,1,0,0)
```

```
C     DETEV, DIVIDES AND TESTS DIFFERENCE                          S.M.K.
```

```
C
      SUBROUTINE DETEV (N,D)
C
      DIMENSION A(50,50,5),B(50,50),C(201),KK(50),NK(50)
C
      COMMON A,B,C,KK,NK
C
      D=1.0
      NLESS=N-1
      DO45I=1,NLESS
      IPLUS=I+1
      IF(B(I,I))30,10,30
   10 DO15J=IPLUS,N
      IF(B(J,I))20,15,20
   15 CONTINUE
      D=0.0
      GO TO 48
   20 D=-D
      DO25K=I,N
      SWAP=B(I,K)
      B(I,K)=B(J,K)
   25 B(J,K)=SWAP
   30 D=D*B(I,I)
   32 DO45L=IPLUS,N
      IF(B(L,I))35,45,35
   35 FAC= B(L,I)/B(I,I)
      DO40M=I,N
      TEMP = B(L,M)
      B(L,M)=TEMP-FAC*B(I,M)
      IF(1.0E-6*ABSF(TEMP)-ABSF(B(L,M)))40,38,38
   38 B(L,M)=0.0
   40 CONTINUF
   45 CONTINUE
      D = D* B(N,N)
   48 CONTINUE
   60 RETURN
      END (2,1,1,0,0)
C     PUNCHA, PUNCH COEFFICIENTS OF CHARACTERISTIC EQUATION  KATZ
C
      SUBROUTINE PUNCHA(CHAR,NL,NPAR,NSH,NMAX)
      DIMENSION CHAR(16,50),I(3),J(3),C(3),NL(16)
    1 FORMAT (I1,I2,4X,I2,5X,I2,3X,16I2)
    2 FORMAT (I1,I2,3(2I2,I1,1PE14.7))
      M=2**NPAR
      L1=2
      L2=1
      PUNCH 1,L1,L2,NSH,NMAX,(NL(I),I=1,16)
      L2=2
      DO 10 KIL=1,3
      I(KIL)=1
      J(KIL)=1
   10 C(KIL)=0.0
      K=0
      NN=3
      DO 100 II=1,M
      LEN=NL(II)
      DO 100 JJ=1,LEN
      IF(CHAR(II,JJ))50,100,50
   50 I(NN)=II
      J(NN)=JJ
```

```
      C(NN)=CHAR(II,JJ)
      IF(NN-1)60,60,55
   55 NN=NN-1
      GO TO 100
   60 NN=3
      PUNCH 2,L1,L2,I(1),J(1),K,C(1),I(2),J(2),K,C(2),I(3),J(3),K,C(3)
  100 CONTINUE
      L2=3
      PUNCH 2,L1,L2,I(1),J(1),K,C(1),I(2),J(2),K,C(2),I(3),J(3),K,C(3)
      RETURN
      END(2,0,1,0,0)
```

```
C     PRICHA, PRINT CHARACTERISTIC EQUATION
C
      SUBROUTINE PRICHA(CHAR,NL,NPAR,NMAX,IPAGE)
      DIMENSION CHAR(16,50),NL(16)
    1 FORMAT (1H121X23HCHARACTERISTIC EQUATION21X4HPAGEI2///)
    2 FORMAT (15X14HNON PARAMETRIC//)
    3 FORMAT (15X14HNON PARAMETRIC16X5HPAR 1//)
    4 FORMAT (15X14HNON PARAMETRIC16X5HPAR 120X5HPAR 215X
     115HPAR 1  X  PAR 2//)
    5 FORMAT (10X4E25.8)
    6 FORMAT (6H   S**I2,2X4E25.8)
    7 FORMAT (1H165X4HPAGEI2///20X5HPAR 315X15HPAR 1  X  PAR 310X
     115HPAR 2  X  PAR 37X21HPAR 1 X PAR 2 X PAR 3//)
    8 FORMAT (1H165X4HPAGEI2///20X5HPAR 415X15HPAR 1  X  PAR 410X
     115HPAR 2  X  PAR 47X21HPAR 1 X PAR 2 X PAR 4//)
    9 FORMAT (1H165X4HPAGEI2///15X15HPAR 3  X  PAR 47X
     121HPAR 1 X PAR 3 X PAR 44X21HPAR 2 X PAR 3 X PAR 42X
     225HPAR1 X PAR2 X PAR3 X PAR4//)
      WRITE OUTPUT TAPE 6,1,IPAGE
      IPAGE=IPAGE+1
      IF(NPAR)100,100,110
  100 M=1
      WRITE OUTPUT TAPE 6,2
      GO TO 125
  110 IF(NPAR-1)115,115,120
  115 M=2
      WRITE OUTPUT TAPE 6,3
      GO TO 125
  120 M=4
      WRITE OUTPUT TAPE 6,4
  125 WRITE OUTPUT TAPE 6,5,(CHAR(I,1),I=1,M)
      L=NMAX-1
      DO 130 J=1,L
  130 WRITE OUTPUT TAPE 6,6, J,(CHAR(I,J+1),I=1,M)
      IF(NPAR-2)200,200,135
  135 WRITE OUTPUT TAPE 6,7,IPAGE
      IPAGE=IPAGE+1
      WRITE OUTPUT TAPE 6,5, (CHAR(I,1),I=5,8)
      DO 140 J=1,L
```

```
  140 WRITE OUTPUT TAPE 6,6,J,(CHAR(I,J+1),I=5,8)
      IF(NPAR-3)200,200,145
  145 WRITE OUTPUT TAPE 6,8, IPAGE
      IPAGE=IPAGE+1
      WRITE OUTPUT TAPE6,5,(CHAR(I,1),I=9,12)
      DO 150 J=1,L
  150 WRITE OUTPUT TAPE 6,6,J,(CHAR(I,J+1),I=9,12)
      WRITE OUTPUT TAPE 6,9,IPAGE
      IPAGE=IPAGE+1
      WRITE OUTPUT TAPE 6,5,(CHAR(I,1),I=13,16)
      DO 160 J=1,L
  160 WRITE OUTPUT TAPE 6,6,J,(CHAR(I,J+1),I=13,16)
  200 RETURN
      END(2,2,1,2,0)




C     STAB, REVISED VERSION OF SEPT., 1962                       S.M.K.
C
      SUBROUTINE STAB(NPAR,NMAX,IPAGE,NER)
C
      DIMENSION A(12500),CHAR(16,50),V(220),C(4,50),E(50),AB(26),BA(26),
     1NSTORE(26,26),KK(4),IN(3),NAME(9),B(50,50),COM(2301)
C
      COMMON A,B,COM,CHAR,V,E
C
    1 FORMAT(1H110X42HSTABILITY REGIONS FOR ONE PARAMETER SYSTEM13X
     14HPAGEI3//////)
    2 FORMAT(I1,I2,I1,4E14.8)
    3 FORMAT(1H1,16X,33HSTABILITY REGIONS FOR PARAMETER 3,16X,4HPAGE,I3
     1//25X,17HWITH PARAMETER 1=,1PE15.7//26X,16HAND PARAMETER 2=,1PE15.
     27//)
    4 FORMAT(34X9HPARAMETERI2,1H=1PE15.7//)
    5 FORMAT(1H1      8X      31HSTABILITY REGIONS FOR PARAMETERI2,
     114H AND PARAMETER,I2,8X,4HPAGE,I3//)
    6 FORMAT(40X28HRANGE EXAMINED FOR PARAMETERI2,3H IS1PE15.7,3H TO
     1E15.7///)
    7 FORMAT (1PE30.7,5X,2A6,E30.7,5X,2A6//)
    8 FORMAT(26X51H+---------+---------+---------+---------+---------+)
    9 FORMAT(16XF8.0,2H +25A2,A1,1H+)
   10 FORMAT(25X1HI25A2,A1,1HI)
   11 FORMAT(8X9HPARAMETER8X1HI25A2,A1,1HI)
   12 FORMAT(12XI1,12X1HI25A2,A1,1HI)
   13 FORMAT(10X5HTIMES10X1HI25A2,A1,1HI)
   14 FORMAT(9X,4H10**,I3,9X,1HI,25A2,A1,1HI)
   15 FORMAT(//18X,6F10.0)
   17 FORMAT(39X,9HPARAMETER,I2,11H TIMES 10**,I3)
   18 FORMAT(//20X2A6,27H OVER ENTIRE RANGE EXAMINED)
C
C     SET UP ALFANUMERIC WORDS IN NAME(I) ARRAY
C
      CALL HOLLER(8,48HSTABLETHRESHUNSTAB    OLD  LE   S    T    ,
     1NAME)
      NAME(9)=NAME(4)
C
C     CLEAR C,E,AB,BA,NSTORE ARRAYS
C
   50 DO 55 I=1,50
```

```
      DO 55 J=1,50
      E(I)=0.0
   55 C(I,J)=0.0
      DO 59 I=1,26
      DO 59 J=1,26
      AB(I)=0.0
      BA(I)=0.0
   59 NSTORE(I,J)=0
      IF(NPAR-2)61,61,75
C
C     1 AND 2 PARAMETER BRANCH, SET PARAMETERS 3 AND 4 TO ZERO, LF=5,
C
   61 X=0.0
      Y=0.0
      LF=5
   63 DO 64 I=1,4
   64 KK(I)=I
C
C     TRANSFER FROM CHAR TO C ARRAYS (REDUCE TO 2 PAR. SYSTEM) FOR PAIR1
C
      DO 70 J=1,NMAX
      DO 70 I=1,4
   70 C(I,J)=CHAR(I,J)+X*CHAR(I+4,J)+Y*CHAR(I+8,J)+X*Y*CHAR(I+12,J)
      IF(NPAR-1)72,72,100
C
C     WRITE PAGE TITLE FOR 1 PARAMETER SYSTEM
C
   72 WRITE OUTPUT TAPE 6,1,IPAGE
   73 NUM=1
      GO TO 105
C
C     3 AND 4 PARAMETER BRANCH, READ AND TEST DATA CARD
C
   75 READ 2,L1,LF,L,X,Y
      IF(L1-3)500,76,500
   76 IF(LF-4)500,78,77
   77 IF(LF-5)78,78,500
   78 IF(L-1)500,63,79
   79 IF(L-2)63,80,500
   80 KK(1)=3
      KK(2)=4
      KK(3)=1
      KK(4)=2
C
C     TRANSFER FROM CHAR TO C (REDUCE TO 2 PAR. SYSTEM) FOR PAIR 2
C
      DO 90 J=1,NMAX
      DO 90 I=1,4
      K=4*I-3
   90 C(I,J)=CHAR(K,J)+X*CHAR(K+1,J)+Y*CHAR(K+2,J)+X*Y* CHAR(K+3,J)
      IF(NPAR-3)92,92,100
C
C     WRITE PAGE TITLE FOR 3 PARAMETER SYSTEM (THIRD PAR.)
C
   92 WRITE OUTPUT TAPE 6,3,IPAGE,X,Y
      GO TO 73
  100 NUM=26
C
C     WRITE PAGE TITLE FOR 2,3, OR 4 PAR. SYSTEM
C
```

```
      WRITE OUTPUT TAPE 6,5,KK(1),KK(2),IPAGE
      IF (NPAR-3) 105,101,102
  101 WRITE OUTPUT TAPE 6,4,KK(3),X
      GO TO 105
  102 WRITE OUTPUT TAPE 6,4,KK(3),X,KK(4),Y
  105 NR=NMAX
C
C     IF COEFFICIENT OF 0 POWER TERM =0 SHIFT C ARRAY LEFT
C
  106 DO 107 I=1,4
      IF(C(I,1))115,107,115
  107 CONTINUE
      DO 110 I=1,4
      DO 110 J=1,NR
  110 C(I,J)=C(I,J+1)
      NR=NR-1
      GO TO 106
C
C     READ SECOND DATA CARD, SET UP VALUES OF CO-ORDINATES
C
  115 IPAGE=IPAGE+1
      READ 2,L1,L2,L,AB(1),AB(26),BA(1),BA(26)
      IF(L1-3)500,117,500
  117 IF(L2-2)500,118,500
  118 DIFF=(AB(26)-AB(1))/25.0
      DO 120 I=2,25
      AB(I)=AB(I-1)+DIFF
      IF(ABSF(AB(I))-ABSF(DIFF)/100.)119,120,120
  119 AB(I)=0.0
  120 CONTINUE
      IF(NUM-1)122,122,124
  122 BA(1)=0
      GO TO 128
  124 DIFF=(BA(26)-BA(1))/25.0
      DO 126 I=2,25
      BA(I)=BA(I-1)+DIFF
      IF (ABSF(BA(I))-ABSF(DIFF)/100.)125,126,126
  125 BA(I)=0.0
  126 CONTINUE
  128 IN(1)=0
      IN(2)=0
      IN(3)=0
C
C     FILL IN NSTORE ARRAY
C
      DO 135 K=1,NUM
      DO 135 I=1,26
      DO 132 J=1,NR
  132 E(J)=C(1,J)+AB(I)*C(2,J)+BA(K)*C(3,J)+AB(I)*BA(K)*C(4,J)
      IF(E(1))133,134,134
  133 DO1133J=1,NR
 1133 E(J)=-E(J)
  134 CALL ROUTH(NR,NCRIT)
      NSTORE(I,K)=NCRIT
  135 IN(NCRIT)=IN(NCRIT)+1
      WRITE OUTPUT TAPE 6,6,KK(1),AB(1),AB(26)
      IF(NUM-1)137,137,150
  137 DO 140 I=1,3
      IF(IN(I)-26)140,200,200
  140 CONTINUE
```

```
C
C     PRINT STABILITY DATA FOR 1 PAR.
C
      DO 144 I=1,13
      K=NSTORE(I,1)
      KI=NSTORE(I+13,1)
  144 WRITE OUTPUT TAPE 6,7,AB(I),NAME(K),NAME(K+3),AB(I+13),NAME(K1),
     1NAME(K1+3)
      GO TO 450
  150 WRITE OUTPUT TAPE 6,6,KK(2),BA(1),BA(26)
      DO 152I=1,3
      IF(IN(I)-676)152,200,200
  152 CONTINUE
C
C     PRINT STABILITY PLOT FOR 2 PARS.
C
      DIFF=ABSF(BA(26)-BA(1))/5.0
      M=0
      IF(DIFF-10.0**M)154,158,156
  154 M=M-1
      IF(DIFF-10.0**M)154,158,158
  156 M=M+1
      IF(DIFF-10.0**M)157,158,156
  157 M=M-1
  158 WRITE OUTPUT TAPE 6,8
      DO 160 I=1,26
      DO 160 J=1,26
      K=NSTORE(I,J)
  160 NSTORE(I,J)=NAME(K+6)
      M=-M
      DO 185 J=1,26
      K=27-J
      IF(J-1)161,161,163
  161 X=BA(K)*10.0**M
      WRITE OUTPUT TAPE 6,9, X,(NSTORE(I,K),I=1,26)
      GO TO 185
  163 IF(J-6)165,161,167
  165 WRITE OUTPUT TAPE 6,10,(NSTORE(I,K),I=1,26)
      GO TO 185
  167 IF(J-8)165,168,169
  168 WRITE OUTPUT TAPE 6,11,(NSTORE(I,K),I=1,26)
      GO TO 185
  169 IF(J-9)170,170,171
  170 WRITE OUTPUT TAPE 6,12,KK(2),(NSTORE(I,K),I=1,26)
      GO TO 185
  171 IF(J-11)165,161,172
  172 IF(J-13)165,173,174
  173 WRITE OUTPUT TAPE 6,13,(NSTORE(I,K),I=1,26)
      GO TO 185
  174 IF(J-14)175,175,176
  175 WRITE OUTPUT TAPE 6,14,M,(NSTORE(I,K),I=1,26)
      GO TO 185
  176 IF(J-16)165,161,177
  177 IF(J-21)165,161,179
  179 IF(J-26)165,161,161
  185 CONTINUE
      WRITE OUTPUT TAPE 6,8
      DIFF=ABSF(AB(26)-AB(1))/5.0
      M=0
      IF(DIFF-10.0**M)194,198,196
```

```
  194 M=M-1
      IF(DIFF-10.0**M)194,198,198
  196 M=M+1
      IF(DIFF-10.0**M)197,198,196
  197 M=M-1
  198 M=-M
      DO 199 I=1,6
  199 AB(5*I-4)=AB(5*I-4)*10.0**M
      WRITE OUTPUT TAPE 6,15,AB(1),AB(6),AB(11),AB(16),AB(21),AB(26)
  193 WRITE OUTPUT TAPE 6,17,KK(1),M
  450 IF(LF-5)50,510,50
C
C     PRINT STATEMENT IF NO CHANGE IN REGION.
C
  200 WRITE OUTPUT TAPE 6,18,NAME(I),NAME(I+3)
      GO TO 450
  500 NER=10
  510 RETURN
      END(2,0,1,0,0)
```

```
      SUBROUTINE HOLLER  (NWDS,B,A)
      DIMENSION A(2),B(2)
      DO 1 N=1,NWDS
      M=32767-(NWDS-2-N)
      L=NWDS+1-N
    1 A(L)=B(M)
      RETURN
      END(2,0,1,0,0)
```

```
C     ROUTH, DETERMINES POSITIONS OF ROOTS OF POLYNOMIAL          S.M.K.
C
C
      SUBROUTINE ROUTH (NR,NCRIT)
C
      DIMENSION A(12500),B(50,50),COM(3321),E(50)
C
      COMMON A,B,COM,E
C
      IF (E(1))15,10,15
   10 NCRIT=1
      DO 12 I=1,NR
   12 E(I)=E(I+1)
      NRO=NR-1
      GO TO 20
   15 NRO=NR
      NCRIT =0
   20 DO 25 I=1,50
      DO 25 J=1,50
   25 B(I,J)=0.0
      NC=(NRO+1)/2
      DO 30 I=1,NC
      IF(E(2*I-1))110,110,30
```

```
 30 B(NRO,I) =E(2*I-1)
    NC=NRO/2
    NRO=NRO-1
    DO 35 I=1,NC
    IF (E(2*I))110,110,35
 35 B(NRO,I)=E(2*I)
    NRO=NRO-1
    DO 100 I=1,NRO
    K=NRO-I+1
    NC=(K+1)/2
    DO 40 J=1,NC
    B(K,J)=B(K+2,J+1)-B(K+2,1)*B(K+1,J+1)/B(K+1,1)
    IF(ABSF(B(K,J))-ABSF(B(K+2,J+1))*1.0E-6) 38,38,40
 38 B(K,J)=0.0
 40 CONTINUE
    L=0
    DO 50 J=1,NC
    IF (B(K,J)) 110,45,50
 45 L=L+1
 50 CONTINUE
    IF (L) 100,100,60
 60 IF (L-NC)110,65,65
 65 IF (K-2) 120,120,70
 70 NCRIT=1
    DO 80 J=1,NC
    F=K+2-2*J
 80 B(K,J)=F*B(K+1,J)
100 CONTINUE
    NCRIT=NCRIT+1
    RETURN
110 NCRIT=3
    RETURN
120 NCRIT=2
    RETURN
    END (2,0,1,0,0)
```

```
C     WTAP2, WRITE LASS2 DATA ON TAPE 3
C
      SUBROUTINE WTAP2 (IPAGE,N,NPAR,NVAR,NER,KEY,NSH,NMAX,NL)
      DIMENSION A(19071),KEY(6),NL(16)
      COMMON A
      WRITE TAPE 3,A,IPAGE,N,NPAR,NVAR,NER,KEY,NSH,NMAX,NL
      RETURN
      END(2,2,1,2,0)
```

```
C                                 LASS 3
C
C     LASS 3 CALLS THE FOLLOWING SUBROUTINES
C            RTAP2
C            RENUM
C            MESS2 (CALLS POLYX THAT CALLS DETEV)
C            PUNNU
C            PRITRA (CALLS ALPH)
C            ALPH
C            FREQ (CALLS THE MICHIGAN PLOT SUBROUTINES, SEE SHARE
C                    DISTRIBUTION NUMBER 1085.  FREQ ALSO CALLS ALPH.)
C            ROOTER
C            EVAL
C            MULLER (SEE SHARE DISTRIBUTION NUMBER 1124.)
C            TRANS
C            ERR
C            LTAP1
C
C
C
      DIMENSION A(50,50,5),B(50,50),C(201),K(50),NK(50),G(50,8,5),
     1CHAR(16,50),NC(4),MA(8),NT(8),V(50,4),NL(16),AL(4),NO(10),E(50),
     2F(4,50),KEY(6),NPN(4),EXC(3)
C
      COMMON A,B,C,K,NK,G,CHAR,NC,MA,NT,V,E
C
    1 FORMAT (I1,I2,4X,I2,3X,10I2)
    2 FORMAT(I1,I2,1X,1P4E14.7)
    3 FORMAT (1H030X,34HPROBLEM COMPLETED. TO NEXT PROBLEM)
      CALL EFM
      REWIND 1
      REWIND 3
      CALL RTAP2(IPAGE,N,NPAR,NVAR,NER,KEY,NSH,NMAX,NL)
      REWIND 3
      IF(KEY(4)+KEY(5)+KEY(6))300,300,30
   30 READ 1,L1,L2,NIV,(NO(I),I=1,10)
C
C     NIV=NO. TRANSFER FUNCTIONS TO EVALUATE
C     NO(I)= ITH INTERNAL VARIABLE TO EVALUATE FOR
C
      IF(L1-4)325,32,325
   32 IF(L2-1)325,33,325
   33 IF(NIV-10)35,35,325
   35 DO 37 I=1,NIV
      IF(NO(I))325,325,37
   37 CONTINUE
      IF(NPAR)145,145,40
   40 READ 2,L1,L2,(AL(I),I=1,4)
      IF(L1-4)325,44,42
   42 IF(L1-9)325,350,325
   44 IF(L2-2)325,45,325
C
C     1 SET OF PARAMETRIC VALUES IN, CALCULATE DENOMINATOR, PUT IN E
C     CONSTANT TERM LAST
C
   45 PUNCH 2,L1,L2,(AL(I),I=1,4)
  145 DO 46 I=1,50
   46 E(I)=0.0
      DO 50 I=1,NMAX
      J=NMAX+1-I
```

```
   50 E(J)    =CHAR(1,I)+AL(1)*CHAR(2,I)+AL(2)*CHAR(3,I)+AL(1)*AL(2)*
     1CHAR(4,I)+AL(3)*CHAR(5,I)+AL(1)*AL(3)*CHAR(6,I)+AL(2)*AL(3)*CHAR
     2(7,I)+AL(1)*AL(2)*AL(3)*CHAR(8,I)+AL(4)*CHAR(9,I)+AL(1)*AL(4)*
     3CHAR(10,I)+AL(2)*AL(4)*CHAR(11,I)+AL(1)*AL(2)*AL(4)*CHAR(12,I)+
     4AL(3)*AL(4)*CHAR(13,I)+AL(1)*AL(3)*AL(4)*CHAR(14,I)+AL(2)*AL(3)*
     5AL(4)*CHAR(15,I)+AL(1)*AL(2)*AL(3)*AL(4)*CHAR(16,I)
   54 DO 250 I=1,NIV
      DO 52 J=1,4
      DO 52 L=1,50
   52 F(J,L)=0.0
C
C     STORAGES FOR NUMERATOR CLEARED,BRANCH ON KEY4
C
      IF(KEY(4))55,55,60
   55 CALL RENUM(NO,I,F,NPN,NER)
      IF(NER) 70,70,330
C
C     NUMERATORS READ IN, NPN(I)=NO. TERMS IN NUMERATOR FOR V(I)
C
   60 CALL MESS2(F,AL,N,NO,NPN,I,NPAR,NVAR,NSH)
      CALL PUNNU(NO,I,F,NPN,NVAR)
   70 CALL BRLTRA(F,NO,I,NPN,NVAR,AL,IPAGE,NPAR,NMAX)
      DO 72 J=1,NVAR
      IF(NMAX-NPN(J))290,72,72
   72 CONTINUE
      IF(KEY(5))80,80,73
   73 CALL FREQ(E,F,AL,NPAR,NO(I),IPAGE,NVAR,NER,NMAX,NPN,I)
      IF(NER) 80,80,330
   80 IF(KEY(6))250,250,85
   85 IF (I-1) 86,86,87
   86 CALL ROOTER(AL,T,EXC,NMAX,NZER,NRS,NRD,NPI,NCS,NPAR,IPAGE,NER,
     1IREF)
   87 IF (NER) 250,88,330
   88 CALL TRANS(AL,T,EXC,F,NPN,NPAR,NVAR,IPAGE,IREF,NMAX,NZER,NRS,NRD,
     1NPI,NCS,NO(I))
  250 CONTINUE
      IF(NPAR)300,300,40
  290 NER=15
      GO TO 330
  300 READ 1,L1
      IF(L1-9)310,350,310
  310 NER=16
      GO TO 330
  325 NER=12
  330 CALL ERR(NER)
  350 REWIND 1
      PRINT 3
      WRITE OUTPUT TAPE 6,3
      CALL LTAP1
      END(2,0,1,0,1)



C     RTAP2, READ LASS2 DATA INTO CORE
C
      SUBROUTINE RTAP2(IPAGE,N,NPAR,NVAR,NER,KEY,NSH,NMAX,NL)
      DIMENSION  A(19071),KEY(6),NL(16)
      COMMON A
```

```
      READ TAPE 3,A,IPAGE,N,NPAR,NVAR,NER,KEY,NSH,NMAX,NL
      RETURN
      END(2,2,1,2,0)



C     RENUM, READS NUMERATOR INTO STORAGE                             S.M.K.
C
      SUBROUTINE RENUM (NO,I,F,NPN,NER)
C
      DIMENSION F(4,50),NPN(4),NO(10)
C
    1 FORMAT (I1,I2,3X,I2,5X,4I2)
    2 FORMAT (I1,I2,3(2I2,1X,E14.8))
      READ 1,L1,L2,L3,(NPN(J),J=1,4)
      IF(L1-4)25,4,25
    4 IF(L2-3)25,5,25
    5 IF(L3-NO(I))25,6,25
    6 READ 2,L1,L2,I1,J1,(F(I1,J1)),I2,J2,(F(I2,J2)),I3,J3,(F(I3,J3))
      IF(L2-NO(I))25,8,25
    8 IF(L1-4)25,6,10
   10 IF(L1-5)12,12,25
   25 NER=14
   12 RETURN
      END(2,2,1,2,0)



C     MESS2,CALCULATE NUMERATORS OF TRANSFER FUNCTION                 S.M.K.
C
      SUBROUTINE MESS2(F,AL,N,NO,NPN,I,NPAR,NVAR,NSH)
C
      DIMENSION A(50,50,5),B(50,50),C(201),K(50),NK(50),G(50,8,5),
     1CHAR(16,50),NC(4),MA(8),NT(8),V(50,4),AL(4),NO(10),E(50),
     2F(4,50),NPN(4),XTERM(50)
C
      COMMON A,B,C,K,NK,G,CHAR,NC,MA,NT,V,E
C
      DO 5 J=1,4
    5 NPN(J)=0
C
C     SET UP A MATRIX WITH VALUES FOR PARAMETERS
C
      NZ=NSH
      DO 20 J=1,NPAR
      L=-NT(2*J)
      NZ=NZ-L
      K1=XMAXOF(MA(2*J-1)+L,MA(2*J))
      K2=NC(J)
      K(K2)=K1
      DO 10 M=1,N
      DO 10 MM=1,K1
   10 A(M,K2,MM)=0.0
      L1=MA(2*J)
      DO 15 M=1,N
      DO 15 MM=1,L1
```

```
   15 A(M,K2,MM)=AL(J)*G(M,2*J,MM)
      L1=MA(2*J-1)
      DO 20 M=1,N
      DO 20 MM=1,L1
      L2=MM+L
   20 A(M,K2,L2)=A(M,K2,L2)+G(M,2*J-1,MM)
      L1=NO(I)
      KSAV=K(L1)
      K(L1)=1
      DO 30 M=1,N
   30 XTERM(M)=A(M,L1,1)
      DO 50 J=1,NVAR
      DO 35 M=1,N
   35 A(M,L1,1)=V(M,J)
      CALL POLYX (N,L)
      IF(L)50,50,40
   40 NPN(J)=NZ+L
      DO 45 II=1,L
      JJ=NZ+II
   45 F(J,JJ)=C(II)
   50 CONTINUE
      DO 60 M=1,N
   60 A(M,L1,1)=XTERM(M)
      K(L1)=KSAV
      RETURN
      END(2,0,1,0,0)
```

```
C     PUNNU, PUNCH NUMERATORS OF TRANSFER FUNCTION
C
      SUBROUTINE PUNNU(NO,I,F,NPN,NVAR)
C
      DIMENSION F(4,50),NPN(4),NO(10),II(3),JJ(3),C(3)
C
    1 FORMAT(I1,I2,3X,I2,5X,4I2)
    2 FORMAT(I1,I2,3(2I2,1X1PE14.7))
      L1=4
      L2=3
      L3=NO(I)
      PUNCH 1,L1,L2,L3,(NPN(J),J=1,4)
      DO 10 IKKY=1,3
      II(IKKY)=IKKY
      JJ(IKKY)=1
   10 C(IKKY)=0.0
      NN=3
      DO 20 L=1,NVAR
      LEN=NPN(L)
      DO 20 J=1,LEN
      IF(F(L,J))12,20,12
   12 II(NN)=L
      JJ(NN)=J
      C(NN)=F(L,J)
      IF(NN-1)17,17,15
   15 NN=NN-1
      GO TO 20
   17 NN=3
      PUNCH2,L1,L3,(II(LL),JJ(LL),C(LL),LL=1,3)
```

```
  20 CONTINUE
     L1=5
     PUNCH2,L1,L3,(II(LL),JJ(LL),C(LL),LL=1,3)
     RETURN
     END(2,0,1,0,0)
```

```
C     PRITRA,PRINT TRANSFER FUNCTION
C
      SUBROUTINE PRITRA(F,NO,I,NPN,NVAR,AL,IPAGE,NPAR,NMAX)
C
      DIMENSION A(50,50,5),B(50,50),C(201),K(50),NK(50),G(50,8,5),
     1CHAR(16,50),NC(4),MA(8),NT(8),V(50,4),AL(4),NO(10),E(50),
     2F(4,50),NPN(4),II(4)
C
      COMMON A,B,C,K,NK,G,CHAR,NC,MA,NT,V,E
C
    1 FORMAT (1H114X37HTRANSFER FUNCTION FOR VARIABLE NUMBERI3,11X4HPAGE
     1I3//)
    6 FORMAT (1H 20X11HDENOMINATOR25X38HNUMERATOR FOR EXCITING VARIABLE
     1NUMBER/22X,4(21X,I1))
    7 FORMAT (6H   S** I2,2X,5(1PE22.7))
    8 FORMAT(10X,1P5E22.7)
      WRITE OUTPUT TAPE 6,1,NO(I),IPAGE
      IPAGE=IPAGE+1
      CALL ALPH(AL,NPAR)
   20 DO 22 J=1,4
   22 II(J)=J
      WRITE OUTPUT TAPE 6,6,(II(J),J=1,NVAR)
      WRITE OUTPUT TAPE 6,8,E(NMAX),(F(J,1),J=1,NVAR)
      L=NMAX-1
      DO 30 J=1,L
      LL=NMAX-J
   30 WRITE OUTPUT TAPE 6,7,J,E(LL),    (F(JJ,J+1),JJ=1,NVAR)
      RETURN
      END(2,0,1,0,0)
```

```
C     PRINT LINE WITH PARAMETER VALUES                    KATZ
C
      SUBROUTINE ALPH(AL,NPAR)
C
      DIMENSION AL(4)
    1 FORMAT(20X,10HPARAMETER=1PE16.7//)
    2 FORMAT(20X,12HPARAMETER 1=1PE16.7,10X,12HPARAMETER 2=E16.7//)
    3 FORMAT(12X,12HPARAMETER 1=1PE16.7,12X,12HPARAMETER 2=E16.7,12X,
     112HPARAMETER 3=E16.7//)
    4 FORMAT(14H  PARAMETER 1=1PE16.7,14H  PARAMETER 2=E16.7,
     114H  PARAMETER 3=E16.7,14H  PARAMETER 4=E16.7//)
    5 FORMAT (1H )
      IF(NPAR)25,25,10
   10 GO TO (11,12,13,14),NPAR
   11 WRITE OUTPUT TAPE 6,1,AL(1)
      GO TO 20
```

```
 12 WRITE OUTPUT TAPE 6,2,AL(1),AL(2)
    GO TO 20
 13 WRITE OUTPUT TAPE 6,3,AL(1),AL(2),AL(3)
    GO TO 20
 14 WRITE OUTPUT TAPE 6,4,AL(1),AL(2),AL(3),AL(4)
 20 RETURN
 25 WRITE OUTPUT TAPE 6,5
    RETURN
    END(2,0,1,0,0)



                                                              KATZ
C      FREQUENCY RESPONSE SUBROUTINE
C
       SUBROUTINE FREQ(E,F,AL,NPAR,NZ,IPAGE,NVAR,NER,NMAX,NPN,NOP)
C
       DIMENSION A(50,50,5),OMEG(100),OLMEG(100),GR(4,100),GI(4,100),
      1GAN(4,100),GADEB(4,100),THET(4,100),YCO(100),FIN(10),FLIN(10),
      2E(50),F(4,50),NPN(4),TD(4),TN(4),AL(4),BL(3643),GRAPH(867),
      3XC(100),WOMEG(120),NSCALE(5)
C
       COMMON A,                GR,GI,GAN,GADEB,THET,YCO,FIN,FLIN,TD,TN,XC,
      1BL,GRAPH
C
     1 FORMAT(2X,I2,2X,I3,2X,I1,3X,I2,3X,10F3.1)
     2 FORMAT(1H1,11X,38HFREQUENCY RESPONSE FOR VARIABLE NUMBERI3,13X,
      14HPAGEI3/  19X,26HEXCITED BY FUNCTION NUMBERI2/ )
     7 FORMAT(5X,9HFREQUENCY16X,1HX19X,1HY17X4HGAIN16X,4HGAIN12X,
      111HPHASE ANGLE/6X7HRAD/SEC14X,6HNUMBER14X,6HNUMBER14X,6HNUMBER
      213X,8HDECIBELS12X,7HDEGREES)
     8 FORMAT(1P6E20.7)
     9 FORMAT(1H117X,30HFREQUENCY RESPONSE (CONTINUED)18X,4HPAGEI3//)
    10 FORMAT(1H111X,38HLOG MAGNITUDE PLOT FOR VARIABLE NUMBERI3,13X,
      14HPAGE I3)
    11 FORMAT(120A1/50X,20HFREQUENCY IN RAD/SEC)
    12 FORMAT(1H113X,36HPHASE ANGLE PLOT FOR VARIABLE NUMBERI3,13X4HPAGE
      1I3//)
    13 FORMAT(1H116X,30HPOLAR PLOT FOR VARIABLE NUMBERI3,16X,4HPAGEI3//)
    14 FORMAT(60X,9HREAL PART)
       IF(NOP-1)15,15,27
    15 NFIR=0
       NF=0
    20 READ 1,L1,L2,N,NO,(FIN(I),I=1,10)
       IF(L1-10)400,21,400
    21 IF(N)26,26,22
    22 IF(NO-10)23,23,400
C
C      SET UP OMEGA VALUES IN OMEG(J) AND LOG OMEGA IN OLMEG (J)
C
    23 DO 223 I=1,NO
   223 FLIN(I)=LOGF(FIN(I))/2.3025851
       DO 25 K=1,N
       NP=L2+K-1
       OL=NP
       NFIR=NF
       NF=NF+NO
       IF(NF-100)24,405,405
    24 DO 25 KK=1,NO
```

```
      NFIRK=NFIR+KK
      OMEG(NFIRK)=FIN(KK)*10.**NP
   25 OLMEG(NFIRK)=FLIN(KK)+OL
      GO TO 20
   26 NF=NF+1
      OMEG(NF)=10.**L2
      OLMEG(NF)=L2
C
C     CALCULATE GAIN IN CARTESIAN AND POLAR COORDINATES
C
   27 DO 85 I=1,NF
      DO 28 J=1,4
   28 TD(J)=0.0
C
C     TD2=POS. REAL PART OF DEN., TD3=POS. IMAG. PART
C     TD4=NEG. REAL PART OF DEN., TD1=NEG. IMAG. PART
C
      IF(OLMEG(I))30,30,47
C
C     OMEGA L.T.E. 1, USE ORIGINAL EQUATION
C     DENOMINATOR FIRST
C
   30 DO 34 J=1,NMAX
      DO 29 K=1,4
   29 TD(K)=TD(K)*OMEG(I)
      NUM=NMAX
      IF(E(J))31,32,32
   31 NUM=2+NMAX
   32 KN= XMODF(NUM-J+1,4) +1
   34 TD(KN)=TD(KN)+ABSF(E(J))
C
C     PUT REAL PART OF DEN. IN TD2, IMAG. IN TD1, SQUARE OF DEN. IN TD3
C
      TD(2)=TD(2)-TD(4)
      TD(1)=TD(3)-TD(1)
      TD(3)=TD(2)**2+TD(1)**2
C
C     NUMERATORS NEXT
C
      DO 45 II=1,NVAR
      NU=NPN(II)
      DO 36 K=1,4
   36 TN(K)=0.0
C
C     TN FOR NUMERATOR TERMS AS TD FOR DENOMINATOR
C
      DO 42 J=1,NU
      DO 38 K=1,4
   38 TN(K)=TN(K)*OMEG(I)
      NUM=NU
      LUM=NU+1-J
      IF(F(II,LUM))39,40,40
   39 NUM=2+NU
   40 KN= XMODF(NUM-J+1,4) +1
   42 TN(KN)=TN(KN)+ABSF(F(II,LUM))
      TN(2)=TN(2)-TN(4)
      TN(1)=TN(3)-TN(1)
      GR(II,I)=(TN(2)*TD(2)+TN(1)*TD(1))/TD(3)
   45 GI(II,I)=(TN(1)*TD(2)-TN(2)*TD(1))/TD(3)
      GO TO 75
```

```
C
C     OMEGA G. T. 1, DIVIDE NUM. AND DENOM BY OMEGA** NMAX
C
   47 DO 49 J=1,4
   49 TD(J)=0.0
C
C     TD1=POS. REAL PART, TD2=POS. IMAG. PART
C     TD3=NEG. REAL PART, TD4=NEG. IMAG. PART
C
      DO 57 J=1,NMAX
      DO 52 K=1,4
   52 TD(K)=TD(K)/OMBG(I)
      NUM=NMAX+1-J
      KJ=J
      IF(E(NUM))53,55,55
   53 KJ=J+2
   55 KN= XMODF(KJ-1,4) +1
   57 TD(KN)=TD(KN)+ABSF(E(NUM))
      TD(3)=TD(1)-TD(3)
      TD(1)=TD(2)-TD(4)
      TD(2)=TD(3)
      TD(3)=TD(1)**2+TD(2)**2
      DO 70 II=1,NVAR
      DO 59 K=1,4
   59 TN(K)=0.0
      DO 67 J=1,NMAX
      DO 61 K=1,4
   61 TN(K)=TN(K)/OMBG(I)
      NUM=J
      IF(F(II,J))62,63,63
   62 NUM=J+2
   63 KN= XMODF(NUM-1,4) +1
   67 TN(KN)=TN(KN)+ABSF(F(II,J))
      TN(3)=TN(1)-TN(3)
      TN(1)=TN(2)-TN(4)
      GR(II,I)=(TN(3)*TD(2)+TN(1)*TD(1))/TD(3)
   70 GI(II,I)=(TN(1)*TD(2)-TN(3)*TD(1))/TD(3)
   75 DO 85 II=1,NVAR
      GAN(II,I)=SQRTF(GR(II,I)**2+GI(II,I)**2)
      GADEB(II,I)=8.6858896*LOGF(GAN(II,I))
      IF(GR(II,I))76,71,78
   71 IF(GI(II,I))72,73,74
   72 THET(II,I)=270.0
      GO TO 85
   73 THET(II,I)=0.0
      GO TO 85
   74 THET(II,I)=90.0
      GO TO 85
   76 THET(II,I)=180.0
      GO TO 83
   78 IF(GI(II,I))80,82,82
   80 THET(II,I)=360.0
      GO TO 83
   82 THET(II,I)=0.0
   83 THET(II,I)=THET(II,I)+57.29578*ATANF(GI(II,I)/GR(II,I))
   85 CONTINUE
C
C     PRINT TABLE OF FREQUENCY RESPONSE
C
      DO 100 I=1,NVAR
```

```
      WRITE OUTPUT TAPE 6,2,NZ,IPAGE,I
      IPAGE=IPAGE+1
      CALL ALPHI(AL,NPAR)
   90 WRITE OUTPUT TAPE 6,7
      IB=1
      IF(NF-50)95,95,91
   91 IE=NF/2
      ID=1
      GO TO 96
   95 IE=NF
      ID=2
   96 DO 97 J=IB,IE
   97 WRITE OUTPUT TAPE 6,8,OMEG(J),GR(I,J),GI(I,J),GAN(I,J),GADEB(I,J),
     1THET(I,J)
      GO TO (98,100),ID
   98 IB=IE+1
      WRITE OUTPUT TAPE 6,9,IPAGE
      IPAGE=IPAGE+1
      GO TO 95
  100 CONTINUE
C
C     TABLE PRINTED, NOW MAKE THREE PLOTS
C
      WRITE OUTPUT TAPE 6,10,NZ,IPAGE
      IPAGE=IPAGE+1
      CALL ALPHI(AL,NPAR)
  111 YMAX=GADEB(1,1)
      YMIN=GADEB(1,1)
      DO 112 I=1,NVAR
      DO 112 J=1,NF
      YMAX=MAX1F(YMAX,GADEB(I,J))
  112 YMIN=MIN1F(YMIN,GADEB(I,J))
      DIV=10.0
      NDIV=10
  115 IF(YMAX)113,116,114
  113 I=YMAX/DIV
      YMAX=NDIV*I
      GO TO 116
  114 I=.99999+YMAX/DIV
      YMAX=NDIV*I
  116 IF(YMIN)117,119,118
  117 I=YMIN/DIV-.99999
      YMIN=NDIV*I
      GO TO 119
  118 I=YMIN/DIV
      YMIN=NDIV*I
  119 NHL=(YMAX-YMIN)/DIV+.01
      IF(NHL-5)120,120,121
  120 NSBH=10
      GO TO 125
  121 IF(NHL-10)122,122,123
  122 NSBH=5
      GO TO 125
  123 DIV=2.0*DIV
      NDIV=2*NDIV
      GO TO 115
  125 NSCALE(1)=1
      NSCALE(2)=0
      NSCALE(3)=-1
      NSCALE(4)=0
```

```
      NSCALE(5)=-1
      IF(NOP-1)124,124,140
  124 I2=0
  124 I2=0
      IF(OLMEG(NF))126,128,127
  126 I2=OLMEG(NF)-.0001
      GO TO 128
  127 I2=OLMEG(NF)+.9999
  128 I1=0
      IF(OLMEG(1))129,131,130
  129 I1=OLMEG(1)-.9999
      GO TO 131
  130 I1=OLMEG(1)+.0001
  131 NVL=I2-I1
      IF(NVL-5)132,132,134
  132 NSBV=20
      GO TO 140
  134 IF(NVL-10)136,136,138
  136 NSBV=10
      GO TO 140
  138 NSBV=101/NVL
  140 CALL PLOT1(NSCALE,NHL,NSBH,NVL,NSBV)
      XMAX=I2
      XMIN=I1
      CALL PLOT2(GRAPH,XMAX,XMIN,YMAX,YMIN)
      FAC= .02*(YMAX-YMIN)
      YCO(1)=YMIN
      DO 141 I=2,51
  141 YCO(I)= YCO(I-1) + FAC
      IC=1
  144 DO 145 I=1,NVL
      FAC=I-1
      DO 142 J=1,51
  142 XC(J)=XMIN+FAC+.3
      BL(2*I-1)=XC(1)
      CALL PLOT 3(1HI,XC,YCO,51)
      DO 143 J=1,51
  143 XC(J)=XMIN+FAC+.7
      BL(2*I)=XC(1)
  145 CALL PLOT 3(1HI,XC,YCO,51)
      GO TO (147,186),IC
  147 TERM=NHL
      ND=NHL+1
      TERM=(YMAX-YMIN)/TERM
      DO 150 J=1,NVL
      DO 146 I=1,ND
      FAC=I-1
      YCO(I)=YMIN+FAC*TERM
  146 XC(I)=BL(2*J-1)
      CALL PLOT 3(1H+,XC,YCO,ND)
      DO 148 I=1,ND
  148 XC(I)=BL(2*J)
  150 CALL PLOT3(1H+,XC,YCO,ND)
      DO 160 I=1,NVAR
      DO 152 J=1,NF
  152 YCO(J)=GADEB(I,J)
      GO TO(156,157,158,159),I
  156 BCD=1HA
      GO TO 160
  157 BCD=1HB
```

```
      GO TO 160
  158 BCD=1HC
      GO TO 160
  159 BCD=1HD
  160 CALL PLOT3(BCD,OLMEG,YCO,NF)
      CALL OMIT(1)
      CALL FPLOT 4(25,25HLOG MAGNITUDE IN DECIBELS)
      IF(NOP-1)163,163,178
B 163 FIN(1)=330000000000
B     FIN(2)=000000000000
B     FIN(3)=010000000000
B     FIN(4)=606060606060
      DO 161 I=1,120
  161 WOMEG(I)=FIN(4)
      J=13-NSBV
      II=NVL+1
      DO 175 I=1,II
      J=J+NSBV
      JJ=I1+I-1
      IF(JJ+1)162,165,168
  162 JP=J+JJ+2
      WOMEG(JP-2)=FIN(1)
      DO 164 KP=JP,J
  164 WOMEG(KP-1)=FIN(2)
      WOMEG(J)=FIN(3)
      GO TO 175
  165 WOMEG(J-1)=FIN(1)
      WOMEG(J)=FIN(3)
      GO TO 175
  168 IF(JJ)169,169,171
  169 WOMEG(J)=FIN(3)
      GO TO 175
  171 JP=J-JJ+1
      WOMEG(JP-1)=FIN(3)
      DO 173 KP=JP,J
  173 WOMEG(KP)=FIN(2)
  175 CONTINUE
  178 WRITE OUTPUT TAPE 6,11,(WOMEG(I),I=1,120)
  180 WRITE OUTPUT TAPE 6,12,NZ,IPAGE
      IC=2
      IPAGE=IPAGE+1
      CALL ALPH(AL,NPAR)
  182 CALL PLOT1(NSCALE,4,10,NVL,NSBV)
      CALL PLOT 2(GRAPH,XMAX,XMIN,360.,0.)
      YCO(1)=0.0
      DO 184 I=2,51
  184 YCO(I)=YCO(I-1)+9.0
      GO TO 144
  186 DO 195 I=1,NVL
      DO 190 J=1,5
      YCO(J)=(J-1)*90
  190 XC(J)=BL(2*I-1)
      CALL PLOT3(1H+,XC,YCO,5)
      DO 194 J=1,5
  194 XC(J)=BL(2*I)
  195 CALL PLOT 3(1H+,XC,YCO,5)
      DO 210 I=1,NVAR
      DO 198 J=1,NF
  198 YCO(J)=THET(I,J)
      GO TO (201,202,203,204),I
```

```
201 BCD=1HA
    GO TO 210
202 BCD=1HB
    GO TO 210
203 BCD=1HC
    GO TO 210
204 BCD=1HD
210 CALL PLOT3(BCD,OLMEG,YCO,NF)
    CALL OMIT(1)
    CALL FPLOT4(22,22HPHASE ANGLE IN DEGREES)
    WRITE OUTPUT TAPE 6,11,(WOMEG(I),I=1,120)
    WRITE OUTPUT TAPE 6,13,NZ,IPAGE
    IPAGE=IPAGE+1
    CALL ALPH(AL,NPAR)
212 XMAX=GR(1,1)
    YMAX=GI(1,1)
    DO 215 I=1,NVAR
    DO 215 J=1,NF
    XMAX=MAX1F(ABSF(GR(I,J)),XMAX)
215 YMAX=MAX1F(ABSF(GI(I,J)),YMAX)
    IX=XMAX
    XMAX=IX+1
    IX=YMAX
    YMAX=IX+1
    CALL PLOT 1(NSCALE,2,25,2,50)
    CALL PLOT 2(GRAPH,XMAX,-XMAX,YMAX,-YMAX)
    DO 240 I=1,NVAR
    DO 230 J=1,NF
    YCD(J)=GI(I,J)
230 XC(J)=GR(I,J)
    GO TO (231,232,233,234),I
231 BCD=1HA
    GO TO 240
232 BCD=1HB
    GO TO 240
233 BCD=1HC
    GO TO 240
234 BCD=1HD
240 CALL PLOT 3(BCD,XC,YCO,NF)
    CALL OMIT(-1)
    CALL FPLOT 4(14,14HIMAGINARY PART)
    WRITE OUTPUT TAPE 6,14
    RETURN
400 NER=17
    RETURN
405 NER=18
    RETURN
    END(2,0,1,0,0)
```

```
C
C     MICHIGAN PLOT SUBROUTINES - SEE SHARE DISTRIBUTION NO. 1085
C
C
```

```
C     ROOTER-FINDS POLES OF TRANSFER FUNCTION              KATZ
C
      SUBROUTINE ROOTER(AL,T,EXC,NMAX,NZER,NRS,NRD,NPI,NCS,
     1NPAR,IPAGE,NER,IREF)
C
      DIMENSION A(18321),E(50),AL(4),ROOTR(50),ROOTI(50),VAL(6,50),
     1GRAPH(867),NMG(3),MIG(6),EXC(3)
C
      COMMON A,B,GRAPH,VAL,ROOTR,ROOTI
    1 FORMAT ( 1H1,13X,38HSINGULARITIES OF THE TRANSFER FUNCTION14X,
     14HPAGEI3/31X,3HFOR)
    2 FORMAT (I1,3X,4E14.8)
    3 FORMAT(1H010X,57HODD NUMBER OF COMPLEX ROOTS FOUND. THEY ARE THE F
     1OLLOWING)
    4 FORMAT (6X,9HREAL PART8X,14HIMAGINARY PART)
    5 FORMAT (1P2E20.7,20X,2E20.7)
    6 FORMAT(1H010X,69HCOMPLEX ROOTS DO NOT OCCUR IN CONJUGATE PAIRS. TH
     1EY ARE THE FOLLOWING)
    7 FORMAT(1H013X,13HSINGULARITIES30X,46HCOEFFICIENTS OF EQUATION (CON
     1STANT TERM FIRST)/11X9HREAL PART6X,14HIMAGINARY PART32X,8HORIGINAL
     26X,14HRE-CONSTITUTED)
    8 FORMAT (60X,1P2E20.7)
    9 FORMAT(1H010X,62HSINGULARITIES NOT AT ORIGIN WITH MULTIPLICITY GRE
     1ATER THAN TWO)
      WRITE OUTPUT TAPE 6,1,IPAGE
      IREF=IPAGE
      IPAGE=IPAGE+1
      CALL ALPH(AL,NPAR)
      READ 2,L1,T,(EXC(I),I=1,3)
      IF(L1-6)500,20,500
   20 N=NMAX-1
      CALL MULLER (E,N,ROOTR,ROOTI)
C
C     ROOTS OF DENOMINATOR FOUND, NOW ARRANGE IN ORDER
C
      J=0
      K=0
      L=0
      CRS=1.E-5/T
      DO 30 I=1,N
      IF(ROOTI(I))25,23,25
   23 J=J+1
      VAL(1,J)=ROOTR(I)
      VAL(4,J)=ROOTI(I)
      GO TO 30
   25 IF(ABSF(ROOTI(I))-CRS)26,26,28
C
C     IF REAL OR IMAGINARY PART OF ROOT IS VERY SMALL(100*T*ROOT PART
C     LESS THAN .001), SET IT TO ZERO - EXPONENTIAL WILL NOT CHANGE
C     DURING TIME INTERVAL EXAMINED
C
   26 K=K+1
      VAL(2,K)=ROOTR(I)
      VAL(5,K)=ROOTI(I)
      GO TO 30
   28 L=L+1
      VAL(3,L)=ROOTR(I)
      VAL(6,L)=ROOTI(I)
```

```
   30 CONTINUE
C
C     J REAL ROOTS IN VAL(1,I),VAL(4,I) J=NMG(1)
C     K ROOTS WITH NEAR ZERO IMAGINARY PART IN VAL(2,I),VAL(5,I),K=NMG(2
C     L COMPLEX ROOTS IN VAL(3,I),VAL(6,I),L=NMG(3)
C
C     K+L TOTAL NUMBER COMPLEX ROOTS. IF ODD PRINT ERROR AND RETURN
C
      KNUM=(K+L)/2
      IF(K+L-2*KNUM)40,40,32
   32 WRITE OUTPUT TAPE 6,3
   33 WRITE OUTPUT TAPE 6,4
      DO 35 I=1,N
   35 WRITE OUTPUT TAPE 6,5,ROOTR(I),ROOTI(I)
   37 NER=-5
      RETURN
   40 NMG(1)=J
      NMG(2)=K
      NMG(3)=L
      I=0
      DO 55 II=1,3
      MIG(2*II-1)=0
      L=0
      J=NMG(II)
      IF(J)49,49,43
   43 DO 48 K=1,J
      IF(ABSF(VAL(II,K))-CRS)45,45,47
   45 MIG(2*II-1)=MIG(2*II-1)+1
      I=I+1
      ROOTR(I)=VAL(II,K)
      ROOTI(I)=VAL(II+3,K)
      GO TO 48
   47 L=L+1
      VAL(II,L)=VAL(II,K)
      VAL(II+3,L)=VAL(II+3,K)
   48 CONTINUE
   49 MIG(2*II)=L
      IF(L)55,55,50
   50 DO 54 K=1,L
      I=I+1
      ROOTR(I)=VAL(II,K)
   54 ROOTI(I)=VAL(II+3,K)
   55 CONTINUE
C
C     MIG(1) REAL ROOTS NEAR ORIGIN
C     MIG(2) REAL ROOTS NOT NEAR ORIGIN        NMG(1)=MIG(1)+MIG(2)
C     MIG(3) COMPLEX ROOTS NEAR ORIGIN         NMG(2)=MIG(3)+MIG(4)
C     MIG(4) COMPLEX ROOTS ALMOST REAL NOT NEAR ORIGIN
C     MIG(5) ROOTS ALMOST PURE IMAGINARY       NMG(3)=MIG(5)+MIG(6)
C     MIG(6) COMPLEX ROOTS NOT NEAR ANY AXIS
C     NOW ARRANGE EACH GROUP IN ORDER
C
      NTOP=MIG(1)
      DO 61 J=2,6
      NP=NTOP
      NPLUS=NP+1
      NTOP=NP+MIG(J)
      NLESS=NTOP-1
      IF (MIG(J)-1)61,61,57
   57 DO 60 K=NPLUS,NLESS
```

```
      IN=K+1
      DO 60 L=IN,NTOP
      IF(ROOTR(K)-ROOTR(L))58,60,60
   58 VAL(1,1)=ROOTR(K)
      VAL(1,2)=ROOTI(K)
      ROOTR(K)=ROOTR(L)
      ROOTI(K)=ROOTI(L)
      ROOTR(L)=VAL(1,1)
      ROOTI(L)=VAL(1,2)
   60 CONTINUE
   61 CONTINUE
C
C     OBTAIN RE-CONSTITUTED EQUATION AND TEST FOR CONJUGATE PAIRS
C
      DO 62 I=1,6
      DO 62 J=1,50
   62 VAL(I,J)=0.0
      VAL(1,1)=E(1)
      NZER=NMG(1)
      IF(NZER)85,85,77
   77 DO 80 I=1,NZER
      TEM1=0.0
      TEM2=0.0
      VAL(2,1)=-ROOTR(I)
      VAL(2,2)=1.0
      II=I+1
      DO 80 J=1,II
      TEM1=TEM2
      TEM2=VAL(1,J)
   80 VAL(1,J)=TEM2*VAL(2,1)+TEM1*VAL(2,2)
   85 NNZ=NMG(2)+NMG(3)
      IF(NNZ)94,94,86
   86 DO 93 I=1,KNUM
      J=NZER+2*I-1
      IF(ABSF(ROOTR(J)-ROOTR(J+1))-ABSF(ROOTR(J))*1.0E-6)87,87,89
   87 IF(ABSF(ROOTI(J)+ROOTI(J+1))-ABSF(ROOTI(J))*1.0E-6)90,90,89
   89 WRITE OUTPUT TAPE 6,6
      GO TO 33
   90 TEM1=0.0
      TEM2=0.0
      TEM3=0.0
      VAL(2,1)=ABSF(ROOTR(J)*ROOTR(J+1))+ABSF(ROOTI(J)*ROOTI(J+1))
      VAL(2,2)=-ROOTR(J)-ROOTR(J+1)
      VAL(2,3)=1.0
      K=NZER+2*I+1
      DO 93 L=1,K
      TEM1=TEM2
      TEM2=TEM3
      TEM3=VAL(1,L)
   93 VAL(1,L)=TEM3*VAL(2,1)+TEM2*VAL(2,2)+TEM1*VAL(2,3)
C
C     COEFFICIENTS OF EQUATION HAVE BEEN RECONSTITUTED, NOW PRINT OUT
C
   94 WRITE OUTPUT TAPE 6,7
      DO 95 I=1,N
      J=N+2-I
   95 WRITE OUTPUT TAPE 6,5,ROOTR(I),ROOTI(I),E(J),VAL(1,I)
      WRITE OUTPUT TAPE 6,8,E(1),VAL(1,N+1)
C
C     TEST FOR MULTIPLICITIES GREATER THAN TWO IN REAL ROOTS
```

```
C
      J=MIG(2)
      K=MIG(4)
      IF(J)115,115,102
  102 IF(K)115,115,104
  104 DO 110 I=1,J
      L=MIG(1)+I
      DO 106 M=1,K
      LL=NMG(1)+MIG(3)+M
      ROOT=ROOTR(L)-ROOTR(LL)
      IF(ABSF(ROOT)-CRS)503,503,105
  105 IF(ROOT)106,106,110
  106 CONTINUE
      GO TO 115
  110 CONTINUE
C
C     TRANSFER ROOTS TO VAL(1,I),VAL(2,I)
C
  115 VAL(1,1)=0.0
      VAL(2,1)=0.0
      I=1
C
C     ARE THERE PAIRED ROOTS IN GROUP 2 (REAL ROOTS NOT AT ORIGIN)
C
      NRS=0
      NRD=0
      L=MIG(1)
      M=L+J
  116 IF(M-L-1)125,122,117
  117 IF(ROOTR(L+1)-ROOTR(L+2)-CRS)119,119,122
  119 DO 120 LL=1,2
      L=L+1
      NRD=NRD+1
      VAL(3,NRD)=ROOTR(L)
  120 VAL(4,NRD)=ROOTI(L)
      IF(M-L)125,125,121
  121 IF(ROOTR(L)-ROOTR(L+1)-CRS)503,503,116
  122 NRS=NRS+1
      I=I+1
      L=L+1
      VAL(1,I)=ROOTR(L)
      VAL(2,I)=ABSF(ROOTI(L))
      GO TO 116
  125 IF(NRD)130,130,126
  126 DO 128 LL=1,NRD
      I=I+1
      VAL(1,I)=VAL(3,NRD)
  128 VAL(2,I)=ABSF(VAL(4,NRD))
  130 L=NMG(L)+MIG(3)
      IF(K)137,137,132
  132 DO 135 LL=1,K
      I=I+1
      L=L+1
      VAL(1,I)=ROOTR(L)
  135 VAL(2,I)=ABSF(ROOTI(L))
  137 L=NMG(1)+NMG(2)
      J=MIG(5)+MIG(6)
      IF(J)145,145,139
  139 DO 140 LL=1,J
      L=L+1
```

```
      I=I+1
      VAL(1,I)=ROOTR(L)
  140 VAL(2,I)=ABSF(ROOTI(L))
C
C     ROOTS HAVE BEEN SHIFTED TO VAL(1,I) AND VAL (2,I)
C
  145 NZER=MIG(1)+MIG(3)
      NRD=NRD+MIG(4)
      NPI=MIG(5)
      NCS=MIG(6)
C
C     NZER=NO. AT ORIGIN. NRS=NO. SIMPLE REAL. NRD=NO. DOUBLE REAL
C     NPI=NO. PURE IMAGINARY.  NCS=NO. COMPLEX
C
C     EVALUATE DERIVATIVES OF POLYNOMIAL
C
      M=NMAX-NZER
      VAL(3,1)=E(M)
      NN=NMAX-2
      DO 150 I=1,NN
      FAC=NMAX-I
      ROOTR(I)=E(I)*FAC
      ROOTI(I)=ROOTR(I)*(FAC-1.0)
  150 VAL(6,I)=ROOTI(I)*(FAC-2.0)
C
C     FIRST DERIVATIVE IN ROOTR, SECOND IN ROOTI, THIRD IN VAL(6,I)
C
      ROOTR(N)=E(N)
      IF(NRS)162,162,152
  152 DO 160 I=1,NRS
      J=I+1
  160 CALL EVAL(ROOTR,N,VAL(1,J),0.0,VAL(3,J),VAL(4,J))
  162 K=NPI+NCS
      IF(K)172,172,165
  165 DO 170 I=1,K,2
      J=1+NRS+NRD+I
      CALL EVAL(ROOTR,N,VAL(1,J),VAL(2,J),VAL(3,J),VAL(4,J))
      VAL(3,J+1)=VAL(3,J)
  170 VAL(4,J+1)=VAL(4,J)
  172 IF(NRD)501,501,173
  173 DO 175 I=1,NN
  175 ROOTR(I)=VAL(6,I)
      NNN=NMAX-3
      DO 178 I=1,NRD
      J=1+NRS+I
      CALL EVAL(ROOTI,NN,VAL(1,J),0.0,VAL(3,J),VAL(4,J))
  178 CALL EVAL(ROOTR,NNN,VAL(1,J),0.0,VAL(5,J),VAL(6,J))
      RETURN
C
C     FOR SINGLE ROOTS VALUE OF FIRST DERIV. IN VAL(3,I),VAL(4,I)
C     FOR DOUBLE ROOTS VALUE OF SECOND DERIV. IN VAL(3,I),VAL(4,I)
C     THIRD DERIV. IN VAL(5,I),VAL(6,I)
C
  500 NER=19
  501 RETURN
  503 WRITE OUTPUT TAPE 6,9
      GO TO 37
      END(2,0,1,0,0)
```

```
C     EVALUATE POLYNOMIAL                                           KATZ
      SUBROUTINE EVAL(C,N,AR,AI,VR,VI)
C
C     N COEFFICIENTS (N-1 ORDER) IN C ARRAY WITH CONSTANT TERM LAST
C     TO BE EVALUATED FOR COMPLEX VALUE OF VARIABLE, REAL PART=AR
C     IMAGINARY=AI, VALUE OF POLYNOMIAL IS VR(REAL),VI(IMAGINARY)
C
      DIMENSION C(1)
      N1=N-1
      IF(N1)5,1,5
    1 VR=C(N)
      VI=0.0
      RETURN
    5 TEMR=C(1)
      TEMI=0.0
      DO 10 I=1,N1
      TE1=TEMR*AR-TEMI*AI
      TEMI=TEMI*AR+TEMR*AI
   10 TEMR=TE1+C(I+1)
      VR=TEMR
      VI=TEMI
      RETURN
      END(2,0,1,0,0)
```

```
C     TRANS SUBROUTINE OBTAINS INVERSE LAPLACE TRANSFORM      KATZ
      SUBROUTINE TRANS(AL,T,EXC,F,NPN,NPAR,NVAR,IPAGE,IREF,NMAX,
     1NZER,NRS,NRD,NPI,NCS,NOP)
C
      DIMENSION A(12500),FUN(4,51),RESP(4,101),TERM(52),TREM(50),
     1EXC(3),AL(4),F(4,50),NPN(4),VAL(6,50),TR(4909),GRAPH(867),TIME(101
     2),E(50),STAT(7),NSCALE(5),YCO(101)
C
      COMMON A,FUN,RESP,TERM,TREM,TIME,YCO,TR,E,GRAPH,VAL
C
C     THREE TYPES OF EXCITATION-1)IMPULSE,2)STEP,3)RAMP
C
    1 FORMAT(1H1,11X,38HTRANSIENT RESPONSE FOR VARIABLE NUMBERI3,13X,
     14HPAGEI3/5X,11HEXCITED BY A5,13H OF MAGNITUDEF15.7,19H IN FUNCTION
     2 NUMBERI2,5X24H(REFER TO PRINT-OUT PAGEI3,1H))
    2 FORMAT(13X,27HF(T) APPROXIMATELY GIVEN BY25X,1HT14X4HF(T)15X1HT
     114X4HF(T))
    3 FORMAT(1PE13.3,34X,5H+    I4E17.5)
    4 FORMAT(1PE13.3,15X,7HX  TIME12X,5H+    I4E17.5)
    5 FORMAT(1PE13.3,11X,9HX  TIME**I3,11X,5H+    I4E17.5)
    6 FORMAT(1PE13.3,6X,4HEXP(E10.3,8H X TIME)6X,5H+    I4E17.5)
    7 FORMAT(1PE13.3,14H X TIME X EXP(E10.3,15H X TIME)  +    I4E17.5)
    8 FORMAT(1PE13.3,4HCOS(E10.3,10H X TIME + E10.3,5H) + I4E17.5)
    9 FORMAT(1PE13.3,6X,4HEXP(E10.3,8H X TIME)6X,5HX    I4E17.5)
   10 FORMAT (6X,4HCOS(1PE10.3,10H X TIME + E10.3,1H)6X,5H+    I4E17.5)
   11 FORMAT (52X,1P4E17.5)
   12 FORMAT(1H1,7X,46HPLOT OF TRANSIENT RESPONSE FOR VARIABLE NUMBERI3,
     19X,4HPAGEI3/5X,11HEXCITED BY A5,13H OF MAGNITUDEF15.7,26X,
     224H(REFER TO PRINT-OUT PAGEI3,1H))
```

```
   13 FORMAT (20X,23HABSCISSA IS TIME X 10**I3,10X,27HORDINATE IS RESPON
     1SE X 10**I3)
C
      STAT(1)=5HPULSE
      STAT(2)=4HSTEP
      STAT(3)=4HRAMP
      STAT(4)=1HA
      STAT(5)=1HB
      STAT(6)=1HC
      STAT(7)=1HD
C
      DO 300 IEX=1,3
      IF(EXC(IEX))300,300,20
C
C     IF EXC(IEX)=0 DO NOT CALCULATE FOR THIS TYPE EXCITATION
C
   20 NZ=NZER+IEX-1
      TIME(1)=0.0
C
C     CALCULATE 101 VALUES OF TIME FROM 0 TO 100*T (IN TIME(J))
C
      DO 19 J=2,101
   19 TIME(J)=TIME(J-1)+T
C
C     NZ=NUMBER POLES AT ORIGIN
C
      XMIN =0.0
      XMAX =0.0
      DO 250 I=1,NVAR
      LTE=NPN(I)
      IF(NZ)30,30,22
C
C     COEFFICIENTS FOR POLES AT ORIGIN
C
   22 FUN(I,NZ)=F(I,1)*EXC(IEX)/VAL(3,1)
      IF(NZ-1)30,30,123
  123 DO 23 J=1,52
   23 TERM(J)=0.0
      DO 25 J=1,LTE
   25 TERM(J)=F(I,J)*EXC(IEX)
      LT=NMAX-NZER
      DO 27 K=2,NZ
      DO 26 L=1,LT
      LK=L+K-2
      NZK=NZ+1-K
      NMK=LT+1-L
   26 TERM(LK)=TERM(LK)-FUN(I,NZK+1)*E(NMK)
   27 FUN(I,NZK)=TERM(K)/VAL(3,1)
  127 COUNT =1.0
      AK=0.0
      DO 28 K=2,NZ
      AK=AK+1.0
      COUNT=COUNT*AK
   28 FUN(I,K)=FUN(I,K)/COUNT
C
C     SET UP COEFFICIENTS FOR NUMERATOR WITH CONSTANT TERM LAST
C
   30 DO 34 L=1,LTE
      K=LTE+1-L
   34 TERM(L)=F(I,K)
```

```
      COUNT=IEX-1
      IF(NRS)40,40,35
C
C     COEFFICIENTS FOR SIMPLE REAL POLES
C
   35 DO 37 J=1,NRS
      K=NZ+J
      CALL EVAL(TERM,LTE,VAL(1,J+1),0.0,FUN(I,K),FUN(I,K+1))
   37 FUN(I,K)=FUN(I,K)*EXC(IEX)/(VAL(1,J+1)**(IEX-1)*VAL(3,J+1))
   40 IF(NRD)50,50,42
C
C     COEFFICIENTS FOR DOUBLE REAL ROOTS-IN FORM(C(J)+D(J)*TIME)*EXPON
C     C(J) IN FUN(I,NZ+NRS+J),D(J) IN NEXT ONE (J=1,3,--(NRD-1)
C
   42 LT=LTE-1
      DO 44 J=1,LT
      FAC=LTE-J
   44 TREM(J)=TERM(J)*FAC
      DO 48 J=1,NRD,2
      L=1+NRS+J
      LL=NZ+NRS+J
      CALL EVAL(TERM,LTE,VAL(1,L),0.0,FUNC,TEMP)
      CALL EVAL(TREM,LT,   VAL(1,L),0.0,DERIV,TEMP)
      FUN(I,LL+1)=2.0*FUNC*EXC(IEX)/(VAL(3,L)*VAL(1,L)**(IEX-1))
   48 FUN(I,LL)=EXC(IEX)*(2.0*VAL(3,L)*(DERIV-COUNT*FUNC/VAL(1,L))-2.0*
     1FUNC*VAL(5,L)/3.0)/(VAL(1,L)**(IEX-1)*VAL(3,L)**2)
   50 NPCS=NPI+NCS
      IF(NPCS)75,75,52
C
C     COEFFICIENTS FOR IMAGINARY AND COMPLEX ROOTS-AS MAGNITUDE AND
C     PHASE ANGLE.  MAG IN FUN(I,NZ+NRS+NRD+J),ANGLE IN NEXT ONE
C     J=1,3,5,-- NPI+NCS-1
C
   52 DO 60 J=1,NPCS,2
      L=1+NRS+NRD+J
      LL=L-1+NZ
      CALL EVAL(TERM,LTE,VAL(1,L),VAL(2,L),FUN(I,LL),FUN(I,LL+1))
      FAC=VAL(3,L)**2+VAL(4,L)**2
      FUNC=(FUN(I,LL)*VAL(3,L)+FUN(I,LL+1)*VAL(4,L))/FAC
      DERIV=(FUN(I,LL+1)*VAL(3,L)-FUN(I,LL)*VAL(4,L))/FAC
      FAC=SQRTF(VAL(1,L)**2+VAL(2,L)**2)
      FUN(I,LL)=(SQRTF(FUNC**2+DERIV**2)/FAC**(IEX-1))*2.0*EXC(IEX)
      IF(FUNC)53,54,53
   53 FAC=ATANF(DERIV/FUNC)+1.5707963-SIGNF(1.5707963,FUNC)
      GO TO 55
   54 FAC=SIGNF(1.5707963,DERIV)
   55 IF(VAL(1,L))56,57,56
   56 FUNC=ATANF(VAL(2,L)/VAL(1,L))+1.5707963-SIGNF(1.5707963,VAL(1,L))
      GO TO 60
   57 FUNC=1.5707963
   60 FUN(I,LL+1)=FAC-COUNT*FUNC
C
C     CALCULATE RESPONSE FOR EXCITING VARIABLE I  AND TIME(J)
C                          (IN RESP(I,J))
C
   75 DO 62 J=1,101
   62 RESP(I,J)=0.0
      DO 100 J=1,101
      IF(NZ)66,66,63
   63 RESP(I,J) =FUN(I,1)
```

```
      IF(NZ-1)66,66,64
   64 DO 65 K=2,NZ
   65 RESP(I,J)=RESP(I,J)+FUN(I,K)*TIME(J)**(K-1)
   66 IF(NRS)74,74,70
   70 DO 72 K=1,NRS
      L=NZ+K
   72 RESP(I,J)=RESP(I,J)+FUN(I,L)*EXPF(VAL(1,K+1)*TIME(J))
   74 IF(NRD)82,82,76
   76 DO 80 K=1,NRD,2
      L=NZ+NRS+K
      LL=1+NRS+K
   80 RESP(I,J)=RESP(I,J)+(FUN(I,L)+FUN(I,L+1)*TIME(J))*EXPF(VAL(1,LL)
     1*TIME(J))
   82 IF(NPI)91,91,85
   85 DO 90 K=1,NPI,2
      LL=1+NRS+NRD+K
      L=LL-1+NZ
   90 RESP(I,J)=RESP(I,J)+FUN(I,L)*COSF(VAL(2,LL)*TIME(J)+FUN(I,L+1))
   91 IF(NCS)97,97,92
   92 DO 95 K=1,NCS,2
      LL=1+NRS+NRD+NPI+K
      L=LL-1+NZ
   95 RESP(I,J)=RESP(I,J)+FUN(I,L)*EXPF(VAL(1,LL)*TIME(J))*COSF
     1 (VAL(2,LL)*TIME(J)+FUN(I,L+1))
   97 XMIN=MIN1F(XMIN,RESP(I,J))
  100 XMAX= MAX1F (XMAX,RESP(I,J))
C
C     PRINT TABLE OF TRANSIENT RESPONSE
C
      WRITE OUTPUT TAPE 6,1,NOP,IPAGE,STAT(IEX),EXC(IEX),I,IREF
      IPAGE=IPAGE+1
      CALL ALPH(AL,NPAR)
      WRITE OUTPUT TAPE 6,2
      LE=2
      IF(NZ)112,112,102
  102 WRITE OUTPUT TAPE 6,3,FUN(I,1),TIME(1),RESP(I,1),TIME(52),
     1RESP(I,52)
      IF (NZ-2)  112,104,104
  104 WRITE OUTPUT TAPE 6,4,FUN(I,2),TIME(2),RESP(I,2),TIME(53),
     1RESP(I,53)
      IF (NZ-3)  112,106,106
  106 DO  110  J=3,NZ
      K=J-1
  110 WRITE OUTPUT TAPE 6,5,FUN(I,J),K,TIME(J),RESP(I,J),TIME(J+51),
     1RESP(I,J+51)
  112 LT=NZ+1
      IF(NRS)118,118,114
  114 DO 116 J=1,NRS
      WRITE OUTPUT TAPE 6,6,FUN(I,LT),VAL(1,LE),TIME(LT),RESP(I,LT),
     1TIME(LT+51),RESP(I,LT+51)
      LT=LT+1
  116 LE=LE+1
  118 IF (NRD) 126,126,120
  120 DO  124  J=1,NRD,2
      WRITE  OUTPUT TAPE 6,6,FUN(I,LT),VAL(1,LE),TIME(LT),RESP(I,LT),
     1TIME(LT+51),RESP(I,LT+51)
      WRITE OUTPUT TAPE 6,7,FUN(I,LT+1),VAL(1,LE),TIME(LT+1),RESP(I,LT+
     11),TIME(LT+52),RESP(I,LT+52)
      LT=LT+2
  124 LE=LE+2
```

```
 126 LJ=LT
     IF (NPI) 132,132,128
 128 DO 130  J=1,NPI,2
     WRITE OUTPUT TAPE 6,8,FUN(I,LT),VAL(2,LE),FUN(I,LT+1),TIME(LJ),
    1RESP(I,LJ),TIME(LJ+51),RESP(I,LJ+51)
     LT=LT+2
     LJ=LJ+1
 130 LE=LE+2
 132 IF (NCS) 138,138,134
 134 DO 136  J=1,NCS,2
     WRITE OUTPUT TAPE 6,9,FUN(I,LT),VAL(1,LE),TIME(LJ),RESP(I,LJ),
    1TIME(LJ+51),RESP(I,LJ+51)
     WRITE OUTPUT TAPE 6,10,VAL(2,LE+1),FUN(I,LT+1),TIME(LJ+1),RESP(I,
    1LJ+1),TIME(LJ+52),RESP(I,LJ+52)
     LT=LT+2
     LJ=LJ+2
 136 LE=LE+2
 138 DO 140 J=LJ,50
 140 WRITE OUTPUT TAPE 6,11,TIME(J),RESP(I,J),TIME(J+51),RESP(I,J+51)
 250 WRITE OUTPUT TAPE 6,11, TIME(51),RESP(I,51)
C
C     PRINT PLOT OF TRANSIENT RESPONSE
C
     WRITE OUTPUT TAPE 6,12,NOP,IPAGE,STAT(IEX),EXC(IEX),IREF
     IPAGE=IPAGE+1
     CALL ALPH(AL,NPAR)
C
C     CALCULATE LIMITS FOR Y CO-ORDINATE
C
     P=MAX1F(ABSF(XMAX),ABSF(XMIN))
     M=0
     IF (P-   10.0**M) 251,255,253
 251 M=M-1
     IF (P-   10.0**M) 251,255,255
 253 M=M+1
     IF (P-   10.0**M) 254,255,253
 254 M=M-1
 255 NMAR = XMAX *10.0**(1-M)
     IMAX = 0
     IF (NMAR) 257,258,256
 256 NMAR = NMAR+1
     IMAX=1
     GO TO 258
 257 IMAX =-1
 258 NMIN = XMIN* 10.0**(1-M)
     IMIN = 0
     IF (NMIN) 259,261,260
 259 NMIN =NMIN-1
     IMIN =-1
     GO TO 261
 260 IMIN = 1
 261 INT = 1
     MAX=NMAR
     MIN=NMIN
 262 IF (MAX-MIN-5) 254,265,263
 263 IF (MAX-MIN-10) 266,266,264
 264 MAX = (NMAR+IMAX*INT)/(INT+1)
     MIN = (NMIN+IMIN*INT)/(INT+1)
     INT = INT+1
     GO TO 262
```

```
  265 NSBH = 10
      GO TO 267
  266 NSBH = 5
  267 NHL=MAX-MIN
      XMAX= MAX* INT
      XMIN = MIN * INT
      XMAX = XMAX * 10.0**(M-1)
      XMIN = XMIN * 10.0**(M-1)
C
C     SET UP FOR PLOT 1
C
      NSCALE(1)=1
      NSCALE(2)=1-M
      NSCALE(3)=-1
      NSCALE(4)=0
      REM=1.000001*T
      IF(MODF(REM,1.0)-1.0E-5*REM)268,268,270
  268 REM=REM/10.0
      NSCALE(4)=NSCALE(4)-1
      IF(MODF(REM,1.0)-1.0E-5*REM)268,268,274
  270 REM=REM*10.0
      NSCALE(4)=NSCALE(4)+1
      IF(MODF(REM,1.0)-1.0E-5*REM)272,272,270
  272 NSCALE(4)=NSCALE(4)-1
  274 NSCALE(5)=-1
      CALL PLOT1(NSCALE,NHL,NSBH,10,10)
      CALL PLOT2 (GRAPH,TIME(101),0.0,XMAX,XMIN)
      DO 280 I=1,NVAR
      DO 273 J=1,101
  273 YCO(J) = RESP(I,J)
  280 CALL PLOT3 (STAT(I+3),TIME,YCO,101)
      CALL FPLOT4 (0,1HA)
      WRITE OUTPUT TAPE 6,13,NSCALE(4),NSCALE(2)
  300 CONTINUE
      RETURN
      END(2,0,1,0,0)
```

```
C     MULLER, OBTAINS ROOTS OF A POLYNOMIAL      KATZ
C
C     THIS SUBROUTINE WAS WRITTEN BY LOCKHEED, MISSILES AND SPACE
C     DIVISION, SUNNYVALE CALIFORNIA.  IT HAS BEEN MODIFIED SLIGHTLY
C     AT SAVANNAH RIVER LABORATORY.
C
      SUBROUTINE MULLER(COE,N1,ROOTR,ROOTI)
      DIMENSION COE(1),ROOTR(1),ROOTI(1)
      N2=N1+1
      N4=0
      I=N1+1
   19 IF(COE(I))9,7,9
    7 N4=N4+1
      ROOTR(N4)=0.
      ROOTI(N4)=0.
      I=I-1
      IF(N4-N1)19,37,19
    9 CONTINUE
   10 AXR=0.8
```

```
      AXI=0.
      PRINT 1001
      L=1
      N3=1
      ALP1R=AXR
      ALP1I=AXI
      M=1
      GOTO99
   11 BET1R=TEMR
      BET1I=TEMI
      AXR=0.85
      ALP2R=AXR
      ALP2I=AXI
      M=2
      GOTO99
   12 BET2R=TEMR
      BET2I=TEMI
      AXR=0.9
      ALP3R=AXR
      ALP3I=AXI
      M=3
      GOTO99
   13 BET3R=TEMR
      BET3I=TEMI
   14 TE1=ALP1R-ALP3R
      TE2=ALP1I-ALP3I
      TE5=ALP3R-ALP2R
      TE6=ALP3I-ALP2I
      TEM=TE5*TE5+TE6*TE6
      TE3=(TE1*TE5+TE2*TE6)/TEM
      TE4=(TE2*TE5-TE1*TE6)/TEM
      TE7=TE3+1.
      TE9=TE3*TE3-TE4*TE4
      TE10=2.*TE3*TE4
      DE15=TE7*BET3R-TE4*BET3I
      DE16=TE7*BET3I+TE4*BET3R
      TE11=TE3*BET2R-TE4*BET2I+BET1R-DE15
      TE12=TE3*BET2I+TE4*BET2R+BET1I-DE16
      TE7=TE9+1.
      TE1=TE9*BET2R-TE10*BET2I
      TE2=TE9*BET2I+TE10*BET2R
      TE13=TE1-BET1R-TE7*BET3R+TE10*BET3I
      TE14=TE2-BET1I-TE7*BET3I-TE10*BET3R
      TE15=DE15*TE3-DE16*TE4
      TE16=DE15*TE4+DE16*TE3
      TE1=TE13*TE13-TE14*TE14-4.*(TE11*TE15-TE12*TE16)
      TE2=2.*TE13*TE14-4.*(TE12*TE15+TE11*TE16)
      TEM=SQRTF(TE1*TE1+TE2*TE2)
      IF(TE1)113,113,112
  113 TE4=SQRTF(.5*(TEM-TE1))
      TE3=.5*TE2/TE4
      GO TO 111
  112 TE3=SQRTF(.5*(TEM+TE1))
      IF(TE2)110,200,200
  110 TE3=-TE3
  200 TE4=.5*TE2/TE3
  111 TE7=TE13+TE3
      TE8=TE14+TE4
      TE9=TE13-TE3
      TE10=TE14-TE4
```

```
      TE1=2.*TE15
      TE2=2.*TE16
      IF(TE7*TE7+TE8*TE8-TE9*TE9-TE10*TE10)204,204,205
  204 TE7=TE9
      TE8=TE10
  205 TEM=TE7*TE7+TE8*TE8
      TE3=(TE1*TE7+TE2*TE8)/TEM
      TE4=(TE2*TE7-TE1*TE8)/TEM
      AXR=ALP3R+TE3*TE5-TE4*TE6
      AXI=ALP3I+TE3*TE6+TE4*TE5
      ALP4R=AXR
      ALP4I=AXI
      M=4
      GO TO 99
   15 N6=1
   38 IF(ABSF(HELL)+ABSF(BELL)-1.E-20)18,18,16
   16 TE7=ABSF(ALP3R-AXR)+ABSF(ALP3I-AXI)
      IF(TE7/(ABSF(AXR)+ABSF(AXI))-1.E-7)18,18,17
   17 N3=N3+1
      ALP1R=ALP2R
      ALP1I=ALP2I
      ALP2R=ALP3R
      ALP2I=ALP3I
      ALP3R=ALP4R
      ALP3I=ALP4I
      BET1R=BET2R
      BET1I=BET2I
      BET2R=BET3R
      BET2I=BET3I
      BET3R=TEMR
      BET3I=TEMI
      IF(N3-100)14,18,18
   18 N4=N4+1
      ROOTR(N4)=ALP4R
      ROOTI(N4)=ALP4I
      N3=0
C
C     MODIFICATION - IMAGINARY PART OF ROOT SET = 0.0  IF IT IS VERY
                     MUCH SMALLER THAN THE REAL PART OF  THE ROOT
      IF(ABSF(ROOTI(N4))-ABSF(ROOTR(N4))*1.0E-4)300,300,41
  300 ROOTI(N4)=0.0
   41 IF(N4-N1)30,37,37
   37 RETURN
   30 IF(ROOTI(N4)) 31,10,31
   31 GO TO(32,10),L
   32 AXR=ALP1R
      AXI=-ALP1I
      ALP1I=-ALP1I
      M=5
      GO TO 99
   33 BET1R=TEMR
      BET1I=TEMI
      AXR=ALP2R
      AXI=-ALP2I
      ALP2I=-ALP2I
      M=6
      GO TO 99
   34 BET2R=TEMR
      BET2I=TEMI
      AXR=ALP3R
```

```
    AXI=-ALP3I
    ALP3I=-ALP3I
    L=2
    M=3
 99 TEMR=COE(1)
    TEMI=0.0
    DO100I=1,N1
    TE1=TEMR*AXR-TEMI*AXI
    TEMI=TEMI*AXR+TEMR*AXI
100 TEMR=      TE1+COE(I+1)
    HELL=TEMR
    BELL=TEMI
 42 IF(N4)102,103,102
102 DO101I=1,N4
    TEM1=AXR-ROOTR(I)
    TEM2=AXI-ROOTI(I)
    TE1=TEM1*TEM1+TEM2*TEM2
    TE2=(TEMR*TEM1+TEMI*TEM2)/TE1
    TEMI=(TEMI*TEM1-TEMR*TEM2)/TE1
101 TEMR=TE2
103 GO TO(11,12,13,15,33,34),M
    END(2,0,1,0,1)
```

APPENDIX II NU LUCY Loader

At the Savannah River Laboratory, the LASS programs have been loaded onto tape by using the NU LUCY loader(21), a 22-card loader that will read binary program cards into core and, with proper sense switch setting, write the program onto Logical Unit 1 as a self-loading program. For LASS 2 and LASS 3, one binary correction card was added to the standard NU LUCY deck. This corrected loader writes programs on, and reads from Logical Unit 2; furthermore, the rewind feature(21) after reading the program from tape is eliminated without requiring sense switch settings. The correction card is made up as follows:

Word	Prefix	Decrement	Tag	Address
9 Left	-	-	-	-
9 Right	-	-	-	-
8 Left	-	-	-	77726_8
8 Right	-	76200_8	-	222_8
7 Left	-	-	-	77614_8
7 Left	-	77200_8	-	202_8
6 Left	-	-	-	77557_8
6 Right	-	76600_8	-	222_8
5 Left	-	-	-	77553_8
5 Right	-	76600_8	-	222_8
4 Left	-	-	-	77526_8
4 Right	-	77200_8	-	202_8
3 Left	-	-	-	77124_8
3 Right	-	77200_8	-	202_8
2 Left	-	-	-	77121_8
2 Right	-	76400_8	-	202_8
1 Left	-	-	-	77120_8
1 Right	-	76400_8	-	202_8
0 Left	-	-	-	77123_8
0 Right	-	2000_8	-	77125_8

BIBLIOGRAPHY

1. Bode, H. W. Network Analysis and Feedback Amplifier Design. Princeton: D. Van Nostrand Co. (1945).

2. Brown, G. S. and D. P. Campbell. Principles of Servomechanisms. New York: John Wiley & Sons (1948).

3. Chestnut, H. and R. W. Mayer. Servomechanisms and Regulating System Design. Volumes I and II, 2d ed., New York: John Wiley & Sons (1959).

4. D'Azzo, J. J. and C. H. Houpis. Feedback Control System Analysis and Synthesis. New York: McGraw-Hill (1960).

5. Fett, G. H. Feedback Control Systems. New York: Prentice-Hall (1954).

6. Grabbe, E. M., S. Ramo, and D. E. Wooldridge. Handbook of Automation, Computation, and Control. Volume I, Control Fundamentals, New York: John Wiley & Sons (1958).

7. Thaler, G. J. and R. G. Brown. Analysis and Design of Feedback Control Systems. New York: McGraw-Hill (1960).

8. Truxall, J. G. Automatic Feedback Control System Synthesis. New York: McGraw-Hill (1955).

9. Harrer, J. M. Nuclear Reactor Control Engineering. Princeton: D. Van Nostrand Co. (1963).

10. Proceedings of the Conference on Transfer Function Measurements and Reactor Stability Analysis Held at Argonne National Laboratory, May 2-3, 1960, USAEC Report ANL-6205.

11. Ryan, J. W. Generalized 704 Program for Linear System Analysis - EA Problem No. 7-0040, Job No. 51011, ANP Program No. 275. General Electric Co., Aircraft Nuclear Propulsion Dept., Cincinnati, Report XDC 60-1-124 (1959).

12. Dight, D. G. and J. A. Warrington. GALS - Generalized Analysis of Linear Systems. Knolls Atomic Power Lab. Schenectady, N. Y. USAEC Report KAPL-M-EC-6 (1961).

13. Dick, J., A General Program for Systems Evaluation (FORTRAN II), October 24, 1961, SHARE Distribution 1244.

14. Lawrence Radiation Laboratory, personal communication.

15. Stephens, G. L., K. F. Grigg, and M. L. Pope. Transfer Function Programs - Mark I and Mark II for Solving Generalized Eigenvalue Problems. General Electric Co., Cincinnati, Ohio, Report TIS R59FPD911 (1960).

16. Ojalvo, M., NA-299 Determinant Evaluation and Root Extraction, September 17, 1958, SHARE Distribution 514.

17. Barlow, D. "Designing Control Systems Automatically". Control Engineering 9, No. 5, 30-31 (1962).

18. M'Pherson, P. K. The Use of Complex Plane Methods in the Analysis of Plant Dynamics and the Design of Automatic Control Systems. United Kingdom Atomic Energy Authority, Winfrith, Dorset, England, Report AEEW-R-120 (1961).

19. Cummins, J. D. A General Digital Computer Procedure for Synthesizing Linear Automatic Control Systems. United Kingdom Atomic Energy Authority, Winfrith, Dorset, England, Report AAEW-R-156 (1961).

20. Guppy, C. B. Computer Programme for the Derivation of Transfer Functions for Multivariable Systems (Solutions of Determinants with Polynomial Elements). United Kingdom Atomic Energy Authority, Winfrith, Dorset, England, Report AEEW-R-189 (1962).

21. Sadowsky, G. and N. Stutz. NU LUCY Extended Binary Symbolic Subroutine Loader, April 1960, SHARE Distribution 902.

22. Gardner, M. F. and J. L. Barnes. Transients in Linear Systems. Volume I, pp 197-201, New York: John Wiley & Sons (1942).

23. Carnahan, B. and L. Evans. UM PLOT Plotting Subroutine, March 1, 1961, SHARE Distribution 1085.

24. Hanson, I. ML HPRS Polynomial Root Finder Subroutine, June 1960, SHARE Distribution 1124.

25. Churchill, R. V. Operational Mathematics. 2d ed. New York: McGraw-Hill (1958).

26. Arnett, L. M., et al. Final Hazards Evaluation of the Heavy Water Components Test Reactor (HWCTR). E. I. du Pont de Nemours & Co., Savannah River Laboratory, Aiken, S. C. USAEC Report DP-600 (1962).

INTERNAL DISTRIBUTION

P. J. Hagelston (3)	SROO, Aiken, S. C.
H. F. Carroll	Declassification Branch USAEC, Oak Ridge, Tenn.
L. Squires - M. H. Wahl	Wilmington AED
S. A. McNeight	"
M. H. Smith - W. H. Holstein - J. B. Tinker	"
L. C. Evans - C. W. J. Wende - J. W. Croach	"
W File	"
A. D. Duff, Jr.	Engineering Department
P. S. Buckley	"
J. A. Monier - F. H. Endorf	Savannah River Plant
W. P. Bebbington - F. E. Kruesi	"
W. P. Overbeck - A. A. Johnson	Savannah River Laboratory
J. O. Morrison	"
J. E. Beach	"
J. W. Morris	"
E. C. Nelson	"
G. Dessauer	"
J. L. Crandall - M. H. Goosey	"
L. M. Arnett	"
J. J. Bugg	"
B. B. Occhipinti	"
D. Randall	"
J. W. Wade	"
TIS File Record Copy	"