UNCLASSIFIED

AD NUMBER
AD871086
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution: Further dissemination only as directed by Army Materiel Command, Washington, DC 20315, MAR 1970, or higher DoD authority.**
AUTHORITY
AMC ltr 31 Aug 1971

THIS PAGE IS UNCLASSIFIED

AMSAA TM 69

AD871086

AD871086L

AD

AMSAA

TECHNICAL MEMORANDUM NO. 69

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS

by

Harold H. Burke
Robert L. Payne, Jr.

March 1970

U.S. ARMY ABERDEEN RESEARCH AND DEVELOPMENT CENTER
ARMY MATERIEL SYSTEMS ANALYSIS AGENCY
ABERDEEN PROVING GROUND, MARYLAND

DISPOSITION

Destroy this report when no longer needed. Do not return it to the originator.

DISCLAIMER

The findings in this report are not to be construed as an official Department of the Army position.

WARNING

Information and data contained in this document are based on the input available at the time of preparation. The results may be subject to change and should not be construed as representing the AMC position unless so specified.

A R M Y   M A T E R I E L   S Y S T E M S   A N A L Y S I S   A G E N C Y

TECHNICAL MEMORANDUM NO. 69

MARCH 1970

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS

Harold H. Burke

Robert L. Payne, Jr.

Combat Support Division

Project No. RDT&E 1P765801M1102

A B E R D E E N   P R O V I N G   G R O U N D,   M A R Y L A N D

ARMY MATERIEL SYSTEMS ANALYSIS AGENCY

TECHNICAL MEMORANDUM NO. 69

HHBurke/RLPayne, Jr./flz
Aberdeen Proving Ground, Md.
March 1970

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS

ABSTRACT

A computer program to perform dynamic systems analyses and plot the results is presented. Linear systems and a large classification of nonlinear systems representing engineering, scientific, and economic disciplines can be modeled to permit application of the computer program. Two examples are given to demonstrate the capabilities of the analysis tool. The mathematical model of a missile guidance and control system is analyzed and a ratio of polynomials representing the closed loop transfer function of a high performance model follower aircraft is evaluated. Linear differential equations to the 100th order having real or complex roots can be studied. System characteristic equation root loci and system transfer functions are plotted.

Next page is blank.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Next page is blank.

1. INTRODUCTION

Systems analyses require that mathematical models be developed and exercised to determine cause-and-effect relationships. One powerful method is to generate linearized equations for each physical phenomenon occurring in the process and combine these into input-output relationships through the application of Laplace transformations.

Studying the characteristic equations of the input-output relations determines parametric stability trends.(1) Steady state frequency response of the system provides another useful characterization of system stability. Application of the inverse Laplace transformation to the process transfer functions provides transient histories of output for specific inputs.(2)

This report discusses a program that mechanizes another part of the analytical procedure, the determination of the frequency response of linear systems and the modified phase-amplitude characteristic for nonlinear systems. Reference 1, published in November 1968, discusses a program that mechanized the determination of the roots of the characteristic equation. Reference 2, discusses a program that determines the transient response of the process when it is subjected to input forcing functions.

Application of this computer program, which is an extension of the material contained in Reference 1, along with the computer program presented in Reference 2, permits one to determine quickly and accurately the stability and performance characteristics of high-order linear dynamic processes and to qualitatively predict the stability and performance of a large class of nonlinear systems.

The techniques developed in this report have been applied to specific studies.(3) (4) A more extensive utilization of the methods will be contained in a forthcoming report which expands Chapter 4.2 of Reference 5. Preliminary work has been commenced to explore the utility of these methods in providing solutions to optimal inventory problems.(6)(7)

A useful linear system analysis method is to determine the steady state frequency response of the system. Evaluation of the polynomials representing the system's transfer function is a time-consuming task. Conversion of the tabulated results to a graphical display is then required before conclusions relevant to system stability can be appreciated.

The analysis of nonlinear systems is a difficult task, but for certain classes of systems it is possible to obtain qualitative indications of stability. Similarly, the task of obtaining the tabular data and converting it to a graphical form before conclusions can be drawn is a laborious process.

These two methods (one linear and one nonlinear) have been coded in a FORTRAN IV program and integrated into the root locus methods discussed in Reference 1. Main features of the program are:

1. FORTRAN IV program. No machine-oriented or object language.
2. No FORTRAN complex type statements necessary.
3. No special graphical plotting equipment.
4. Order of polynomial may be up to 100.
5. Number of variations of given parameter may be up to 100.

Main features of the graphical displays are:

ROOT LOCUS Method[1]

1. Log plot of the third and fourth quadrants of the complex frequency plane from 0 to 10,000 radians/second.

2. Linear plots of selected regions of the third and fourth quadrants of the complex frequency plane with arbitrary scales.

LINEAR FREQUENCY RESPONSE Method

1. Plot of Magnitude (in decibels) and phase (in degrees) of transfer function vs 6 decades of frequency, beginning at a selectable minimum frequency.

2. Linear frequency plots of magnitude (in decibels) and phase (in degrees) of transfer function for selectable ranges of frequency.

NONLINEAR FREQUENCY RESPONSE Method

1. Plot of "Modified Magnitude Characteristic" (in decibels) and "Modified Phase Characteristic" (in degrees) of transfer function vs 6 decades of frequency, beginning at same selectable minimum frequency as linear frequency response method.

2. Linear frequency plots of "Modified Magnitude Characteristics" (in decibels) and "Modified Phase Characteristic" (in degrees) of transfer function for same selectable ranges of frequency as linear frequency response methods.

2. THE PROBLEM

2.1 Linear System.

Regardless of the complexity of a linear closed-loop system, its transfer function can be reduced to the equivalent form shown in Figure 1. For multiple loop systems, the G's and H's are readily expressed as sums of products of polynomials which are identified with individual elements making up the complete system.

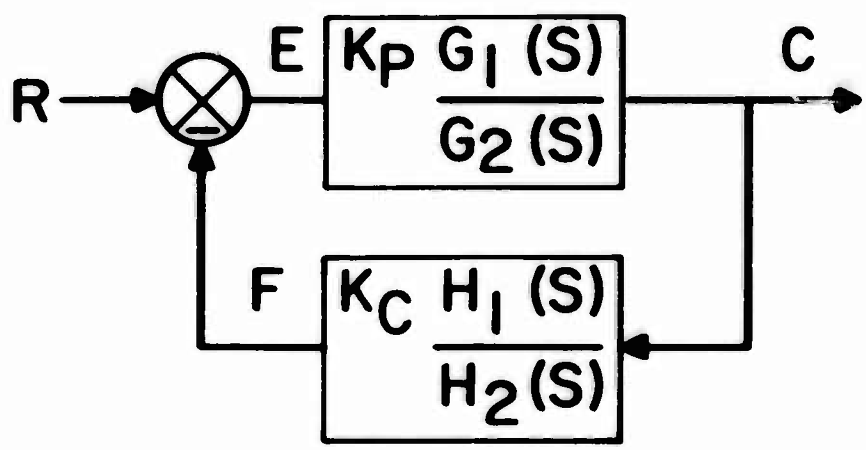


Figure 1. Linear Closed Loop System.

R = system input,
C = system output,
F = system feedback,
E = system error,
K_p = process gain, and
K_c = controller gain.

The fractions $G_1(s)/G_2(s)$ and $H_1(s)/H_2(s)$ are equivalent transfer fractions of the system and are represented by ratios of polynomials which upon expansion can be put into the form

$$\frac{\sum_{\mu=o}^{n-k} a_\mu \, s^{n-\mu}}{\sum_{i=o}^{m+n-\ell} b_i \, s^{m+n-i}} , \qquad (1)$$

or in factored form

$$\frac{s^k \prod_{\mu=1}^{v} (s+z_\mu) \prod_{q=v+1}^{y} \left[s+(\sigma_q+j\omega_q)\right]\left[s+(\sigma_q-j\omega_q)\right]}{s^\ell \prod_{i=1}^{g} (s+p_i) \prod_{t=g+1}^{r} \left[s+(\sigma_t+j\omega_t)\right]\left[s+(\sigma_t-j\omega_t)\right]} \qquad (2)$$

where $y = \frac{n-k-v}{2}$, $r = \frac{m+n-\ell-g}{2}$

where the a's and b's are real, the poles and zeros can be either real or complex in conjugate pairs, and $m \geq 1$.

There are two transfer functions of fundamental importance.

OPEN LOOP TRANSFER FUNCTION:

$$\frac{F}{E} = \left[K_p \frac{G_1(s)}{G_2(s)}\right] \left[K_c \frac{H_1(s)}{H_2(s)}\right] . \qquad (3)$$

CLOSED LOOP TRANSFER FUNCTION:

$$\frac{C}{R} = \frac{K_p G_1(s) H_2(s)}{K_c K_p G_1(s) H_1(s) + G_2(s) H_2(s)} . \qquad (4)$$

The linear system's closed-loop characteristic equation, which determines stability is

$$K_c K_p G_1(s)H_1(s) + G_2(s)H_2(s) = 0. \qquad (5)$$

Dividing both sides of Equation (5) by $G_2(s)H_2(s)$ gives

$$K_c K_p \frac{G_1(s)H_1(s)}{G_2(s)H_2(s)} + 1 = 0 \qquad (6)$$

If we let

$$K_c K_p = K^* , \qquad (7)$$

$$G_1(s)H_1(s) = A(s), \text{ and} \qquad (8)$$

$$G_2(s)H_2(s) = B(s), \text{ then Equation (6) becomes} \qquad (9)$$

$$\frac{K^* A(s)}{B(s)} + 1 = 0. \qquad (10)$$

In terms of the system open-loop transfer function,

$$\frac{F}{E} + 1 = 0. \qquad (11)$$

This expression relates the system's closed-loop characteristic equation to the system's open-loop transfer function.

The steady state frequency response of the open-loop transfer is obtained by evaluating Equation (3) along the imaginary axis for values of ω between zero and infinity rad/s.

The result is a vector quantity,

$$\left.\frac{F}{E}\right|_{0 \leq \omega \leq \infty} = \left. K^* \frac{A(s)}{B(s)}\right|_{0 \leq s \leq \infty} = M_{oL} \angle \phi_{oL} , \qquad (12)$$

where

M_{oL} = scalar magnitude of open-loop transfer function between $0 \leq \omega \leq \infty$, and

ϕ_{oL} = direction of magnitude of open-loop transfer function between $0 \leq \omega \leq \infty$.

Equation (10) can be rewritten as

$$M_{oL} \angle \phi_{oL} + 1 \angle 0^{o} = 0, \quad (13)$$

or

$$M_{oL} \angle \phi_{oL} = 1 \angle 180^{o}. \quad (14)$$

Equation 14 is the form required to provide quantitative stability information for a closed-loop system.

There are two figures of merit which provide a measure of system stability. The only restriction is that all roots of the system's open-loop transfer function denominator, B(s), be ≤ 0.

GAIN MARGIN is the amount that M_{oL} must be increased or decreased to make it equal to 1, only when the orientation of magnitude of the open-loop transfer function, $\angle \frac{A(s)}{B(s)}$, is 180^{o}.

PHASE MARGIN is the amount that ϕ_{oL} must be increased or decreased to make it equal to 180^{o}, only when the magnitude of the open-loop transfer function, $\left|\frac{A(s)}{B(s)}\right|$,is 1.

These concepts can be related directly to Equation 14. Incipient instability is identified with either a zero gain or zero phase margin.

The scalar magnitude of the open-loop transfer function is generally expressed in decibels. The decibel equivalent, M_{oL} (db), of a number, M_{oL}, is M_{oL} (db) = 20 $\log_{10} M_{oL}$.

The gain margin and phase margin of a linear closed-loop system as shown in Figure 1 and Equation (4) can be readily determined by analyzing the open-loop transfer functions of Equation (3) in accordance with Equation (14).

2.2 Nonlinear System.

V.M. Popov's plots of system transfer functions give some insight into stability of nonlinear systems.[8][9] Figure 2 shows the system configurations considered in this analysis. It contains a linear part and one nonlinear element and is subject to the following restrictions.

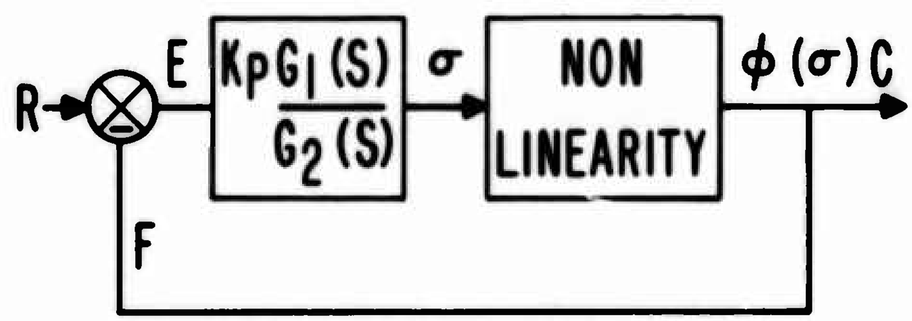


Figure 2. Nonlinear Closed Loop System

1. The roots of $G_2(s)$ have negative real parts and simple or multiple roots on the imaginary axis (one root permitted at origin).

2. $$\epsilon \leq \frac{\phi(\sigma)}{\sigma} \leq K. \tag{15}$$

3. The linear system of Figure 2, obtained by substituting $\phi=\epsilon\sigma$, is stable.

If there exists a non-negative real number q such that

$$R_e\left[(1+jq\omega)\,G(j\omega)\right] + \frac{1}{K} > 0 \qquad (16)$$

for all $\omega>0$, where G is defined by $\frac{G_1(s)}{G_2(s)}$, the nonlinear system is

asympotically stable in the large.

Since G has been defined as the ratio of two polynomials, it may be expressed as

$$G(j\omega) = \frac{G_{1R}(\omega) + jG_{1I}(\omega)}{G_{2R}(\omega) + jG_{2I}(\omega)}, \qquad (17)$$

where

G_{1R}, G_{2R} = real parts of G_1, and G_2, respectively, and

G_{1I}, G_{2I} = imaginary parts of G_1 and G_2, respectively.

Redefining $G(j\omega)$, we have

$$G(j\omega) = G_R(\omega) + jG_I(\omega), \qquad (18)$$

where

$$G_R = \frac{G_{1R}G_{2R} + G_{1I}\,G_{2I}}{(G_{2R})^2 + (G_{2I})^2}\text{, and}$$

$$G_I = \frac{G_{1I}\,G_{2R} - G_{1R}\,G_{2I}}{(G_{2R})^2 + (G_{2I})^2}.$$

As a result, the inequality,

$$R_e\left[(1 + jq\omega)\;G\;(j\omega)\right] + \frac{1}{K} > 0\;, \qquad (19)$$

becomes

$$G_R(\omega) - q\omega G_I(\omega) + \frac{1}{K} > 0. \qquad (20)$$

To aid in a graphical interpretation of these results, we define

$$U = G_R(\omega), \text{ and}$$
$$W = \omega G_I(\omega). \qquad (21)$$

The inequality of Equation (20) may be written as

$$U - q\,W + \frac{1}{K} > 0 \,,$$

where (22)

$$q \geq 0, \; K > 0 \,.$$

If we replace the inequality of Equation (22) by an equal sign,

$$U - q\,W + \frac{1}{K} = 0 \,. \qquad (23)$$

Then it may be seen that inequality of Equation (22) represents those points in the U-W plane of the modified phase amplitude characteristic which are to the right of the line of Equation 23, as shown in Figure 3.

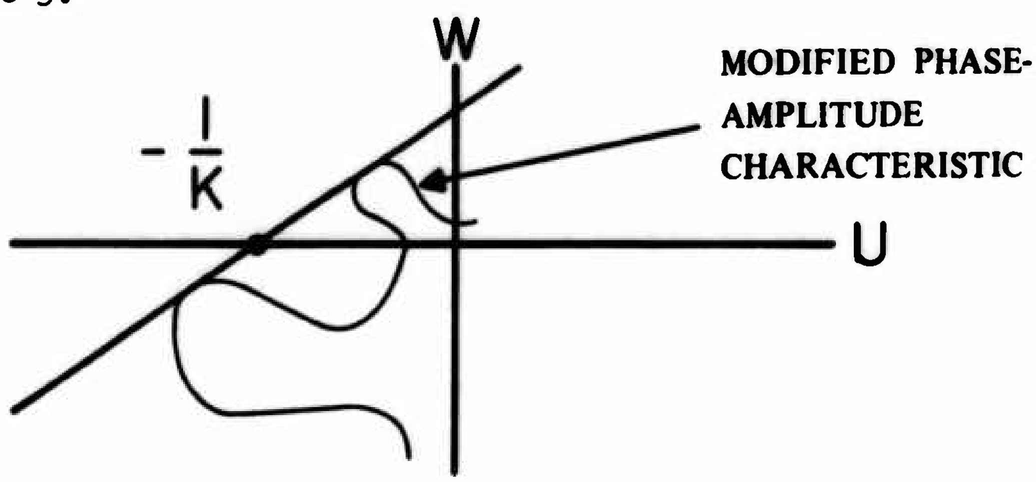


Figure 3. Stability Boundary-Popov Modified Phase Amplitude Characteristic.

The nonlinear system synthesis problem will be concerned with determining the variation of K for a given linear G(s). Figure 4 shows the boundary constraints of the nonlinearity $\phi(\sigma)/\sigma$ of the system shown in Figure 2.

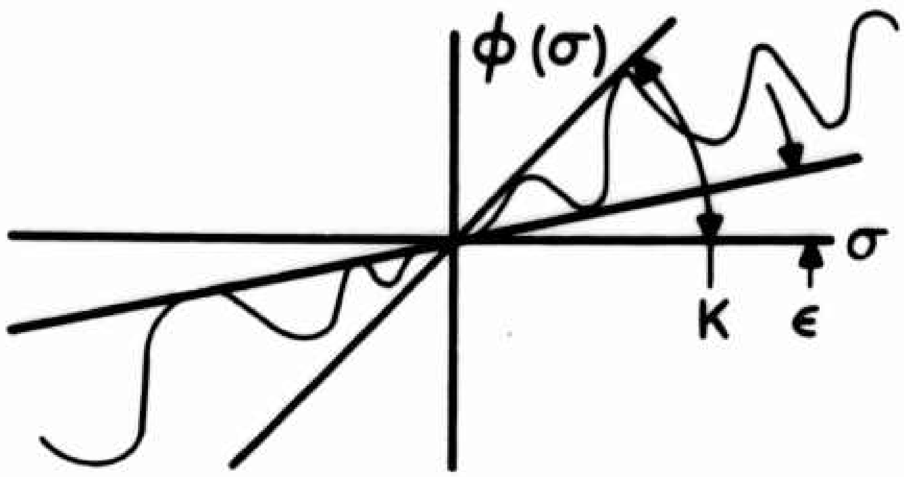


Figure 4. System Nonlinearity.

To maintain symmetry with the Bode Plots, the modified magnitude and phase characteristics are plotted on identical axes. Values of the modified amplitude associated with the modified phase characteristic for $90^o \leq \phi \leq 270^o$ (-90^o) establish the contour described in Figure 3.

Once this contour is established in the U-W plane, the maximum K obtainable is determined by inspection from the intersection of the U axis and a straight line tangent to the modified phase amplitude characteristic.

3. GRAPHICAL METHODS

An on-line printout of the Bode diagram for an open-loop transfer function, F/E, or for a closed-loop transfer function, C/R, is generated for the linear system. Similarly, printouts of the Popov "modified amplitude-phase characteristics" are generated. The phase and amplitude axes are scaled in degrees and decibels with an arbitrarily selectable scale between $\pm \phi$ max degrees $\pm$ M max db, respectively. The frequency axes are logarithmically scaled in radians per second over a range of 6 decades, beginning with an arbitrarily selected lowest decade. Values of amplitude and phase are calculated for 55 uniformly distributed frequencies over the 6-decade range, plus 17 additional points centered about each root of $\frac{A(s)}{K^*A(s) + B(s)}$. This additional distribution of printouts expressed as a multiple of each frequency is:

(0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 2.0, 4.0, 6.0, 8.0, 10).

An option of the program permits the density of the Bode plot to be either sparse or dense. The sparse plot requires K* = 0 and only one set of frequencies are available, from which the amplitude and phase of A(s)/B(s) are calculated. The dense plot permits K* to have values consistent with the root locus subroutine.

For each value of K*, a different set of frequencies will exist for K* A(s) + B(s).

A larger number of amplitude and phase values will be generated. Use of the dense plot capability may be necessary to obtain precise definition of the graphical plot in some cases.

Any number of simultaneous expands of different portions of the 6-decade frequency range may be plotted.

4. DATA FORMATS

4.1 General Description.

This is a subroutine designed to supplement the methods presented in AMSAA TM No. 21, A Linear Closed Loop System Analysis Procedure Using Line Printer Plots of Characteristic Equation Root Loci. This subroutine utilizes the polynomial multiplication and root locus methods discussed in AMSAA TM No. 21, and extends the analysis capabilities to include the following available options:

1. Root Locus Plots.
2. Frequency Response Plots.
3. Popov Modified Phase-Amplitude Characteristics Plots.

For the root locus option, the output is identical with the output obtained from AMSAA TM No. 21.

For the frequency-response option, the output is a log-log plot of steady state amplitude vs frequency and a semi-log plot of steady state phase angle vs frequency. Linear expands of amplitude and phase for any frequency range may be plotted.

For the Popov modified phase-amplitude characteristic, the output is a log-log plot of steady state amplitude vs frequency and semi-log plot of steady state phase angle vs frequency. Linear expands of amplitude and phase for any frequency range may be plotted.

4.2 Input Description.

a. Program option control card that precedes all data (see below).

b. Program option control card.

To run the program in its entirety with a dense plot in the frequency response, i.e.: Root Locus plus Expands and Frequency Response and Popov Plots plus expands place a one (1) in columns 10, 20, 30, and 40.

To run only the root locus log plot, place a one (1) in column 10 and zeros (0) in columns 20, 30, 40.

To run the root locus expand plot, place a one (1) in column 20 and zeros (0) in columns 10, 30, 40.

To run the sparse frequency response Popov plot only place a one (1) in column 30 and zeros in columns 10, 20, and 40.

To run the dense frequency response plot only, place a one (1) in columns 30 and 40 and zeros in columns 10 and 20.

To void all plot routines place a zero in columns 10, 20, 30, and 40.

c. Comment Card that follows the program option control card for identification purposes.

d. Program Data Cards.

Same format as described in Chapter IV (Data Formats), paragraph A2 of AMSAA TM No. 21. Control Card used to call All Plot Subroutines is same as control card used to call log plot subroutine of AMSAA TM No. 21 (Section IV, paragraph B2).

Control cards used to call root locus option are same as control cards used to call root locus expand option, paragraph B2, section IV of AMSAA TM No. 21. Table 1 demonstrates this input format for the dense frequency response option. Control cards used to call frequency response, Popov plots, and expands of both are additional features of this report.

DESCRIPTION	COLUMNS	DATA
Data appears only once	a. - 1-10	Smallest frequency, rad/s (real)
	b. - 11-20	Max phase angle, deg, (real)
	c. - 21-30	Max amplitude, DB (real)
Frequency response, Popov	a. - 1-10	Zero, if no expands desired (integer), One, if expands desired
Number of expands desired	a. - 1-10	No. of freq, response phase angle expands (integer)
	b. - 11-20	No. of freq response amplitude expands (integer)
	c. - 21-30	No. of Popov phase angle expands (integer)
	d. - 31-40	No. of Popov amplitude expands (integer)
Region of expand, one card for each expand in consecutive order with a,b,c,d, above.	1-10	Smallest freq, rad/s (real)
	11-20	Largest freq, rad/s (real)

4.3. Output Description.

Depending on the options specified, the output consists of:

Root Locus Plot Option

a. A list of all of the roots plotted on the complex frequency plane log plot.

b. Two log plots of the roots calculated by the polynomial multiplication and root locus program.

c. The specified number of linear expand plots of selectable regions in the complex frequency plane.

FREQUENCY RESPONSE/POPOV PLOT OPTION

a. A list of amplitude, amplitude in decibels, and phase angle in degrees for various frequencies specified in Section 3.

b. A log-log plot of amplitude vs frequency and a semi-log plot of phase angle vs frequency.

c. The specified number of linear expands of the frequency axis for the amplitude and phase angles of the specified transfer function.

5. EXAMPLES

The two examples discussed in AMSAA TM No. 21 will be analyzed, using the frequency response/Popov plot option. The root loci of A(s)/B(s) using K* as a parameter have previously been determined. The characteristic equation of the system shown in Figure 2 of Reference 1 is

$$K^*A(s) + B(s) = 0.$$

The open-loop transfer function for this expression, as given in Equation (12) is

$$\frac{F}{E} = \frac{K^*A_1(s)+A_2(s)+A_3(s)}{B(s)} \tag{24}$$

<u>EXAMPLE 1</u>.

where

$$K^* = -0.005$$

$$A_1(s) = 2092(S^2+1.2S-1672)(S^2+3.2S+25{,}600)+491S^2(S^2+6S+400)$$

$$A_2(s) = 0.081(464)(S+2.7)(S^2+3.2S+25{,}600)$$

$$A_3(s) = 0.95(464)(S+2.7)(S^2+3.2S+25{,}600)$$

$$B(s) = S(S^2+6S+400)(S^2+3.2S+25{,}600)$$

Table 1 shows the input card format for this open-loop transfer function.

Lines 2 through 24 apply to the polynomial multiplication and root locus methods and lines 26 through 36 apply to the frequency response and Popov condition plot subroutines.

The basic tabulated numerical output is shown in Tables 2 and 3 and is similar to Tables 2 and 3 of Reference 1. A tabulation of the open-loop frequency response gain and phase and Popov modified phase amplitude characteristics are shown in Table 4. Figures 5 and 6 are the Bode plots of F/E and Figures 7 and 8 are the Popov plots of F/E. Two regions are expanded to demonstrate capability of this program. First the regions of frequencies between 1.0 rad/s and 300 rad/s are shown in Figures 9 and 10 (frequency response), 11 and 12 (Popov response) and secondly the region of frequencies between 120 rad/s and 200 rad/s shown in Figures 13 and 14 (frequency response), 15 and 16 (Popov response) are plotted. Slope information consistent with Bode principles may be applied directly to these plots.

TABLE 1. INPUT FOR EXAMPLE 1

COLUMNS 1-10	11-20	21-30	31-40	Remarks
0	0	1	1	Program option control card
	Frequency Response			Comment card
0	4	1	504	Problem data cards (Polynomial multiplication and Root Locus program). Same format as AMSAA TM No. 21, Chapter IV, Paragraph A2.
0	1.x10^{-9}	1.x10^{-9}		
4	3	4	4	
3				
1	.005			
3	25600.	3.2	1.	
3	-1672.	1.2	1.	
1	-2092.			
1	.005			
3	0.	0.	-491.5	
3	400.	6.	1.	
1	.081			
2	0.	464.		
2	2.7	1.		
3	25600.	3.2	1.	
1	.95			
1	464.			
2	2.7	1.		
3	25600.	3.2	1.	
2	0.	1.		
3	400.	6.	1.	
3	25600.	3.2	1.	
0	0	0	10000	Same as paragraph B2, Chapter IV, TM No. 21. Control card used to call all plot subroutines.
.01	180.	100.		Frequency response and Popov response plot subroutine.
1				
2	2	2	2	
1.	300.			
120.	200.			
1.	300.			
120.	200.			
1.	300.			
120.	200.			
1.	300.			
120.	200.			

EXAMPLE 2

Table 6, and Figures 12 through 14 of Reference 1 are the A(s)/B(s) for a 24th order linear system. Table 5 gives the F/E data input for this system, and Table 6 lists open-loop transfer function. Figures 17 and 18 show the Bode plot and Figures 19 and 20 show the Popov plot. An expand plot of the region between 1 rad/s $\leq \omega \leq$ 200 rad/s is shown in Figures 21 and 22 for the frequency response and in Figures 23 and 24 for the Popov response.

These two examples demonstrate the significant systems analysis capability that can be realized by using the tools developed in this document and the related programs contained in References 1 and 2. Detailed conclusions can be realized when high order feedback control systems are encountered.

6. CONCLUSIONS

The method described in this memorandum for computing linear system root loci, steady state amplitude and phase, and the nonlinear system steady state modified phase-amplitude characteristic has the following merits:

1. Accepts input in the form of sums of products of polynomials.
2. Produces a log plot of the entire complex plane in two graphical displays.
3. Produces selectable scaled linear plots of regions in complex frequency plane.
4. Relates the closed-loop gain to the graphical display.
5. Produces a magnitude (db) vs log frequency steady state response plot.
6. Produces a phase (deg) vs log frequency steady state response plot.
7. Produces selectable scaled linear plots of magnitude and phase of the steady state frequency response.

Next page is blank.

TABLE 2. MIRROR OF INPUT, EXAMPLE 1.

```
POLYNOMIAL MULTIPLICATION AND ROOT LOCUS                                          PROBLEM NO. 504

 DELTAK=          0      POLY.ADDED IN A(S)=        4     POLY.ADDED IN B(S)=       1     PROB.NO.=       504

K-INITIAL=         .0000000000     INCREMENT K=         .0000000010     K-TERMINATE=        .0000000010
    NUMBER OF POLY. IN GROUP  1 OF NUMERATOR=        4
    NUMBER OF POLY. IN GROUP  2 OF NUMERATOR=        3
    NUMBER OF POLY. IN GROUP  3 OF NUMERATOR=        4
    NUMBER OF POLY. IN GROUP  4 OF NUMERATOR=        4
    NUMBER OF POLY. IN GROUP  1 OF DENOMINATOR=        3
    C(  1)=          .0050000000
    C(  1)=     25600.0000000000
    C(  2)=         3.2000000000
    C(  3)=         1.0000000000
    C(  1)=     -1672.0000000000
    C(  2)=         1.2000000000
    C(  3)=         1.0000000000
    C(  1)=     -2092.0000000000


    C(  1)=          .0050000000
    C(  1)=          .0000000000
    C(  2)=          .0000000000
    C(  3)=      -491.5000000000
    C(  1)=       400.0000000000
    C(  2)=         6.0000000000
    C(  3)=         1.0000000000


    C(  1)=          .0810000000
    C(  1)=          .0000000000
    C(  2)=       464.0000000000
    C(  1)=         2.7000000000
    C(  2)=         1.0000000000
    C(  1)=     25600.0000000000
    C(  2)=         3.2000000000
    C(  3)=         1.0000000000


    C(  1)=          .9500000000
    C(  1)=       464.0000000000
    C(  1)=         2.7000000000
    C(  2)=         1.0000000000
    C(  1)=     25600.0000000000
    C(  2)=         3.2000000000
    C(  3)=         1.0000000000


    C(  1)=          .0000000000
    C(  2)=         1.0000000000
    C(  1)=       400.0000000000
    C(  2)=         6.0000000000
    C(  3)=         1.0000000000
    C(  1)=     25600.0000000000
    C(  2)=         3.2000000000
    C(  3)=         1.0000000000
```

TABLE 3. EQUIVALENT SYSTEM OPEN LOOP POLYNOMIALS AND ROOTS, EXAMPLE 1.

```
COEFFICIENTS ARE GIVEN IN ASCENDING ORDER


   THE COEFFICIENTS OF POLYNOMIAL A (ORDER =   4)
 4.7818957E 08      1.3620729E 07      7.1376580E 05      6.0177660E 02      2.4666500E 01

   THE ROOTS OF A
-9.7472771E 00  +I  2.4364008E 01      -9.7472771E 00  +I -2.4364008E 01      -2.4509797E 00  +I  1.6776908E 02
-2.4509797E 00  +I -1.6776908E 02


   THE COEFFICIENTS OF POLYNOMIAL B (ORDER =   5)
 0.0000000E 00      1.0240000E 07      1.5488000E 05      2.6019200E 04      9.2000000E 00      1.0000000E 00
   THE ROOTS OF B
 0.0000000E 00  +I  0.0000000E 00      -3.0000000E 00  +I  1.9773720E 01      -3.0000000E 00  +I -1.9773720E 01
-1.6000000E 00  +I -1.5999200E 02      -1.6000000E 00  +I  1.5999200E 02
```

TABLE 4. STANDARD FREQUENCY RESPONSE AND POPOV RESPONSE, EXAMPLE 1.

OMEGA	/M/	/M/-DB	PHI	/M/P	/M/P-DB	PHI-P
1.00000E-02	4.66982E 03	7.33860E 01	-8.99923E 01	4.67023E 01	3.33868E 01	-8.92385E 01
2.00000E-02	2.33491E 03	6.73654E 01	-8.99847E 01	4.67024E 01	3.33868E 01	-8.92366E 01
3.00000E-02	1.55661E 03	6.38436E 01	-8.99770E 01	4.67024E 01	3.33868E 01	-8.92359E 01
4.00000E-02	1.16746E 03	6.13448E 01	-8.99694E 01	4.67024E 01	3.33868E 01	-8.92356E 01
5.00000E-02	9.33967E 02	5.94066E 01	-8.99617E 01	4.67025E 01	3.33868E 01	-8.92354E 01
6.00000E-02	7.78307E 02	5.78230E 01	-8.99541E 01	4.67026E 01	3.33868E 01	-8.92353E 01
7.00000E-02	6.67122E 02	5.64841E 01	-8.99464E 01	4.67026E 01	3.33868E 01	-8.92352E 01
8.00000E-02	5.83733E 02	5.53243E 01	-8.99388E 01	4.67027E 01	3.33868E 01	-8.92351E 01
9.00000E-02	5.18875E 02	5.43012E 01	-8.99311E 01	4.67028E 01	3.33869E 01	-8.92351E 01
1.00000E-01	4.66988E 02	5.33861E 01	-8.99235E 01	4.67029E 01	3.33869E 01	-8.92350E 01
2.00000E-01	2.33504E 02	4.73659E 01	-8.98469E 01	4.67047E 01	3.33872E 01	-8.92348E 01
3.00000E-01	1.55679E 02	4.38446E 01	-8.97704E 01	4.67076E 01	3.33878E 01	-8.92348E 01
4.00000E-01	1.16771E 02	4.13467E 01	-8.96938E 01	4.67117E 01	3.33885E 01	-8.92347E 01
5.00000E-01	9.34277E 01	3.94095E 01	-8.96173E 01	4.67170E 01	3.33895E 01	-8.92347E 01
6.00000E-01	7.78679E 01	3.78272E 01	-8.95408E 01	4.67234E 01	3.33907E 01	-8.92347E 01
7.00000E-01	6.67556E 01	3.64897E 01	-8.94643E 01	4.67310E 01	3.33921E 01	-8.92347E 01
8.00000E-01	5.84229E 01	3.53317E 01	-8.93878E 01	4.67398E 01	3.33937E 01	-8.92348E 01
9.00000E-01	5.19433E 01	3.43106E 01	-8.93112E 01	4.67498E 01	3.33956E 01	-8.92348E 01
1.00000E 00	4.67609E 01	3.33977E 01	-8.92348E 01	4.67609E 01	3.33977E 01	-8.92348E 01
2.00000E 00	2.34755E 01	2.74123E 01	-8.84706E 01	4.69385E 01	3.34306E 01	-8.92352E 01
3.00000E 00	1.57581E 01	2.39501E 01	-8.77089E 01	4.72408E 01	3.34863E 01	-8.92360E 01
4.00000E 00	1.19354E 01	2.15367E 01	-8.69521E 01	4.76782E 01	3.35664E 01	-8.92374E 01
5.00000E 00	9.67357E 00	1.97117E 01	-8.62039E 01	4.82660E 01	3.36728E 01	-8.92397E 01
6.00000E 00	8.19588E 00	1.82719E 01	-8.54702E 01	4.90260E 01	3.38085E 01	-8.92435E 01
7.00000E 00	7.17053E 00	1.71110E 01	-8.47602E 01	4.99883E 01	3.39774E 01	-8.92494E 01
8.00000E 00	6.43296E 00	1.61682E 01	-8.40883E 01	5.11943E 01	3.41844E 01	-8.92585E 01
9.00000E 00	5.89331E 00	1.54072E 01	-8.34762E 01	5.27006E 01	3.44363E 01	-8.92720E 01
1.00000E 01	5.49962E 00	1.48066E 01	-8.29572E 01	5.45854E 01	3.47415E 01	-8.92922E 01
2.00000E 01	5.49063E 00	1.47924E 01	-1.26454E 02	8.83866E 01	3.89277E 01	-9.21153E 01
3.00000E 01	1.06161E 00	5.19260E-01	-1.40243E 02	2.03845E 01	2.61860E 01	-9.22945E 01
4.00000E 01	6.68701E-01	-3.49537E 00	-1.19123E 02	2.33687E 01	2.73727E 01	-9.07980E 01
5.00000E 01	5.31239E-01	-5.49419E 00	-1.10005E 02	2.49600E 01	2.79449E 01	-9.04172E 01
6.00000E 01	4.47049E-01	-6.99290E 00	-1.05283E 02	2.58746E 01	2.82575E 01	-9.02609E 01
7.00000E 01	3.87739E-01	-8.22921E 00	-1.02396E 02	2.65091E 01	2.84679E 01	-9.01799E 01
8.00000E 01	3.43619E-01	-9.27844E 00	-1.00430E 02	2.70354E 01	2.86387E 01	-9.01318E 01
9.00000E 01	3.09902E-01	-1.01755E 01	-9.89894E 01	2.75487E 01	2.88020E 01	-9.01007E 01
1.00000E 02	2.83921E-01	-1.09360E 01	-9.78767E 01	2.81243E 01	2.89816E 01	-9.00793E 01
2.00000E 02	1.01425E-01	-1.98771E 01	-9.61141E 01	2.01696E 01	2.60940E 01	-9.00307E 01
3.00000E 02	7.88747E-02	-2.20612E 01	-9.31037E 01	2.36277E 01	2.74684E 01	-9.00104E 01
4.00000E 02	6.04551E-02	-2.43713E 01	-9.22471E 01	2.41635E 01	2.76632E 01	-9.00056E 01
5.00000E 02	4.87509E-02	-2.62403E 01	-9.17748E 01	2.43638E 01	2.77349E 01	-9.00036E 01
6.00000E 02	4.07848E-02	-2.77900E 01	-9.14698E 01	2.44628E 01	2.77701E 01	-9.00025E 01
7.00000E 02	3.50364E-02	-2.91096E 01	-9.12553E 01	2.45196E 01	2.77903E 01	-9.00018E 01
8.00000E 02	3.06998E-02	-3.02573E 01	-9.10959E 01	2.45554E 01	2.78029E 01	-9.00014E 01
9.00000E 02	2.73144E-02	-3.12722E 01	-9.09727E 01	2.45794E 01	2.78114E 01	-9.00011E 01
1.00000E 03	2.45992E-02	-3.21816E 01	-9.08745E 01	2.45963E 01	2.78174E 01	-9.00009E 01
2.00000E 03	1.23250E-02	-3.81843E 01	-9.04358E 01	2.46493E 01	2.78361E 01	-9.00002E 01
3.00000E 03	8.21973E-03	-4.17029E 01	-9.02904E 01	2.46589E 01	2.78395E 01	-9.00001E 01
4.00000E 03	6.16560E-03	-4.42005E 01	-9.02177E 01	2.46622E 01	2.78406E 01	-9.00001E 01
5.00000E 03	4.93277E-03	-4.61382E 01	-9.01742E 01	2.46638E 01	2.78412E 01	-9.00000E 01
6.00000E 03	4.11078E-03	-4.77215E 01	-9.01451E 01	2.46646E 01	2.78415E 01	-9.00000E 01
7.00000E 03	3.52359E-03	-4.90603E 01	-9.01244E 01	2.46651E 01	2.78417E 01	-9.00000E 01
8.00000E 03	3.08318E-03	-5.02200E 01	-9.01088E 01	2.46654E 01	2.78418E 01	-9.00000E 01
9.00000E 03	2.74063E-03	-5.12430E 01	-9.00967E 01	2.46657E 01	2.78419E 01	-9.00000E 01
1.00000E 04	2.46658E-03	-5.21581E 01	-9.00871E 01	2.46658E 01	2.78419E 01	-9.00000E 01
2.43640E-01	1.91684E 02	4.56517E 01	-8.98135E 01	4.67058E 01	3.33874E 01	-8.92348E 01

TABLE 4. Continued

OMEGA	/M/	/M/-DB	PHI	/M/P	/M/P-DB	PHI-P
1.21820E 00	3.84103E 01	3.16889E 01	-8.90679E 01	4.67894E 01	3.34029E 01	-8.92348E 01
2.43640E 00	1.93216E 01	2.57209E 01	-8.81378E 01	4.70546E 01	3.34520E 01	-8.92355E 01
4.87280E 00	9.90778E 00	1.99195E 01	-8.62984E 01	4.81822E 01	3.36577E 01	-8.92394E 01
7.30920E 00	6.91694E 00	1.67983E 01	-8.45476E 01	5.03329E 01	3.40370E 01	-8.92518E 01
9.74560E 00	5.58789E 00	1.49450E 01	-8.30780E 01	5.40646E 01	3.46583E 01	-8.92863E 01
1.21820E 01	5.01793E 00	1.40105E 01	-8.24340E 01	6.05999E 01	3.56494E 01	-8.93753E 01
1.46184E 01	4.99486E 00	1.39705E 01	-8.42032E 01	7.26453E 01	3.72241E 01	-8.96021E 01
1.70548E 01	5.51717E 00	1.48343E 01	-9.32164E 01	9.39466E 01	3.94576E 01	-9.01888E 01
1.94912E 01	5.74973E 00	1.51929E 01	-1.19288E 02	9.77842E 01	3.98054E 01	-9.16483E 01
2.19276E 01	3.81459E 00	1.16290E 01	-1.47424E 02	4.51501E 01	3.30932E 01	-9.40827E 01
2.43640E 01	2.22777E 00	6.95739E 00	-1.54187E 02	2.37191E 01	2.75020E 01	-9.48502E 01
4.87280E 01	5.44700E-01	-5.27685E 00	-1.10831E 02	2.48079E 01	2.78918E 01	-9.04474E 01
9.74560E 01	2.89893E-01	-1.07552E 01	-9.81357E 01	2.79675E 01	2.89331E 01	-9.00840E 01
1.46184E 02	2.68970E-01	-1.14059E 01	-9.57300E 01	3.91226E 01	3.18486E 01	-9.00393E 01
1.94912E 02	1.00522E-01	-1.99547E 01	-9.66996E 01	1.94593E 01	2.57825E 01	-9.00345E 01
2.43640E 02	9.34485E-02	-2.05886E 01	-9.40766E 01	2.27102E 01	2.71244E 01	-9.00168E 01
1.67769E 00	2.79405E 01	2.89247E 01	-8.87167E 01	4.68680E 01	3.34175E 01	-8.92350E 01
8.38845E 00	6.20329E 00	1.58524E 01	-8.38419E 01	5.17400E 01	3.42765E 01	-8.92631E 01
1.67769E 01	5.43532E 00	1.47045E 01	-9.15162E 01	9.11560E 01	3.91957E 01	-9.00904E 01
3.35538E 01	8.47953E-01	-1.43257E 00	-1.30640E 02	2.15969E 01	2.66878E 01	-9.14653E 01
5.03307E 01	5.27868E-01	-5.54949E 00	-1.09800E 02	2.49979E 01	2.79581E 01	-9.04098E 01
6.71076E 01	4.03039E-01	-7.89307E 00	-1.03111E 02	2.63421E 01	2.84130E 01	-9.01989E 01
8.38845E 01	3.29467E-01	-9.64377E 00	-9.98217E 01	2.72321E 01	2.87016E 01	-9.01182E 01
1.00661E 02	2.82415E-01	-1.09816E 01	-9.78117E 01	2.81665E 01	2.89947E 01	-9.00781E 01
1.17438E 02	2.53446E-01	-1.19223E 01	-9.64261E 01	2.95772E 01	2.94191E 01	-9.00549E 01
1.34215E 02	2.44201E-01	-1.22451E 01	-9.54978E 01	3.26247E 01	3.02709E 01	-9.00411E 01
1.50992E 02	3.09045E-01	-1.01996E 01	-9.71819E 01	4.62973E 01	3.33111E 01	-9.00478E 01
1.67769E 02	4.62744E-02	-2.66932E 01	-1.73228E 02	9.16547E-01	-7.56909E-01	-9.28738E 01
3.35538E 02	7.12907E-02	-2.29393E 01	-9.27262E 01	2.38937E 01	2.75657E 01	-9.00081E 01
6.71076E 02	3.65269E-02	-2.87477E 01	-9.13105E 01	2.45059E 01	2.77854E 01	-9.00020E 01
1.00661E 03	2.44385E-02	-3.22385E 01	-9.08687E 01	2.45973E 01	2.78177E 01	-9.00009E 01
1.34215E 03	1.83508E-02	-3.47269E 01	-9.06503E 01	2.46280E 01	2.78286E 01	-9.00005E 01
1.67769E 03	1.46886E-02	-3.66604E 01	-9.05198E 01	2.46420E 01	2.78335E 01	-9.00003E 01
1.97737E-01	2.36175E 02	4.74647E 01	-8.98487E 01	4.67046E 01	3.33872E 01	-8.92348E 01
9.88686E-01	4.72946E 01	3.34962E 01	-8.92434E 01	4.67596E 01	3.33974E 01	-8.92348E 01
1.97737E 00	2.37412E 01	2.75101E 01	-8.84878E 01	4.69331E 01	3.34296E 01	-8.92352E 01
3.95474E 00	1.20658E 01	2.16311E 01	-8.69862E 01	4.76553E 01	3.35622E 01	-8.92373E 01
5.93212E 00	8.27937E 00	1.83599E 01	-8.55194E 01	4.89684E 01	3.37983E 01	-8.92432E 01
7.90949E 00	6.49057E 00	1.62457E 01	-8.41471E 01	5.10737E 01	3.41640E 01	-8.92575E 01
9.88686E 00	5.53794E 00	1.48670E 01	-8.30098E 01	5.43500E 01	3.47040E 01	-8.92895E 01
1.18642E 01	5.06005E 00	1.40831E 01	-8.24330E 01	5.95146E 01	3.54925E 01	-8.93585E 01
1.38416E 01	4.94359E 00	1.38809E 01	-8.31900E 01	6.79470E 01	3.66434E 01	-8.95057E 01
1.58190E 01	5.18828E 00	1.43005E 01	-8.71953E 01	8.19754E 01	3.82737E 01	-8.98226E 01
1.77963E 01	5.73111E 00	1.51648E 01	-9.89839E 01	1.00746E 02	4.00645E 01	-9.05090E 01
1.97737E 01	5.62277E 00	1.49990E 01	-1.23276E 02	9.30043E 01	3.93701E 01	-9.19009E 01
3.95474E 01	6.77500E-01	-3.38181E 00	-1.19733E 02	2.32685E 01	2.73354E 01	-9.08274E 01
7.90949E 01	3.47132E-01	-9.19010E 00	-1.00582E 02	2.69895E 01	2.86239E 01	-9.01353E 01
1.18642E 02	2.52016E-01	-1.19714E 01	-9.63434E 01	2.97168E 01	2.94600E 01	-9.00537E 01
1.58190E 02	6.52422E-01	-3.70943E 00	-1.22404E 02	8.71371E 01	3.88041E 01	-9.02299E 01
1.97737E 02	1.01144E-01	-1.99012E 01	-9.63511E 01	1.98773E 01	2.59671E 01	-9.00323E 01
1.59992E 00	2.92886E 01	2.93340E 01	-8.87761E 01	4.68529E 01	3.34147E 01	-8.92350E 01
7.99960E 00	6.43321E 00	1.61686E 01	-8.40886E 01	5.11937E 01	3.41843E 01	-8.92584E 01
1.59992E 01	5.22917E 00	1.43687E 01	-8.78503E 01	8.36039E 01	3.84445E 01	-8.98656E 01
3.19984E 01	9.23181E-01	-6.94260E-01	-1.34545E 02	2.10632E 01	2.64705E 01	-9.17618E 01
4.79976E 01	5.52811E-01	-5.14847E 00	-1.11339E 02	2.47154E 01	2.78594E 01	-9.04663E 01
6.39968E 01	4.21072E-01	-7.51288E 00	-1.03979E 02	2.61494E 01	2.83492E 01	-9.02229E 01
7.99960E 01	3.43635E-01	-9.27806E 00	-1.00431E 02	2.70352E 01	2.86386E 01	-9.01319E 01

TABLE 4. Continued

OMEGA	/M/	/M/-DB	PHI	/M/P	/M/P-DB	PHI-P
9.59952E 01	2.93512E-01	-1.06475E 01	-9.82913E 01	2.78812E 01	2.89062E 01	-9.00870E 01
1.11994E 02	2.61031E-01	-1.16662E 01	-9.68262E 01	2.90268E 01	2.92560E 01	-9.00612E 01
1.27994E 02	2.44497E-01	-1.22345E 01	-9.57759E 01	3.11352E 01	2.98650E 01	-9.00453E 01
1.43993E 02	2.59742E-01	-1.17092E 01	-9.55129E 01	3.72280E 01	3.14174E 01	-9.00384E 01
1.59992E 02	8.01243E-01	-1.92472E 00	-1.67605E 02	2.75277E 01	2.87954E 01	-9.16290E 01
3.19984E 02	7.44499E-02	-2.25627E 01	-9.28779E 01	2.37927E 01	2.75289E 01	-9.00090E 01
6.39968E 02	3.82770E-02	-2.83412E 01	-9.13757E 01	2.44890E 01	2.77794E 01	-9.00022E 01
9.59952E 02	2.56193E-02	-3.18287E 01	-9.09113E 01	2.45902E 01	2.78152E 01	-9.00009E 01
1.27994E 03	1.92399E-02	-3.43160E 01	-9.06821E 01	2.46241E 01	2.78272E 01	-9.00005E 01
1.59992E 03	1.54011E-02	-3.62489E 01	-9.05451E 01	2.46395E 01	2.78326E 01	-9.00003E 01

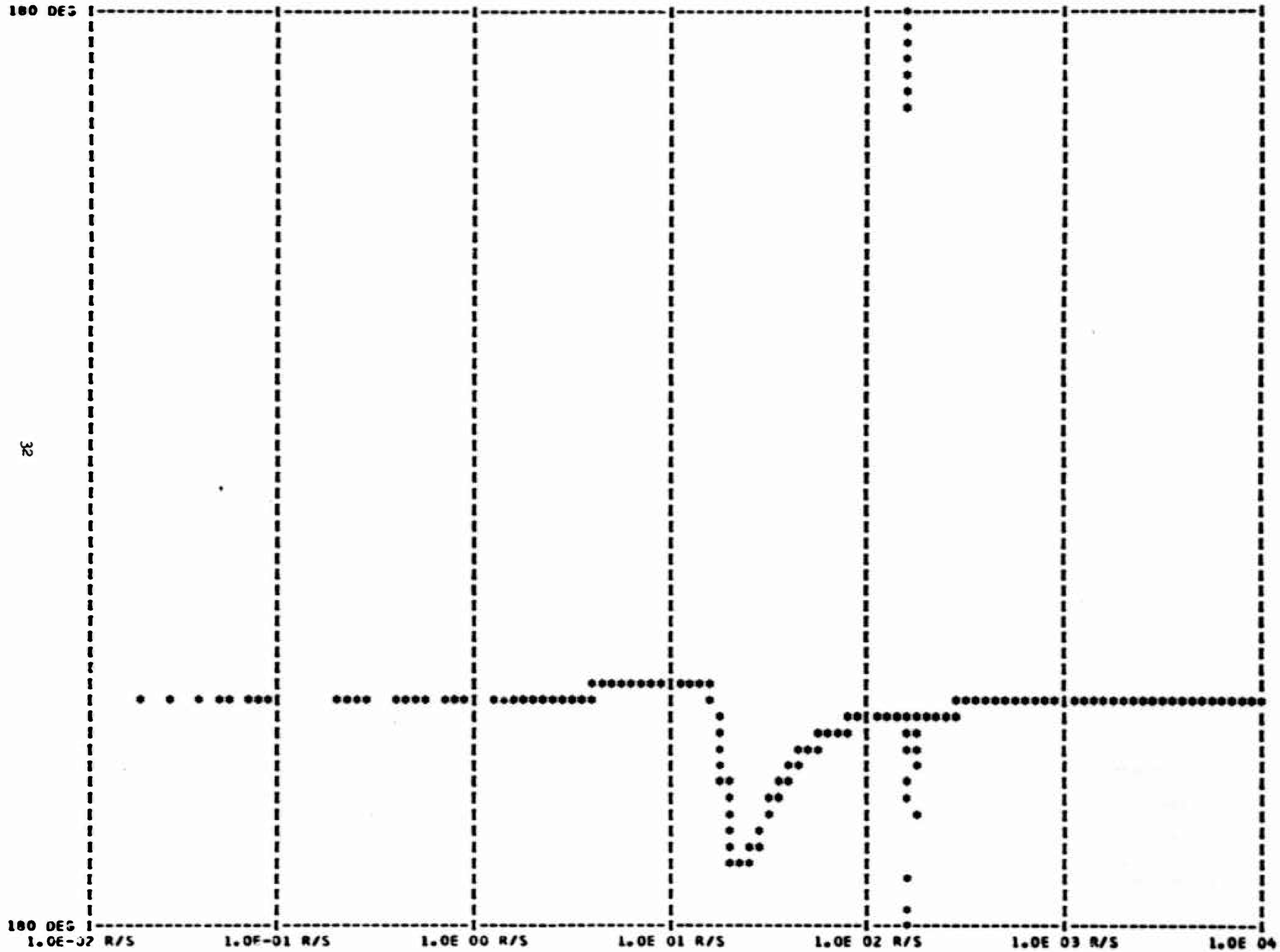


FIGURE 5. EXAMPLE 1, LINEAR SYSTEM BODE PLOT, PHASE ANGLE

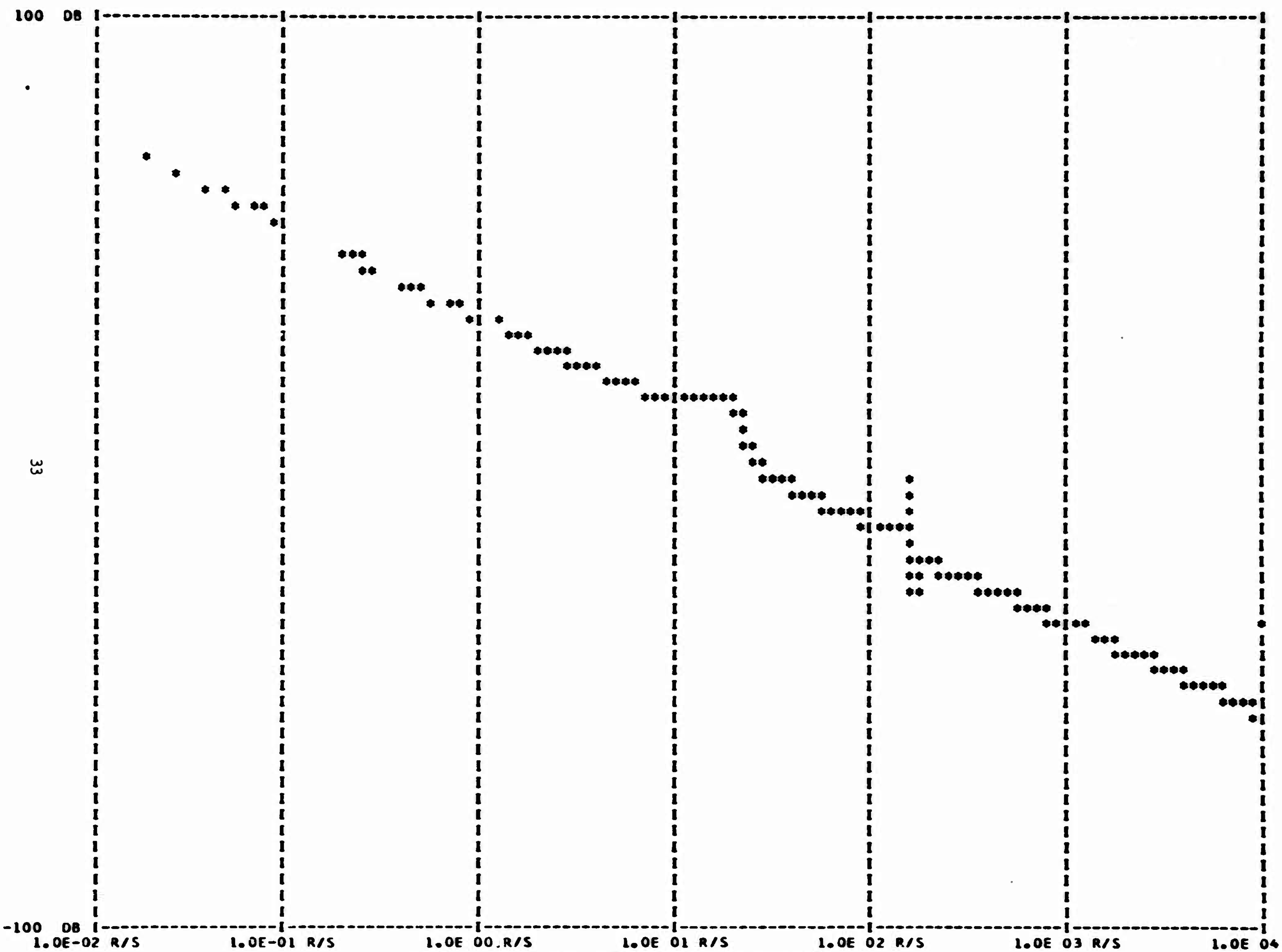


FIGURE 6. EXAMPLE 1, LINEAR SYSTEM BODE, MAGNITUDE

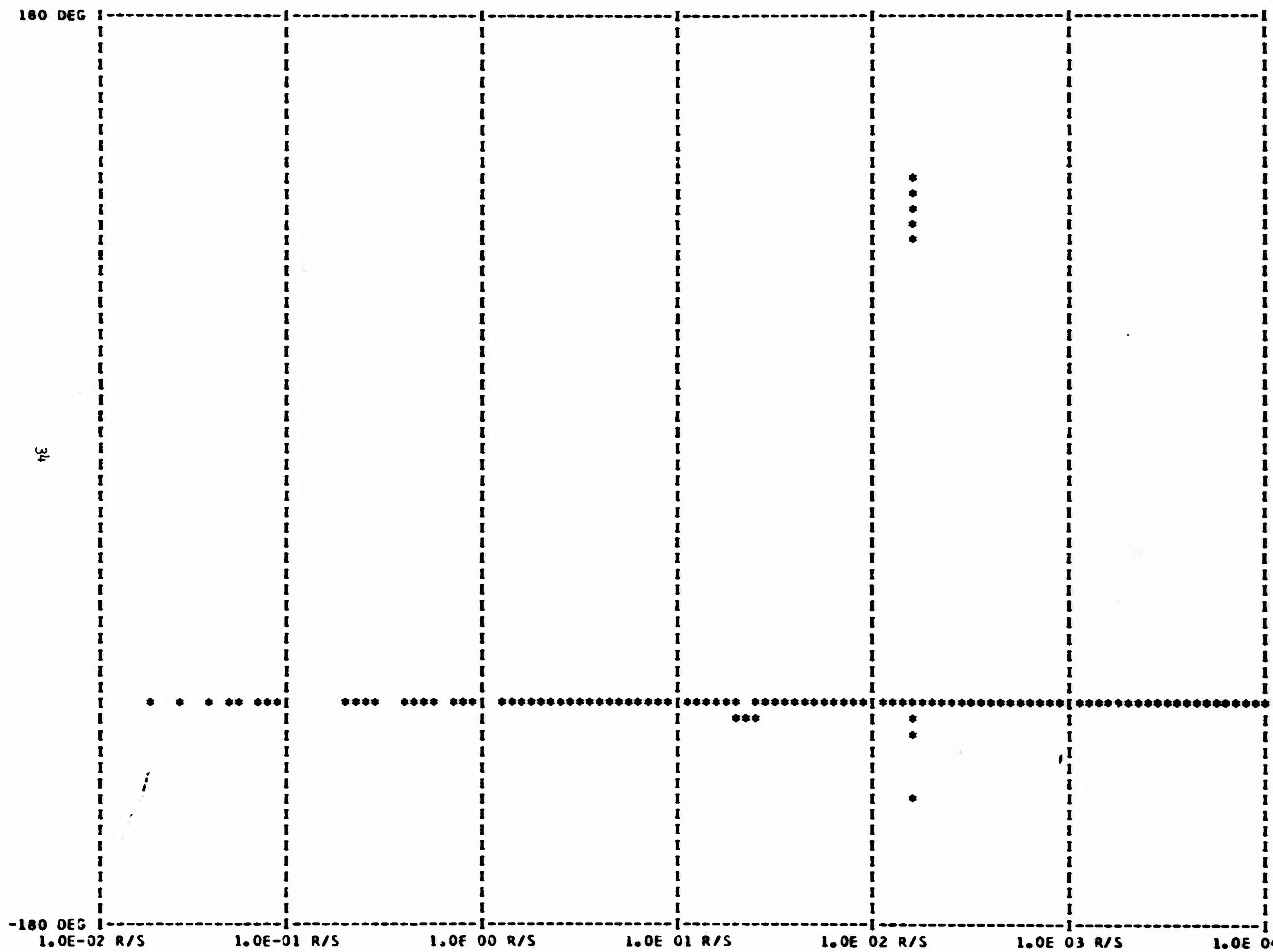


FIGURE 7. EXAMPLE 1. NONLINEAR SYSTEM, BODE PLOT, PHASE ANGLE

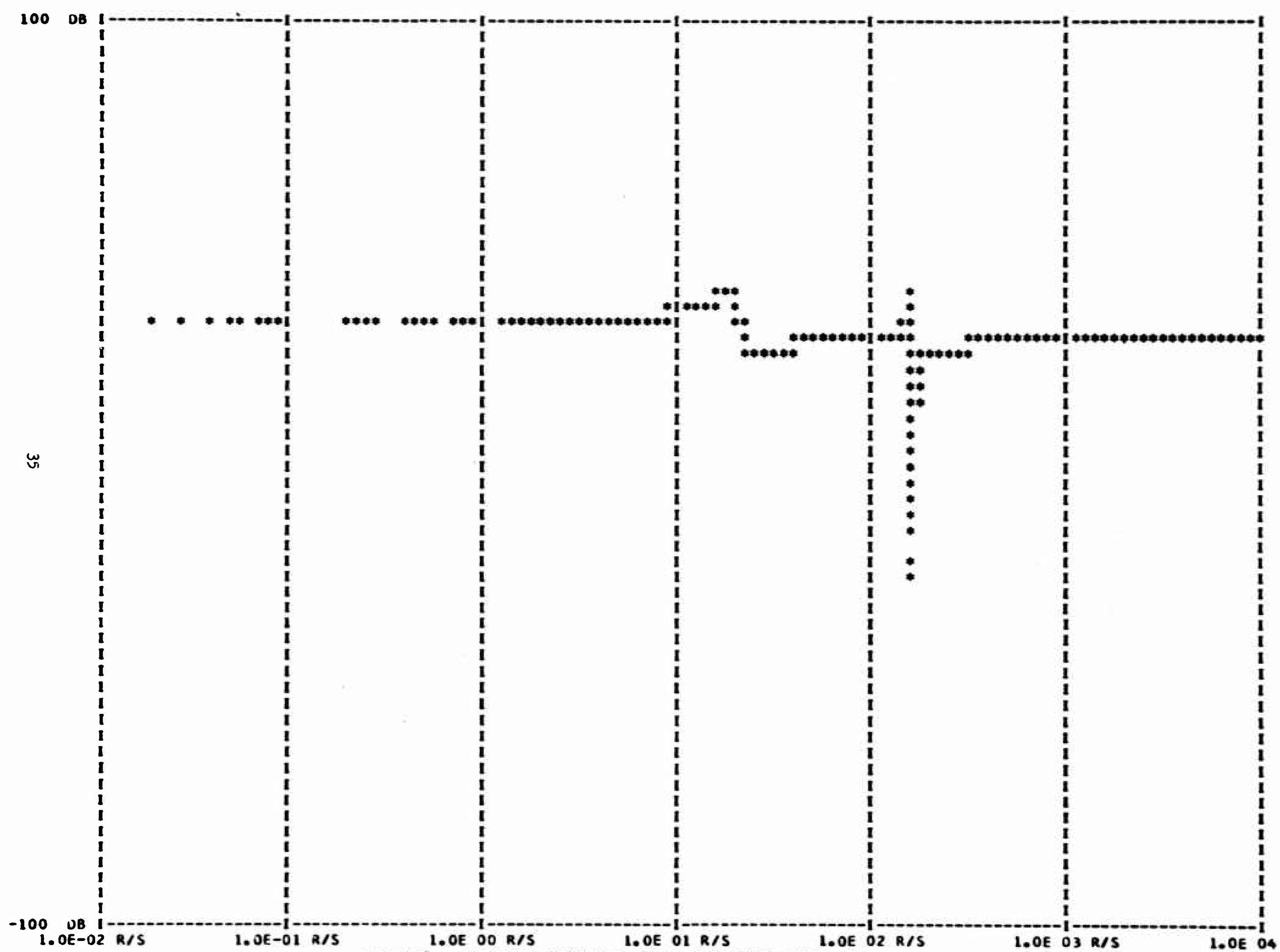


FIGURE 8. EXAMPLE 1, NONLINEAR SYSTEM, BODE PLOT, MAGNITUDE

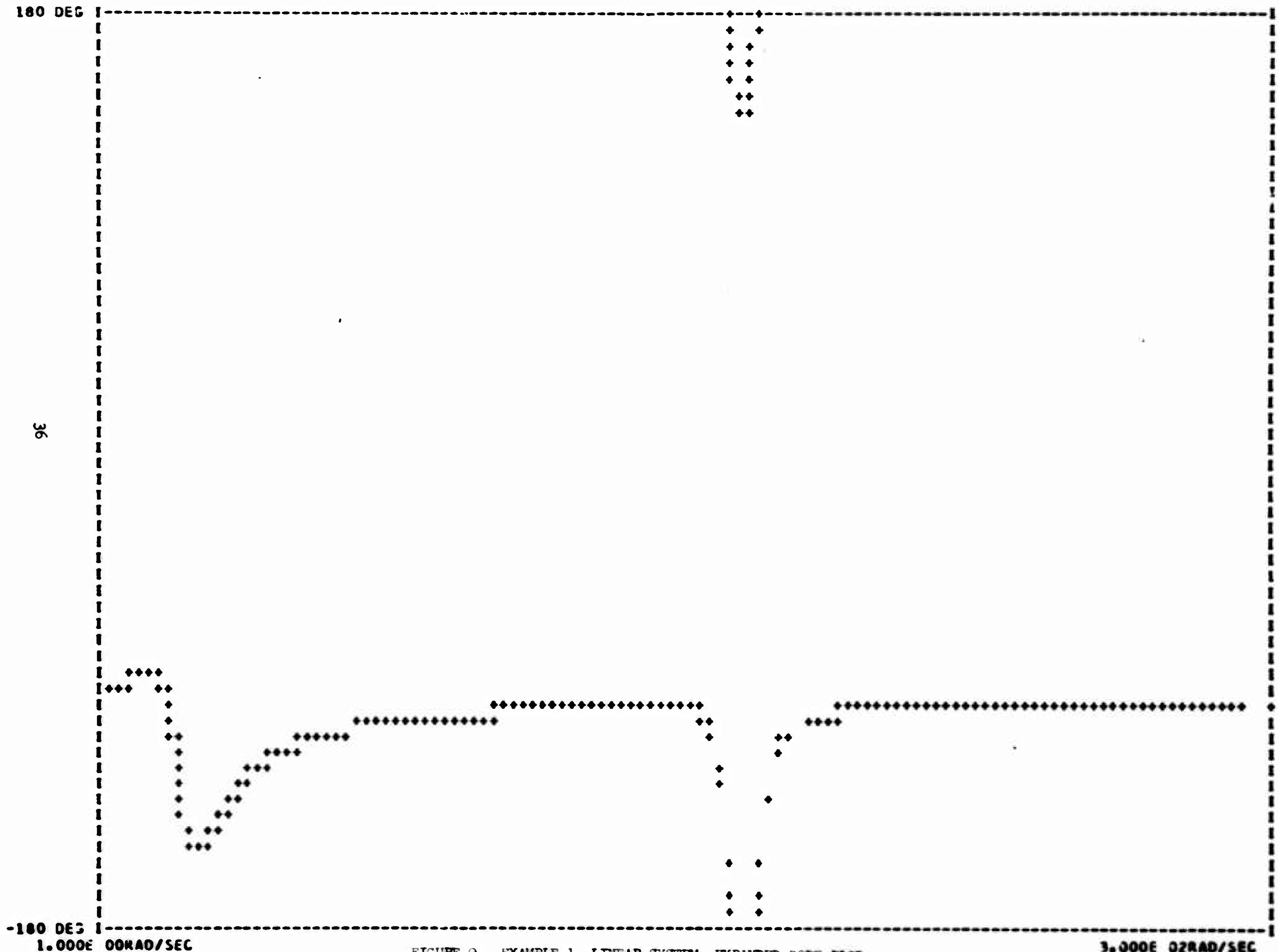


FIGURE 9. EXAMPLE 1, LINEAR SYSTEM, EXPANDED BODE PLOT
(1 to 300 RAD/S) PHASE ANGLE

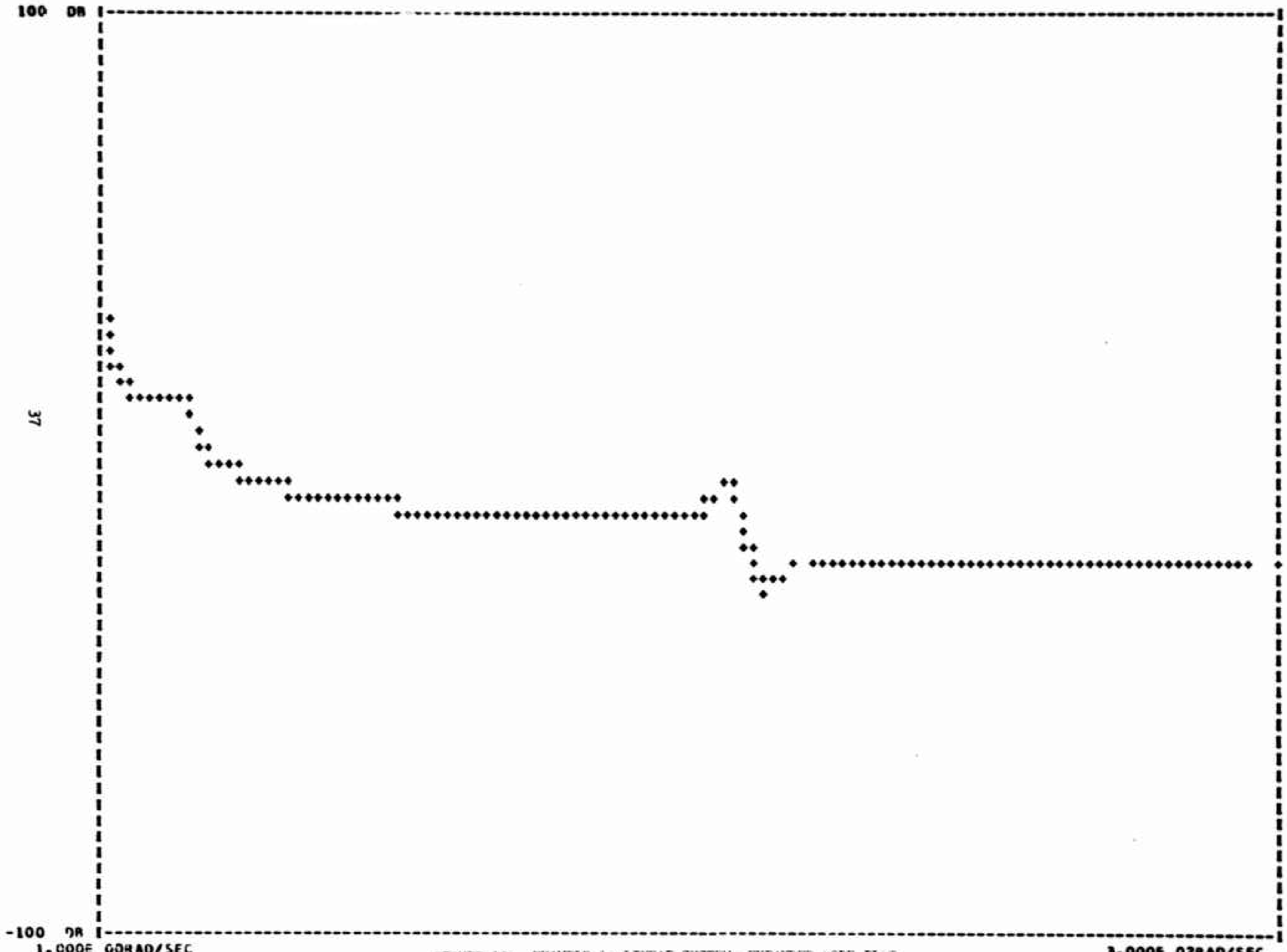


FIGURE 10. EXAMPLE 1, LINEAR SYSTEM, EXPANDED BODE PLOT
(1 TO 300 RAD/S) MAGNITUDE

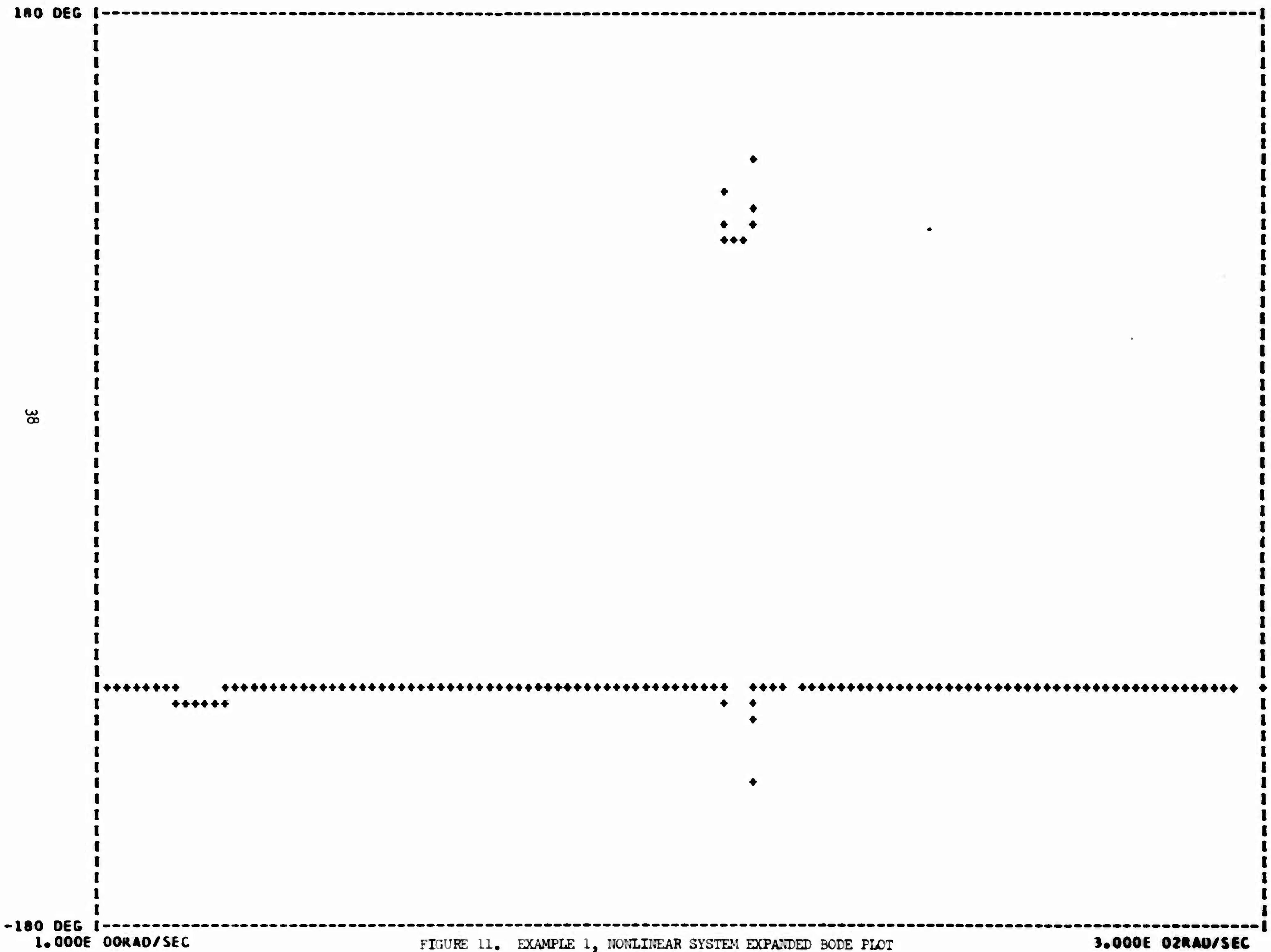


FIGURE 11. EXAMPLE 1, NONLINEAR SYSTEM EXPANDED BODE PLOT
(1 TO 300 RAD/S) PHASE ANGLE

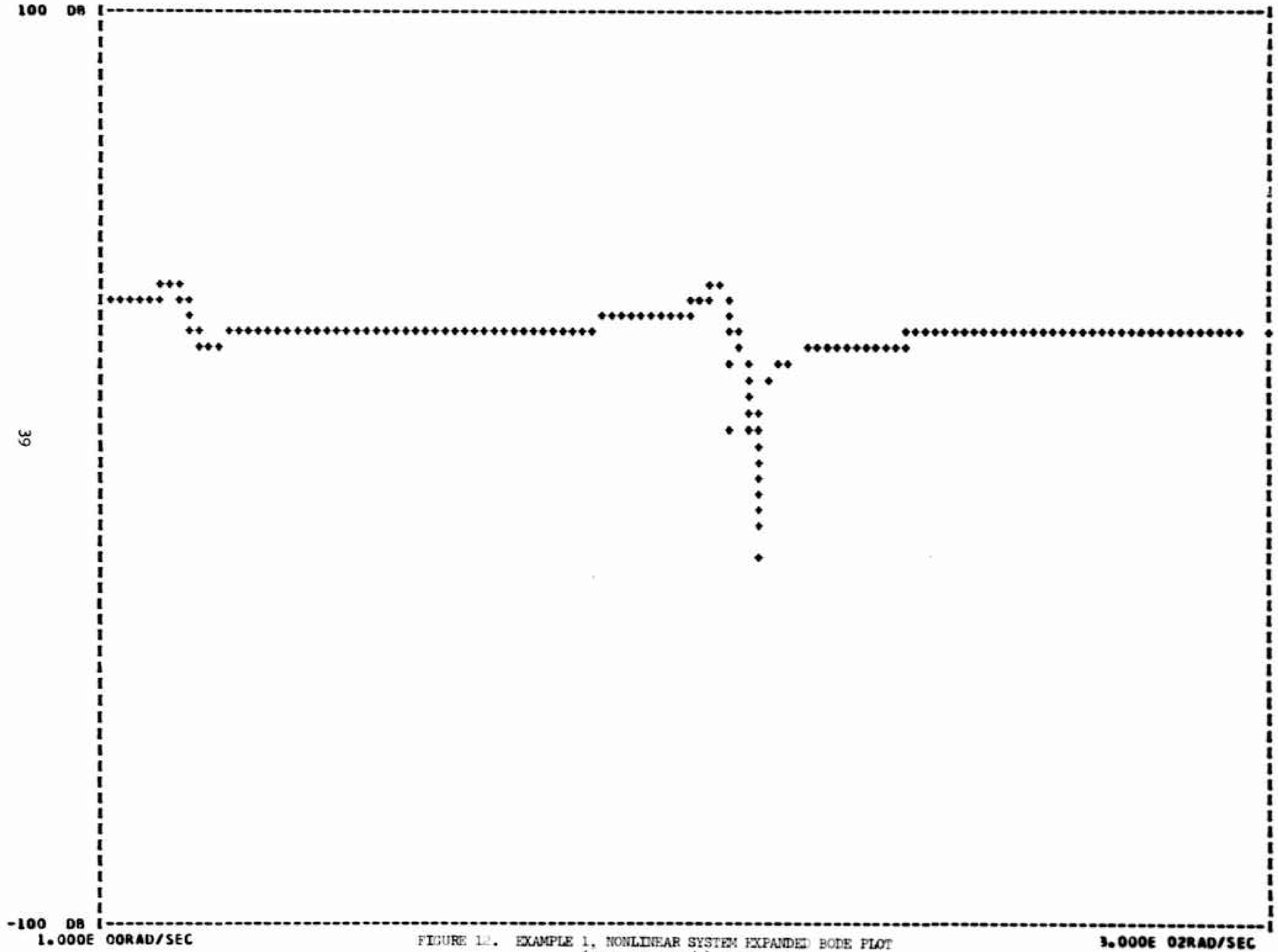


FIGURE 12. EXAMPLE 1, NONLINEAR SYSTEM EXPANDED BODE PLOT
(1 TO 300 RAD/S) MAGNITUDE

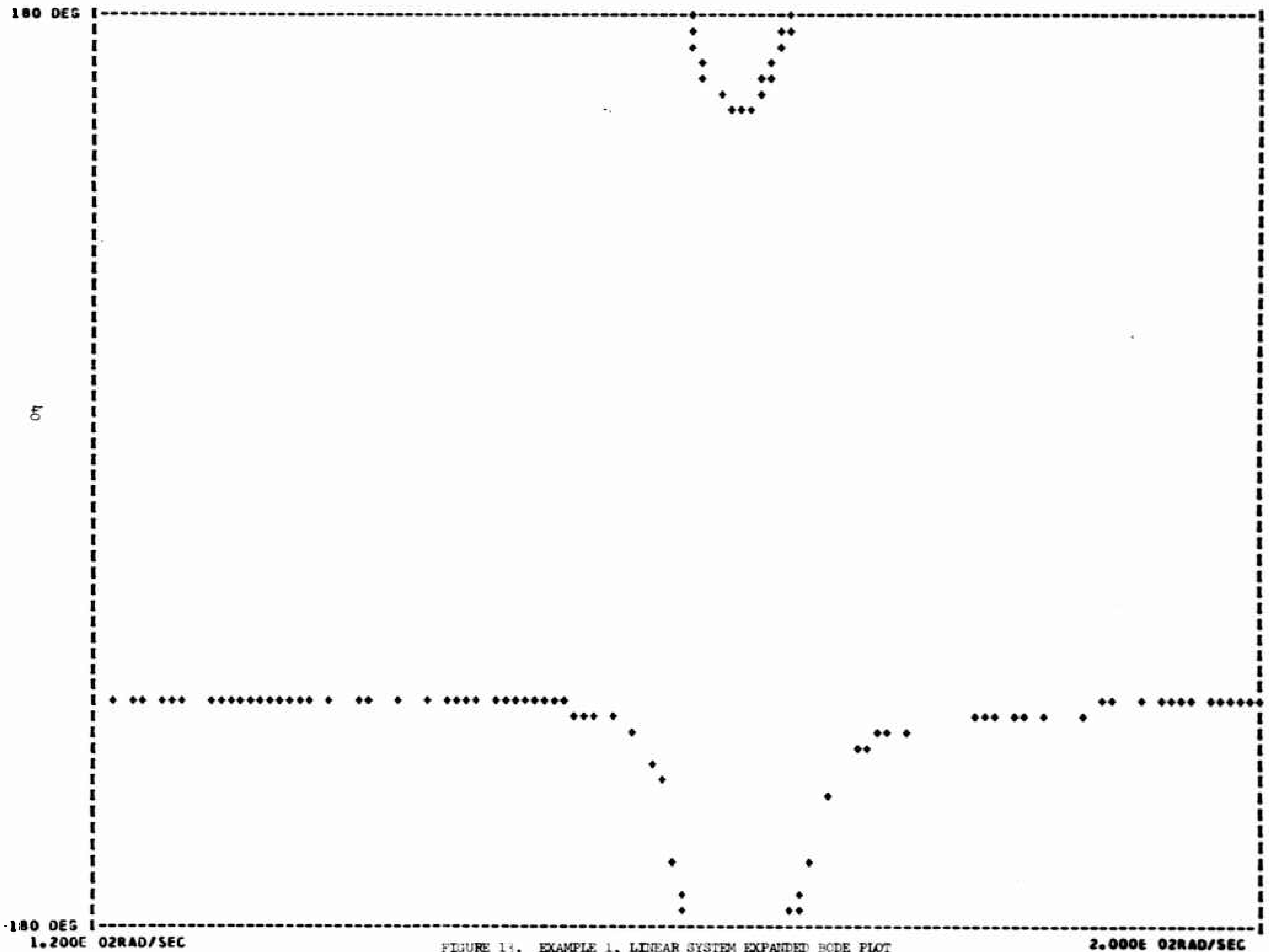


FIGURE 13. EXAMPLE 1, LINEAR SYSTEM EXPANDED BODE PLOT
(120 TO 200 RAD/S) PHASE ANGLE

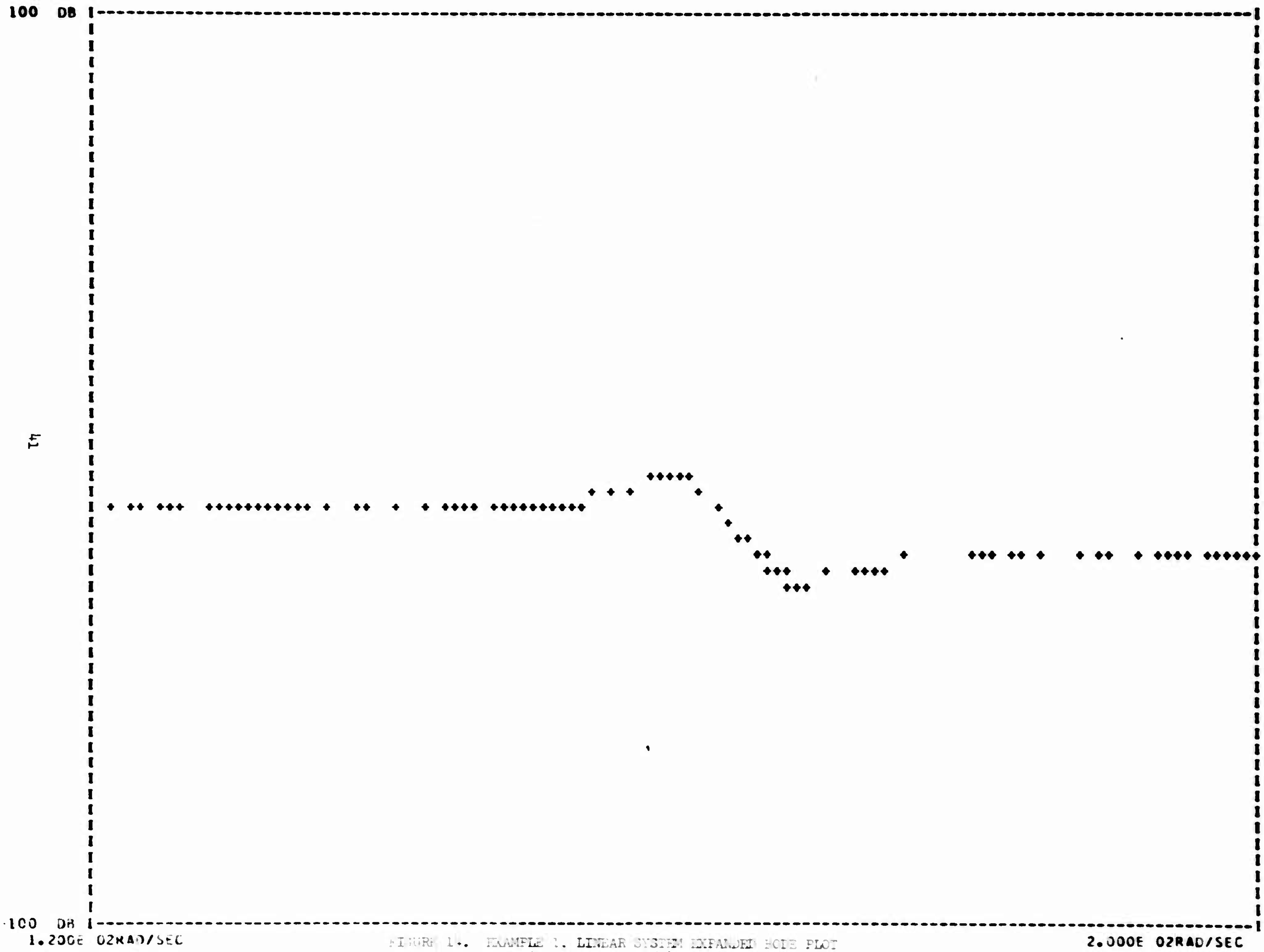


FIGURE 14. EXAMPLE 1. LINEAR SYSTEM EXPANDED BODE PLOT
(120 TO 200 RAD/S) MAGNITUDE

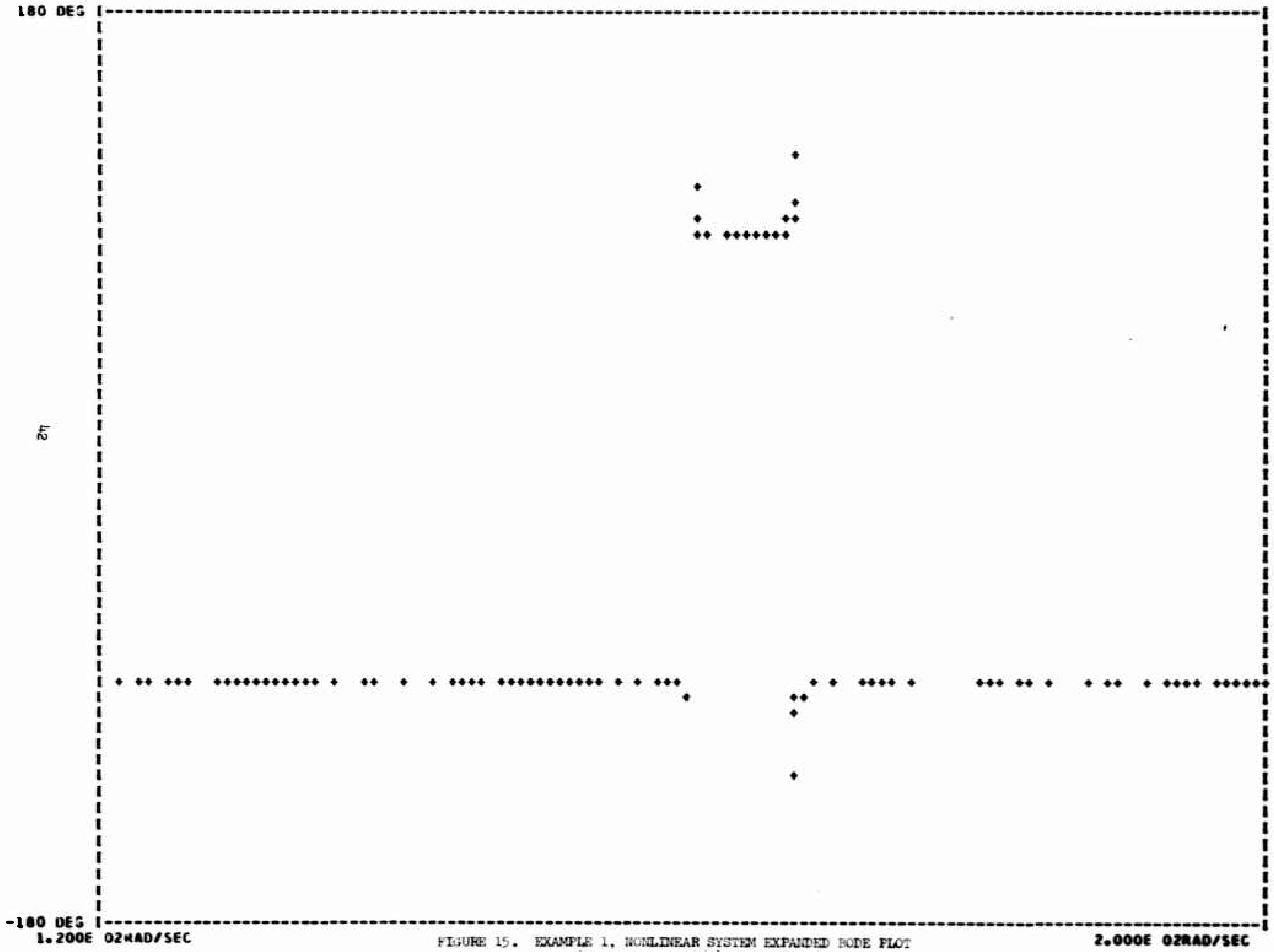


FIGURE 15. EXAMPLE 1, NONLINEAR SYSTEM EXPANDED BODE PLOT
(120 TO 200 RAD/S) PHASE ANGLE

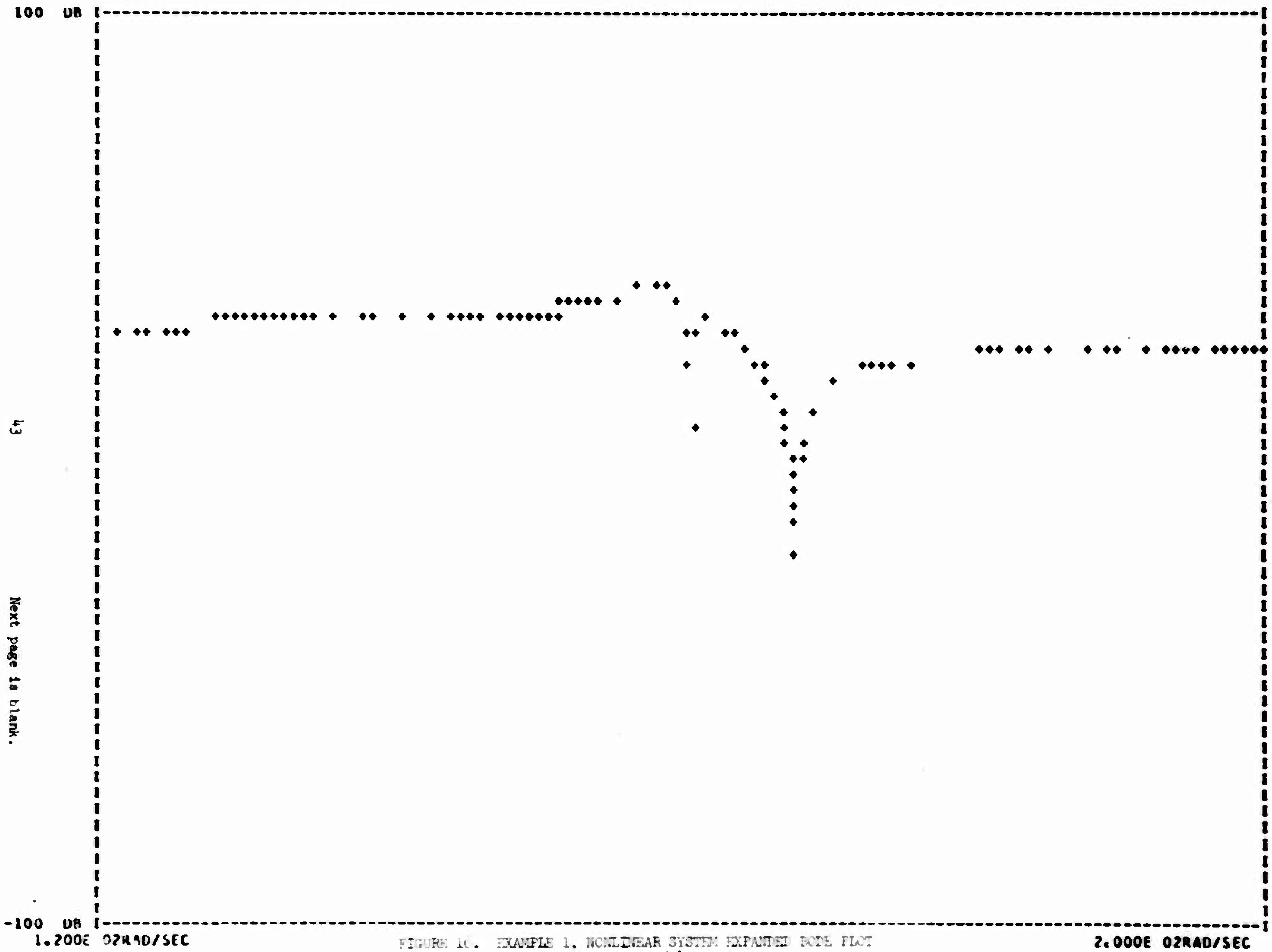


FIGURE 16. EXAMPLE 1, NONLINEAR SYSTEM EXPANDED BODE PLOT
(120 TO 200 RAD/S) MAGNITUDE

Next page is blank.

TABLE 5. MIRROR OF INPUT, EXAMPLE 2.

```
POLYNOMIAL MULTIPLICATION ANC RCCT LOCUS                                          PROBLEM NO.   37

DELTAK=          0     PCLY.ADDEC IN A(S)=        2     PCLY.ADDED IN B(S)=        1     PROB.NO.=        37

K-INITIAL=        .0000000000     INCREMENT K=        .0000000010     K-TERMINATE=        .0000000010
   NUMBER OF POLY. IN GROUP  1 OF NUMERATOR=        8
   NUMBER OF POLY. IN GROUP  2 OF NUMERATOR=        9
   NUMBER OF POLY. IN GROUP  1 OF CENOMINATCR=       7
```

C(1)=	.0015000000	C(1)=	-1.0000000000	C(1)=	.0432850000
C(1)=	-.3981900000	C(1)=	.1500000000	C(2)=	.1699800000
C(2)=	61.1407000000	C(1)=	.0000000000	C(3)=	5.9049100000
C(3)=	9194.1800000000	C(2)=	-.1754300000	C(4)=	1.7491100000
C(4)=	45.7742000000	C(3)=	-12.0360000000	C(5)=	1.0072300000
C(5)=	-2.6617000000	C(4)=	-13.2940000000	C(6)=	.0028104000
C(6)=	.0815440000	C(5)=	-.0247200000	C(7)=	.0008209900
C(7)=	.0285990000	C(6)=	-.0088740000	C(8)=	.0000012601
C(8)=	.0000673410	C(7)=	-.0000085040	C(9)=	.0000002245
C(9)=	.0000159750	C(8)=	-.0000009454	C(10)=	.0000000002
C(10)=	.0000000172	C(9)=	-.0000000001	C(11)=	.0000000000
C(11)=	.0000000026	C(10)=	.0000000002	C(12)=	.0000000000
C(12)=	.0000000000	C(11)=	.0000000000	C(13)=	.0000000000
C(13)=	.0000000000	C(12)=	.0000000000	C(14)=	.0000000000
C(14)=	.0000000000	C(13)=	.0000000000	C(15)=	.0000000000
C(15)=	.0000000000	C(14)=	.0000000000	C(1)=	119300.0000000000
C(1)=	119300.0000000000	C(1)=	35344.0000000000	C(2)=	207.3000000000
C(1)=	37.5000000000	C(1)=	.0000000000	C(3)=	1.0000000000
C(2)=	12.5000000000	C(2)=	1.0000000000	C(1)=	37.5000000000
C(1)=	35344.0000000000	C(1)=	119300.0000000000	C(2)=	3.0000000000
C(2)=	62.8000000000	C(2)=	207.3000000000	C(1)=	35344.0000000000
C(3)=	1.0000000000	C(3)=	1.0000000000	C(2)=	62.8000000000
C(1)=	1.0000000000	C(1)=	37.5000000000	C(3)=	1.0000000000
C(2)=	2.0000000000	C(2)=	3.0000000000	C(1)=	1.0000000000
C(1)=	180000.0000000000	C(1)=	180000.0000000000	C(2)=	2.0000000000
C(2)=	4000.0000000000	C(2)=	4000.0000000000	C(1)=	180000.0000000000
C(1)=	3600.0000000000	C(1)=	3600.0000000000	C(2)=	13000.0000000000
C(2)=	30.0000000000	C(2)=	30.0000000000	C(3)=	246.0000000000
				C(4)=	2.3000000000
				C(1)=	3600.0000000000
				C(2)=	120.0000000000

TABLE 6. EQUIVALENT SYSTEM OPEN LOOP POLYNOMIALS AND ROOTS, EXAMPLE 2.

COEFFICIENTS ARE GIVEN IN ASCENDING ORDER

THE COEFFICIENTS OF POLYNOMIAL A (ORDER = 20)

-6.1198957E 16	9.2521124E 18	1.4379605E 21	3.5421717E 21	1.2905790E 21	5.9459794E 19
9.0365506E 17	3.3735915E 16	3.9371677E 15	1.1777247E 14	2.9117079E 12	6.2795230E 10
8.1833541E 08	1.1560946E 07	1.0646439E 05	9.3728004E 02	6.6668516E 00	3.3289968E-02
1.8402498E-04	4.2351181E-07	1.7475004E-09			

THE ROOTS OF A

-1.0694428E-02	+I	0.0000000E 00	-4.8391511E-01	+I	1.0342048E-14	4.0455430E-03	+I	0.0000000E 00
-2.6281354E 00	+I	0.0000000E 00	-1.9524174E 01	+I	-1.2192916E 01	-1.9524174E 01	+I	1.2192916E 01
1.8019734E 01	+I	-1.8023020E 01	1.8019734E 01	+I	1.8023020E 01	-4.4083790E-01	+I	5.8588405E 01
-4.4083790E-01	+I	-5.8588405E 01	-1.0455319E 01	+I	1.0028177E 02	-1.0455319E 01	+I	-1.0028177E 02
-4.5000000E 01	+I	0.0000000E 00	9.5034570E 00	+I	9.8992955E 01	9.5034570E 00	+I	-9.8992955E 01
-3.2752621E 01	+I	1.8552017E 02	-3.2752621E 01	+I	-1.8552017E 02	-1.2000000E 02	+I	0.0000000E 00
-1.4673431E 00	+I	1.6720316E 02	-1.4673431E 00	+I	-1.6720316E 02			

THE COEFFICIENTS OF POLYNOMIAL B (ORDER = 24)

4.4350635E 18	2.7125958E 19	6.4489032E 20	1.5105939E 21	7.3257871E 20	3.1217410E 20
4.6274362E 19	3.1607955E 18	1.2877217E 17	3.9926549E 15	1.0008113E 14	1.9949714E 12
3.4274584E 10	4.7956796E 08	5.9506779E 06	5.8881896E 04	5.5416836E 02	3.8551691E 00
2.8002674E-02	1.3359390E-04	7.3606134E-07	2.2365276E-09	8.7580072E-12	1.3306326E-14
3.0508488E-17					

THE ROOTS OF B

-5.0000000E-01	+I	0.0000000E 00	-1.3437380E-02	+I	8.4953454E-02	-1.3437380E-02	+I	-8.4953454E-02
-8.5697546E-01	+I	2.2620353E 00	-8.5697546E-01	+I	-2.2620353E 00	-2.0000000E 01	+I	-8.0754456E-14
-1.2500000E 01	+I	-8.7707630E-14	-4.6644787E-01	+I	5.1981022E 01	-4.6644787E-01	+I	-5.1981022E 01
-4.3478261E 01	+I	4.4974263E 01	-4.3478261E 01	+I	-4.4974263E 01	-3.0000000E 01	+I	0.0000000E 00
-6.0901678E-01	+I	6.0974886E 01	-6.0901678E-01	+I	-6.0974886E 01	-1.5634793E 00	+I	1.1108566E 02
-1.5634793E 00	+I	-1.1108566E 02	-2.6156110E 00	+I	1.3130735E 02	-2.6156110E 00	+I	-1.3130735E 02
-1.9225814E 00	+I	1.5928538E 02	-1.9225814E 00	+I	-1.5928538E 02	-3.1400000E 01	+I	1.8535922E 02
-3.1400000E 01	+I	-1.8535922E 02	-1.0365000E 02	+I	3.2947940E 02	-1.0365000E 02	+I	-3.2947940E 02

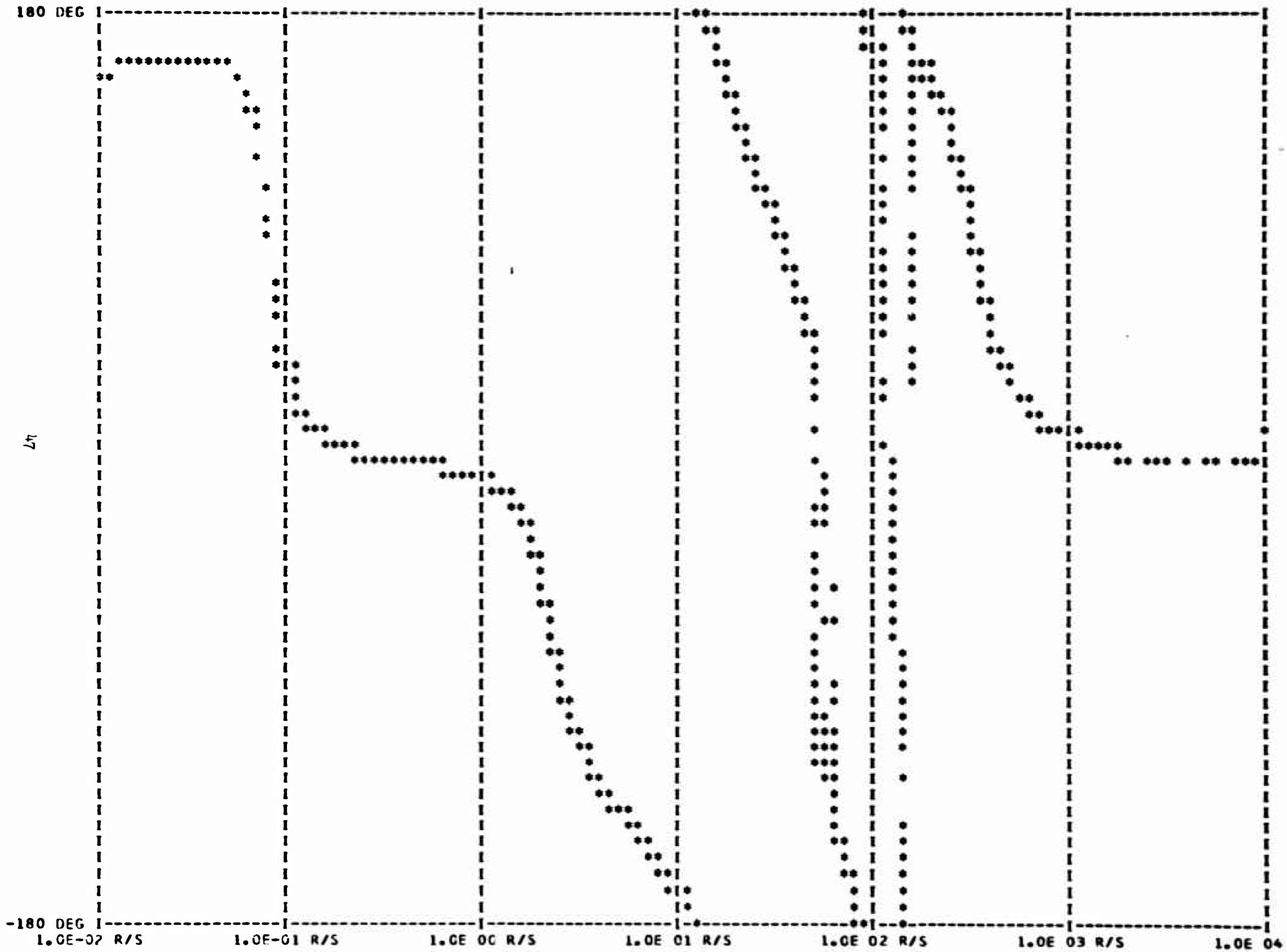


FIGURE 17. EXAMPLE 2. LINEAR SYSTEM, BODE PLOT, PHASE ANGLE

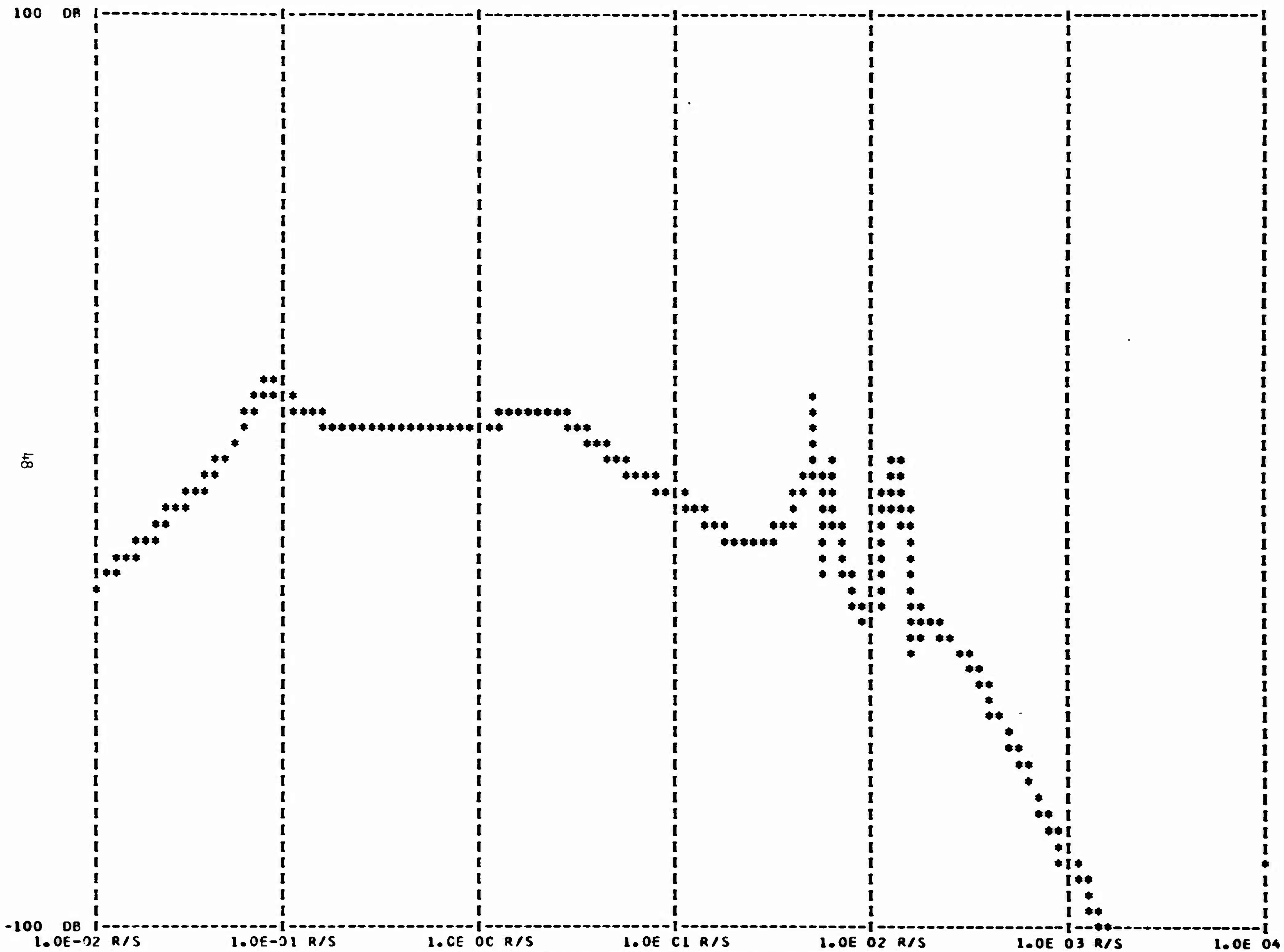


FIGURE 18. EXAMPLE 2. LINEAR SYSTEM BODE PLOT, MAGNITUDE

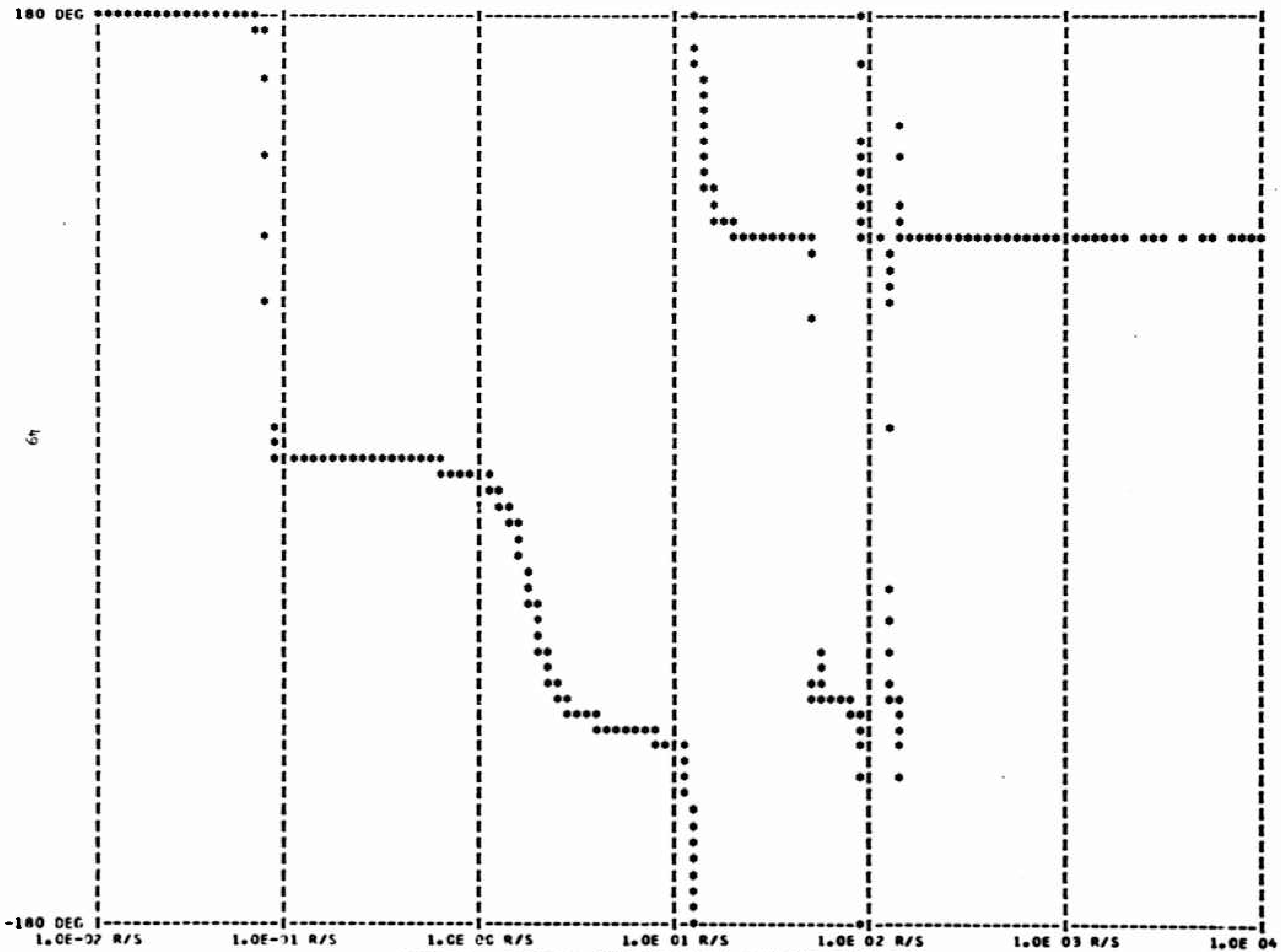


FIGURE 19. EXAMPLE 2, NONLINEAR SYSTEM, BODE PLOT, PHASE ANGLE

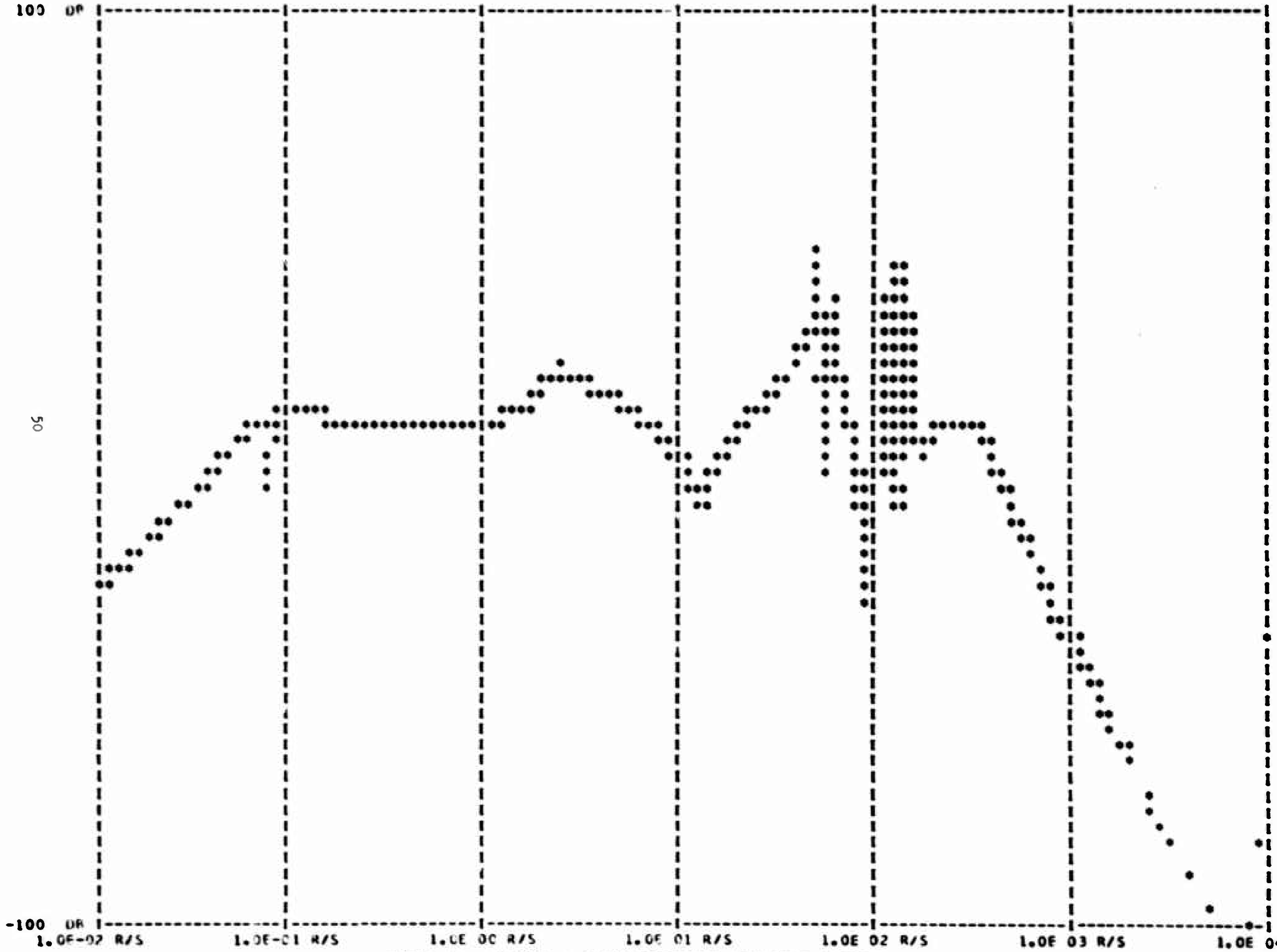


FIGURE [illegible]. EXAMPLE [illegible]. NONLINEAR SYSTEM, BODE PLOT, MAGNITUDE

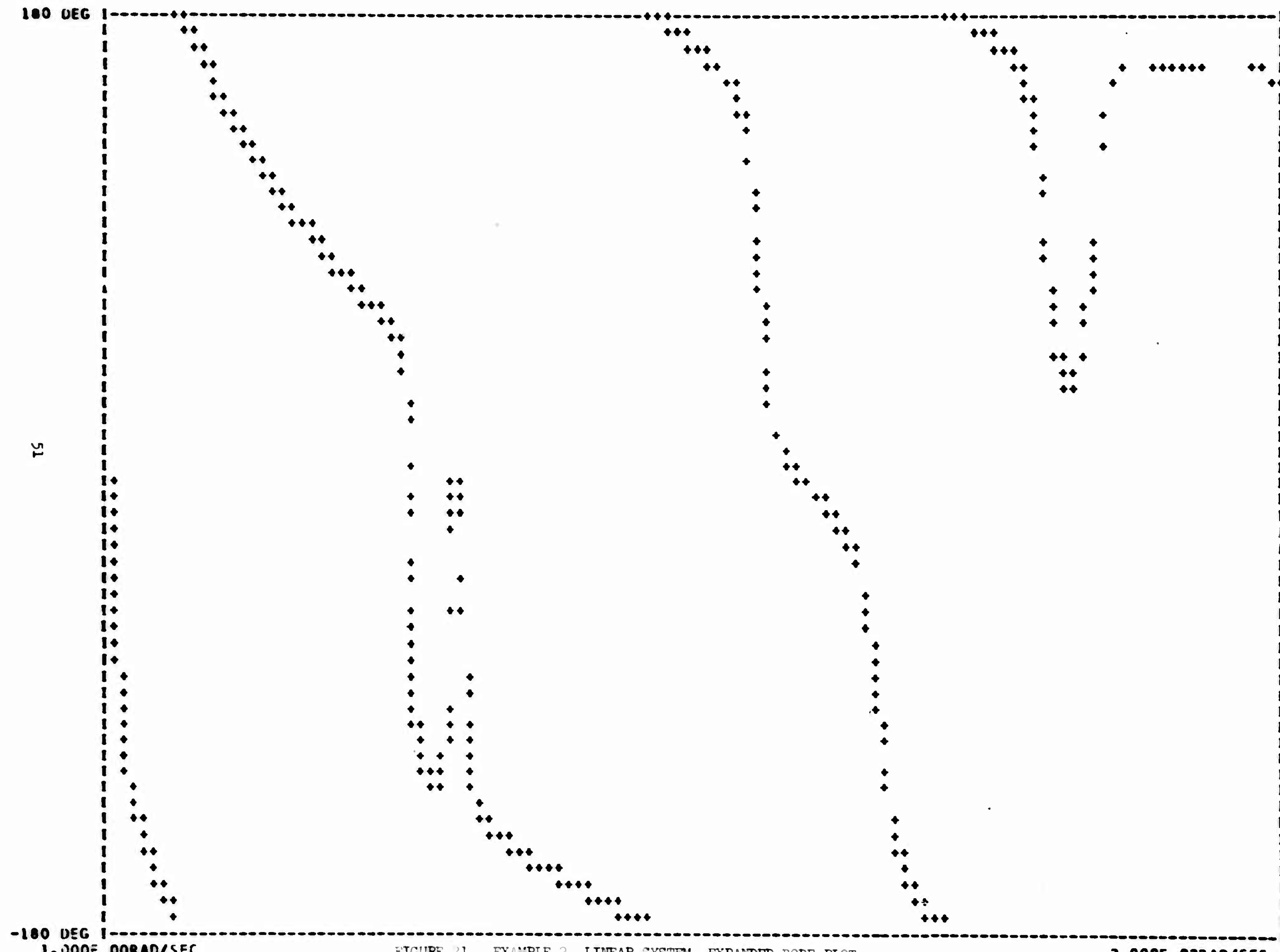


FIGURE 21. EXAMPLE 2, LINEAR SYSTEM, EXPANDED BODE PLOT (1 TO 200 RAD/S) PHASE ANGLE

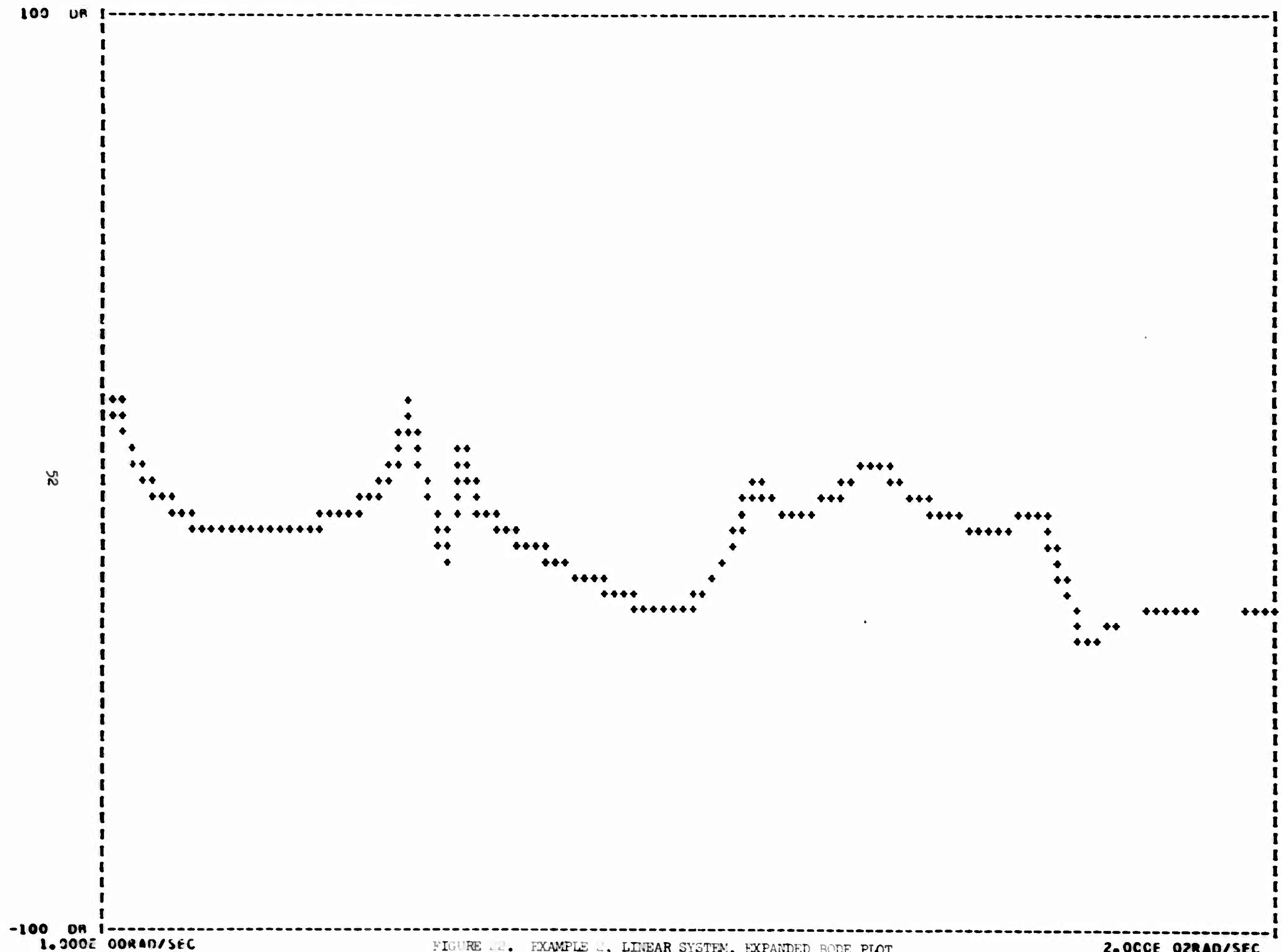


FIGURE 12. EXAMPLE 2, LINEAR SYSTEM, EXPANDED BODE PLOT
(1 TO 200 RAD/S) MAGNITUDE

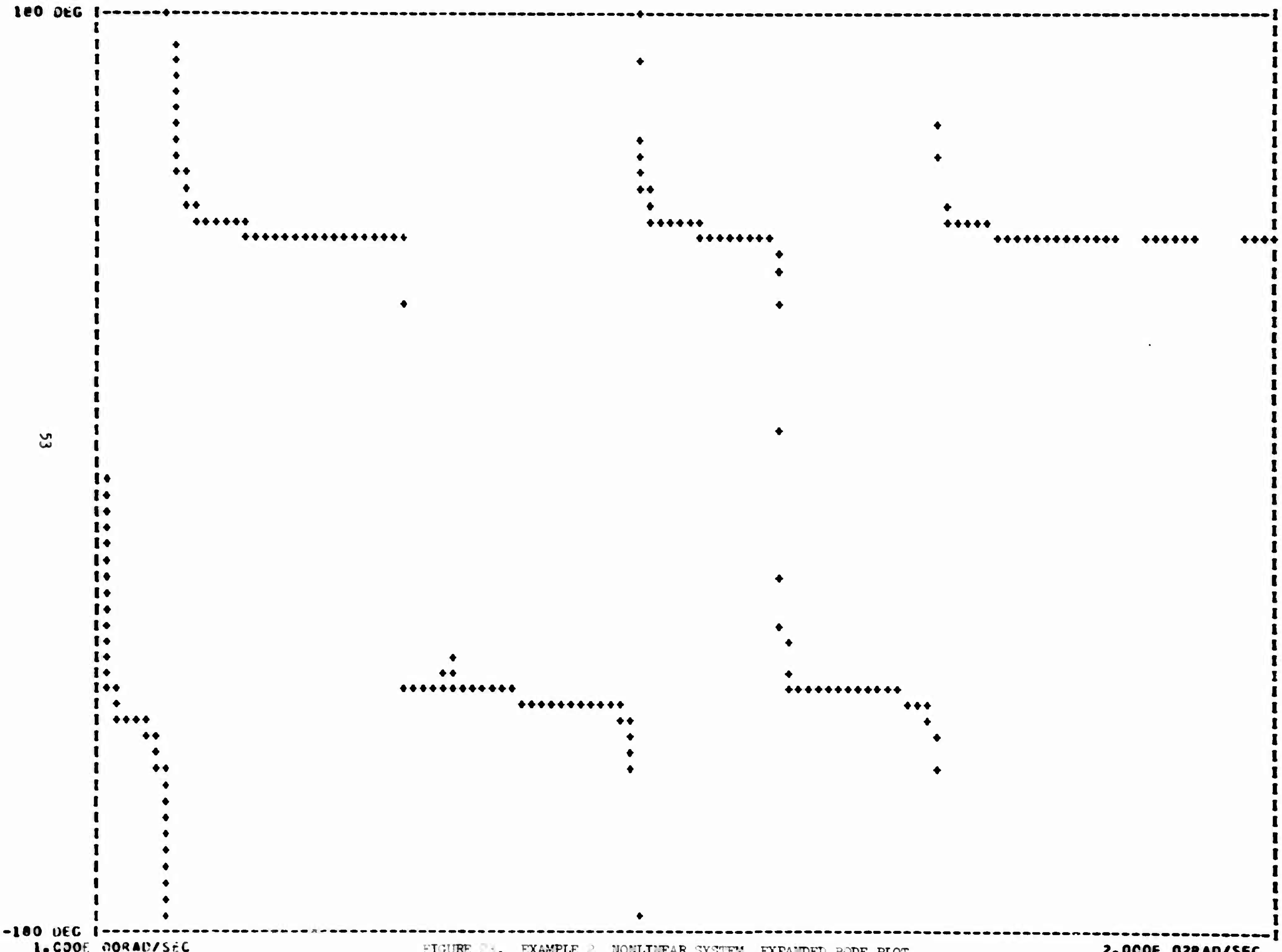


FIGURE [illegible]. EXAMPLE 2. NONLINEAR SYSTEM, EXPANDED BODE PLOT
(1 TO 200 RAD/S) PHASE ANGLE

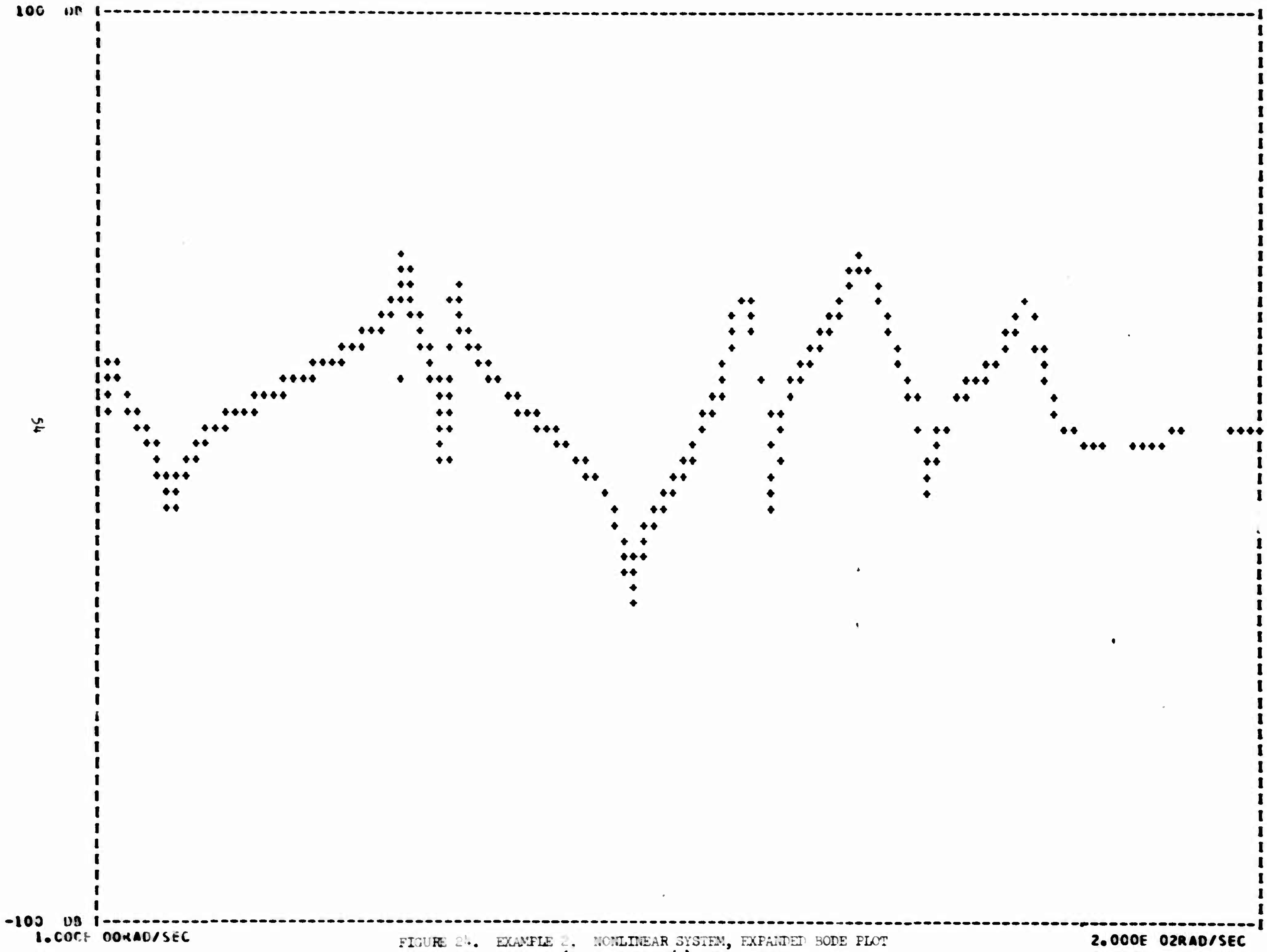


FIGURE 24. EXAMPLE 2. NONLINEAR SYSTEM, EXPANDED BODE PLOT
(1 TO 200 RAD/S) MAGNITUDE

REFERENCES

1. Burke, Harold H., Payne, Robert L. Jr.; A Linear Closed Loop Systems Analysis Procedure Using Line Printer Plots of Characteristic Equation Root Loci, AMSAA Technical Memorandum No. 21, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md., November 1968, UNCLASSIFIED.

2. Burke, Harold H., Andrese, Joseph A.; A Fortran IV Program to Compute the Inverse Laplace Transform and Plot the Response of a Linear System. AMSAA Technical Memorandum No. 60, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md., (In publication), UNCLASSIFIED.

3. Burke, Harold H.; Mathematical Model and Simulation Studies of a Pneumatic Valve-Actuator-Load Impedance Configuration, AMSAA Technical Report No. 25, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md., April 1969, UNCLASSIFIED.

4. Burke, Harold H., Ogorzalek, M. J.; A Surface-To-Air Tactical Missile Guidance and Control Concept for the Advanced Forward Area Air Defense (AFAAD) Requirement, AMSAA Technical Memorandum No. 31, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md., September 1969 (CONFIDENTIAL).

5. Brown, James W.; A Preliminary Study of Shillelagh in a Combat, Optical Countermeasure Environment, AMSAA Technical Memorandum No. 62, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md., (In publication) (SECRET).

6. Sivaglian, B. D.; An Asymptotic Study of a Stationary (σ-s) Inventory Problem, Technical Report No. 8, June 1968, Themis Project, Systems Engineering Department, University of Florida, Gainsville, Fla.

7. Leon, Jesus, Herrero, Emilio E.; Determination of the Time-Domain Equivalents of Higher Order Laplace Transfer Functions By Analog Computation, Themis Project, Systems Research Center Industrial and Systems Engineering Department, University of Florida, Gainsville, Fla., November 1968.

8. Lefschetz, Solomon, Stability of Nonlinear Control Systems, Academic Press, New York, 1965.

9. Greensite, Arthur L.; Analysis and Design of Space Vehicle Flight Control Systems, Volume IV, Nonlinear Systems, NASA CR 823, July 1967.

Next page is blank.

APPENDIX A

LISTING OF SOURCE DECK

Next page is blank.

```
$     MAXT(10)MINS
         DO 10 J=1,100
      CALL RTLOCS
   10 CONTINUE
      END
C
C
C
C
      SUBROUTINE RTLOCS
      DIMENSION SAVE1(100,100),SAVE2(100,100),XA(2000),XB(2000)
      DIMENSION X(100),IGA(100),IGB(100),C(100),D(100),ANS(100),
     1SAVE(100),ERA(100),AS(100),BS(100),A(100),B(100),ROOTR(100),
     2ROOTI(100),ATK(100),CK(100),CKS(100)
      INTEGER COMENT(20),DENSE,ROTLCS,EXPND,FREQSR,STAR,STAR1,STAR2,
     *        STAR3,ONE,ZERO,DOT,VEE,AYE,DASH,BLANK,Q,
     *        POINT(130,100),SPOT(130,50)
      DATA ONE/1H1/,ZERO/1H0/,DOT/1H./,AYE/1HA/,VEE/1HV/,BLANK/1H /,
     *     DASH/1H-/,Q/1HI/,STAR1/1HO/,STAR2/1H*/,STAR3/1H./
      LOGICAL PRNT
      EQUIVALENCE (SPOT(1,1),POINT(1,1)),(STAR1,ZERO),(STAR3,DOT),
     *            (JACKIE,KJ,I),(JOANN,KK,J)
      COMMON/FREEK/A,B
      COMMON /INFO4/ Q,BLANK,DASH
      COMMON /INFO7/ ICOUNT
      COMMON SAVE1,SAVE2,POINT
      COMMON /INFO8/ROTLCS,EXPND,FREQSR
      COMMON /INFO9/ DENSE
      READ (5,105) ROTLCS,EXPND,FREQSR,DENSE
      READ(5,11) COMENT
      WRITE(6,13) COMENT
      IF(ROTLCS.NE.0.AND.EXPND.NE.0.AND.FREQSR.NE.0) GO TO 37
      IF(ROTLCS.EQ.0.AND.EXPND.EQ.0) WRITE(6,30)
      IF(FREQSR.EQ.0.AND.EXPND.EQ.0) WRITE(6,31)
      IF(FREQSR.EQ.0.AND.ROTLCS.EQ.0) WRITE(6,32)
      IF(ROTLCS.NE.0.AND.EXPND.NE.0) WRITE(6,33)
      IF(FREQSR.NE.0.AND.EXPND.NE.0) WRITE(6,34)
      IF(FREQSR.NE.0.AND.ROTLCS.NE.0) WRITE(6,35)
   38 DO277 IOU=1,2000
      XA(IOU)=0.0
  277 XB(IOU)=0.0
      DO 401 KK=1,100
      DO 401 KJ=1,130
  401 POINT(KJ,KK)=BLANK
      DO 402 KK=1,130
      POINT(KK,51)=DASH
  402 POINT(KK,50)=DASH
      DO 403 KK=1,100
      DO 403 KJ=34,125,13
  403 POINT(KJ,KK)=Q
      DO 404 KJ=34,124
      DO 404 KK=7, 93,7
      IF(KK.EQ.49) KK=58
  404 POINT(KJ,KK)=DASH
      DO 405 KK=7, 93,7
      IF(KK.EQ.49) KK=58
  405 POINT(25,KK)=ONE
      DO 406 KK=26,28
  406 POINT(KK,7)=ZERO
```

```
      DO 407 KK=26,27
  407 POINT(KK,14)=ZERO
      POINT(26,21)=ZERO
      DO 408 KK=35,65,7
      IF(KK.EQ.49) KK=58
  408 POINT(24,KK)=DOT
      POINT(25,42)=ZERO
      POINT(25,58)=ZERO
      POINT(26,42)=ONE
      POINT(26,58)=ONE
      POINT(26,79)=ZERO
      POINT(26,86)=ZERO
      POINT(27,86)=ZERO
      DO 409 KK=26,28
  409 POINT(KK,93)=ZERO
      DO 420 KK=1,100
      IF(KK.EQ.4) KK=98
  420 POINT(26,KK)=O
      POINT(26,4)=VEE
      POINT(26,97)=AYE
      DO 7681 JACKIE=1,100
      DO 7681 JOANN=1,100
      SAVE1(JACKIE,JOANN)=0.0
 7681 SAVE2(JACKIE,JOANN)=0.0
      NO=1
      ICOUNT=0
      MI=0
      JZO=1
      K1=1
      III=1
  140 FORMAT(10X,7HDELTAK=,I10,5X,19HPOLY.ADDED IN A(S)=,I10,5X,19HPOLY.
     *ADDED IN B(S)=,I10,5X,9HPROB.NO.=,I10)
   12 READ (5,101) N,IA,IB,IPROB
      IF(IB.NE.0) GO TO 10
      IF(IA.NE.0) GO TO 10
      IF( N.NE.0) GO TO 10
      IF(IPROB.EQ.10000) GO TO 1016
101   FORMAT (7I10)
   10 WRITE(6,1)
      DO 5 I = 1,100
      A(I) = 0.0
5     B(I) = 0.0
      FORK1=0.0
      FORK2 = 0.0
      FORK3 = 0.0
      WRITE (6 ,102)IPROB
  102 FORMAT (1H1, 9X,40HPOLYNOMIAL MULTIPLICATION AND ROOT LOCUS,44X,11
     1HPROBLEM NO.,I5//)
      IF (N) 21, 20, 21
   20 READ(5,103)Y,DY,YT
      WRITE(6,140)N,IA,IB,IPROB
      WRITE(6,141)Y,DY,YT
  141 FORMAT(1H0,5X,10HK-INITIAL=F20.10,5X,12HINCREMENT K=,F20.10,5X,12H
     *K-TERMINATE=,F20.10)
103   FORMAT (4F10.0)
      GO TO 25
   21 READ(5,104)(X(I),I=1,N)
      WRITE(6,142) (I,X(I),I=1,N)
  104 FORMAT(7E10.0)
25    READ (5 ,105)(IGA (I), I = 1, IA)
```

```
      WRITE(6,143) (I,IGA(I),I=1,IA)
  143 FORMAT(10X,24HNUMBER OF POLY. IN GROUP, I[illegible]F NUMERATOR=,I10)
  142 FORMAT(10X,2HX(,I3,2H)=,F20.10)
      READ (5 ,105)(IGB(J), J= 1, IB)
      WRITE(6,144) (J,IGB(J),J=1,IB)
  144 FORMAT(10X,24HNUMBER OF POLY. IN GROUP, I3,16H OF DENOMINATOR=,I10
     *)
105   FORMAT(7I10)
 300  DO 15 I =  1,100
15    SAVE(I) = 0.0
      ISDEG = 0
      JJ = 1
200   READ (5 ,106)MDEG,( C(I), I=1, 6)
      IF(MDEG.GT.6) GO TO 48
      WRITE(6,145) (I,C(I),I=1,MDEG)
      IF (MDEG - 6) 49,  49,  48
48    READ (5 ,104)(C(I), I = 7, MDEG)
      WRITE(6,145) (I,C(I),I=1,MDEG)
 106  FORMAT(I10,6E10.0)
49    IF (IGA(JJ)-1)  50, 51, 50
51    DO 56  I= 1, MDEG
56    ANS(I) = C(I)
      IADEG = MDEG
      IF (ISDEG - MDEG)  52, 52, 53
53    INDEG = ISDEG
      GO TO 68
52    INDEG = MDEG
      GO TO 68
50    READ (5 ,106)NDEG, (D(I), I =1, 6)
      IF(NDEG.GT.6) GO TO 54
      WRITE(6,145) (I,D(I),I=1,NDEG)
      IF (NDEG - 6)  55, 55, 54
54    READ (5 ,104)(D(I), I=7, NDEG)
      WRITE(6,145) (I,D(I),I=1,NDEG)
55    IADEG = NDEG + MDEG -1
      CALL POLMPY (C,MDEG,D,NDEG,ANS)
      IGA (JJ) = IGA(JJ) -1
      IF (IGA(JJ) -1) 65, 65, 64
64    DO 60  I = 1, IADEG
60    C(I) = ANS(I)
      MDEG =IADEG
      GO TO 50
65    IF (ISDEG - IADEG) 66,66,67
66    INDEG = IADEG
      GO TO 68
67    INDEG = ISDEG
68    CALL POLADD (SAVE,ISDEG,ANS,IADEG,ERA)
  145 FORMAT(10X,2HC(,I3,2H)=,F20.10)
      WRITE(6,1)
    1 FORMAT(1H1//)
6800  IF (ERA(INDEG))6803, 6802, 6803
6802  INDEG = INDEG - 1
      IF (INDEG)  6801, 6801, 6800
6801  INDEG = 1
6803  JJ = JJ + 1
      DO 70 I = 1, INDEG
70    SAVE(I) = ERA(I)
      ISDEG = INDEG
      IA = IA -1
      IF (IA) 201, 201, 200
```

```
 201  IF (FORK1) 202, 202, 203
C     SAVE NUMERATOR.
 202  DO 220 I = 1, ISDEG
 220  A(I) = SAVE(I)
      IDA = ISDEG
      IA = IB
      FORK1 = 1.0
      DO 230 I=1, IA
 230  IGA(I) = IGB(I)
C     START DENOMINATOR
      GO TO 300
C     SAVE DENOM.
 203  DO 240 I = 1, ISDEG
 240  B(I) = SAVE(I)
      IDB = ISDEG
      WRITE (6 ,109)
 109  FORMAT (10X,41HCOEFFICIENTS ARE GIVEN IN ASCENDING ORDER/////)
 339  IF (A(IDA)) 340, 341, 340
 341  IDA = IDA - 1
      IF (IDA)  345, 345, 339
 345  WRITE (6 ,120)
 120  FORMAT (1H0,10X,20HPOLYNOMIAL A IS ZERO//)
      FORK2 = 1.0
      GO TO 410
 340  IF (IDA - 2) 346, 347, 335
  346 WRITE(6,121)A(1)
      STAR=STAR1
      PRNT=.TRUE.
 121  FORMAT (1H0,10X,28HPOLYNOMIAL A IS A CONSTANT =,1P1E16.7//)
      GO TO  410
 347  ROOT = - A(1) / A(2)
      WRITE (6 ,133)A(1), A(2)
 133  FORMAT (10X,21HTHE COEFFICIENTS OF A/1P2E20.7)
      WRITE (6 ,122)ROOT
      STAR=STAR1
      PRNT=.TRUE.
      ANUMB1=ROOT
      ANUMB2=0.0
      CALL PLOTER(SAVE1,SAVE2,ANUMB1,ANUMB2,POINT,XA,XB,MI,PRNT,STAR,III
     *,NO)
 122  FORMAT (1H0,10X,23HROOT OF POLYNOMIAL A IS,1P1E16.7//)
      GO TO  410
C     WRITE POLYS
 335  ID1A =IDA -1
      WRITE (6,107)ID1A,(A(I),I=1,IDA)
      K = IDA
      DO 800 I = 1,IDA
      AS(I) = A(K)
 800  K = K-1
      IDP2A=IDA *2
      ID2A= 2 *ID1A
      CALL MULLER (AS,ID1A,ROOTR,ROOTI)
      DO 805 I = 1,ID1A
      SAM = 100. * AMAX1(ABS(ROOTR(I)),ABS(ROOTI(I)))
      IF (SAM + ABS(ROOTR(I)).EQ. SAM) ROOTR(I)= 0.0
      IF (SAM + ABS(ROOTI(I)).EQ. SAM) ROOTI(I)= 0.0
  805 CONTINUE
 400  WRITE (6,111) (ROOTR(I),ROOTI(I),I=1,ID1A)
      CALL ERCHEK(ROOTI,ID1A)
      PRNT=.TRUE.
```

```
      STAR=STAR1
      DO 2 III=1,ID1A
      ANUMB1=ROOTR(III)
      ANUMB2=ABS(ROOTI(III))
      CALL PLOTER(SAVE1,SAVE2,ANUMB1,ANUMB2,POINT,XA,XB,MI,PRNT,STAR,III
     *,NO)
    2 CONTINUE
410   IF(B(IDB)) 411, 412, 411
412   IDB = IDB - 1
      IF (IDB)  445, 445, 410
445   WRITE (6 ,123)
123   FORMAT (1H0,10X,20HPOLYNOMIAL B IS ZERO//)
      IF (FORK2)12,450,12
450   FORK3 = 1.0
      GO TO 698
411   IF (IDB - 2)  451, 452, 499
451   WRITE (6 ,124)B(IDB)
      STAR=STAR2
      PRNT=.FALSE.
      STAR=STAR3
124   FORMAT (1H0,10X,28HPOLYNOMIAL B IS A CONSTANT =,1P1E16.7//)
      GO TO 698
452   ROOT = -B(1) / B(2)
      WRITE (6 ,134)B(1), B(2)
134   FORMAT (10X,21HTHE COEFFICIENTS OF B/1P2E20.7)
      WRITE (6 ,125)ROOT
      STAR=STAR2
      PRNT=.TRUE.
      ANUMB1=ROOT
      ANUMB2=0.0
      CALL PLOTER(SAVE1,SAVE2,ANUMB1,ANUMB2,POINT,XA,XB,MI,PRNT,STAR,III
     *,NO)
      PRNT=.FALSE.
      STAR=STAR3
125   FORMAT (1H0,10X,23HROOT OF POLYNOMIAL B IS,1P1E16.7//)
      GO TO 698
  107 FORMAT (10X,42HTHE COEFFICIENTS OF POLYNOMIAL A (ORDER = I3,1H)/ (
     11P6E20.7))
  499 ID1B = IDB -1
      WRITE (6,108)ID1B,(B(I),I=1,IDB)
  108 FORMAT (////10X,42HTHE COEFFICIENTS OF POLYNOMIAL B (ORDER = I3,1H
     1)/ (1P6E20.7))
      K = IDB
      DO 801 I = 1,IDB
      BS(I) = B(K)
801   K = K-1
      IDP2B= IDB * 2
      ID2B = 2 * ID1B
      CALL MULLER (BS,ID1B,ROOTR,ROOTI)
      DO 806 I = 1,ID1B
      SAM = 100. * AMAX1(ABS(ROOTR(I)),ABS(ROOTI(I)))
      IF (SAM + ABS(ROOTR(I)).EQ. SAM) ROOTR(I)= 0.0
      IF (SAM + ABS(ROOTI(I)).EQ. SAM) ROOTI(I)= 0.0
  806 CONTINUE
500   WRITE (6,112)(ROOTR(I),ROOTI(I),I= 1,ID1B)
      CALL ERCHEK(ROOTI,ID1B)
      STAR=STAR2
      PRNT=.TRUE.
      DO 3 III=1,ID1B
      ANUMB1=ROOTR(III)
```

```
      ANUMB2=ABS(ROOTI(III))
      CALL PLOTER(SAVE1,SAVE2,ANUMB1,ANUMB2,POINT,XA,XB,MI,PRNT,STAR,III
     *,NO)
    3 CONTINUE
      PRNT=.FALSE.
      STAR=STAR3
  111 FORMAT (1H0,11X,14HTHE ROOTS OF A/ (1P1E20.7,6H    +I ,1P1E14.7,1P1
     1E20.7,6H    +I ,1P1E14.7,1P1E20.7,6H    +I ,1P1E14.7))
  112 FORMAT (1H0,11X,14HTHE ROOTS OF B/ (1P1E20.7,6H    +I ,1P1E14.7,1P1
     1E20.7,6H    +I ,1P1E14.7,1P1E20.7,6H    +I ,1P1E14.7))
  698 IF (FORK2)12,699,12
  699 IF (FORK3)12,6991,12
 6991 WRITE (6 ,102)IPROB
      MSHEET = 5
C     START K CALCULATIONS
      IF (N) 702,702,533
  533 DO 550 I= 1, N
      DO 541 J= 1, IDA
  541 ATK(J) =  X(I) * A(J)
C     COMPUTE ROOTS OF K * A + B
      IDC= MAX0( IDA, IDB)
      CALL POLADD (ATK,IDA,B,IDB,CK)
      IDS = IDC
  554 IF (CK(IDS))555, 557, 555
  557 IDS = IDS - 1
      IF (IDS) 558,558, 554
  558 WRITE (6 ,129)X(I)
  129 FORMAT (1H0,10X,35HPOLYNOMIAL K*A + B  IS ZERO FOR K =,1P1E16.7//)
      GO TO 550
  555 IF (IDS - 2)  559, 560, 561
  559 WRITE (6 ,130)CK(IDS), X(I)
  130 FORMAT (1H0,10X,35HPOLYNOMIAL K*A + B IS A CONSTANT = ,1P1E15.7,10
     1H FOR K =  ,1P1E14.7//)
      GO TO 550
  560 ROOT = -CK(1) / CK(2)
      WRITE (6 ,131)ROOT, X(I)
  131 FORMAT (1H0,10X,18HROOT OF K*A + B = ,1P1E15.7,10H FOR K =  ,1P1E1
     14.7//)
      GO TO 550
  561 K = IDS
      DO 803 J = 1,IDS
      CKS(J) = CK(K)
  803 K = K - 1
      ID1C = IDS - 1
      IDP2C = IDS * 2
      ID2C  = 2  * ID1C
      CALL MULLER (CKS,ID1C,ROOTR,ROOTI)
      DO 807 J = 1,ID1C
      SAM = 100. * AMAX1(ABS(ROOTR(J)),ABS(ROOTI(J)))
      IF (SAM + ABS(ROOTR(J)).EQ.SAM)  ROOTR(J) = 0.0
      IF (SAM + ABS(ROOTI(J)).EQ.SAM)  ROOTI(J) = 0.0
  807 CONTINUE
      WRITE (6,808)ID1C,X(I),(CK(J),J=1,IDS)
  808 FORMAT (///10X,48HTHE COEFFICIENTS OF POLYNOMIAL K*A + B (ORDER =
     1I3,7H)  K = 1P1E16.7/(1P6E20.7))
  545 WRITE (6,115)(ROOTR(J),ROOTI(J),J=1,ID1C)
      CALL ERCHEK(ROOTI,ID1C)
      CALL SAVER(ROOTR,ROOTI,ID1C,SAVE1,SAVE2,JZO,K1)
  115 FORMAT (1H0,9X,16HROOTS OF K*A + B/(1P1E20.7,6H  + I ,1P1E14.7,1P1
     1E20.7,6H  + I ,1P1E14.7,1P1E20.7,6H  + I ,1P1E14.7))
```

```
5452 MSHEET = MSHEET - 1
     IF (MSHEET) 546, 546, 550
 546 WRITE (6 ,102)IPROB
     MSHEET = 5
 550 CONTINUE
     GO TO 12
 702 DO  705  J = 1, IDA
 705 ATK(J) = Y * A(J)
C    COMPUTE ROOTS OF K * A + B
     IDC= MAXO(IDA, IDB)
     CALL POLADD (ATK,IDA,B,IDB,CK)
     IDS = IDC
 754 IF (CK(IDS))755, 757, 755
 757 IDS = IDS - 1
     IF (IDS) 758, 758, 754
 758 WRITE (6 ,129)Y
     GO TO 711
 755 IF (IDS - 2) 759, 760, 761
 759 WRITE (6 ,130)CK(IDS), Y
     GO TO 711
 760 ROOT = -CK(1) / CK(2)
     WRITE (6 ,131)ROOT, Y
     GO TO 711
 761 K = IDS
     DO 804 I = 1,IDS
     CKS(I) = CK(K)
 804 K = K -1
     ID1C = IDS - 1
     IDP2C = IDS * 2
     ID2C  = 2 * ID1C
     CALL MULLER (CKS,ID1C,ROOTR,ROOTI)
     DO 809 I = 1,ID1C
     SAM = 100. * AMAX1(ABS(ROOTR(I)),ABS(ROOTI(I)))
     IF (SAM + ABS(ROOTR(I)).EQ. SAM) ROOTR(I)= 0.0
     IF (SAM + ABS(ROOTI(I)).EQ. SAM) ROOTI(I)= 0.0
 809 CONTINUE
     WRITE (6,808)ID1C,Y,(CK(I),I=1,IDS)
     WRITE (6,115) (ROOTR(J),ROOTI(J),J=1,ID1C)
     CALL ERCHEK(ROOTI,ID1C)
     CALL SAVER(ROOTR,ROOTI,ID1C,SAVE1,SAVE2,JZO,K1)
 711 Y = Y + DY
     IF (Y - YT) 712,712,12
 712 MSHEET = MSHEET - 1
     IF (MSHEET) 713, 713, 702
 713 WRITE (6 ,102)IPROB
     MSHEET = 5
     GO TO 702
  37 WRITE(6,36)
     GO TO 38
1016 CALL PLOTER(SAVE1,SAVE2,ANUMB1,ANUMB2,POINT,XA,XB,MI,PRNT,STAR,III
    *,NO)
1018 IF(EXPND.EQ.0) GO TO 1017
     READ(5,22)L
  22 FORMAT(I10)
     IF(L.EQ.0) GO TO 1017
     CALL EXPAND(L,XA,XB)
1017 IF(FREQSR.EQ.0) RETURN
     CALL FREQRS(XA,XB)
  11 FORMAT(20A4)
  13 FORMAT(///////////,20X,20A4)
```

```
   30 FORMAT(////20X,44HONLY THE FREQUENCY RESPONSE HAS BEEN PLOTTED    )
   31 FORMAT(////20X,45HONLY THE ROOT LOCUS LOG PLOT HAS BEEN PLOTTED   )
   32 FORMAT(////20X,39HONLY THE LINEAR EXPAND HAS BEEN PLOTTED    )
   33 FORMAT(////20X,59HTHE ROOT LOCUS LOG PLOT AND LINEAR EXPAND HAVE B
     *EEN PLOTTED      )
   34 FORMAT(////20X,62HTHE LINEAR EXPAND AND THE FREQUENCY RESPONSE HAV
     *E BEEN PLOTTED    )
   35 FORMAT(////20X,68HTHE ROOT LOCUS LOG PLOT AND THE FREQUENCY RESPON
     *SE HAVE BEEN PLOTTED     )
   36 FORMAT(////20X,28HEVERYTHING HAS BEEN PLOTTED      )
      RETURN
      END
C
C
C
C
      SUBROUTINE POLMPY (A,N,B,M,C)
      DIMENSION A(1),B(1),C(1)
      K = M+N
      DO 5 I=1,K
 5    C(I) = 0.0
      DO 10 I=1,N
      L = I-1
      DO 10 J=1,M
      L = L+1
 10   C(L) = C(L)+A(I)*B(J)
      RETURN
      END
C
C
C
C
      SUBROUTINE POLADD (A,N,B,M,C)
      DIMENSION A(1),B(1),C(1)
      IF (N-M) 1,1,2
 1    NK = N
      GO TO 5
 2    NK = M
 5    DO 10 I=1,NK
 10   C(I) = A(I)+B(I)
      NK = NK+1
      IF (N-M) 11,25,15
 11   DO 20 I=NK,M
 20   C(I) = B(I)
 25   RETURN
 15   DO 30 I=NK,N
 30   C(I) = A(I)
      RETURN
      END
C
C
C
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.
      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=C.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 = SQRT (TE1*TE1+TE2*TE2)
      IF(TE1)113,113,112
113    TE4 = SQRT (.5 * (TEM - TE1))
      TE3=.5*TE2/TE4
      GO TO 111
112    TE3 = SQRT (.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 (ABS (HELL) + ABS (BELL) - 1.E-20) 18,18,16
16   TE7 = ABS (ALP3R - AXR) + ABS (ALP3I - AXI)
     IF (TE7 / (ABS (AXR) + ABS (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
 41 IF(N4-N1)30,37,37
 37 RETURN
30   IF (ABS (ROOTI(N4)) - 1.E-5)10,10,31
 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
      RELL=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
C
C
C
C
      SUBROUTINE EXPLOT(XARRAY,IDIMEN,YARRAY,KDIMEN,BLIMIT,LET)
      INTEGER POINT(130,100),GRAPHL(120,58),STAR
      DIMENSION XARRAY(IDIMEN),YARRAY(KDIMEN)
      DIMENSION ENCRM1(120),ENCRM2(58)
      DIMENSION SAVE1(100,100),SAVE2(100,100)
      COMMON SAVE1,SAVE2,POINT
      EQUIVALENCE(POINT(1,1),GRAPHL(1,1))
      DATA STAR/1H*/
      WRITE(6,1)
      READ(5,10)D1,D2
      D3=BLIMIT
      D4=-BLIMIT
      IF(D2.GT.D1) CALL SWITCH(D1,D2)
      IF(D4.GT.D3) CALL SWITCH(D3,D4)
      IC1=D3
      JDIMEN=KDIMEN
      CALL BLANKR(GRAPHL,120,58,120)
      DO 22 I=1,JDIMEN
      X=XARRAY(I)
      Y=YARRAY(I)
      IF(X.EQ.0.0.AND.Y.EQ.0.0) GO TO 22
      CALL LINAR (X,LIMITX,120,D1,D2,ENCRM1)
      CALL LINAR (Y,LIMITY,58,D3,D4,ENCRM2)
      IF(LIMITX.EQ.0.OR.LIMITY.EQ.0) GO TO 22
      LIMITX=121-LIMITX
      GRAPHL(LIMITX,LIMITY)=STAR
   22 CONTINUE
      WRITE(6,30) IC1,LET,(GRAPHL(I,1),I=1,120)
      IC1=D4
      WRITE(6,34)((GRAPHL(I,J),I=1,120),J=2,57)
      WRITE(6,30) IC1,LET,(GRAPHL(I,58),I=1,120)
      WRITE(6,35) D2,D1
    1 FORMAT(1H1)
   10 FORMAT(6F10.0,20X)
```

```
   30 FORMAT(1X,I4,A4,1X,120A1)
   34 FORMAT(10X,120A1)
   35 FORMAT(3X,1P1E10.3,7HRAD/SEC,91X,1P1E10.3,7HRAD/SEC)
   61 RETURN
      END
C
C
C
C
      SUBROUTINE LINAR (Y,LIMITY,IDIMEN,DIMNS1,DIMNS2,ENCRMT)
      DIMENSION ENCRMT(IDIMEN)
      DIMENS=DIMNS1-DIMNS2
      LIMITY=0
      J=IDIMEN-1
      A=J
      DELTA=DIMENS/A
      ENCRMT(1)=DIMNS1
      ENCRMT(IDIMEN)=DIMNS2
      DO 10 I=2,J
   10 ENCRMT(I)=ENCRMT(I-1)-DELTA
      DO 11 I=1,J
   11 IF(Y.LE.ENCRMT(I).AND.Y.GE.ENCRMT(I+1))GO TO 20
      GO TO 40
   20 LIMITY=I
   40 RETURN
      END
C
C
      SUBROUTINE SWITCH(X,Y)
      TEST=Y
      Y=X
      X=TEST
      RETURN
      END
C
C
      SUBROUTINE ERCHEK(X,I)
      DIMENSION X(100)
      DATA ERRLIM/0.1E-8/
      DO 10 J=1,I
   10 IF(ABS(X(J)).LT.ERRLIM) X(J)=0.0
      RETURN
      END
C
C
C
C
      SUBROUTINE EXPAND(ILOVE,X,Y)
      DIMENSION SAVE1(100,100),SAVE2(100,100),X2(2000),Y2(2000),X(2000),
     *          Y(2000),X1(2000),Y1(2000),Z(125),Z1(87)
      INTEGER SPOT(130,50),DASH,Q,BLANK,POINT(130,100)
      DATA Q/1HI/, DASH/1H-/,BLANK/1H /
      DATA APLUS/0.0/,BPLUS/0.0/,AMINUS/0.0/,BMINUS/0.0/
      COMMON /INFO4/ Q,BLANK,DASH
      COMMON /PAYNE/AM,AP,BM,BP
      COMMON /PJ/X1,Y1
      COMMON SAVE1,SAVE2,POINT
      EQUIVALENCE(AP,APLUS),(AM,AMINUS),(BM,BMINUS),(BP,BPLUS)
      EQUIVALENCE(X1(1),Z(1)),(Y1(1),Z1(1))
      EQUIVALENCE (SPOT(1,1),POINT(1,1)),(I1,I2)
```

```
      DO 40 J=1,ILOVE
      DO 50 M=1,2000
      X1(M)=0.0
   50 Y1(M)=0.0
      CALL  BLANKR(SPOT,130,50,126)
      DO 54 M=1,2000
      X2(M)=0.0
   54 Y2(M)=0.0
      READ(5,10) OMEGA,ENCRMT,SIGMA,DELTA
      OMEGA=ABS(OMEGA)
   10 FORMAT(4F10.0)
      A=ENCRMT
      PERCNT=A*0.01
      B=OMEGA
      B=B*PERCNT
      APLUS=B+OMEGA
      D=AMINUS
      I1=1
      AMINUS=OMEGA-B
      C=APLUS
      DO 41 L=1,2000
      IF(Y(L).LT.D)GO TO 41
      IF(Y(L).GT.C)GO TO 41
   23 X1(I1)=X(L)
      Y1(I1)=Y(L)
      I1=I1+1
   41 CONTINUE
      A=DELTA
  901 PERCNT=A*0.01
      B=SIGMA
  903 B=B*PERCNT
      BPLUS=SIGMA+B
      BMINUS=SIGMA-B
      C=BPLUS
      D=BMINUS
      I2=1
  910 DO 42 L=1,2000
      IF(C.GT.0.0) GO TO 60
      IF(X1(L).GT.D) GO TO 42
      IF(X1(L).LT.C) GO TO 42
      GO TO 25
   60 IF(X1(L).GT.C) GO TO 42
      IF(X1(L).LT.D) GO TO 42
   25 X2(I2)=X1(L)
      Y2(I2)=Y1(L)
      I2=I2+1
   42 CONTINUE
      CALL SPLIT(X2,Y2,SPOT,APLUS,AMINUS,BPLUS,BMINUS)
   40 CONTINUE
      RETURN
      END
C
C
C
C
      SUBROUTINE SPLIT(X,Y,SPOT,APLUS,AMINUS,BPLUS,BMINUS)
      DIMENSION X1(2000),Y1(2000)
      DIMENSION X(2000),Y(2000),Z(125),Z1(87)
      INTEGER SPOT(130,50)
      COMMON /PJ/X1,Y1
```

```
      EQUIVALENCE(X1(1),Z(1)),(Y1(1),Z1(1))
      A=APLUS
      B=AMINUS
      C=A-B
      D=BPLUS
      E=BMINUS
      G=ABS(D)
      H=ABS(E)
      F=G-H
      DELTA=C/124.0
      DIFF=F/50.0
      DO 11 J=1,124
      Z(J)=A
   11 A=A-DELTA
      Z(125)=B
      IF(D.LT.0.0)DIFF=-DIFF
      DO 12 J=1,49
      Z1(J)=D
   12 D=D-DIFF
      Z1(50)=E
      CALL BRAKUP(Z,X,Y,SPOT,Z1)
      RETURN
      END
C
C
C
C
      SUBROUTINE BRAKUP(YY,X,Y,SPCT,XX)
      LOGICAL SKIP,SKIP1
      INTEGER SPOT(130,50),                    XPT,YPT,STAR
      DATA STAR/1HX/
      DIMENSION X(2000),Y(2000),YY(125),XX(87)
      COMMON /PAYNE/AM,AP,BM,BP
      EQUIVALENCE(AP,APLUS),(AM,AMINUS),(BM,BMINUS),(BP,BPLUS)
      SKIP=.FALSE.
      SKIP1=.FALSE.
      DO 43 J=1,2000
      DO 43 I=1,125
      IF(Y(J).NE.0.0) GO TO 30
      IF(X(J).NE.0.0) GO TO 30
      L=J+1
      M=J+6
      DO 60 N=L,M
      IF(N.GT.2000) GO TO 60
      IF(Y(N).NE.0.0) GO TO 30
      IF(X(N).NE.0.0) GO TO 30
   60 CONTINUE
      GO TO 40
   30 IF(SKIP) GO TO 20
      IF(Y(J).LT.YY(I)) GO TO 20
      YPT=I
      SKIP=.TRUE.
   20 IF(SKIP1)GO TO 48
      IF(I.GT.50) GO TO 43
      IF(BPLUS.GT.0.0) GO TO 10
      IF(X(J).GT.XX(I)) GO TO 43
      GO TO 11
   10 IF(X(J).LT.XX( I)) GO TO 43
   11 IF(BPLUS.LT.0.0) XPT=I
      IF(BPLUS.GE.0.) XPT=51-I
```

```
      SKIP1=.TRUE.
      IF(.NOT.SKIP)GO TO 43
   50 I=125
      SPOT(YPT,XPT)=STAR
      SKIP=.FALSE.
      SKIP1=.FALSE.
   48 IF(SKIP)GO TO 50
   43 CONTINUE
   40 CALL RITEIT(SPOT)
      RETURN
      END
C
C
C
C
      SUBROUTINE RITEIT(SPOT)
      COMMON /PAYNE/AM,AP,BM,BP
      EQUIVALENCE(AP,APLUS),(AM,AMINUS),(BM,BMINUS),(BP,BPLUS)
      INTEGER SPOT(130,50)
      IF(BP.LT.BM) GO TO 40
      A=BP
      B=BM
      WRITE(6,1)
    1 FORMAT(1H1,40X,43HCOMPLEX FREQUENCY PLANE,RIGHT HAND QUADRANT   )
      GO TO 30
   40 A=BM
      B=BP
      WRITE(6,2)
    2 FORMAT(1H1,40X,43HCOMPLEX FREQUENCY PLANE, LEFT HAND QUADRANT   )
   30 WRITE(6,12)AP,AM
   12 FORMAT(114X,15H<-------J-OMEGA,/1X,F8.2,112X,F8.2)
      WRITE(6,14)B
   14 FORMAT(124X,5HSIGMA,/126X,1HI,/126X,1HV,/121X,F8.2)
      WRITE(6,11) SPOT
   11 FORMAT(1X,130A1)
      WRITE(6,15)A
   15 FORMAT(60X,27HLINEAR EXPANC PLOT(RAD/SEC) ,34X,F8.2)
      RETURN
      END
C
C
      SUBROUTINE SAVER(ROOTR,ROOTI,IDIC,SAVE1,SAVE2,JZO,K1)
      DIMENSION SAVE1(100,100),SAVE2(100,100),ROOTR(100),ROOTI(100)
      IF(K1)30,9,10
    9 K1=1
   10 ID1=IDIC
      IDIC=IDIC+(K1-1)
      IF(IDIC.GE.100)GO TO 30
   50 DO 40 IZAP=K1,IDIC
      IZA=IZAP-(K1-1)
      SAVE1(JZO,IZAP)=ROOTR(IZA)
      SAVE2(JZO,IZAP)=ROOTI(IZA)
   40 CONTINUE
      K1=IDIC+1
      GO TO 20
   30 JZO=JZO+1
      IDIC=ID1
      GO TO 9
   20 RETURN
      END
```

```
C
C
C
C
      SUBROUTINE PLOTER(SAVE1,SAVE2,ANUMB1,ANUMB2,POINT,XA,XB,MI,PRNT,
     *STAR,NBB,NO)
      DIMENSION SAVE1(100,100),SAVE2(100,100),XA(2000),XB(2000)
      INTEGER POINT(130,100),DASH,BLANK,STAR,ROTLCS,EXPND,FREQSR,DENSE
      COMMON /INFO7/ ICOUNT
      COMMON /INFO8/ROTLCS,EXPND,FREQSR
      COMMON /INFO9/ DENSE
      LOGICAL PRNT
      IF(PRNT) GO TO 43
      DO 40 NAB=1,100
      DO 40 NBB=1,100
      IF(SAVE1(NAB,NBB))41,42,41
   42 IF(SAVE2(NAB,NBB))41,40,41
   41 ANUMB1=SAVE1(NAB,NBB)
      ANUMB2=SAVE2(NAB,NBB)
      ANUMB2=ABS(ANUMB2)
   43 AZZ=ANUMB2
      AZZ=AZZ*100000.0
      NZZ=AZZ
      IF(NZZ.EQ.0) GO TO 45
   50 NBB=NBB+1
   45 CALL EXCUTE(ANUMB1,ANUMB2,POINT,XA,XB,MI,NO,STAR)
      IF (DENSE.NE.1) GO TO 10
      ICOUNT=ICOUNT+1
      GO TO 11
   10 IF(PRNT) ICOUNT=ICOUNT+1
   11 IF(PRNT) RETURN
   40 CONTINUE
      IF(ROTLCS.EQ.0) RETURN
   70 CALL WRITIT(XA,XB)
      CALL PREPAR(POINT)
      RETURN
      END
C
C
C
C
      SUBROUTINE EXCUTE(ANUMB1,ANUMB2,POINT,XA,XB,MI,NO,STAR)
      LOGICAL SKIP1,SKIP2 ,LESS
      INTEGER ROTLCS,EXPND,FREQSR
      DIMENSION XA(2000),XB(2000)
      INTEGER POINT(130,100),DASH,BLANK,STAR
      COMMON /INFO8/ROTLCS,EXPND,FREQSR
      DATA        K1/10/,K2/100/,K3/1000/,K4/10000/,NEGONE/-1/
      LESS=.FALSE.
      I=0
      J=0
      XA(NO)=ANUMB1
      XB(NO)=ABS(ANUMB2)
      NO=NO+1
      IF(ROTLCS.EQ.0) RETURN
      IF(ABS(ANUMB1).GT.10000.0) GO TO 50
      IF(ABS(ANUMB2).GT.10000.0) GO TO 50
      IF(ABS(ANUMB1).EQ.0.0) GO TO 500
      IF(ABS(ANUMB2).EQ.0.0) GO TO 500
      IF(ABS(ANUMB2).LT.0.001) GO TO 50
```

```
      IF(ABS(ANUMB1).LT.0.001) GO TO 50
  500 CALL SCALE1(K1,K2,K3,K4,NEGONE,I,ANUMB1,ICONS,LESS,SKIP1)
      CALL SCALE2(K1,K2,K3,K4,J,JCONS,ANUMB2,SKIP2)
      CALL WPOINT(J,JCONS,L,ANUMB2,SKIP2)
      CALL SPOINT(I,ANUMB1,LESS,ICONS,L,POINT,SKIP1,NO,XA,XB,STAR)
   50 RETURN
      END
C
C
C
C
      SUBROUTINE SCALE1(K1,K2,K3,K4,NEGONE,I,ANUMB1,ICONS,LESS,SKIP1)
      LOGICAL LESS
      LOGICAL SKIP1
      SKIP1=.FALSE.
      ICONS=1
      I=1
      AKEEP=ANUMB1
   21 NUMB1=ANUMB1
      NUMB=IABS(NUMB1)
      IF(NUMB.EQ.0)GOTO 12
      IF(NUMB1)31,40,40
   40 IF(NUMB1.GE.10)GOTO 11
      GO TO 50
   31 LESS=.TRUE.
      XXX=-NUMB1
      NUMB1=XXX
      GOTO 40
   12 GOTO(1,2,3,4),I
    1 ANUMB1=AKEEP
      ICONS=K1
      RK1=K1
      ANUMB1=ANUMB1*RK1
      I=2
      GOTO 21
    2 ANUMB1=AKEEP
      ICONS=K2
      RK2=K2
      ANUMB1=ANUMB1*RK2
      I=3
      GOTO 21
    3 ANUMB1=AKEEP
      ICONS=K3
      RK3=K3
      ANUMB1=ANUMB1*RK3
      I=4
      GOTO 21
    4 ANUMB1=AKEEP
      ICONS=K4
      RK4=K4
      ANUMB1=ANUMB1*RK4
      NUMB1=ANUMB1
      IF(NUMB1.EQ.0) GO TO 51
      GO TO 50
   11 SKIP1=.TRUE.
      GO TO (6,7,8,9),I
    6 ANUMB1=AKEEP
      ICONS=K1
      RK1=K1
      ANUMB1=ANUMB1/RK1
```

```
      I=2
      GOTO 21
    7 ANUMB1=AKEEP
      ICONS=K2
      RK2=K2
      ANUMB1=ANUMB1/RK2
      I=3
      GOTO 21
    8 ANUMB1=AKEEP
      ICONS=K3
      RK3=K3
      ANUMB1=ANUMB1/RK3
      I=4
      GOTO 21
    9 ANUMB1=AKEEP
      ICONS=K4
      RK4=K4
      ANUMB1=ANUMB1/RK4
      GO TO 50
   51 I=5
   50 RETURN
      END
C
C
C
C
      SUBROUTINE SCALE2(K1,K2,K3,K4,J,JCONS,ANUMB2,SKIP2)
      LOGICAL SKIP2
      SKIP2=.FALSE.
      JCONS=1
      J=1
      ANUMB2=ABS(ANUMB2)
      BKEEP=ANUMB2
   20 NUMB2=ANUMB2
      IF(NUMB2.EQ.0)GO TO 10
      IF(NUMB2.GE.10)GO TO 11
      GO TO 50
   10 GOTO(1,2,3,4),J
    1 ANUMB2=BKEEP
      JCONS=K1
      RK1=K1
      ANUMB2=ANUMB2*RK1
      J=2
      GO TO 20
    2 ANUMB2=BKEEP
      JCONS=K2
      RK2=K2
      ANUMB2=ANUMB2*RK2
      J=3
      GO TO 20
    3 ANUMB2=BKEEP
      JCONS=K3
      RK3=K3
      ANUMB2=ANUMB2*RK3
      J=4
      GO TO 20
    4 ANUMB2=BKEEP
      JCONS=K4
      RK4=K4
      ANUMB2=ANUMB2*RK4
```

```
      NUMB2=ANUMB2
      IF(NUMB2.EQ.0) GO TO 51
      GO TO 50
   11 SKIP2=.TRUE.
      GO TO (6,7,8,9),J
    6 JCONS=K1
      ANUMB2=BKEEP
      RK1=K1
      ANUMB2=ANUMB2/RK1
      J=2
      GO TO 20
    7 JCONS=K2
      ANUMB2=BKEEP
      RK2=K2
      ANUMB2=ANUMB2/RK2
      J=3
      GO TO 20
    8 JCONS=K3
      ANUMB2=BKEEP
      RK3=K3
      ANUMB2=ANUMB2/RK3
      J=4
      GO TO 20
    9 JCONS=K4
      ANUMB2=BKEEP
      RK4=K4
      ANUMB2=ANUMB2/RK4
      GO TO 50
   51 J=5
   50 RETURN
      END
C
C
C
C
      SUBROUTINE WPOINT(J,JCONS,L,ANUMB2,SKIP2)
      LOGICAL SKIP2
      L=0
      IDELTA=0
      IF(ANUMB2.GE.9.2) IDELTA=12
      IF(ANUMB2.GE.8.0) GO TO 200
      IF(ANUMB2.GE.6.9) GO TO 201
      IF(ANUMB2.GE.5.9) GO TO 202
      IF(ANUMB2.GE.5.0) GO TO 203
      IF(ANUMB2.GE.4.2) GO TO 204
      IF(ANUMB2.GE.3.5) GO TO 205
      IF(ANUMB2.GE.2.9) GO TO 206
      IF(ANUMB2.GE.2.4) GO TO 207
      IF(ANUMB2.GE.1.9) GO TO 208
      IF(ANUMB2.GE.1.5) GO TO 209
      IF(ANUMB2.GE.1.2) GO TO 210
      GO TO 211
  200 IF(ANUMB2.LT.9.2) IDELTA=11
  201 IF(ANUMB2.LT.8.0) IDELTA=10
  202 IF(ANUMB2.LT.6.9) IDELTA=9
  203 IF(ANUMB2.LT.5.9) IDELTA=8
  204 IF(ANUMB2.LT.5.0) IDELTA=7
  205 IF(ANUMB2.LT.4.2) IDELTA=6
  206 IF(ANUMB2.LT.3.5) IDELTA=5
  207 IF(ANUMB2.LT.2.9) IDELTA=4
```

```
  208 IF(ANUMB2.LT.2.4) IDELTA=3
  209 IF(ANUMB2.LT.1.9) IDELTA=2
  210 IF(ANUMB2.LT.1.5) IDELTA=1
  211 IF(ANUMB2.LT.1.2)IDELTA=0
      IDELTA=13-IDELTA
      IF(J.EQ.5) GO TO 50
      IF(SKIP2) GO TO 41
      IF(JCONS.EQ.1)GOTO 1
      IF(JCONS.EQ.10)GOTO 10
      IF(JCONS.EQ.100)GOTO 100
      IF(JCONS.EQ.1000)GOTO 1000
      IF(JCONS.EQ.10000)GOTO 10000
1     L=73+IDELTA
      GO TO 40
   10 L=86+IDELTA
      GO TO 40
  100 L=99+IDELTA
      GO TO 40
1000  L=112+IDELTA
      GO TO 40
10000 L=125
      GO TO 40
   41 KCONS=JCONS+1
      SKIP2=.FALSE.
      IF(KCONS.EQ.11)GOTO 11
      IF(KCONS.EQ.101)GOTO 101
      IF(KCONS.EQ.1001)GOTO 1001
      IF(KCONS.EQ.10001)GOTO 10001
   11 L=60+IDELTA
      GO TO 40
  101 L=47+IDELTA
      GO TO 40
 1001 L=34+IDELTA
      GO TO 40
   50 L=125
      GO TO 40
10001 L=34
   40 RETURN
      END
C
C
C
C
      SUBROUTINE SPOINT(I,ANUMB1,LESS,ICONS,L,POINT,SKIP1,NO,XA,XB,STAR)
      DIMENSION XA(2000),XB(2000)
      INTEGER POINT(130,100),DASH,BLANK,STAR,Q
      INTEGER STAR1,STAR2
      DATA STAR1/1HO/,STAR2/1H*/
      DATA BLANK/1H /,DASH/1H-/,          Q/1HI/
      LOGICAL SKIP1         ,LESS
      AKEEP=ANUMB1
      ANUMB1=ABS(AKEEP)
      KOOL=0
      IF(I.EQ.5)GO TO 1000
      IF(ANUMB1.GE.7.1)KOOL=7
      IF(ANUMB1.GE.5.0)GO TO 1011
      IF(ANUMB1.GE.4.0)GO TO 1111
      IF(ANUMB1.GE.3.1) GO TO 1211
      IF(ANUMB1.GE.2.3)GOTO1311
      IF(ANUMB1.GE.1.6) GO TO 1411
```

```
      IF(ANUMB1.GE.1.0)GO TO 1511
      GO TO 4011
1011  IF(ANUMB1.LT.7.1)KOOL=6
1111  IF(ANUMB1.LT.5.0)KOOL=5
1211  IF(ANUMB1.LT.4.0)KOOL=4
1311  IF(ANUMB1.LT.3.1)KOOL=3
1411  IF(ANUMB1.LT.2.3)KOOL=2
1511  IF(ANUMB1.LT.1.6)KOOL=1
 4011 NCRMNT=8-KOOL
      IF(LESS)GO TO 40
      IF(SKIP1)GO TO 41
      IF(ICONS.EQ.1)GOTO 1
      IF(ICONS.EQ.10)GOTO 10
      IF(ICONS.EQ.100)GOTO 100
      IF(ICONS.EQ.1000)GOTO 1000
    1 KOOL=71+KOOL
      GO TO 50
   10 KOOL=64+KOOL
      GO TO 50
  100 KOOL=57+KOOL
      GO TO 50
 1000 KOOL=50+KOOL
      GO TO 50
   41 LCONS=ICONS+1
      SKIP1=.FALSE.
      IF(LCONS.EQ.11)GOTO 11
      IF(LCONS.EQ.101)GOTO 101
      IF(LCONS.EQ.1001)GOTO 1001
   11 KOOL=78+KOOL
      GO TO 50
  101 KOOL=85+KOOL
      GO TO 50
 1001 KOOL=92+KOOL
   50 LESS=.FALSE.
      IF(POINT(L,KOOL).EQ.DASH) GO TO 51
      IF(POINT(L,KOOL).EQ.Q    ) GO TO 51
   14 IF(POINT(L,KOOL).EQ.STAR1) GO TO 15
      IF(POINT(L,KOOL).EQ.STAR2) GO TO 15
      IF(STAR.EQ.STAR1) GO TO 51
      IF(STAR.EQ.STAR2) GO TO 51
      IF(POINT(L,KOOL).NE.BLANK) GO TO 70
   51 POINT(L,KOOL)=STAR
      ANUMB1=AKEEP
      GO TO 70
   15 L=L+1
      IF(L.EQ.130) GO TO 51
      GO TO 14
   40 IF(SKIP1) GO TO 42
      IZ=ICONS+2
      IF(IZ.EQ.3)GOTO 3
      IF(IZ.EQ.12)GOTO 12
      IF(IZ.EQ.102)GOTO 102
      IF(IZ.EQ.1002)GOTO 1002
    3 KOOL=21+NCRMNT
      GO TO 50
   12 KOOL=28+NCRMNT
      GO TO 50
  102 KOOL=35+NCRMNT
      GO TO 50
 1002 KOOL=42+NCRMNT
```

```
      GO TO 50
   42 IY=ICONS+3
      SKIP1=.FALSE.
      IF(IY.EQ.13)GOTO 13
      IF(IY.EQ.103)GOTO 103
      IF(IY.EQ.1003)GOTO 1003
      IF(IY.EQ.10003)GOTO 10003
   13 KOOL=14+NCRMNT
      GO TO 50
  103 KOOL=7+NCRMNT
      GO TO 50
 1003 KOOL=NCRMNT
      GO TO 50
10003 POINT(L,1)=STAR
      GO TO 70
   70 RETURN
      END
C
C
C
C
      SUBROUTINE PREPAR(POINT)
      INTEGER POINT(130,100),PRT
      DATA PRT /6/
      WRITE(6,1)
    1 FORMAT(1H1)
      WRITE(6,11)
   11 FORMAT(///)
      WRITE(PRT,14)
   14 FORMAT(75X,8HLOG PLOT,/58X,42HCOMPLEX FREQUENCY PLANE,LEFT HAND CU
     *ADRANT,/75X,9H(RAD/SEC)   )
      WRITE(6,12)
   12 FORMAT(32X,5H10000,9X,4H1000,9X,3H100,10X,2H10,12X,1H1,11X,2H.1,10
     *X,14H.01<---J-OMEGA,/ 18X,11HMINUS SIGMA)
   13 FORMAT(32X,5H10000,9X,4H1000,9X,3H100,10X,2H10,12X,1H1,11X,2H.1,10
     *X,14H.01<---J-OMEGA,/ 18X,11H PLUS SIGMA)
      DO 50 I=1,130
   50 POINT(I,50)=POINT(I,51)
      WRITE(PRT,10)POINT
   10 FORMAT(1X,130A1)
      WRITE(6,13)
      WRITE(PRT,15)
   15 FORMAT(75X,8HLOG PLOT,/58X,43HCOMPLEX FREQUENCY PLANE,RIGHT HAND Q
     *UADRANT,/75X,9H(RAD/SEC)   )
      RETURN
      END
C
C
C
C
C
      SUBROUTINE WRITIT(XA,XB)
      DIMENSION XA(2000),XB(2000)
      WRITE(6,1)
    1        FORMAT(1H1)
      WRITE(6,11)
   11 FORMAT(5X,48HTHE FOLLOWING  ROOTS ARE PLOTTED ON THE LOG PLOT,/
     15X,99HROOTS AT THE ORIGIN ARE NOT PRINTED OR PLOTTED, ROOTS ON THE
     * J-OMEGA AXIS ARE NOT PLOTTED.                                   ,
     2,//16X,5HSIGMA,25X,7HJ-OMEGA,//)
```

```
      DO 77 IZO=1,2000
      XC=XB(IZO)
      IF(XC.GT.0.0) XC=-XC
      IF(XC.NE.0.0) GO TO 22
   20 IF(XA(IZO))22,40,22
   40 K1=IZO
      K2=IZO+12
      DO 50 K=K1,K2
      IF(XA(K).NE.0.0) GO TO 77
   50 IF(XB(K).NE.0.0) GO TO 77
      GO TO 60
   22 WRITE(6,10)XA(IZO),XC
   10 FORMAT(5X,F20.9,10X,F20.9)
   77 CONTINUE
   60 RETURN
      END
C
C
      SUBROUTINE FREQRS(SIGMAX,OMEGAY)
      DIMENSION SAVE1(100,100),SAVE2(100,100),ACOEFF(100),BCOEFF(100),
     *          AAAAAA(99),BBBBBB(99),SIGMAX(2000),OMEGAY(2000),DEC(9),
     *          OMEGAS(55),OMEGA(17),WLOG(3000),PHILIN(3000),MDBLIN(3000
     *          ),PHIPLN(3000),MPDBLN(3000),ALIM(7),BIGSEM(3000),
     *          EMSUBS(3000)
      REAL MDBLIN,MPDBLN
      INTEGER FRQEXP,IEXP(4)
      INTEGER POINT(130,100)
      COMMON/FREEK/ACOEFF,BCOEFF
      COMMON/INFO2/ALIM
      COMMON/INFO3/OMEGA
      COMMON/INFO6/EMSUBS
      COMMON /INFO7/ ICOUNT
      COMMON SAVE1,SAVE2,POINT
      EQUIVALENCE (WLOG(1),SAVE1(1)),(PHILIN(1),SAVE2(1)),
     1            (SAVE1(3001),MDBLIN(1)),(SAVE2(3001),PHIPLN(1)),
     2            (SAVE1(6001),MPDBLN(1)),(SAVE2(6001),BIGSEM(1)),
     3            (ALIM(1),XSMALL),(AAAAAA(1),ACOEFF(2)),(BBBBBB(1),
     4            BCOEFF(2))
      TENS(X)=X*10.0
      DATA(DEC(I),I=1,9)/1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0/
      DATA LET1/4H  DB/,LET2/4H DEG/
      K1=0
      KOUNT=1
      DO 61 I=1,4
   61 IEXP(I)=0
      DO 60 I=1,3000
   62 WLOG(I)=0.0
      PHILIN(I)=0.0
      BIGSEM(I)=0.0
      EMSUBS(I)=0.0
      MDBLIN(I)=0.0
      PHIPLN(I)=0.0
   60 MPDBLN(I)=0.0
      WRITE(6,51)
      READ(5,19)XSMALL,DEGREE,DBLIM
      READ(5,12)FRQEXP
      IF(FRQEXP.NE.0)READ(5,12)(IEXP(K),K=1,4)
   12 FORMAT(4I10)
   19 FORMAT(3E10.0,50X)
      DO 90 I=2,7
```

```
 90 ALIM(I)=TENS(ALIM(I-1))
    DO 70 J=1,6
    DO 71 I=1,9
    K=K1+I
 71 OMEGAS(K)=DEC(I)*ALIM(J)
 70 K1=K1+9
    OMEGAS(55)=ALIM(7)
    K=1
    CALL FIGURR(55,OMEGAS,KOUNT)
 41 Y=ABS(OMEGAY(K))
    X=ABS(SIGMAX(K))
    IF(Y.EQ.0.0) GO TO 13
    YOMEGA=Y
    GO TO 20
 13 IF(X.EQ.0.0) GO TO 10
    YOMEGA=X
 20 CALL SWEEPR(YOMEGA)
    CALL FIGURR(17,OMEGA,KOUNT)
    IF(KOUNT.GE.2984)GO TO 40
    GO TO 80
 10 KK=K+1
    KI=K+11
    IF(KI.GT.2000) GO TO 40
    DO 11 L=KK,KI
    Y=ABS(OMEGAY(L))
    X=ABS(SIGMAX(L))
    IF(X.NE.0.0.OR.Y.NE.0.0) GO TO 80
 11 CONTINUE
    GO TO 40
 80 IF(K.GE.ICOUNT) GO TO 40
    K=K+1
    GO TO 41
 40 DO 68 I=1,3000
    IF(WLOG(I).EQ.0.0.AND.PHILIN(I).EQ.0.0.AND.PHIPLN(I).EQ.0.0.AND.
   1MDBLIN(I).EQ.0.0.AND.MPDBLN(I).EQ.0.0.AND.BIGSEM(I).EQ.0.0.AND.
   2EMSUBS(I).EQ.0.0) GO TO 69
    JJ=MOD(I     ,57)
    IF(JJ.EQ.0) WRITE(6,51)
 68 WRITE(6,50) WLOG (I),BIGSEM(I),MDBLIN(I),PHILIN(I),EMSUBS(I),
   1MPDBLN(I),PHIPLN(I)
 69 WRITE(6,14)
    CALL NEWPLT(DEGREE,WLOG,PHILIN,LET2 )
    IF(IEXP(1).EQ.0) GO TO 30
    JJ=IEXP(1)
    DO 34 I=1,JJ
 34 CALL EXPLOT(WLOG,3000,PHILIN,3000,DEGREE,LET2)
 30 CALL NEWPLT(DBLIM,WLOG,MDBL IN,LET1)
    IF(IEXP(2).EQ.0) GO TO 31
    JJ=IEXP(2)
    DO 35 I=1,JJ
 35 CALL EXPLOT(WLOG,3000,MDBLIN,3000,DBLIM,LET1)
 31 WRITE(6,15)
    CALL NEWPLT(DEGREE,WLOG,PHIPLN,LET2)
    IF(IEXP(3).EQ.0) GO TO 32
    JJ=IEXP(3)
    DO 36 I=1,JJ
 36 CALL EXPLOT(WLOG,3000,PHIPLN,3000,DEGREE,LET2)
 32 CALL NEWPLT(DBLIM,WLOG,MPDBLN,LET1)
    IF(IEXP(4).EQ.0) GO TO 33
    JJ=IEXP(4)
```

```
      DO 37 I=1,JJ
   37 CALL EXPLOT(WLOG,3000,MPDBLN,3000,DBLIM,LET1)
   51 FORMAT(1H1,5X,5HOMEGA,13X,3H/M/,16X,6H/M/-DB,10X,3HPHI,20X,4H/M/P,
     *10X,7H/M/P-DB,14X,5HPHI-P,/)
   50 FORMAT(2X,1P1E12.5,5X,1P1E12.5,6X,1P1E12.5,5X,1P1E12.5,11X,
     *1P1E12.5,3X,1P1E12.5,9X,1P1E12.5)
   14 FORMAT(1H1,////////////////////,25X,60HTHE FOLLOWING PLOTS ARE NOR
     *MAL FREQUENCY RESPONSE DATA                                       )
   15 FORMAT(1H1,////////////////////,25X,36HTHE FOLLOWING PLOTS ARE POP
     *OV DATA                                                        )
   33 RETURN
      END
C
C
      SUBROUTINE FIGURR(KONT,OMEGA,KOUNT)
      DIMENSION SAVE1(100,100),SAVE2(100,100)
      DIMENSION ACOEFF(100),BCOEFF(100),AAAAAA(99),BBBBBB(99)
      DIMENSION WLOG(3000),PHILIN(3000),MDBLIN(3000),PHIPLN(3000),
     *MPDBLN(3000),BIGSEM(3000),EMSUBS(3000),OMEGA(KONT)
      REAL MDBLIN,MPDBLN
      INTEGER POINT(100,100)
      COMMON/FREEK/ACOEFF,BCOEFF
      EQUIVALENCE(AAAAAA(1),ACOEFF(2)),(BBBBBB(1),BCOEFF(2))
      COMMON/INFO6/EMSUBS
      COMMON SAVE1,SAVE2,POINT
      EQUIVALENCE (WLOG(1),SAVE1(1)),(PHILIN(1),SAVE2(1)),
     1            (SAVE1(3001),MDBLIN(1)),(SAVE2(3001),PHIPLN(1)),
     2            (SAVE1(6001),MPDBLN(1)),(SAVE2(6001),BIGSEM(1))
      DO 21 I=1,KONT
      CALL FIGURE(OMEGA(I),ACOEFF(1),AAAAAA,EMOFAS,PHIAS)
      CALL FIGURE(OMEGA(I),BCOEFF(1),BBBBBB,EMOFBS,PHIBS)
      BIGEMM=ABS(EMOFAS/EMOFBS     )
      BIGSEM(KOUNT)=BIGEMM
      WLOG(KOUNT)=OMEGA(I)
      PHIEND=PHIAS-PHIBS
      IF(PHIEND.LT.-180.) PHIEND=PHIEND+360.
      IF(PHIEND.GT. 180.) PHIEND=PHIEND-360.
      PHILIN(KOUNT)=PHIEND
      PHIEND = PHIEND * 0.0174533
      REEL = BIGEMM * COS ( PHIEND )
      EIMAG= BIGEMM * SIN ( PHIEND )
      EMSUBP = SQRT ( ( REEL** 2 ) + ( ( OMEGA ( I ) * EIMAG ) ** 2 ) )
      EMSUBS(KOUNT)=EMSUBP
      QUANZ=OMEGA(I) * EIMAG
      PHIPEE=ATAN2(QUANZ,REEL)
      PHIPEE = PHIPEE * 57.2957795131
      IF(PHIPEE . LT . 0.) PHIPEE = PHIPEE + 360.
      IF(PHIPEE.LT.-180.) PHIPEE=PHIPEE+360.
      IF(PHIPEE.GT. 180.) PHIPEE=PHIPEE-360.
      PHIPLN(KOUNT)=PHIPEE
      DEEBEE=20.0*(ALOG10(BIGEMM))
      MDBLIN(KOUNT)=DEEBEE
      DEEBEP=20.0*(ALOG10(EMSUBP))
      MPDBLN(KOUNT)=DEEBEP
      KOUNT=KOUNT+1
   21 CONTINUE
      RETURN
      END
C
C
```

```
      SUBROUTINE FIGURE(X,Y,Z,EMOFAS,PHIAS)
      DIMENSION Z(99)
      REELAS=Y
      I=2
      A=-1.0
   21 REELAS=((Z(I)*(X**I))*A)+REELAS
   25 I=I+2
      IF(I.GT.99 ) GO TO 22
      A=-A
      IF(Z(I).EQ.0.)GO TO 25
      GO TO 21
   22 I=1
      A=1.0
      AIMAGA=0.0
   24 AIMAGA=(Z(I)*(X**I)*A)+AIMAGA
   26 I=I+2
      IF(I.GT.99 ) GO TO 23
      A=-A
      IF(Z(I).EQ.0.)GO TO 26
      GO TO 24
   23 EMOFAS=SQRT((REELAS**2)+(AIMAGA**2))
      PHIAS=ATAN2(AIMAGA,REELAS)
      PHIAS = PHIAS*57.2957795131
      IF(PHIAS.LT.0.) PHIAS=PHIAS+360.
      RETURN
      END
C
C
      SUBROUTINE LOGGER(XX,LIMITW)
      DIMENSION  ALIM(7),DIMS(20)
      EQUIVALENCE(WASTE,AXX)
      COMMON/INFO2/ALIM
      TENS(X)=X*10.0
      DATA (DIMS(I),I=1,20) /1.0,1.15,1.3,1.5,1.7,1.95,2.25,2.5,2.85,
     *3.15,3.5,3.95,4.5,4.85,5.4,6.05,6.8,7.7,8.8,9.999999/
      LIMITW=0
      LIMITX=0
      LIMIT2=0
      DO 20 I=1,6
      AA=2*(I-1)
   20 IF(XX.LT.ALIM(I+1).AND.XX.GE.ALIM(I)) GO TO 21
      RETURN
   21 LIMITX=TENS(AA)
      WASTE=ABS(XX)
      IF(WASTE.EQ.0.0) RETURN
   30 IF(AXX.GE.1.0.AND.AXX.LT.10.0) GO TO 22
      IF(AXX.LT.10.0)GO TO 23
      AXX=AXX/10.0
      GO TO 30
   23 AXX=TENS(AXX)
      GO TO 30
   22 DO 24 I=1,19
   24 IF(AXX.GE.DIMS(I).AND.AXX.LT.DIMS(I+1)) GO TO 25
   25 LIMIT2=I
      LIMITW=LIMITX+LIMIT2
      RETURN
      END
C
C
      SUBROUTINE BLANKR (POINT,N1,N2,LIM1)
```

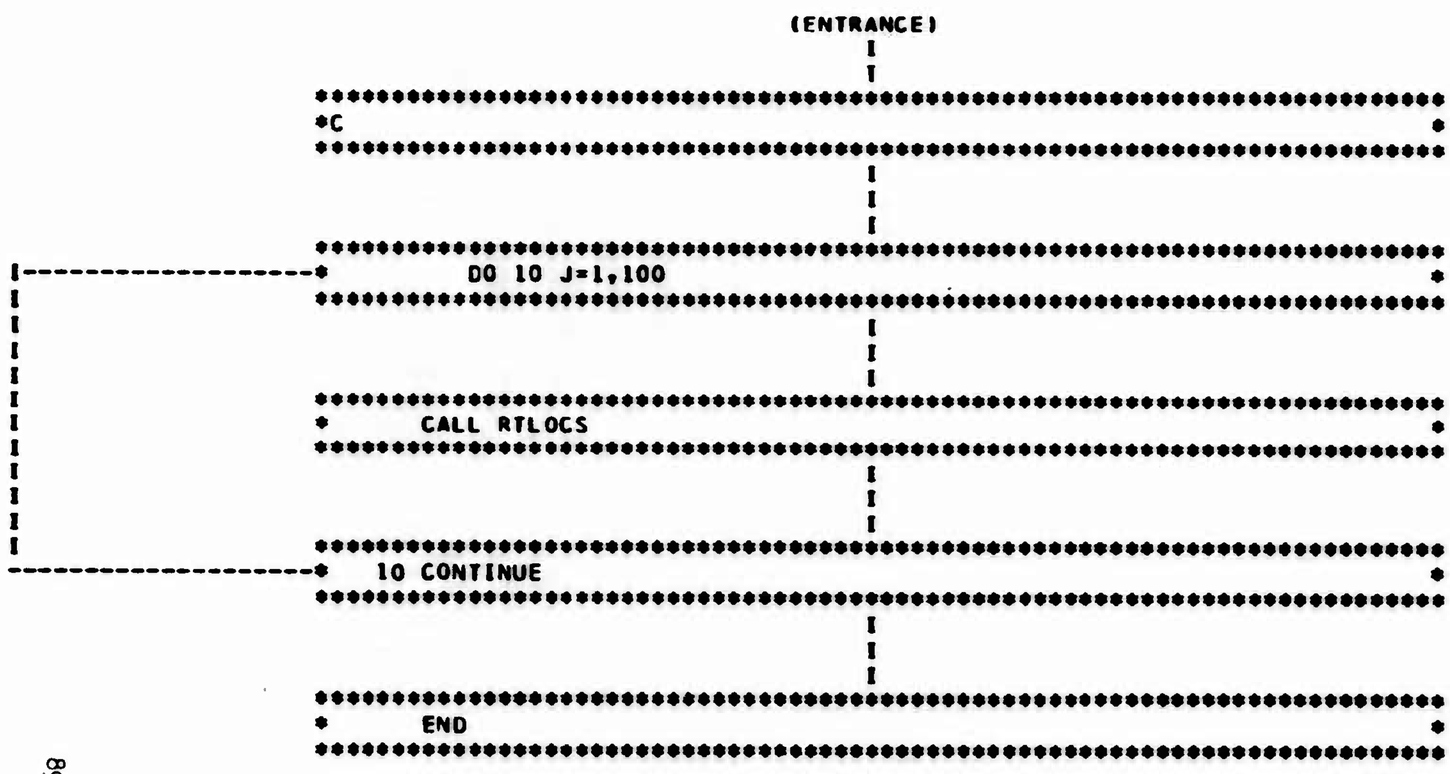
```
      INTEGER POINT(N1,N2),BLANK,DASH,Q
      COMMON /INFO4/ Q,BLANK,DASH
      DATA BLANK/1H /,DASH/1H-/,Q/1HI/
      DO 51 M=1,N1
      DO 51 N=1,N2
   51 POINT(M,N)=BLANK
      DO 52 M=1,N1
      POINT(M,1)=DASH
   52 POINT(M,N2)=DASH
      DO 53 M=1,N2
      POINT(1,M)=Q
   53 POINT(LIM1,M)=C
      RETURN
      END
C
C
      SUBROUTINE NEWPLT(X,Y,Z,LET)
      INTEGER POINT(130,100),POINTF(120,58),STAR,Q
      DIMENSION ALIM(7)
      DIMENSION Y(3000),Z(3000)
      DIMENSION SAVE1(100,100),SAVE2(100,100)
      COMMON /INFO4/ Q,BLANK,DASH
      COMMON/INFO2/ALIM
      COMMON SAVE1,SAVE2,POINT
      EQUIVALENCE(POINT(1,1),POINTF(1,1))
      DATA STAR/1H*/,Q/1HI/
    1 FORMAT(1H1)
      CALL BLANKR(POINTF,120,58,120)
      DO 10 I=1,3000
      IF(Y(I).EQ.0.0) GO TO 11
      GO TC 12
   11 IF(Z(I).EQ.0.0) GO TO 10
   12 CALL LOGGER(Y(I),LIMITW)
      IF(LIMITW.EQ.0) GO TO 10
      CALL LINEAR(X,LIMITY,Z(I))
      IF(LIMITY.EQ.0) GO TO 10
      POINTF(LIMITW,LIMITY)=STAR
   10 CONTINUE
      DO 40 I=20,100,20
      DO 40 J=1,58
   40 POINTF(I,J)=Q
      WRITE(6,1)
   30 FORMAT(1X,6(1P1E10.1,1X,3HR/S,6X),1P1E10.1)
      J=X
      WRITE(6,51)J,LET,(POINTF(I,1),I=1,120)
   51 FORMAT(1X,I4, A4,1X,120A1)
      WRITE(6,20)((POINTF(I,J),I=1,120),J=2,57)
      J=-X
      WRITE(6,51)J,LET,(POINTF(I,58),I=1,120)
   20 FORMAT(10X,120A1)
      WRITE(6,30)ALIM
      RETURN
      END
C
C
      SUBROUTINE LINEAR(X,LIMITY,Y)
      DIMENSION ENCRMT(59)
      LIMITY=0
      FLIMIT=X*2.
      DELTA=FLIMIT/58.
```

```
      ENCRMT(1)=X
      DO 10 I=2,58
   10 ENCRMT(I)=ENCRMT(I-1)-DELTA
      ENCRMT(59)=-X
      DO 11 I=1,58
   11 IF(Y.LE.ENCRMT(I).AND.Y.GE.ENCRMT(I+1))GO TO 20
      GO TO 40
   20 LIMITY=I
   40 RETURN
      END
C
C
      SUBROUTINE SWEEPR(XOMEGA)
      DIMENSION OMEGA(17),SWEPER(17)
      COMMON/INFO3/OMEGA
      DATA (SWEPER(I),I=1,17) /C.C1,0.05,0.1,0.2,0.3,0.4,0.5,0.6,0.7,
     *C.8,0.9,1.0,2.0,4.0,6.0,8.C,10.0/
      DO 70 K=1,17
   70 OMEGA(K)= SWEPER(K)*XOMEGA
      RETURN
      END
C
C
```

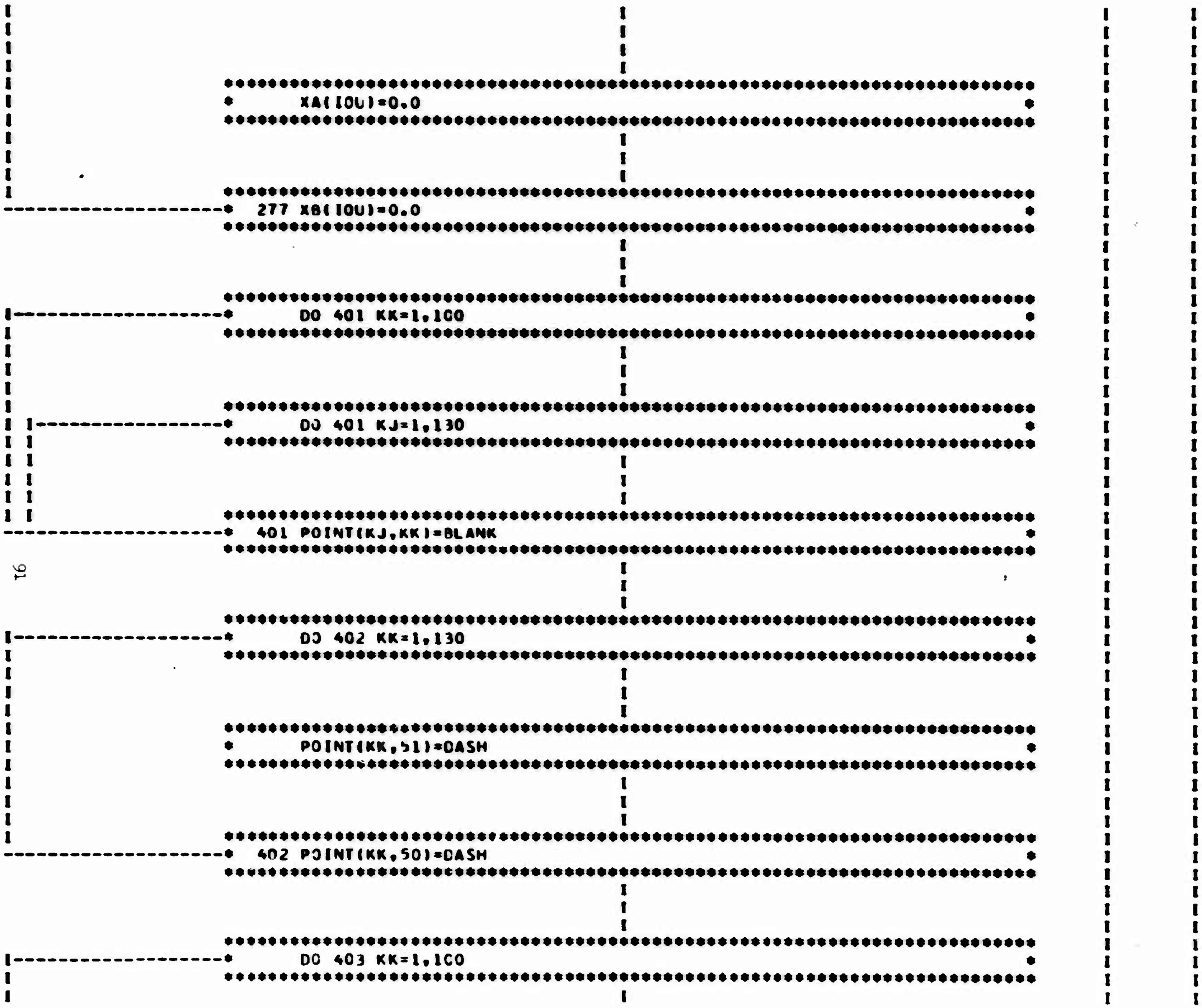
APPENDIX B

FLOW CHART OF SOURCE DECK

 Next page is blank.


(ENTRANCE)
C
DO 10 J=1,100
CALL RTLOCS
10 CONTINUE
END

```
                         (ENTRANCE)
                              I
                              I
*********************************************************************
*C                                                                  *
*C                                                                  *
*C                                                                  *
*C                                                                  *
*      SUBROUTINE RTLOCS                                            *
*      DIMENSION SAVE1(10C,100),SAVE2(100,100),XA(2000),XB(2000)    *
*      DIMENSION X(1CO),IGA(100),IGB(100),C(10C),D(100),ANS(100),   *
*     1SAVE(100),ERA(100),AS(100),RS(100),A(10C),B(100),ROOTR(100), *
*     2ROOTI(100),ATK(1CO),CK(100),CKS(10C)                         *
*      INTEGER COMENT(20),DENSE,ROTLCS,EXPND,FREQSR,STAR,STAR1,STAR2,*
*     *        STAR3,ONE,ZERO,DOT,VEE,AYE,DASH,BLANK,Q,             *
*     *        POINT(130,100),SPOT(130,50)                          *
*      DATA ONE/1H1/,ZERC/1HO/,DOT/1H./,AYE/1HA/,VEE/1HV/,BLANK/1H /,*
*     *     CASH/1H-/,Q/1HI/,STAR1/1HO/,STAR2/1H*/,STAR3/1H./       *
*      LOGICAL PRNT                                                 *
*      EQUIVALENCE (SPOT(1,1),POINT(1,1)),(STAR1,ZERO),(STAR3,DOT), *
*     *            (JACKIE,KJ,I),(JOANN,KK,J)                       *
*      COMMON/FREEK/A,B                                             *
*      COMMON /INFO4/ Q,BLANK,DASH                                  *
*      COMMON /INFO7/ ICOUNT                                        *
*      COMMON SAVE1,SAVE2,POINT                                     *
*      COMMON /INFO8/ROTLCS,EXPND,FREQSR                            *
*      COMMON /INFO9/ DENSE                                         *
*      READ (5,105) ROTLCS,EXPND,FREQSR,DENSE                       *
*      READ(5,11) COMENT                                            *
*      WRITE(6,13) COMENT                                           *
*********************************************************************
                              I
                              I
                              I
*********************************************************************
*      IF(ROTLCS.NE.0.AND.EXPND.NE.0.AND.FREQSR.NE.0) GO TO 37      *------O
*********************************************************************      I
                              I                                            I
                              I                                            I
                              I                                            I
*********************************************************************      I
*      IF(ROTLCS.EQ.0.AND.EXPND.EQ.0) WRITE(6,30)                   *      I
*      IF(FREQSR.EQ.0.AND.EXPND.EQ.0) WRITE(6,31)                   *      I
*      IF(FREQSR.EQ.0.AND.ROTLCS.EQ.0) WRITE(6,32)                  *      I
*      IF(ROTLCS.NE.0.AND.EXPND.NE.0) WRITE(6,33)                   *      I
*      IF(FREQSR.NE.0.AND.EXPND.NE.0) WRITE(6,34)                   *      I
*      IF(FREQSR.NE.0.AND.ROTLCS.NE.0) WRITE(6,35)                  *      I
*********************************************************************      I
                              I                                            I
                              O<-------------------------------------------I-----O
                              I                                            I     I
*********************************************************************      I     I
I-----------*   38 DO277 IOU=1,2C00                                            *      I     I
*********************************************************************      I     I
I                             I                                            I     I
```


XA(IOU)=0.0
277 XB(IOU)=0.0
DO 401 KK=1,100
DO 401 KJ=1,130
401 POINT(KJ,KK)=BLANK
DO 402 KK=1,130
POINT(KK,51)=DASH
402 POINT(KK,50)=DASH
DO 403 KK=1,100

```
      DO 403 KJ=34,125,13
  403 POINT(KJ,KK)=Q
      DO 404 KJ=34,124
      DO 404 KK=7, 93,7
      IF(KK.EQ.49) KK=58
  404 POINT(KJ,KK)=CASH
      DO 405 KK=7, 93,7
      IF(KK.EQ.49) KK=58
  405 POINT(25,KK)=ONE
```

```
      DO 406 KK=26,28
  406 POINT(KK,7)=ZERO
      DO 407 KK=26,27
  407 POINT(KK,14)=ZERO
      POINT(26,21)=ZERO
      DO 408 KK=35,65,7
      IF(KK.EQ.49) KK=58
  408 POINT(24,KK)=DOT
      POINT(25,42)=ZERO
      POINT(25,58)=ZERO
      POINT(26,42)=ONE
      POINT(26,58)=ONE
      POINT(26,79)=ZERO
      POINT(26,86)=ZERO

      POINT(27,86)=ZERO
```

```
      DO 409 KK=26,28
  409 POINT(KK,93)=ZERO
      DO 420 KK=1,100
      IF(KK.EQ.4) KK=98
  420 POINT(26,KK)=Q
      POINT(26,4)=VEE
      POINT(26,97)=AYE
      DO 7681 JACKIE=1,100
      DO 7681 JOANN=1,100
      SAVE1(JACKIE,JOANN)=0.0
```

```
 7681 SAVE2(JACKIE,JOANN)=0.0

      NO=1
      ICOUNT=0
      MI=C
      JZO=1
      K1=1
      III=1
  140 FORMAT(10X,7HDELTAK=,I10,5X,19HPOLY.ADDED IN A(S)=,I10,5X,19HPOLY.
     *ADDED IN B(S)=,I10,5X,9HPROB.NO.=,I10)

   12 READ (5,101) N,IA,IB,IPROB

      IF(IB.NE.0) GO TO 10

      IF(IA.NE.0) GO TO 10

      IF( N.NE.0) GO TO 10

      IF(IPROB.EQ.10000) GO TO 1016

  101 FORMAT (7I10)
```

```
   10 WRITE(6,1)

      DO 5 I = 1,100

      A(I) = 0.0

 5    B(I) = 0.0

      FORK1=0.0
      FORK2 = 0.0
      FORK3 = 0.0
      WRITE (6 ,102)IPROB
  102 FORMAT (1H1, 9X,40HPOLYNOMIAL MULTIPLICATION AND ROOT LOCUS,44X,11
     1HPROBLEM NO.,I5//)

      IF (N) 21, 20, 21

   20 READ(5,103)Y,DY,YT
      WRITE(6,140)N,IA,IB,IPROB
      WRITE(6,141)Y,DY,YT
  141 FORMAT(1H0,5X,10HK-INITIAL=F20.10,5X,12HINCREMENT K=,F20.10,5X,12H
     *K-TERMINATE=,F20.10)
  103 FORMAT (4F10.0)
```

```
      GO TO 25

   21 READ(5,104)(X(I),I=1,N)
      WRITE(6,142) (I,X(I),I=1,N)
  104 FORMAT(7E10.0)

 25   READ (5 ,105)(IGA (I), I = 1, IA)
      WRITE(6,143) (I,IGA(I),I=1,IA)
  143 FORMAT(10X,24HNUMBER OF POLY. IN GROUP, I3,14H OF NUMERATOR=,I10)
  142 FORMAT(10X,2HX(,I3,2H)=,F20.10)
      READ (5 ,105)(IGB(J), J= 1, IB)
      WRITE(6,144) (J,IGB(J),J=1,IB)
  144 FORMAT(10X,24HNUMBER OF POLY. IN GROUP, I3,16H OF DENOMINATOR=,I10
     *)
 105  FORMAT(7I10)

  300 DO 15 I =  1,100

 15   SAVE(I) = 0.0

      ISDEG = 0
      JJ = 1

 200  READ (5 ,106)MDEG,( C(I), I=1, 6)
```

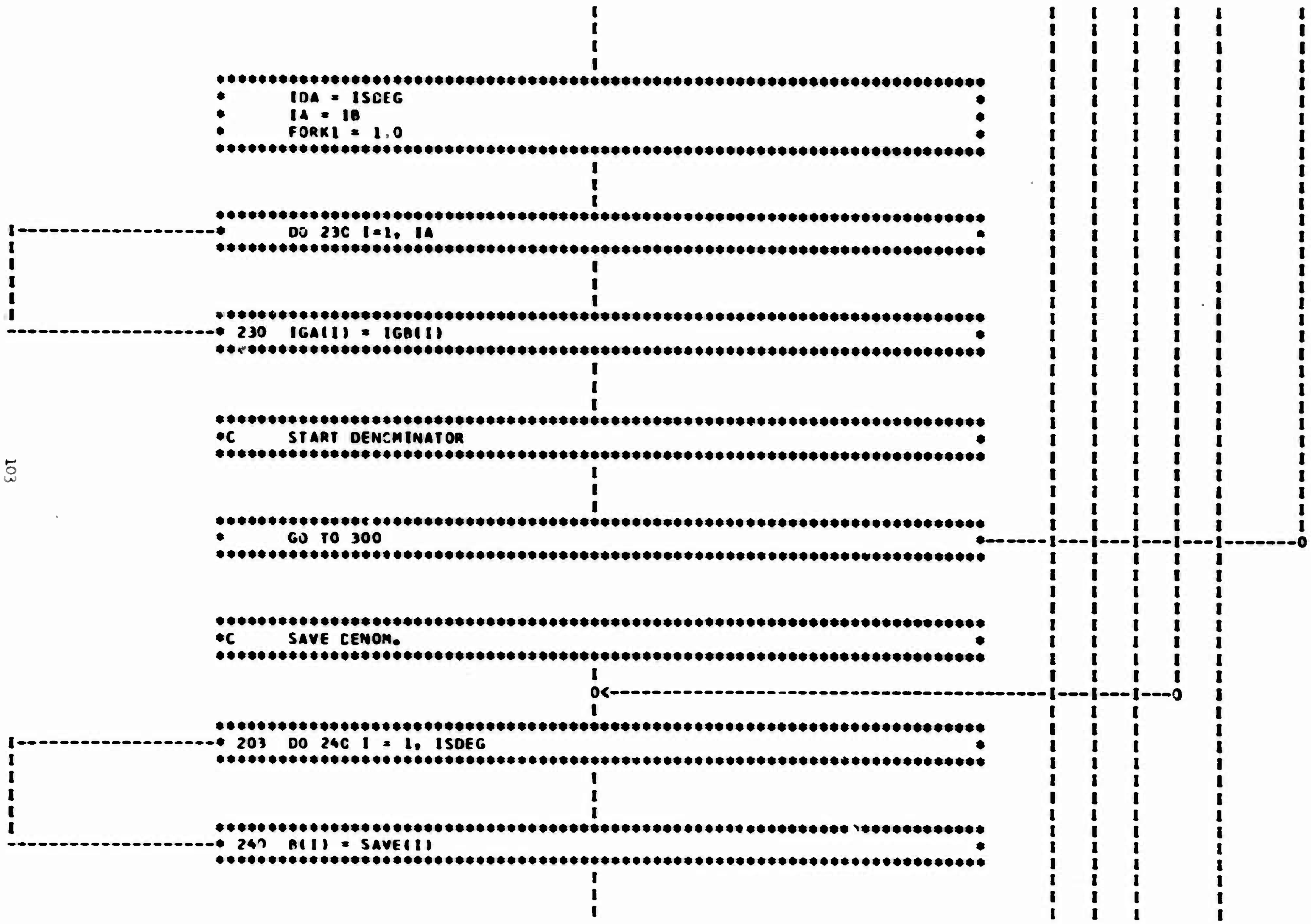
```
      IF(MDEG.GT.6) GO TO 48

      WRITE(6,145) (I,C(I),I=1,MDEG)

      IF (MCEG - 6) 49,  49,  48

 48   READ (5 ,104)(C(I), I = 7, MDEG)
      WRITE(6,145) (I,C(I),I=1,MDEG)
  106 FORMAT(I1C,6E10.0)

 49   IF (ICA(JJ)-1)  50, 51, 50

 51   DO 56  I= 1, MDEG

 56   ANS(I) = C(I)

      IADEG = MCEG
```

```
      IF (ISDEG - MDEG)  52, 52, 53

 53   INDEG = ISDEG

      GO TC 68

 52   INDEG = MDEG

      GO TO 68

 50   READ (5 ,106)NDEG, (D(I), I =1, 6)

      IF(NDEG.GT.6) GO TO 54

      WRITE(6,145) (I,D(I),I=1,NDEG)

      IF (NCEG - 6)  55, 55, 54
```

```
54   REAC (5 ,104)(D(I), I=7, NDEG)
     WRITE(6,145) (I,D(I),I=1,NDEG)

55   IADEG = NDEG + MDEG -1
     CALL POLMPY (C,MDEG,D,NDEG,ANS)
     IGA (JJ) = IGA(JJ) -1

     IF (IGA(JJ) -1) 65, 65, 64

64   DO 60  I = 1, IADEG

60   C(I) = ANS(I)

     MDEG =IADEG

     GO TO 50

65   IF (ISDEG - IADEG) 66,66,67
```

```
 66   INDEG = IADEG

      GO TO 68

 67   INDEG = ISDEG

 68   CALL POLACD (SAVE,ISDEG,ANS,IADEG,ERA)
  145 FORMAT(10X,2HC(,I3,2H)=,F20.10)
      WRITE(6,1)
    1 FORMAT(1H1//)

 6800 IF (ERA(INDEG))6803, 6802, 6803

 6802 INDEG = INDEG - 1

      IF (INDEG)  6801, 6801, 6800

 6801 INDEG = 1
```

```
 6803 JJ = JJ + 1

      DO 70 I = 1, INDEG

 70   SAVE(I) = ERA(I)

      ISDEG = INDEG
      IA = IA -1

      IF (IA) 201, 201, 200

 201  IF (FCRK1) 202, 202, 203

C     SAVE NUMERATOR.

 202  DO 220 I = 1, ISDEG

 220  A(I) = SAVE(I)
```


IDA = ISDEG
IA = IB
FORK1 = 1.0
DO 230 I=1, IA
230 IGA(I) = IGB(I)
C START DENOMINATOR
GO TO 300
C SAVE DENOM.
203 DO 240 I = 1, ISDEG
240 B(I) = SAVE(I)

```
      IDB = ISDEG
      WRITE (6 ,109)
 109  FORMAT (10X,41HCOEFFICIENTS ARE GIVEN IN ASCENDING ORDER/////)

 339  IF (A(IDA)) 340, 341, 340

 341  IDA = IDA - 1

      IF (IDA)  345, 345, 339

 345  WRITE (6 ,120)
 120  FORMAT (1H0,10X,20HPOLYNOMIAL A IS ZERO//)
      FORK2 = 1.0

      GO TO 410

 340  IF (IDA - 2) 346, 347, 335

  346 WRITE(6,121)A(1)
      STAR=STAR1
      PRNT=.TRUE.
 121  FORMAT (1H0,10X,28HPOLYNOMIAL A IS A CONSTANT =,1P1E16.7//)
```

```
      GO TO  410

  347 ROOT = - A(1) / A(2)
      WRITE (6 ,133)A(1), A(2)
  133 FORMAT (10X,21HTHE COEFFICIENTS OF A/1P2E20.7)
      WRITE (6 ,122)ROOT
      STAR=STAR1
      PRNT=.TRUE.
      ANUMB1=ROOT
      ANUMB2=0.0
      CALL PLOTER(SAVE1,SAVE2,ANUMB1,ANUMB2,PCINT,XA,XB,MI,PRNT,STAR,III
     *,NO)
  122 FORMAT (1H0,10X,23HROOT OF POLYNOMIAL A IS,1P1E16.7//)

      GO TO  410

C     WRITE POLYS

  335 IDIA =IDA -1
      WRITE (6,107)IDIA,(A(I),I=1,IDA)
      K = IDA

      DO 800 I = 1,IDA

      AS(I) = A(K)
```

```
  800  K = K-1

       IDP2A=IDA *2
       ID2A= 2 *IDIA
       CALL MULLER (AS,IDIA,ROOTR,ROOTI)

       DO 805 I = 1,IDIA

       SAM = 100. * AMAX1(ABS(ROOTR(I)),ABS(ROOTI(I)))
       IF (SAM + ABS(ROOTR(I)).EQ. SAM) ROOTR(I)= 0.0
       IF (SAM + ABS(ROOTI(I)).EQ. SAM) ROOTI(I)= 0.0

  805  CONTINUE

  400  WRITE (6,111) (ROOTR(I),ROOTI(I),I=1,IDIA)
       CALL ERCHEK(ROOTI,IDIA)
       PRNT=.TRUE.
       STAR=STAR1

       DO 2 III=1,IDIA

       ANUMB1=ROOTR(III)
       ANUMB2=ABS(ROOTI(III))
       CALL PLOTER(SAVE1,SAVE2,ANUMB1,ANUMB2,PCINT,XA,XB,MI,PRNT,STAR,III
      *,NO)
```

```
2 CONTINUE

410  IF(B(IDB)) 411, 412, 411

412  IDB = IDB - 1

     IF (ICB)  445, 445, 410

445  WRITE (6 ,123)
123  FORMAT (1HO,10X,20HPOLYNOMIAL B IS ZERO//)

     IF (FCRK2)12,450,12

450  FORK3 = 1.0

     GO TO 698

411  IF (ICB - 2)  451, 452, 499
```

```
* 451  WRITE (6 ,124)B(IDB)
*      STAR=STAR2
*      PRNT=.FALSE.
*      STAR=STAR3
* 124  FORMAT (1H0,10X,28HPOLYNOMIAL B IS A CONSTANT =,1P1E16.7//)

*      GO TO 698

* 452  ROOT = -B(1) / B(2)
*      WRITE (6 ,134)B(1), B(2)
* 134  FORMAT (10X,21HTHE COEFFICIENTS OF B/1P2E20.7)
*      WRITE (6 ,125)ROOT
*      STAR=STAR2
*      PRNT=.TRUE.
*      ANUMB1=ROOT
*      ANUMB2=0.0
*      CALL PLOTER(SAVE1,SAVE2,ANUMB1,ANUMB2,PCINT,XA,XB,MI,PRNT,STAR,III
*     *,NO)
*      PRNT=.FALSE.
*      STAR=STAR3
* 125  FORMAT (1H0,10X,23HROOT OF POLYNOMIAL B IS,1P1E16.7//)

*      GO TO 698

*  107 FORMAT (10X,42HTHE COEFFICIENTS OF POLYNOMIAL A (ORDER = I3,1H)/ (
*     11P6E20.7))

*  499 IDIB = IDB -1
*      WRITE (6,108)IDIB,(B(I),I=1,IDB)
*  108 FORMAT (////10X,42HTHE COEFFICIENTS OF POLYNOMIAL B (ORDER = I3,1H
*     1)/ (1P6E20.7))
*      K = IDB
```

```
      DO 801 I = 1,IDB

      BS(I) = B(K)

 801  K = K-1

      IDP2B= IDB * 2
      ID2B = 2 * ID1B
      CALL MULLER (BS,ID1B,ROOTR,ROOTI)

      DO 806 I = 1,ID1B

      SAM = 100. * AMAX1(ABS(ROOTR(I)),ABS(ROOTI(I)))
      IF (SAM + ABS(ROOTR(I)).EQ. SAM) ROOTR(I)= 0.0
      IF (SAM + ABS(ROOTI(I)).EQ. SAM) ROOTI(I)= 0.0

 806  CONTINUE

 500  WRITE (6,112)(ROOTR(I),ROOTI(I),I= 1,ID1B)
      CALL ERCHEK(ROOTI,ID1B)
      STAR=STAR2
      PRNT=.TRUE.
```

```
      DO 3 III=1,IDIB

      ANUMB1=ROOTR(III)
      ANUMB2=ABS(ROOTI(III))
      CALL PLOTER(SAVE1,SAVE2,ANUMB1,ANUMB2,POINT,XA,XB,MI,PRNT,STAR,III
     *,NO)

    3 CONTINUE

      PRNT=.FALSE.
      STAR=STAR1
  111 FORMAT (1H0,11X,14HTHE ROOTS OF A/ (1P1E20.7,6H    +I ,1P1E14.7,1P1
     1E20.7,6H    +I ,1P1E14.7,1P1E20.7,6H    +I ,1P1E14.7))
  112 FORMAT (1H0,11X,14HTHE ROOTS OF B/ (1P1E20.7,6H    +I ,1P1E14.7,1P1
     1E20.7,6H    +I ,1P1E14.7,1P1E20.7,6H    +I ,1P1E14.7))

  698 IF (FCRK2)12,699,12

 6991 WRITE (6 ,102)IPROB

  699 IF (FCRK3)12,6991,12
```

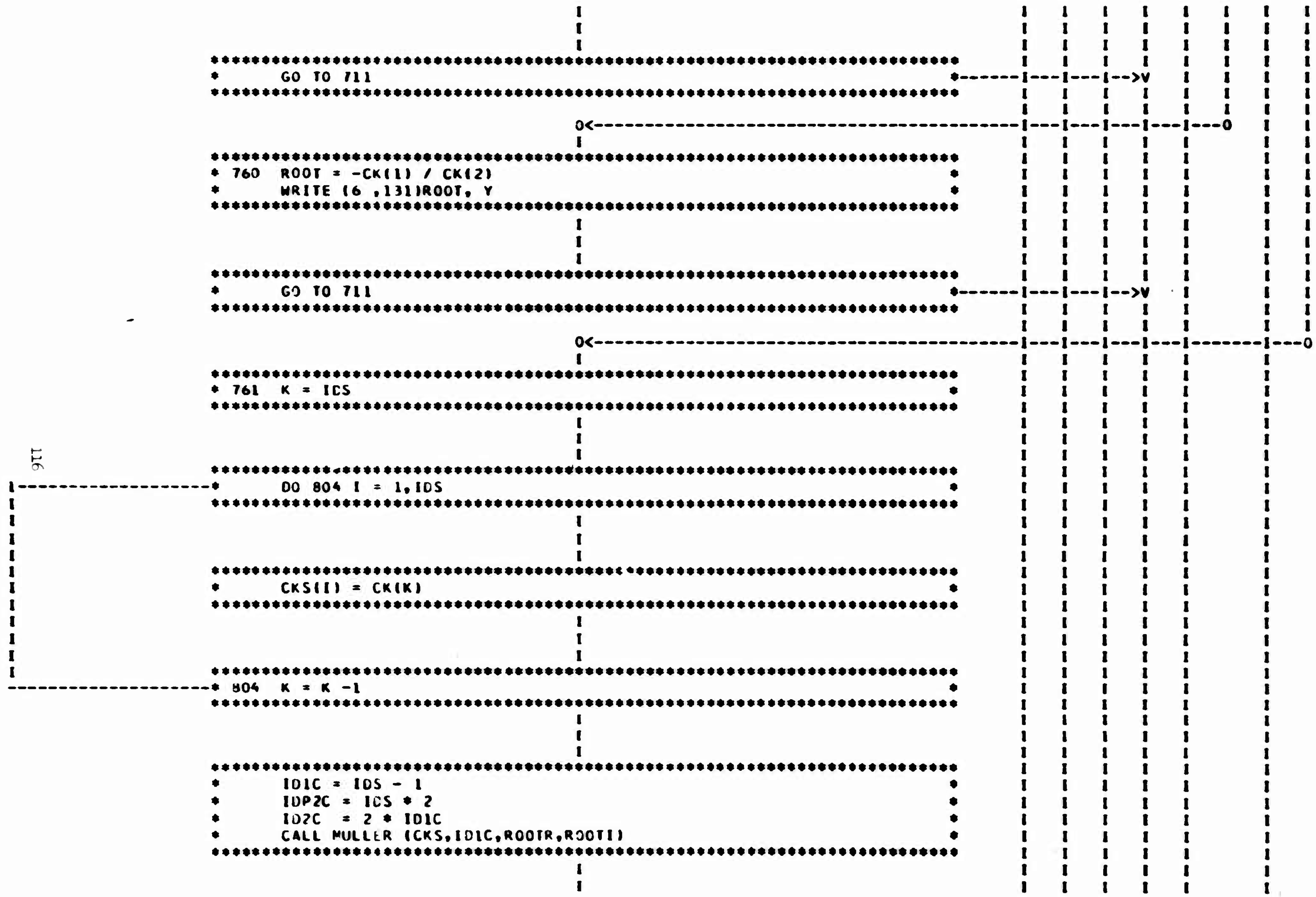
```
*      MSHEET = 5
*C     START K CALCULATIONS

*      IF (N) 702,702,533

* 533  DO 55C I= 1, N

*      DO 541 J= 1, IDA

* 541  ATK(J) =  X(I) * A(J)

*C     COMPUTE RCOTS OF K * A + B
*      IDC= MAXO( IDA, IDB)
*      CALL POLACD (ATK,IDA,B,IDB,CK)
*      IDS = IDC

C<

* 554  IF (CK(IDS))555, 557, 555

* 557  IDS = IDS - 1
```

```
      IF (ICS) 558,558, 554

558   WRITE (6 ,129)X(I)
129   FORMAT (1H0,10X,35HPOLYNOMIAL K*A + B  IS ZERO FOR K =,1P1E16.7//)

      GO TO 550

555   IF (ICS - 2)  559, 560, 561

559   WRITE (6 ,130)CK(IDS), X(I)
130   FORMAT (1H0,10X,35HPOLYNOMIAL K*A + B IS A CONSTANT = ,1P1E15.7,10
     1H FOR K =   ,1P1E14.7//)

      GO TO 550

560   ROOT = -CK(1) / CK(2)
      WRITE (6 ,131)ROOT, X(I)
131   FORMAT (1H0,10X,18HROOT OF K*A + B = ,1P1E15.7,10H FOR K =   ,1P1E1
     14.7//)

      GO TO 550
```

```
561  K = IDS

     DO 803 J = 1,IDS

     CKS(J) = CK(K)

803  K = K - 1

     ID1C = IDS - 1
     IDP2C = IOS * 2
     ID2C  = 2  * ID1C
     CALL MULLER (CKS,ID1C,ROOTR,ROOTI)

     DO 807 J = 1,ID1C

     SAM = 100. * AMAX1(ABS(ROOTR(J)),ABS(ROOTI(J)))
     IF (SAM + ABS(ROOTR(J)).EQ.SAM)  ROOTR(J) = 0.0
     IF (SAM + ABS(ROOTI(J)).EQ.SAM)  ROOTI(J) = 0.0

807 CONTINUE
```

```
      WRITE (6,808)IDIC,X(I),(CK(J),J=1,IDS)
  808 FORMAT (///10X,48HTHE COEFFICIENTS OF POLYNOMIAL K*A + B (ORDER =
     1I3,7H)  K = 1P1E16.7/(1P6E20.7))
  545 WRITE (6,115)(ROOTR(J),ROOTI(J),J=1,IDIC)
      CALL ERCHEK(ROOTI,IDIC)
      CALL SAVER(ROOTR,ROOTI,IDIC,SAVE1,SAVE2,JZC,K1)
  115 FORMAT (1H0,9X,16HROOTS OF K*A + B/(1P1E20.7,6H  + I ,1P1E14.7,1P1
     1E20.7,6H  + I ,1P1E14.7,1P1E20.7,6H  + I ,1P1E14.7))
 5452 MSHEET = MSHEET - 1

      IF (MSHEET) 546, 546, 550

 546  WRITE (6 ,102)IPROB
      MSHEET = 5

 550  CONTINUE

      GO TO 12

 702  DO  705  J = 1, IDA

 705  ATK(J) = Y * A(J)
```

```
C     COMPUTE ROOTS OF K * A + B
      IDC= MAXO(IDA, IDB)
      CALL POLADD (ATK,IDA,B,IDB,CK)
      IDS = IDC

754   IF (CK(IDS))755, 757, 755

757   IDS = IDS - 1

      IF (IDS) 758, 758, 754

758   WRITE (6 ,129)Y

      GO TO 711

755   IF (IDS - 2) 759, 760, 761

759   WRITE (6 ,130)CK(IDS), Y
```


GO TO 711
760 ROOT = -CK(1) / CK(2)
WRITE (6 ,131)ROOT, Y
GO TO 711
761 K = ICS
DO 804 I = 1,IDS
CKS(I) = CK(K)
804 K = K -1
IDIC = IDS - 1
IDP2C = ICS * 2
ID2C = 2 * IDIC
CALL MULLER (CKS,IDIC,ROOTR,ROOTI)

```
      DO 809 I = 1,IDIC

      SAM = 100. * AMAX1(ABS(ROOTR(I)),ABS(ROOTI(I)))
      IF (SAM + ABS(ROOTR(I)).EQ. SAM) ROOTR(I)= 0.0
      IF (SAM + ABS(ROOTI(I)).EQ. SAM) ROOTI(I)= 0.0

  809 CONTINUE

      WRITE (6,808)IDIC,Y,(CK(I),I=1,IDS)
      WRITE (6,115) (ROOTR(J),ROOTI(J),J=1,IDIC)
      CALL ERCHEK(ROOTI,IDIC)
      CALL SAVER(ROOTR,ROOTI,IDIC,SAVE1,SAVE2,JZO,K1)

  711 Y = Y + DY

      IF (Y - YT) 712,712,12

  712 MSHEET = MSHEET - 1

      IF (MSHEET) 713, 713, 702
```

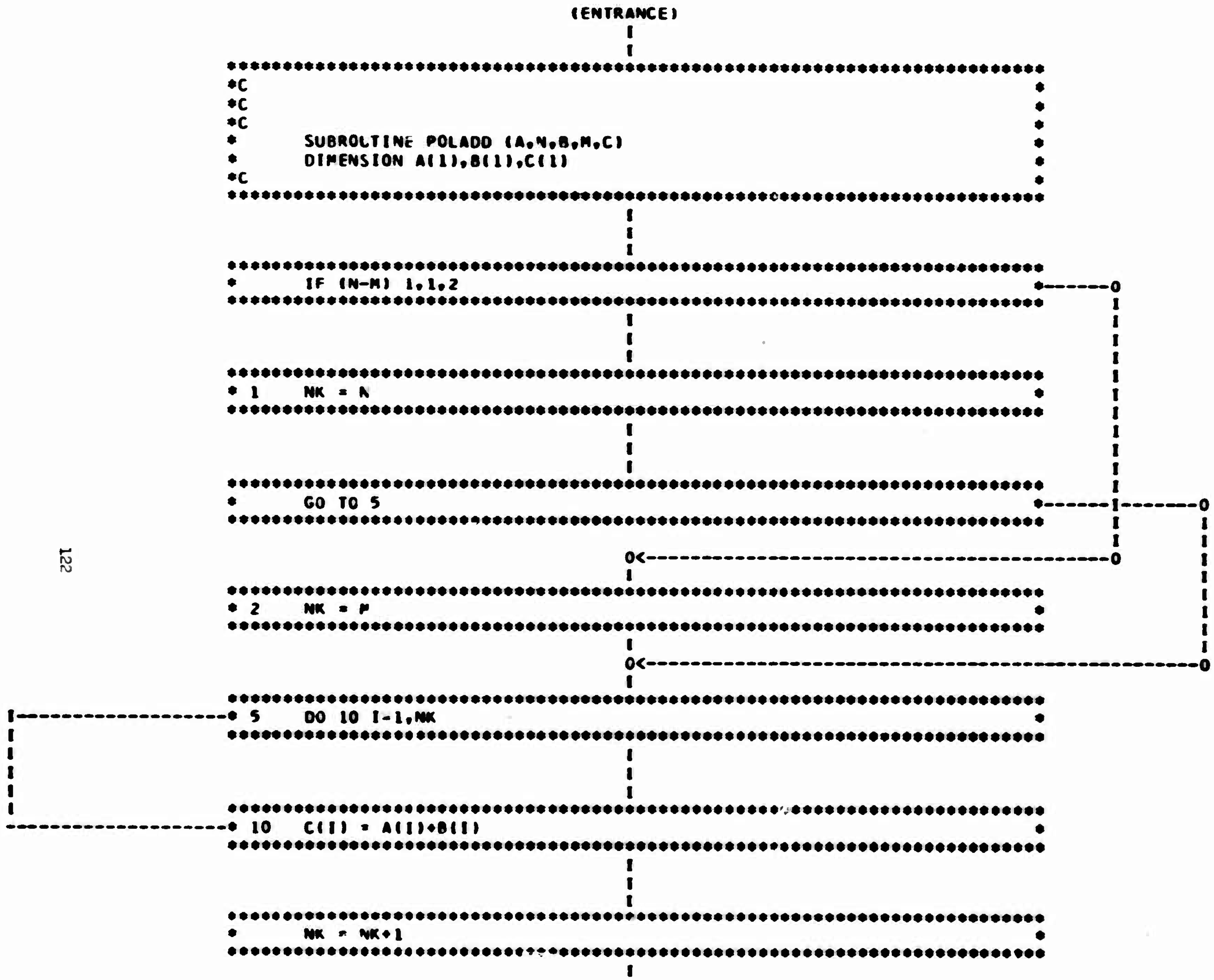
```
  713  WRITE (6 ,102)IPROB
       MSHEET = 5

       GO TO 702

   37 WRITE(6,36)

       GO TO 38

  1016 CALL PLOTER(SAVE1,SAVE2,ANUMB1,ANUMB2,PCINT,XA,XB,MI,PRNT,STAR,III
     *,NO)

  1018 IF(EXPND.EQ.0) GO TO 1017

       READ(5,22)L
   22 FORMAT(I10)

       IF(L.EQ.0) GO TO 1017
```

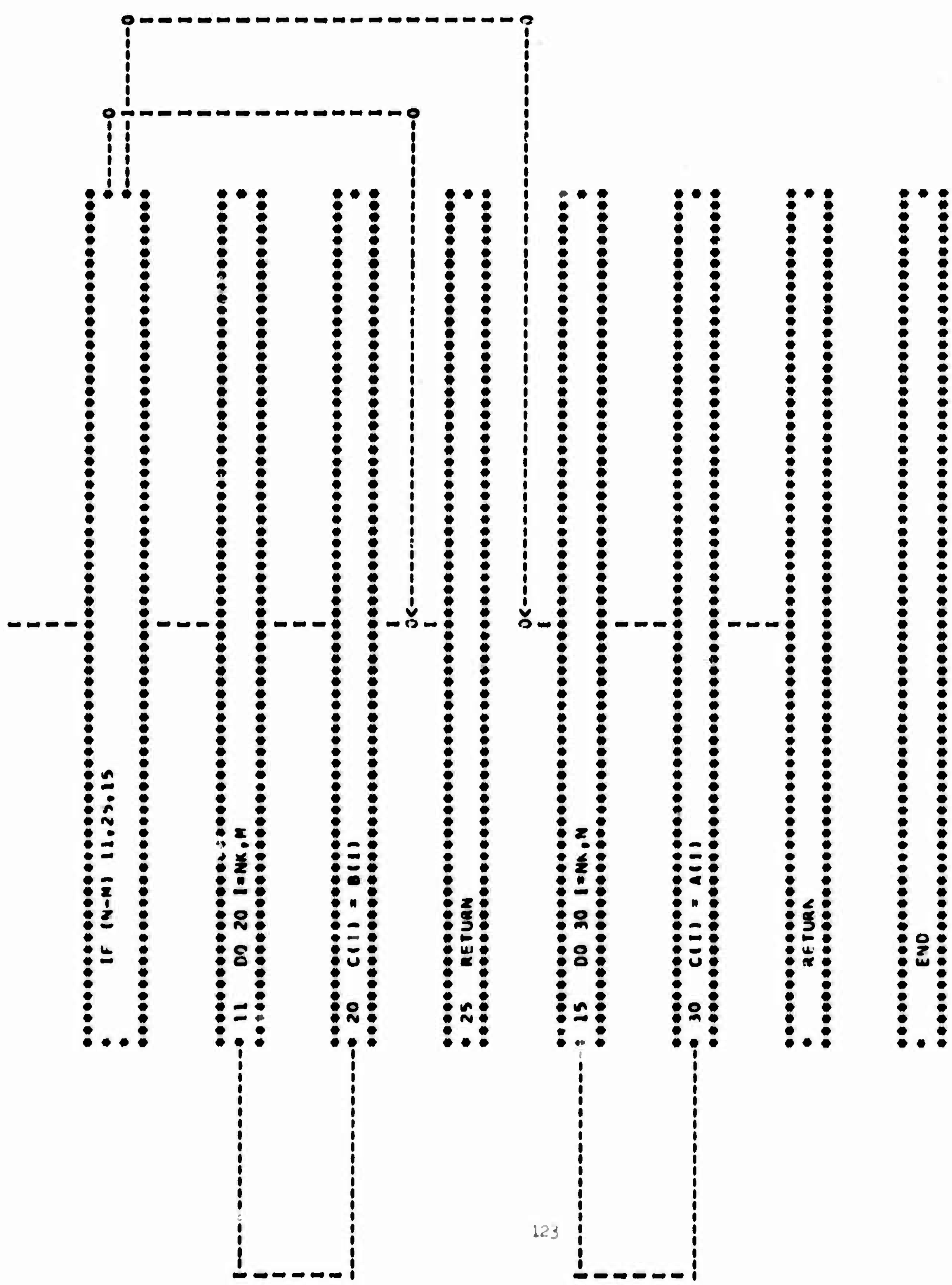
```
                                                I                         I
                                                I                         I
                                                I                         I
*******************************************************************       I
*      CALL EXPAND(L,XA,XB)                                       *       I
*******************************************************************       I
                                                I                         I
                                                O<------------------------O
                                                I
*******************************************************************
* 1017 IF(FREQSR.EQ.C) RETURN                                     *
*      CALL FREQRS(XA,XB)                                         *
*   11 FORMAT(20A4)                                               *
*   13 FORMAT(///////////,20X,20A4)                               *
*   30 FORMAT(////20X,44HONLY THE FREQUENCY RESPONSE HAS BEEN PLOTTED   )*
*   31 FORMAT(////20X,45HONLY THE ROOT LOCUS LOG PLOT HAS BEEN PLOTTED  )*
*   32 FORMAT(////20X,39HONLY THE LINEAR EXPAND HAS BEEN PLOTTED    )  *
*   33 FORMAT(////20X,59HTHE ROOT LOCUS LOG PLOT AND LINEAR EXPAND HAVE B*
*     *EEN PLOTTED      )                                         *
*   34 FORMAT(////20X,62HTHE LINEAR EXPAND AND THE FREQUENCY RESPONSE HAV*
*     *E BEEN PLOTTED    )                                        *
*   35 FORMAT(////20X,68HTHE ROOT LOCUS LOG PLOT AND THE FREQUENCY RESPON*
*     *SE HAVE BEEN PLOTTED     )                                 *
*   36 FORMAT(////20X,28HEVERYTHING HAS BEEN PLOTTED      )       *
*******************************************************************
                                                I
                                                I
                                                I
*******************************************************************
*      RETURN                                                     *
*******************************************************************


*******************************************************************
*      END                                                        *
*******************************************************************
```

(ENTRANCE)

```
C
C
C
C
      SUBROUTINE POLMPY (A,N,B,M,C)
      DIMENSION A(1),B(1),C(1)
      K = M+N
      DO 5 I=1,K
 5    C(I) = 0.0
      DO 10 I=1,N
      L = I-1
      DO 10 J=1,M
      L = L+1
 10   C(L) = C(L)+A(I)*B(J)
```

```
                                  |
                                  |
                                  |
                                  |
*************************************************************************
*       RETURN                                                          *
*************************************************************************


*************************************************************************
*       END                                                             *
*************************************************************************
```


(ENTRANCE)
C
C
C
SUBROUTINE POLADD (A,N,B,M,C)
DIMENSION A(1),B(1),C(1)
C
IF (N-M) 1,1,2
1 NK = N
GO TO 5
2 NK = M
5 DO 10 I=1,NK
10 C(I) = A(I)+B(I)
NK = NK+1


IF (N-M) 11,25,15
11 DO 20 I=NK,M
20 C(I) = B(I)
25 RETURN
15 DO 30 I=NK,N
30 C(I) = A(I)
RETURN
END

```
(ENTRANCE)

C
C
C
C
      SUBROUTINE MULLER(COE,N1,ROOTR,ROOTI)
       DIMENSION COE(1),ROOTR(1),ROOTI(1)
      N2=N1+1
      N4=C
      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=C.
      L=1
      N3=1
      ALP1R=AXR
      ALP1I=AXI
      M=1
```

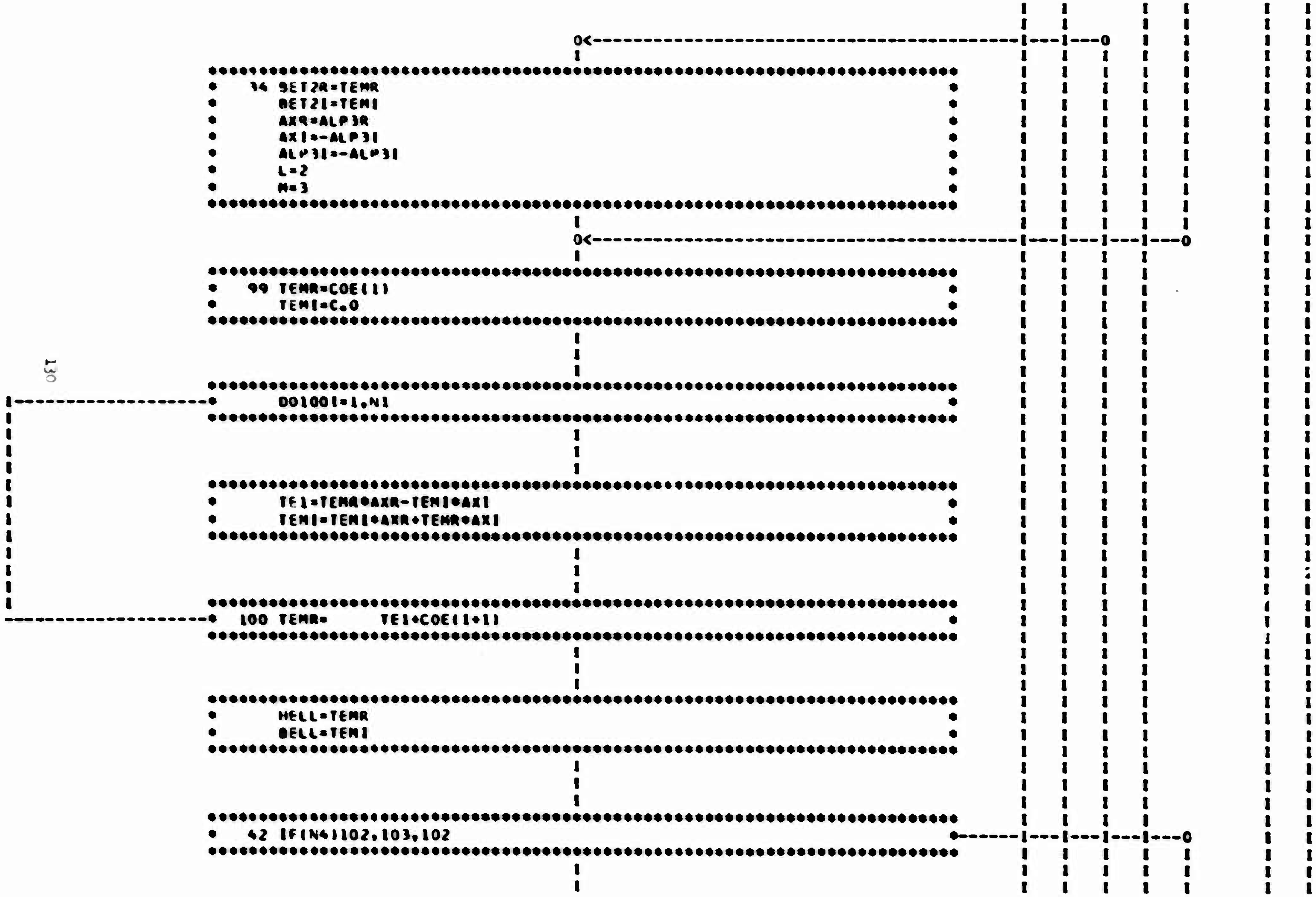
```
      GOTO99

   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 = SQRT (TE1*TE1+TE2*TE2)

      IF(TE1)113,113,112

  113  TE4 = SQRT (.5 * (TEM - TE1))
      TE3=.5*TE2/TE4

      GO TO 111

  112  TE3 = SQRT (.5 * (TEM + TE1))

      IF(TE2)110,200,200

  110 TE3=-TE3
```

```
  230 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 (ABS (HELL) + ABS (BELL) - 1.E-20) 18,18,16

16    TE7 = ABS (ALP3R - AXR) + ABS (ALP3I - AXI)

      IF (TE7 / (ABS (AXR) + ABS (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=TEM3
    BET3I=TEMI

    IF(N3-100)14,18,18

 18 N4=N4+1
    ROOTR(N4)=ALP4R
    ROOTI(N4)=ALP4I
    N3=0
```

```
41 IF(N4-N1)30,37,37

37 RETURN

30   IF (ABS (ROOTI(N4)) - 1.E-5)10,10,31

31 GO TO(32,10),L

32 AXR=ALP1R
   AXI=-ALP1I
   ALP1I=-ALP1I
   M=5

   GO TO 99

33 RET1R=TEMR
   RET1I=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=C.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

```
(ENTRANCE)

C
C
C
C
      SUBROUTINE EXPLOT(XARRAY,IDIMEN,YARRAY,KDIMEN,BLIMIT,LET)
      INTEGER POINT(130,100),GRAPHL(120,58),STAR
      DIMENSION XARRAY(IDIMEN),YARRAY(KDIMEN)
      DIMENSION ENCRM1(120),ENCRM2(58)
      DIMENSION SAVE1(100,100),SAVE2(100,100)
      COMMON SAVE1,SAVE2,POINT
      EQUIVALENCE(POINT(1,1),GRAPHL(1,1))
      DATA STAR/1H*/
      WRITE(6,1)
      READ(5,10)D1,D2
      D3=BLIMIT
      D4=-BLIMIT
      IF(D2.GT.D1) CALL SWITCH(D1,D2)
      IF(D4.GT.D3) CALL SWITCH(D3,D4)
      IC1=D3
      JDIMEN=KDIMEN
      CALL BLANKR(GRAPHL,120,58,120)

      DO 22 I=1,JDIMEN

      X=XARRAY(I)
      Y=YARRAY(I)

      IF(X.EQ.0.0.AND.Y.EQ.0.0) GO TO 22

      CALL LINAR (X,LIMITX,120,D1,D2,ENCRM1)
      CALL LINAR (Y,LIMITY,58,D3,D4,ENCRM2)
```

```
      IF(LIMITX.EQ.C.OR.LIMITY.EQ.0) GO TO 22

      LIMITX=121-LIMITX
      GRAPHL(LIMITX,LIMITY)=STAR

   22 CONTINUE

      WRITE(6,30) IC1,LET,(GRAPHL(I,1),I=1,120)
      IC1=C4
      WRITE(6,34)((GRAPHL(I,J),I=1,120),J=2,57)
      WRITE(6,30) IC1,LET,(GRAPHL(I,58),I=1,120)
      WRITE(6,35) D2,D1
    1 FORMAT(1H1)
   10 FORMAT(6F10.0,20X)
   30 FORMAT(1X,I4,A4,1X,120A1)
   34 FORMAT(10X,120A1)
   35 FORMAT(3X,1P1E10.3,7HRAD/SEC,91X,1P1E10.3,7HRAD/SEC)

   61 RETURN

      END
```

```
(ENTRANCE)

C
C
C
C
      SUBROUTINE LINAR (Y,LIMITY,IDIMEN,DIMNS1,DIMNS2,ENCRMT)
      DIMENSION ENCRMT(IDIMEN)
      DIMENS=DIMNS1-DIMNS2
      LIMITY=0
      J=IDIMEN-1
      A=J
      DELTA=DIMENS/A
      ENCRMT(1)=DIMNS1
      ENCRMT(IDIMEN)=DIMNS2

      DO 10 I=2,J

   10 ENCRMT(I)=ENCRMT(I-1)-DELTA

      DO 11 I=1,J

   11 IF(Y.LE.ENCRMT(I).AND.Y.GE.ENCRMT(I+1))GO TO 20

      GO TO 40

   20 LIMITY=I

   40 RETURN

      END
```

```
(ENTRANCE)
C
C
      SUBROUTINE SWITCH(X,Y)
      TEST=Y
      Y=X
      X=TEST

      RETURN

      END
```

```
(ENTRANCE)
C
C
      SUBROUTINE ERCHEK(X,I)
      DIMENSION X(1C0)
      DATA ERRLIM/0.1E-8/

      DO 10 J=1,I

   10 IF(ABS(X(J)).LT.ERRLIM) X(J)=0.0

      RETURN

      END
```

(ENTRANCE)

```
C
C
C
C
      SUBROUTINE EXPAND(ILOVE,X,Y)
      DIMENSION SAVE1(100,100),SAVE2(100,100),X2(2000),Y2(2000),X(2000),
     *          Y(2000),X1(2000),Y1(2000),Z(125),Z1(87)
      INTEGER SPOT(130,50),DASH,Q,BLANK,POINT(130,100)
      DATA Q/1HI/, DASH/1H-/,BLANK/1H /
      DATA APLUS/0.0/,BPLUS/0.0/,AMINUS/0.0/,BMINUS/0.0/
      COMMON /INFO4/ Q,BLANK,DASH
      COMMON /PSYNE/AM,AP,BM,BP
      COMMON /PJ/X1,Y1
      COMMON SAVE1,SAVE2,POINT
      EQUIVALENCE(AP,APLUS),(AM,AMINUS),(BM,BMINUS),(BP,BPLUS)
      EQUIVALENCE(X1(1),Z(1)),(Y1(1),Z1(1))
      EQUIVALENCE (SPOT(1,1),POINT(1,1)),(I1,I2)
```

```
      DO 40 J=1,ILOVE
```

```
      DO 50 M=1,2000
```

```
      X1(M)=0.0
```

```
   50 Y1(M)=0.0
```

```
      CALL  BLANKR(SPOT,130,50,126)
```

```
      DO 54 M=1,2000

      X2(M)=0.0

   54 Y2(M)=0.0

      READ(5,10) OMEGA,ENCRMT,SIGMA,DELTA
      OMEGA=ABS(OMEGA)
   10 FORMAT(4F10.0)
      A=ENCRMT
      PERCNT=A*0.01
      B=OMEGA
      B=B*PERCNT
      APLUS=B+OMEGA
      AMINUS=OMEGA-B
      C=APLUS
      D=AMINUS
      I1=1

      DO 41 L=1,2000

      IF(Y(L).LT.D)GO TO 41        ------O

      IF(Y(L).GT.C)GO TO 41        ----->V
```

```
   23 X1(I1)=X(L)
      Y1(I1)=Y(L)
      I1=I1+1

   41 CONTINUE

      A=DELTA
  901 PERCNT=A*C.01
      B=SIGMA
  903 B=B*PERCNT
      BPLUS=SIGMA+B
      BMINUS=SIGMA-B
      C=BPLUS
      D=BMINUS
      I2=1

  910 DO 42 L=1,2C00

      IF(C.GT.0.0) GO TO 60

      IF(X1(L).GT.D) GO TC 42

      IF(X1(L).LT.C) GO TC 42
```

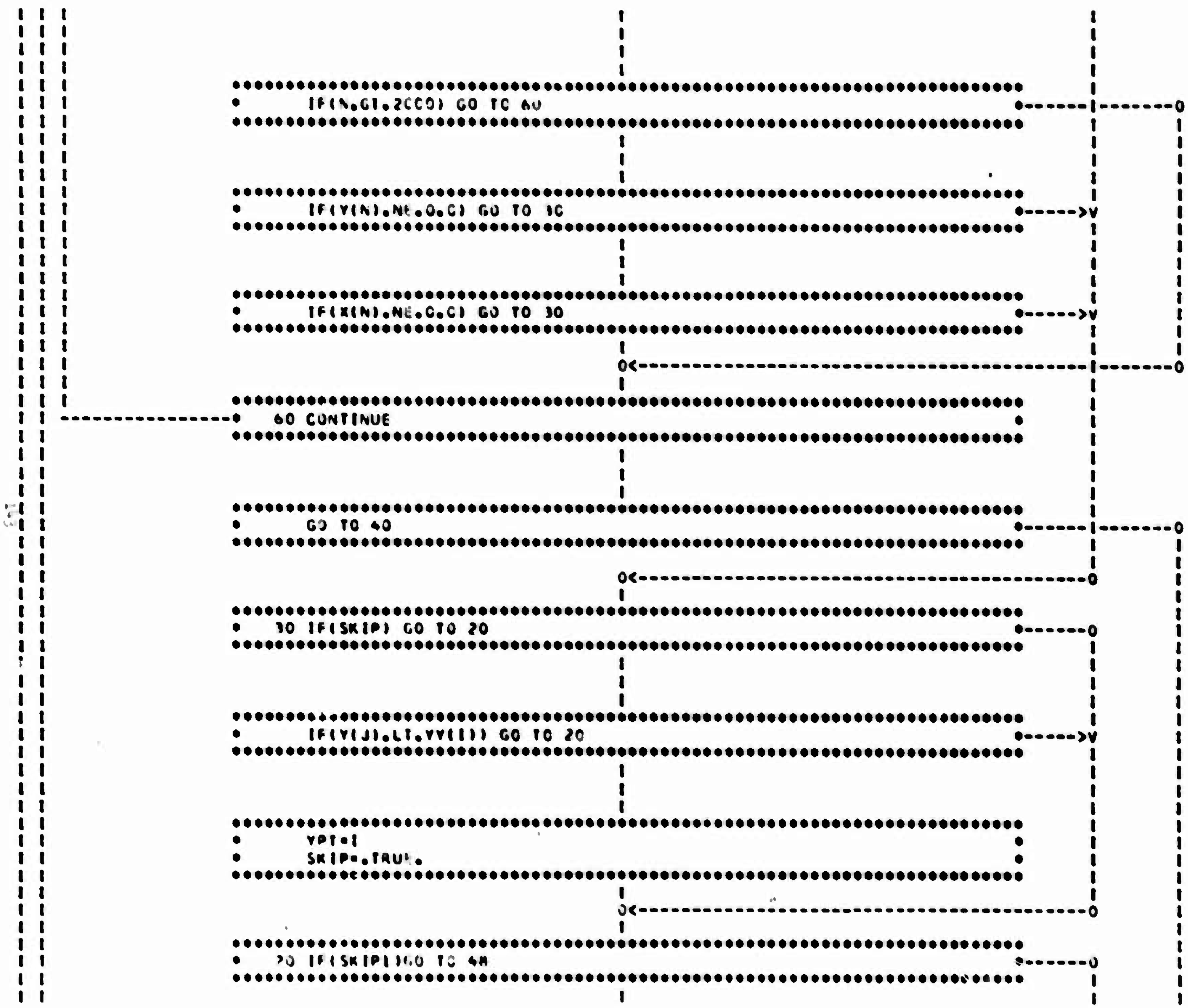
```
      GO TO 25
   60 IF(X1(L).GT.C) GO TO 42
      IF(X1(L).LT.D) GO TO 42
   25 X2(I2)=X1(L)
      Y2(I2)=Y1(L)
      I2=I2+1
   42 CONTINUE
      CALL SPLIT(X2,Y2,SPOT,APLUS,AMINUS,BPLUS,BPINUS)
   40 CONTINUE
      RETURN
      END
```

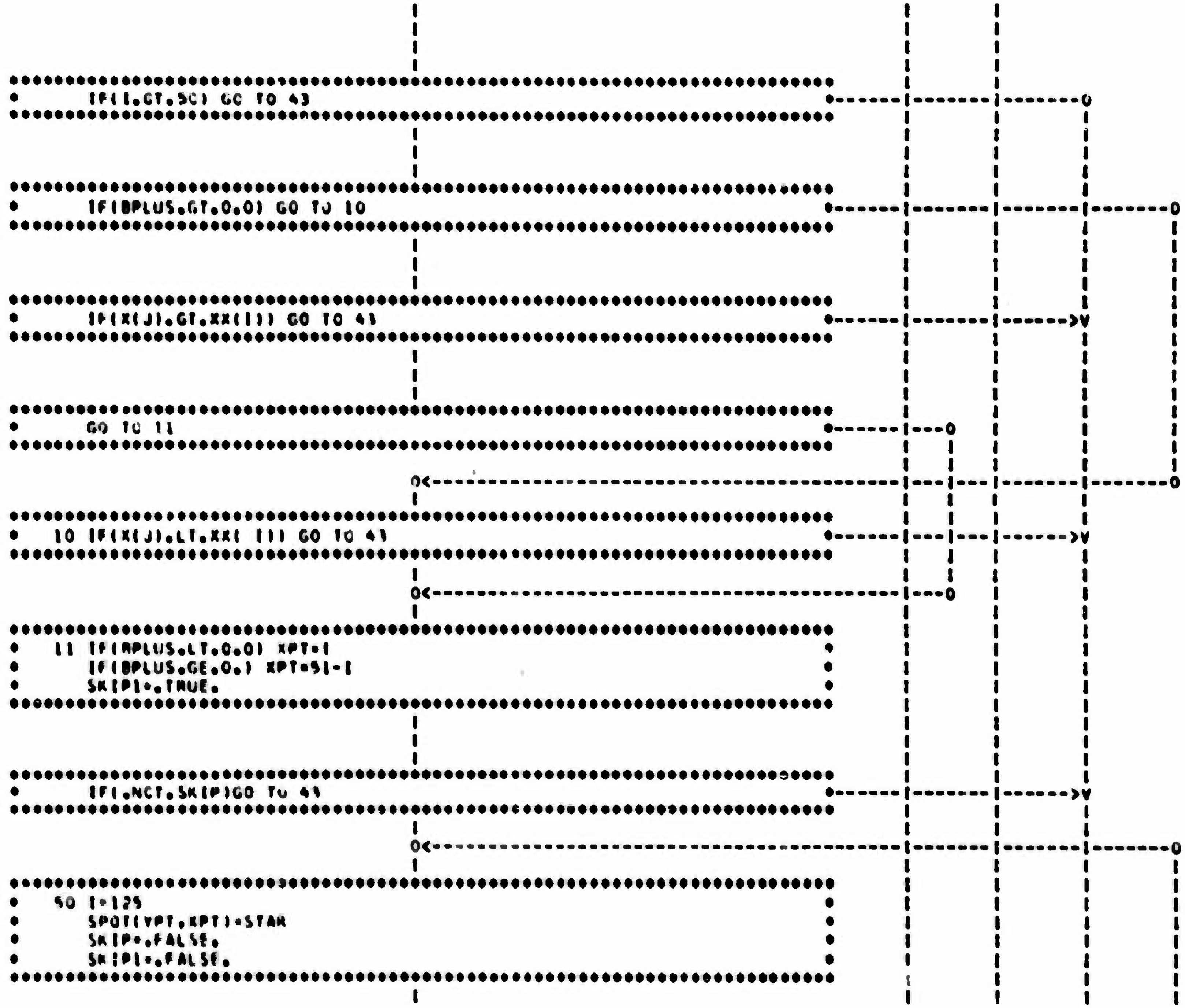
(ENTRANCE)

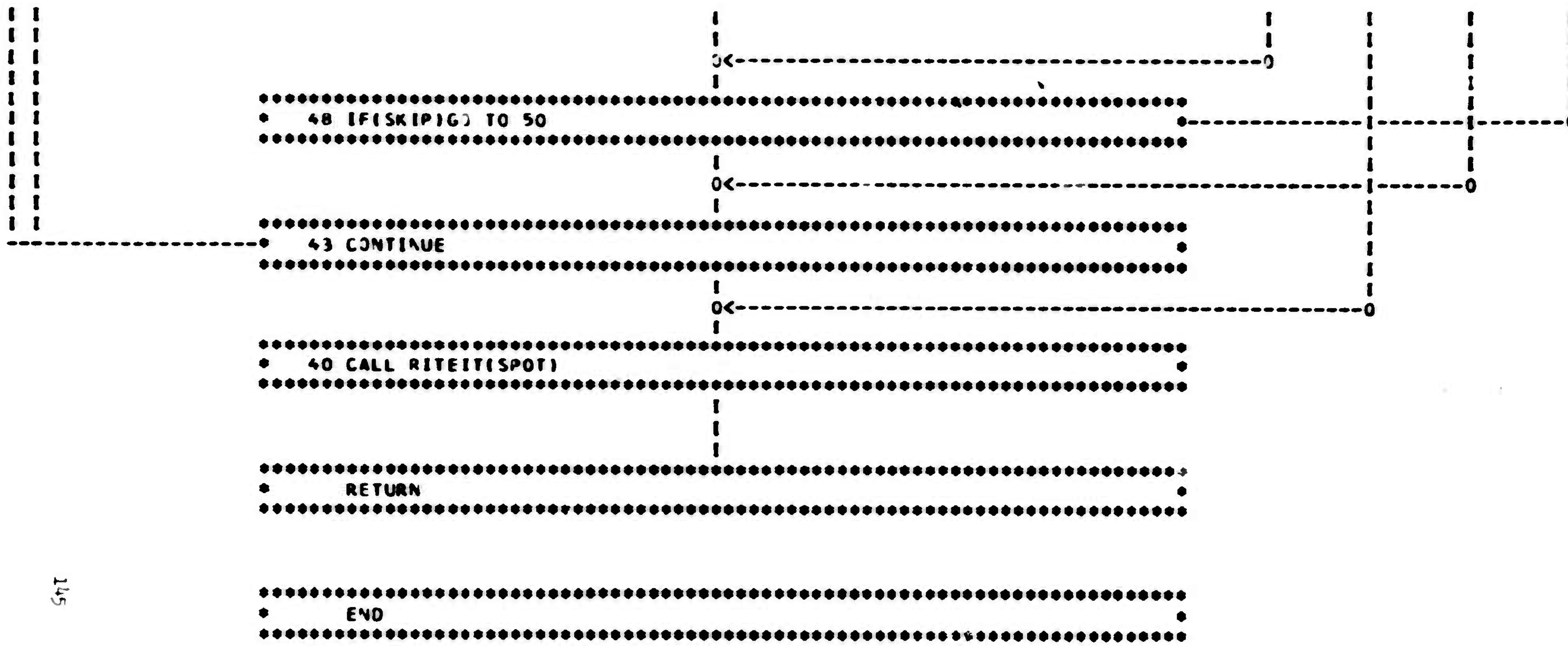
```
C
C
C
C
      SUBROUTINE SPLIT(X,Y,SPOT,APLUS,AMINUS,BPLUS,BMINUS)
      DIMENSION X1(2000),Y1(2000)
      DIMENSION X(2000),Y(2000),Z(125),Z1(87)
      INTEGER SPOT(130,50)
      COMMON /PJ/X1,Y1
      EQUIVALENCE(X1(1),Z(1)),(Y1(1),Z1(1))
      A=APLUS
      B=AMINUS
      C=A-B
      D=BPLUS
      E=BMINUS
      G=ABS(D)
      H=ABS(E)
      F=G-H
      DELTA=C/124.0
      DIFF=F/50.0

      DO 11 J=1,124

      Z(J)=A

   11 A=A-DELTA

      Z(125)=B
      IF(C.LT.0.0)DIFF=-DIFF

      DO 12 J=1,49
```

```
      Z1(J)=D

   12 D=D-DIFF

      Z1(50)=E
      CALL BRAKUP(Z,X,Y,SPOT,Z1)

      RETURN

      END
```

```
(ENTRANCE)

C
C
C
C
      SUBROUTINE WRAKUP(YY,X,Y,SPOT,XX)
      LOGICAL SKIP,SKIP1
      INTEGER SPOT(130,90),                XPT,YPT,STAR
      DATA STAR/1H*/
      DIMENSION X(2000),Y(2000),YY(125),XX(87)
      COMMON /PAYNE/AM,AP,BM,BP
      EQUIVALENCE(AP,APLUS),(AM,AMINUS),(BM,BMINUS),(BP,BPLUS)
      SKIP=.FALSE.
      SKIP1=.FALSE.

      DO 43 J=1,2000

      DO 43 I=1,125

      IF(Y(J).NE.0.0) GO TO 30

      IF(X(J).NE.0.0) GO TO 30

      L=J+1
      M=J+6

      DO 60 N=L,M
```


IF(N.GT.2CCO) GO TO 60
IF(Y(N).NE.O.C) GO TO 3C
IF(X(N).NE.O.C) GO TO 30
60 CONTINUE
GO TO 40
30 IF(SKIP) GO TO 20
IF(Y(J).LT.YY(I)) GO TO 20
YPT=I
SKIP=.TRUE.
20 IF(SKIP1)GO TO 48


IF(I.GT.50) GO TO 43
IF(BPLUS.GT.0.0) GO TO 10
IF(X(J).GT.XX(I)) GO TO 41
GO TO 11
10 IF(X(J).LT.XX(I)) GO TO 41
11 IF(BPLUS.LT.0.0) XPT=I
IF(BPLUS.GE.0.) XPT=51-I
SKIP1=.TRUE.
IF(.NOT.SKIP)GO TO 41
50 I=125
SPOT(YPT,XPT)=STAR
SKIP=.FALSE.
SKIP1=.FALSE.

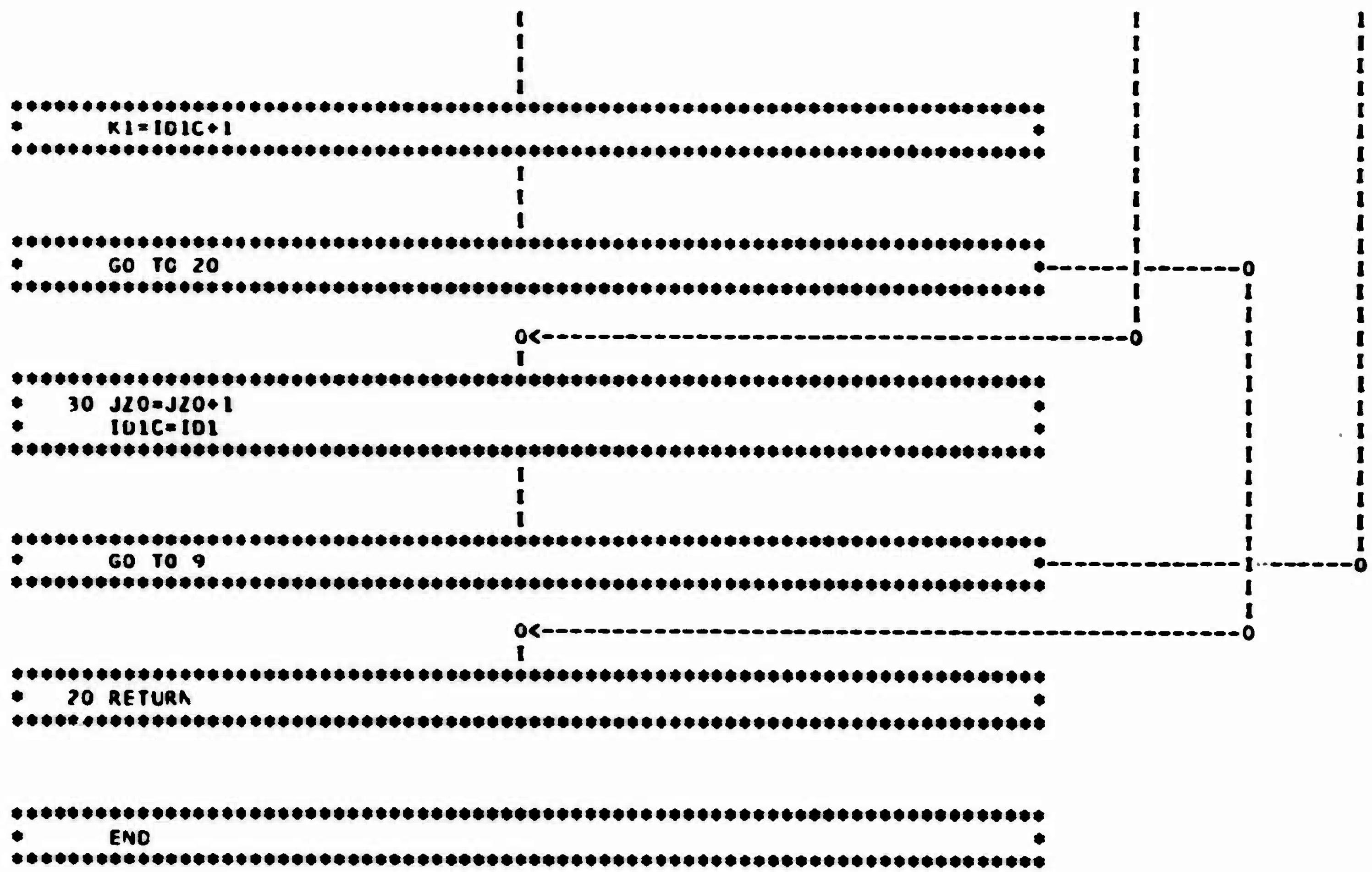

48 IF(SKIP(G)) TO 50
43 CONTINUE
40 CALL RITEIT(SPOT)
RETURN
END

```
                    (ENTRANCE)

C
C
C
C
      SUBROUTINE RITEIT(SPOT)
      COMMON /PAYNE/AM,AP,BM,BP
      EQUIVALENCE(AP,APLUS),(AM,AMINUS),(BM,BMINUS),(BP,BPLUS)
      INTEGER SPOT(130,5C)

      IF(BP.LT.BM) GO TO 40

      A=BP
      B=BM
      WRITE(6,1)
    1 FORMAT(1H1,40X,43HCOMPLEX FREQUENCY PLANE,RIGHT HAND QUADRANT   )

      GO TO 30

0<-
   40 A=BM
      B=BP
      WRITE(6,2)
    2 FORMAT(1H1,40X,43HCOMPLEX FREQUENCY PLANE, LEFT HAND QUADRANT   )

0<-
   30 WRITE(6,12)AP,AM
   12 FORMAT(114X,15H<-------J-OMEGA,/1X,F8.2,112X,F8.2)
      WRITE(6,14)B
   14 FORMAT(124X,5HSIGMA,/126X,1HI,/126X,1HV,/121X,F8.2)
      WRITE(6,11) SPOT
   11 FORMAT(1X,130A1)
      WRITE(6,15)A
   15 FORMAT(60X,27HLINEAR EXPAND PLOT(RAD/SEC) ,34X,F8.2)
```

RETURN

END

```
(ENTRANCE)

C
C
      SUBROUTINE SAVER(ROOTR,ROOTI,IDIC,SAVE1,SAVE2,JZO,K1)
      DIMENSION SAVE1(100,100),SAVE2(100,100),ROOTR(100),ROOTI(100)

      IF(K1)30,9,10

    9 K1=1

   10 ID1=IDIC
      IDIC=IDIC+(K1-1)

      IF(IDIC.GE.100)GO TO 30

   50 DO 40 IZAP=K1,IDIC

      IZA=IZAP-(K1-1)
      SAVE1(JZO,IZAP)=ROOTR(IZA)
      SAVE2(JZO,IZAP)=ROOTI(IZA)

   40 CONTINUE
```


K1=IDIC+1
GO TC 20
30 JZO=JZO+1
IDIC=IDL
GO TO 9
20 RETURN
END

```
(ENTRANCE)
C
C
C
C
      SUBROUTINE PLOTER(SAVE1,SAVE2,ANUMB1,ANUMB2,POINT,XA,XB,PI,PRNT,
     *STAR,NBB,NO)
      DIMENSION SAVE1(100,100),SAVE2(10C,100),XA(2000),XB(2000)
      INTEGER POINT(130,100),DASH,BLANK,STAR,ROTLCS,EXPND,FREQSR,DENSE
      COMMON /INFO7/ ICOUNT
      COMMON /INFO8/ROTLCS,EXPND,FREQSR
      COMMON /INFO9/ DENSE
      LOGICAL PRNT

      IF(PRNT) GO TC 43

      DO 40 NAB=1,1C0

      DO 40 NBB=1,1C0

      IF(SAVE1(NAB,NBB))41,42,41

   42 IF(SAVE2(NAB,NBB))41,40,41

   41 ANUMB1=SAVE1(NAB,NBB)
      ANUMB2=SAVE2(NAB,NBB)
      ANUMB2=ABS(ANUMB2)
```

```
43 AZZ=ANUMB2
   AZZ=AZZ*100000.0
   NZZ=AZZ

   IF(NZZ.EQ.0) GO TO 45

50 NBR=NBB+1

45 CALL EXCUTE(ANUMB1,ANUMB2,POINT,XA,XB,MI,NO,STAR)

   IF (DENSE.NE.1) GO TO 10

   ICOUNT=ICOUNT+1

   GO TO 11

10 IF(PRNT) ICOUNT=ICOUNT+1
```

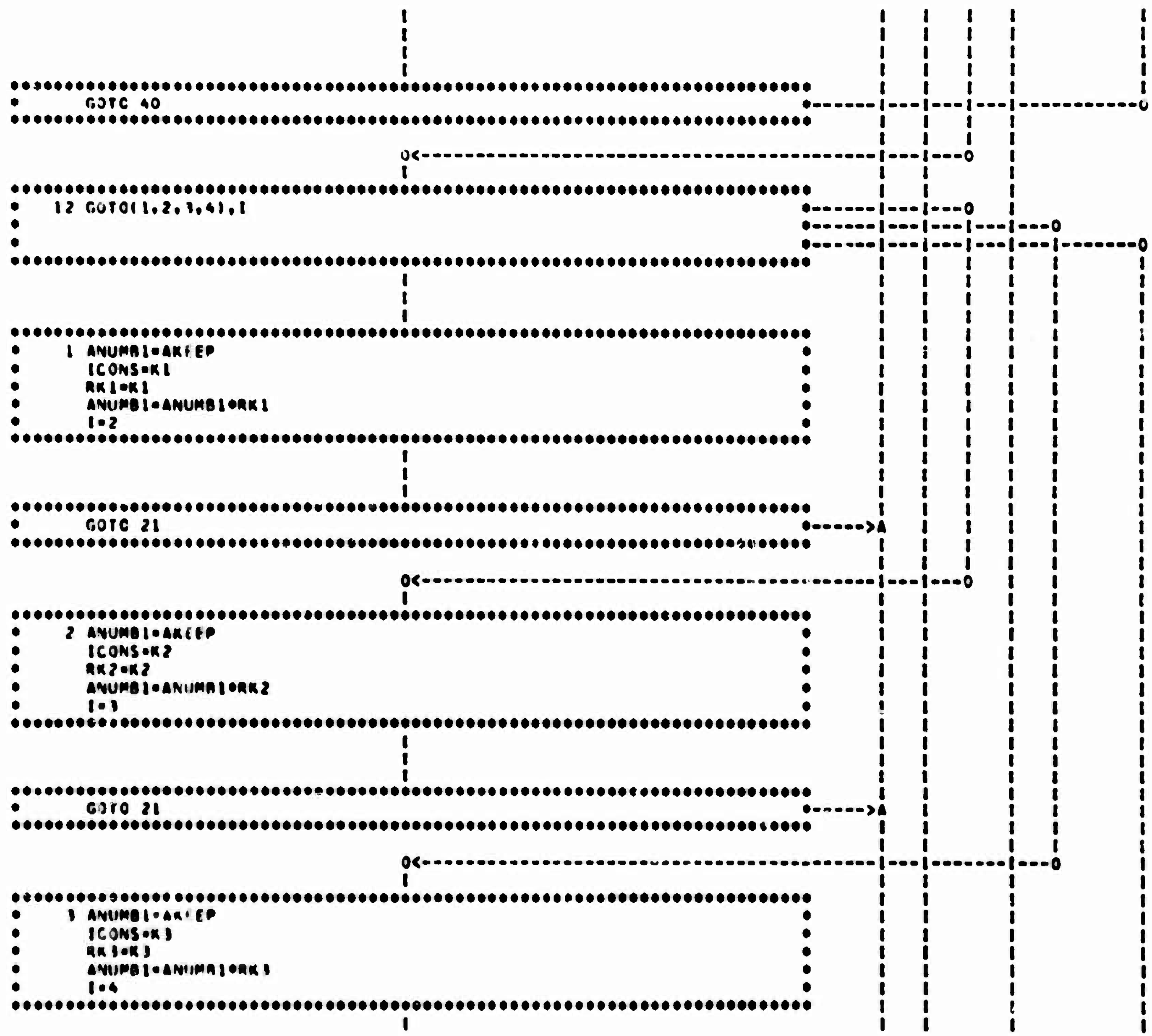
```
   11 IF(PRNT) RETURN

   40 CONTINUE

      IF(ROTLCS.EQ.C) RETURN
   70 CALL WRITIT(XA,XB)
      CALL PREPAR(POINT)

      RETURN

      END
```

```
(ENTRANCE)
    |
*C
*C
*C
*C
*       SUBROUTINE EXCUTE(ANUMB1,ANUMB2,POINT,XA,XB,MI,NO,STAR)
*       LOGICAL SKIP1,SKIP2 ,LESS
*       INTEGER RCTLCS,EXPND,FREQSR
*       DIMENSION XA(2000),XB(2000)
*       INTEGER POINT(130,100),DASH,BLANK,STAR
*       COMMON /INFOB/RCTLCS,EXPND,FREQSR
*       DATA        K1/10/,K2/100/,K3/1000/,K4/10000/,NEGONE/-1/
*       LESS=.FALSE.
*       I=0
*       J=0
*       XA(NO)=ANUMB1
*       XB(NO)=ABS(ANUMB2)
*       NO=NO+1
*       IF(RCTLCS.EQ.0) RETURN

*       IF(ABS(ANUMB1).GT.10000.0) GO TO 50          *------0

*       IF(ABS(ANUMB2).GT.10000.0) GO TO 50          *----->V

*       IF(ABS(ANUMB1).EQ.0.0) GO TO 500             *------|-------0

*       IF(ABS(ANUMB2).EQ.0.0) GO TO 500             *------|------>V

*       IF(ABS(ANUMB2).LT.0.001) GO TO 50            *----->V
```

```
                                         I                      I       I
                                         I                      I       I
                                         I                      I       I
                                         I                      I       I
*************************************************************** I       I
*     IF(ABS(ANUMB1).LT.0.001) GO TO 50                         *----->V       I
*************************************************************** I       I
                                         I                      I       I
                                         0<---------------------I-------0
                                         I                      I
*************************************************************** I
*  500 CALL SCALE1(K1,K2,K3,K4,NEGONE,I,ANUMB1,ICONS,LESS,SKIP1) * I
*      CALL SCALE2(K1,K2,K3,K4,J,JCONS,ANUMB2,SKIP2)            * I
*      CALL WPOINT(J,JCONS,L,ANUMB2,SKIP2)                      * I
*      CALL SPOINT(I,ANUMB1,LESS,ICONS,L,POINT,SKIP1,NO,XA,XB,STAR) * I
*************************************************************** I
                                         I                      I
                                         0<---------------------0
                                         I
***************************************************************
*   50 RETURN                                                   *
***************************************************************


***************************************************************
*      END                                                      *
***************************************************************
```

(ENTRANCE)

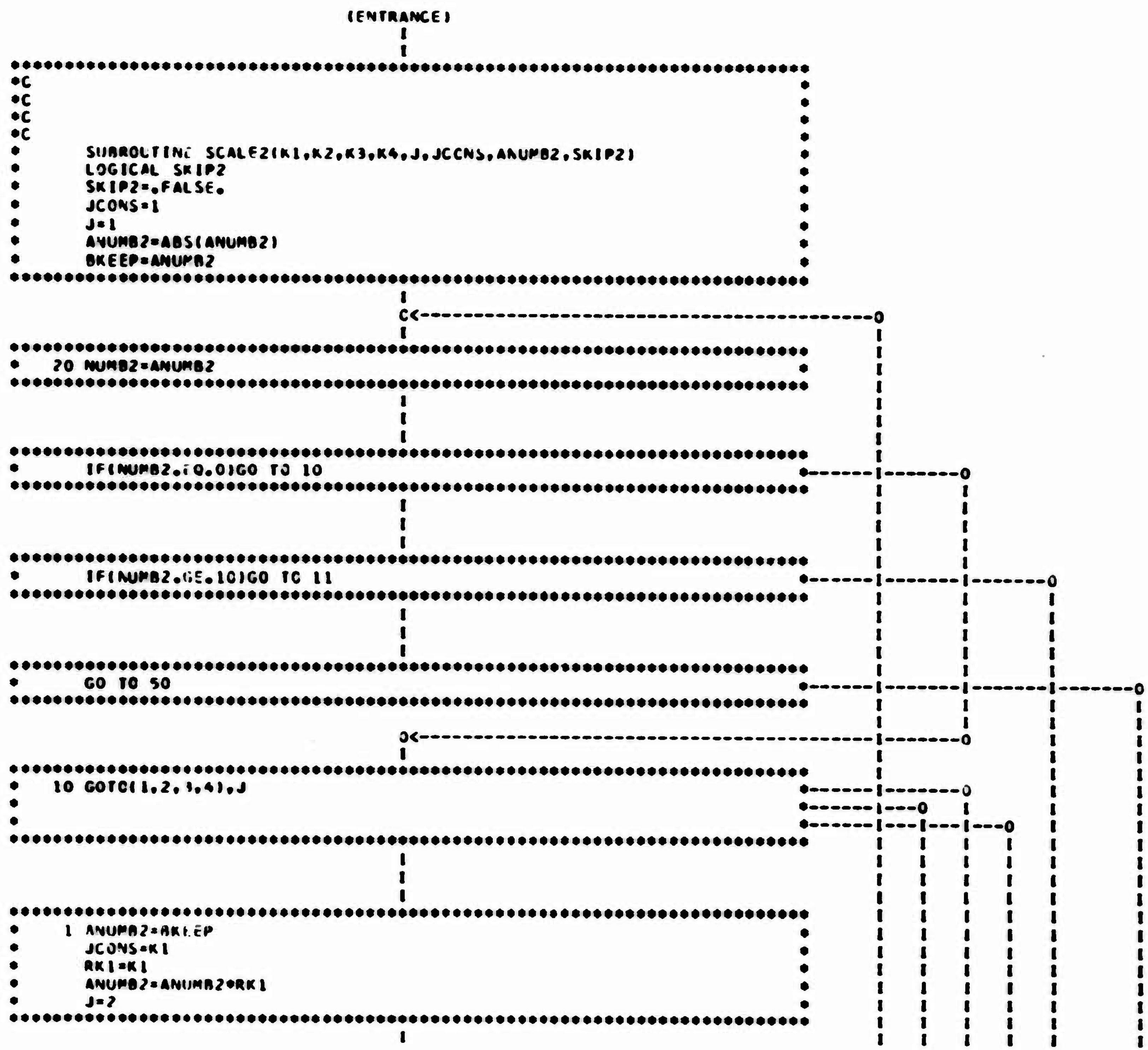
```
C
C
C
C
      SUBROUTINE SCALE1(K1,K2,K3,K4,NEGONE,I,ANUMB1,ICONS,LESS,SKIP1)
      LOGICAL LESS
      LOGICAL SKIP1
      SKIP1=.FALSE.
      ICONS=1
      I=1
      AKEEP=ANUMB1

   21 NUMB1=ANUMB1
      NUMB=IABS(NUMB1)

      IF(NUMB.EQ.0)GOTO 12

      IF(NUMB1)31,40,40

   40 IF(NUMB1.GE.10)GOTO 11

      GO TO 50

   31 LESS=.TRUE.
      XXX=-NUMB1
      NUMB1=XXX
```


GOTC 40
12 GOTO(1,2,3,4),I
1 ANUMB1=AKEEP
ICONS=K1
RK1=K1
ANUMB1=ANUMB1*RK1
I=2
GOTC 21
2 ANUMB1=AKEEP
ICONS=K2
RK2=K2
ANUMB1=ANUMB1*RK2
I=3
GOTO 21
3 ANUMB1=AKEEP
ICONS=K3
RK3=K3
ANUMB1=ANUMB1*RK3
I=4

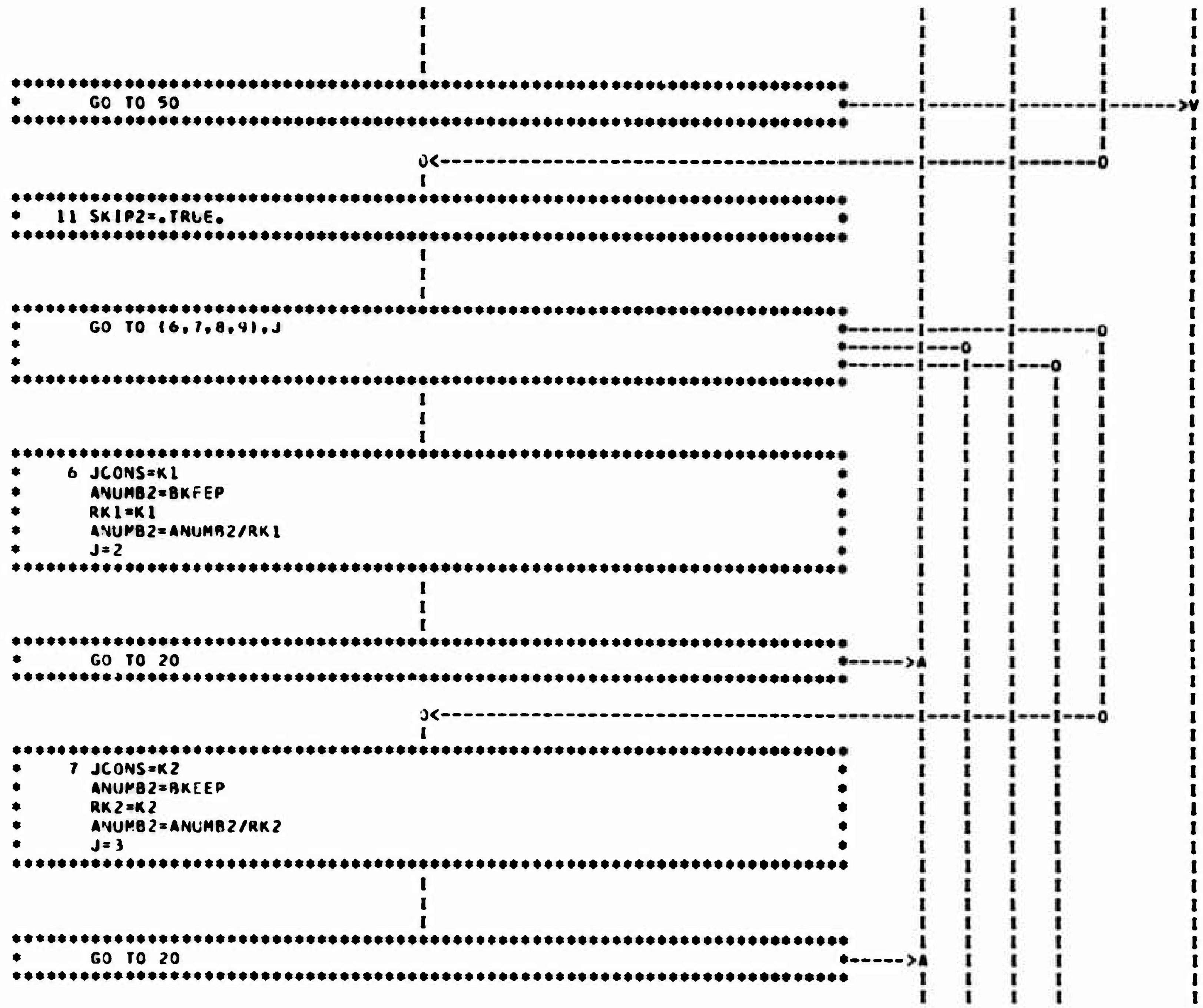
```
      GOTC 21

    4 ANUMB1=AKEEP
      ICONS=K4
      RK4=K4
      ANUMB1=ANUMB1*RK4
      NUMB1=ANUMB1

      IF(NUMB1.EQ.0) GO TO 51

      GO TO 50

   11 SKIP1=.TRUE.

      GO TO (6,7,8,9),I

    6 ANUMB1=AKEEP
      ICONS=K1
      RK1=K1
      ANUMB1=ANUMB1/RK1
      I=2
```

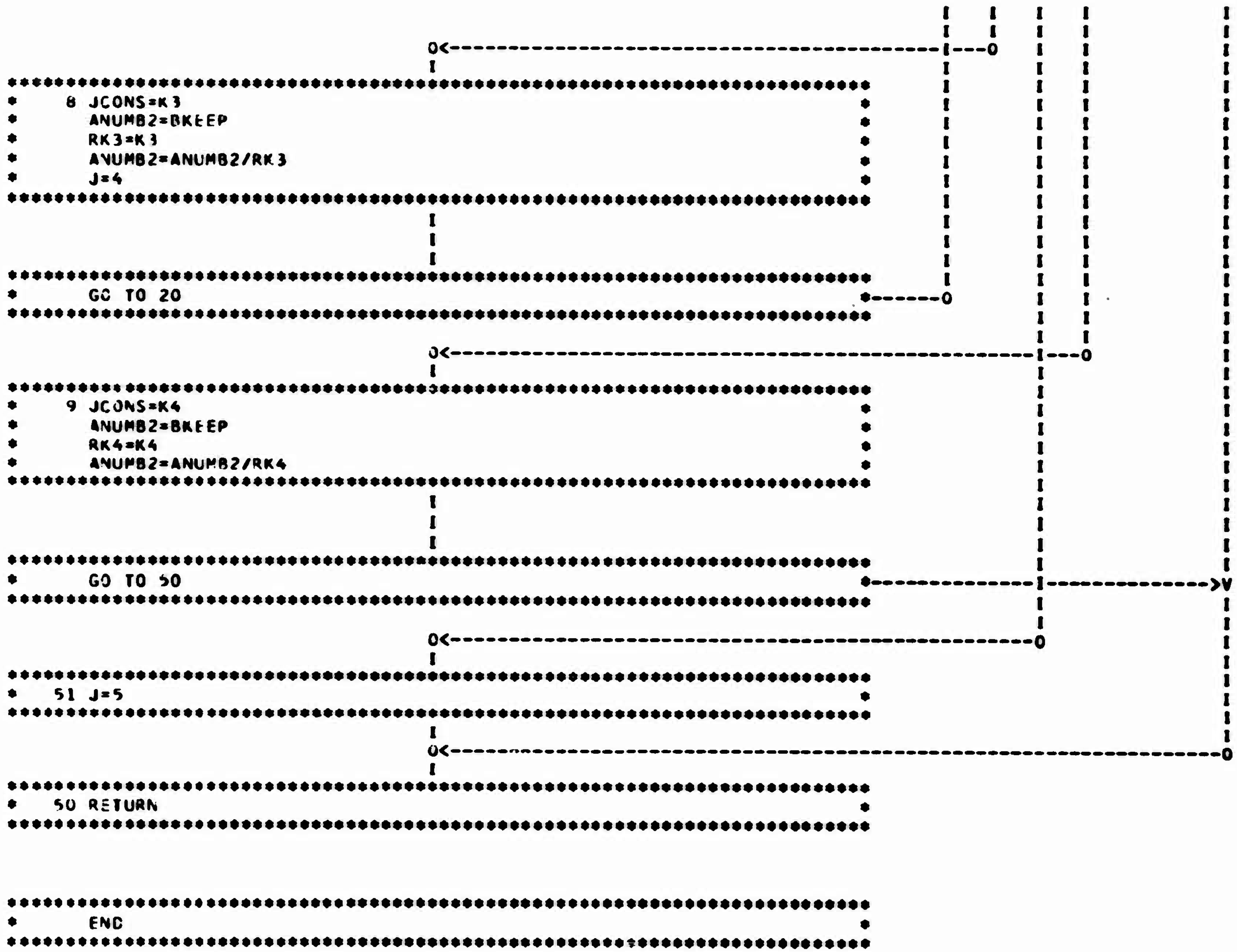
```
      GOTO 21

    7 ANUMB1=AKEEP
      ICONS=K2
      RK2=K2
      ANUMB1=ANUMB1/RK2
      I=3

      GOTO 21

    8 ANUMB1=AKEEP
      ICONS=K3
      RK3=K3
      ANUMB1=ANUMB1/RK3
      I=4

      GOTO 21

    9 ANUMB1=AKEEP
      ICONS=K4
      RK4=K4
      ANUMB1=ANUMB1/RK4

      GO TO 50
```

```
                                                                                 I   I
                                                                                 I   I
                                   0<--------------------------------------------0   I
                                   I                                                 I
***************************************************************************         I
*   51 I=5                                                                *         I
***************************************************************************         I
                                   I                                                 I
                                   0<------------------------------------------------0
                                   I
***************************************************************************
*   50 RETURN                                                             *
***************************************************************************


***************************************************************************
*      END                                                                *
***************************************************************************
```


(ENTRANCE)
C
C
C
C
SUBROUTINE SCALE2(K1,K2,K3,K4,J,JCONS,ANUMB2,SKIP2)
LOGICAL SKIP2
SKIP2=.FALSE.
JCONS=1
J=1
ANUMB2=ABS(ANUMB2)
BKEEP=ANUMB2
20 NUMB2=ANUMB2
IF(NUMB2.EQ.0)GO TO 10
IF(NUMB2.GE.10)GO TO 11
GO TO 50
10 GOTO(1,2,3,4),J
1 ANUMB2=BKEEP
JCONS=K1
RK1=K1
ANUMB2=ANUMB2*RK1
J=2

```
      GO TO 20

    2 ANUMB2=BKEEP
      JCONS=K2
      RK2=K2
      ANUMB2=ANUMB2*RK2
      J=3

      GO TO 20

    3 ANUMB2=BKFEP
      JCONS=K3
      RK3=K3
      ANUMB2=ANUMB2*RK3
      J=4

      GO TO 20

    4 ANUMB2=BK:EP
      JCONS=K4
      RK4=K4
      ANUMR2=ANUMR2*RK4
      NUMR2=ANUMR2

      IF(NUMR2.EQ.0) GO TC 51
```

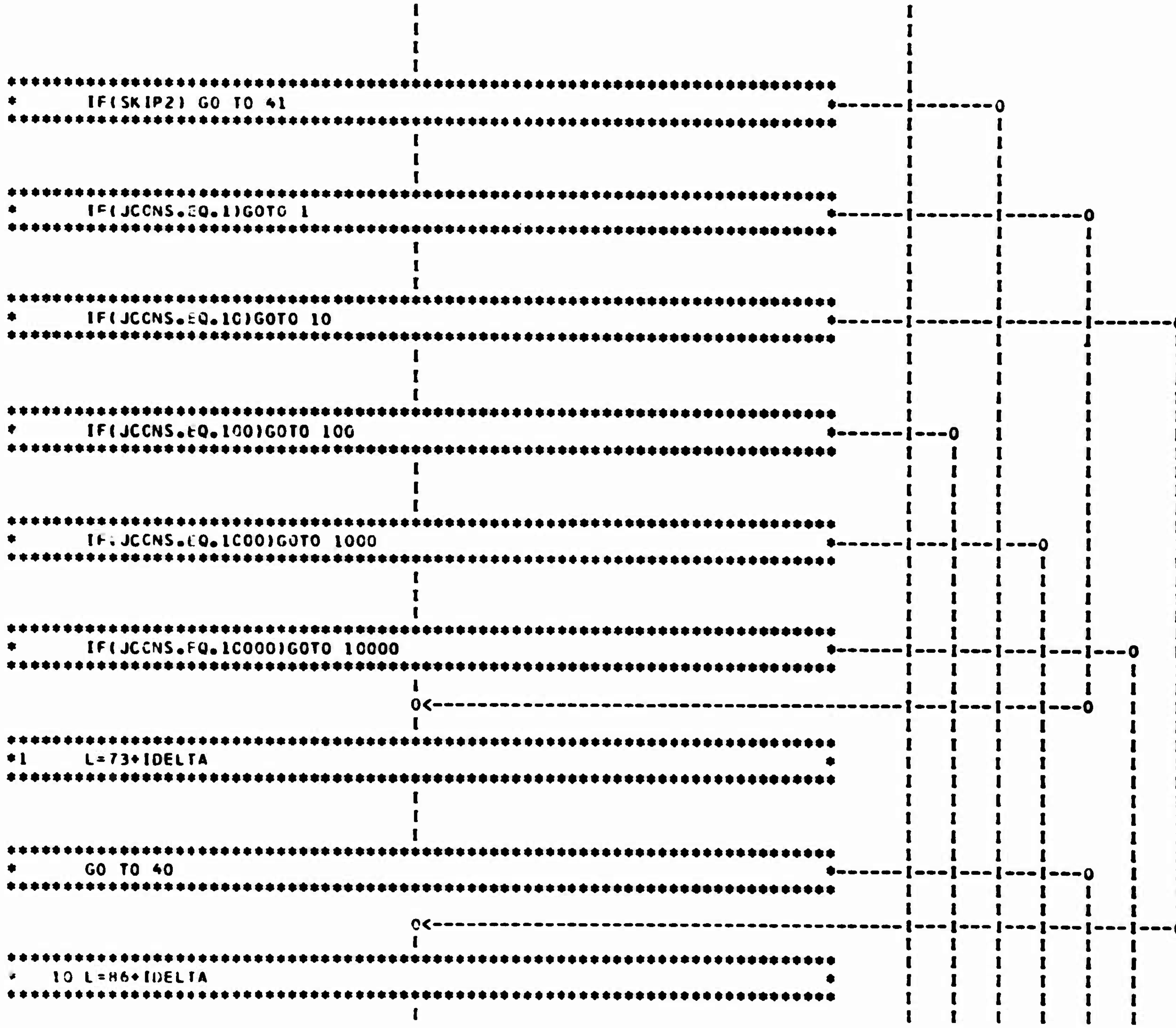

GO TO 50
11 SKIP2=.TRUE.
GO TO (6,7,8,9),J
6 JCONS=K1
ANUMB2=BKEEP
RK1=K1
ANUMB2=ANUMB2/RK1
J=2
GO TO 20
7 JCONS=K2
ANUMB2=BKEEP
RK2=K2
ANUMB2=ANUMB2/RK2
J=3
GO TO 20


8 JCONS=K3
ANUMB2=BKEEP
RK3=K3
ANUMB2=ANUMB2/RK3
J=4
GO TO 20
9 JCONS=K4
ANUMB2=BKEEP
RK4=K4
ANUMB2=ANUMB2/RK4
GO TO 50
51 J=5
50 RETURN
END

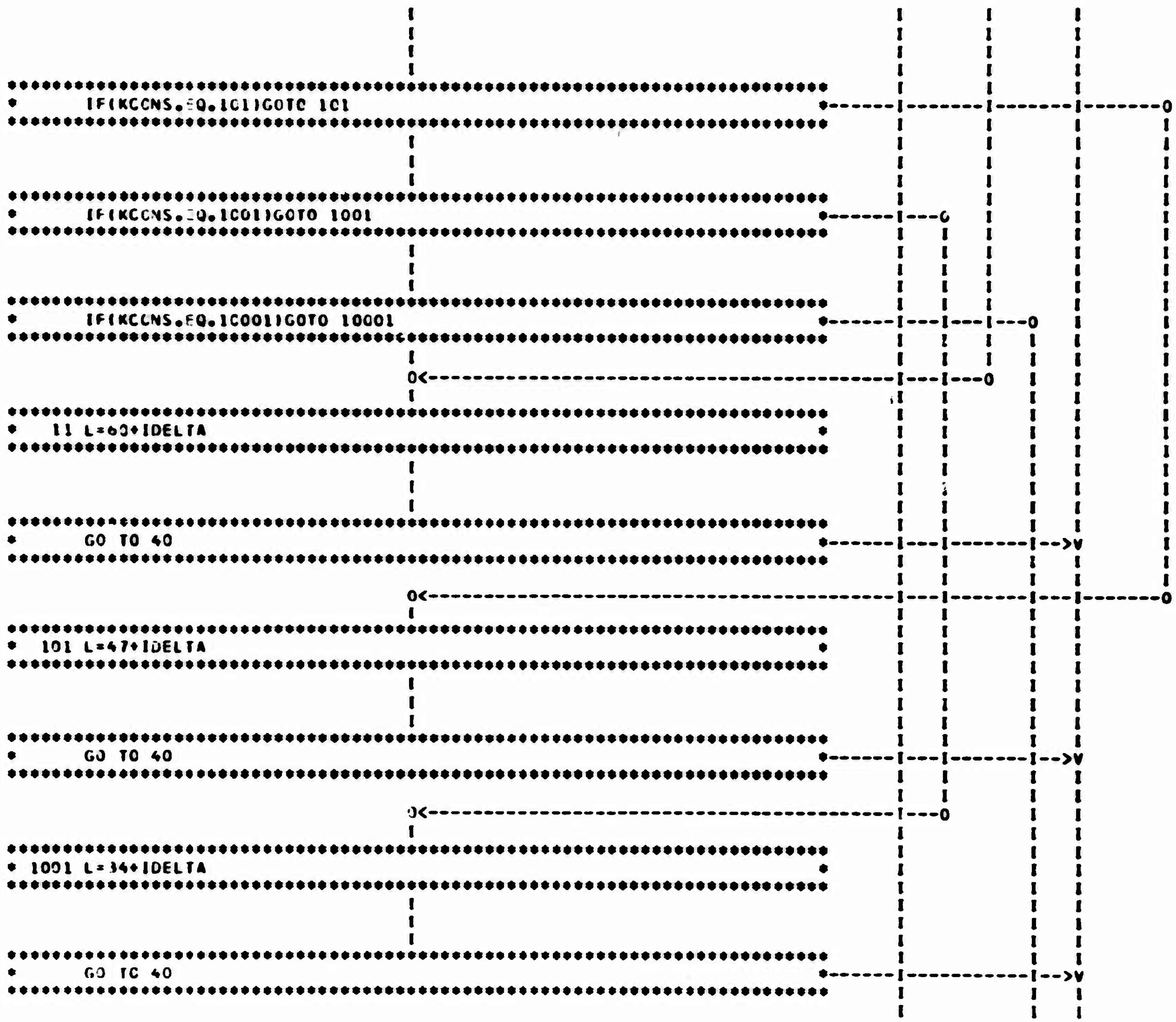
```
(ENTRANCE)

C
C
C
C
      SUBROUTINE WPOINT(J,JCONS,L,ANUMB2,SKIP2)
      LOGICAL SKIP2
      L=0
      IDELTA=0
      IF(ANUMB2.GE.9.2) IDELTA=12

      IF(ANUMB2.GE.8.0) GO TO 200

      IF(ANUMB2.GE.6.9) GO TO 201

      IF(ANUMB2.GE.5.9) GO TO 202

      IF(ANUMB2.GE.5.0) GO TO 203

      IF(ANUMB2.GE.4.2) GO TO 204

      IF(ANUMB2.GE.3.5) GO TO 205

      IF(ANUMB2.GE.2.9) GO TO 206
```

```
      IF(ANUMB2.GE.2.4) GO TO 207

      IF(ANUMB2.GE.1.9) GO TO 208

      IF(ANUMB2.GE.1.5) GC TO 209

      IF(ANUMB2.GE.1.2) GC TO 210

      GO TO 211

  200 IF(ANUMB2.LT.9.2) IDELTA=11

  201 IF(ANUMB2.LT.8.0) IDELTA=10

  202 IF(ANUMB2.LT.6.9) IDELTA=9

  203 IF(ANUMB2.LT.5.9) IDELTA=8
```

```
204 IF(ANUMB2.LT.5.0) IDFLTA=7

205 IF(ANUMB2.LT.4.2) IDELTA=6

206 IF(ANUMB2.LT.3.5) IDELTA=5

207 IF(ANUMB2.LT.2.9) IDELTA=4

208 IF(ANUMB2.LT.2.4) IDELTA=3

209 IF(ANUMB2.LT.1.9) IDELTA=2

210 IF(ANUMB2.LT.1.5) IDELTA=1

211 IF(ANUMB2.LT.1.2)IDELTA=0
    IDELTA=13-IDELTA

    IF(J.EQ.5) GO TO 50
```

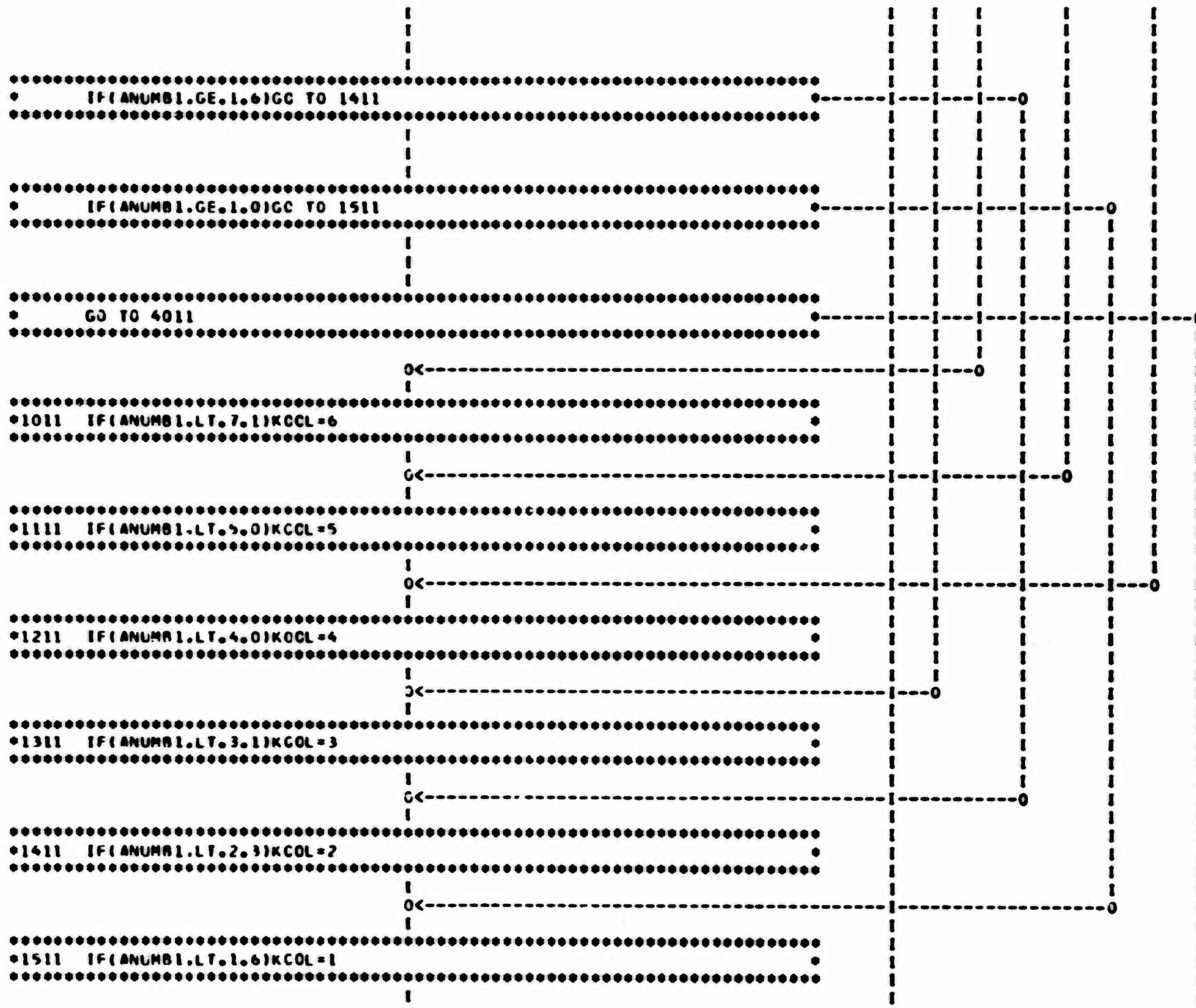

IF(SKIP2) GO TO 41
IF(JCCNS.EQ.1)GOTO 1
IF(JCCNS.EQ.10)GOTO 10
IF(JCCNS.EQ.100)GOTO 100
IF(JCCNS.EQ.1000)GOTO 1000
IF(JCCNS.EQ.10000)GOTO 10000
1 L=73+IDELTA
GO TO 40
10 L=H6+IDELTA


GO TO 40
100 L=99+IDELTA
GO TO 40
1000 L=112+IDELTA
GO TO 40
10000 L=125
GO TO 40
41 KCONS=JCONS+1
SKIP2=.FALSE.
IF(KCCNS.EQ.11)GOTO 11

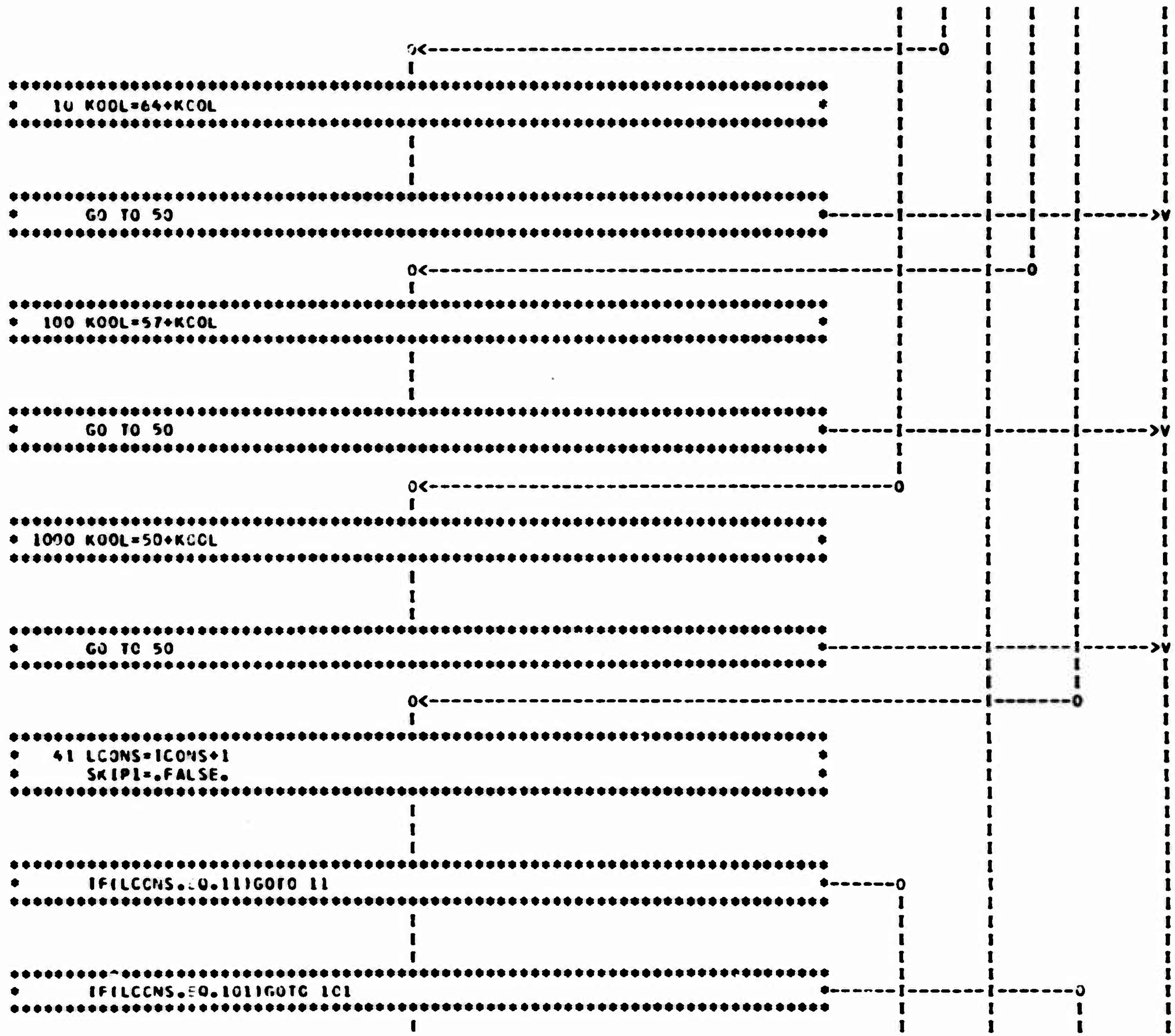

IF(KCONS.EQ.101)GOTO 101
IF(KCONS.EQ.1001)GOTO 1001
IF(KCONS.EQ.10001)GOTO 10001
11 L=60+IDELTA
GO TO 40
101 L=47+IDELTA
GO TO 40
1001 L=34+IDELTA
GO TO 40

```
*   50 L=125        *
*      GO TO 40     *
*10001 L=34         *
*   40 RETURN       *
*      END          *
```

```
                              (ENTRANCE)
                                  I
                                  I
*****************************************************************************
*C                                                                          *
*C                                                                          *
*C                                                                          *
*C                                                                          *
*      SUBROUTINE SPCINT(I,ANUMB1,LESS,ICONS,L,POINT,SKIP1,NO,XA,XB,STAR)*
*      DIMENSION XA(2C00),XB(20GC)                                          *
*      INTEGER PCINT(130,100),DASH,BLANK,STAR,C                             *
*      INTEGER STAR1,STAR2                                                  *
*      DATA STAR1/1HC/,STAR2/1H*/                                           *
*      DATA BLANK/1H /,DASH/1H-/,          C/1HI/                           *
*      LOGICAL SKIP1        ,LESS                                           *
*      AKEEP=ANUMB1                                                         *
*      ANUMB1=ABS(AKEEP)                                                    *
*      KOOL=C                                                               *
*****************************************************************************
                                  I
                                  I
                                  I
*****************************************************************************
*      IF(I.EQ.5)GO TO 100C                                                 *------O
*****************************************************************************      I
                                  I
                                  I
                                  I
*****************************************************************************
*      IF(ANUMB1.GE.7.1)KCCL=7                                              *
*****************************************************************************
                                  I
                                  I
                                  I
*****************************************************************************
*      IF(ANUMB1.GE.5.0)GO TO 1011                                          *------I-------O
*****************************************************************************
                                  I
                                  I
                                  I
*****************************************************************************
*      IF(ANUMB1.GE.4.0)GO TO 1111                                          *------I-------I-------O
*****************************************************************************
                                  I
                                  I
                                  I
*****************************************************************************
*      IF(ANUMB1.GE.3.1)GO TO 1211                                          *------I-------I-------I-------O
*****************************************************************************
                                  I
                                  I
                                  I
*****************************************************************************
*      IF(ANUMB1.GE.2.3)GO TO 1311                                          *------I---O
*****************************************************************************
                                  I
```


IF(ANUMB1.GE.1.6)GC TO 1411
IF(ANUMB1.GE.1.0)GC TO 1511
GO TO 4011
1011 IF(ANUMB1.LT.7.1)KCCL=6
1111 IF(ANUMB1.LT.5.0)KCCL=5
1211 IF(ANUMB1.LT.4.0)KOCL=4
1311 IF(ANUMB1.LT.3.1)KCOL=3
1411 IF(ANUMB1.LT.2.3)KCOL=2
1511 IF(ANUMB1.LT.1.6)KCOL=1

```
4011 NCRMNT=8-KOOL
     IF(LESS)GC TO 40
     IF(SKIP1)GO TC 41
     IF(ICCNS.EQ.1)GOTO 1
     IF(ICCNS.EQ.1C)GOTO 10
     IF(ICCNS.EQ.1C0)GOTC 100
     IF(ICCNS.EQ.1C00)GOTO 1000
   1 KOOL=71+KOOL
     GO TC 50
```

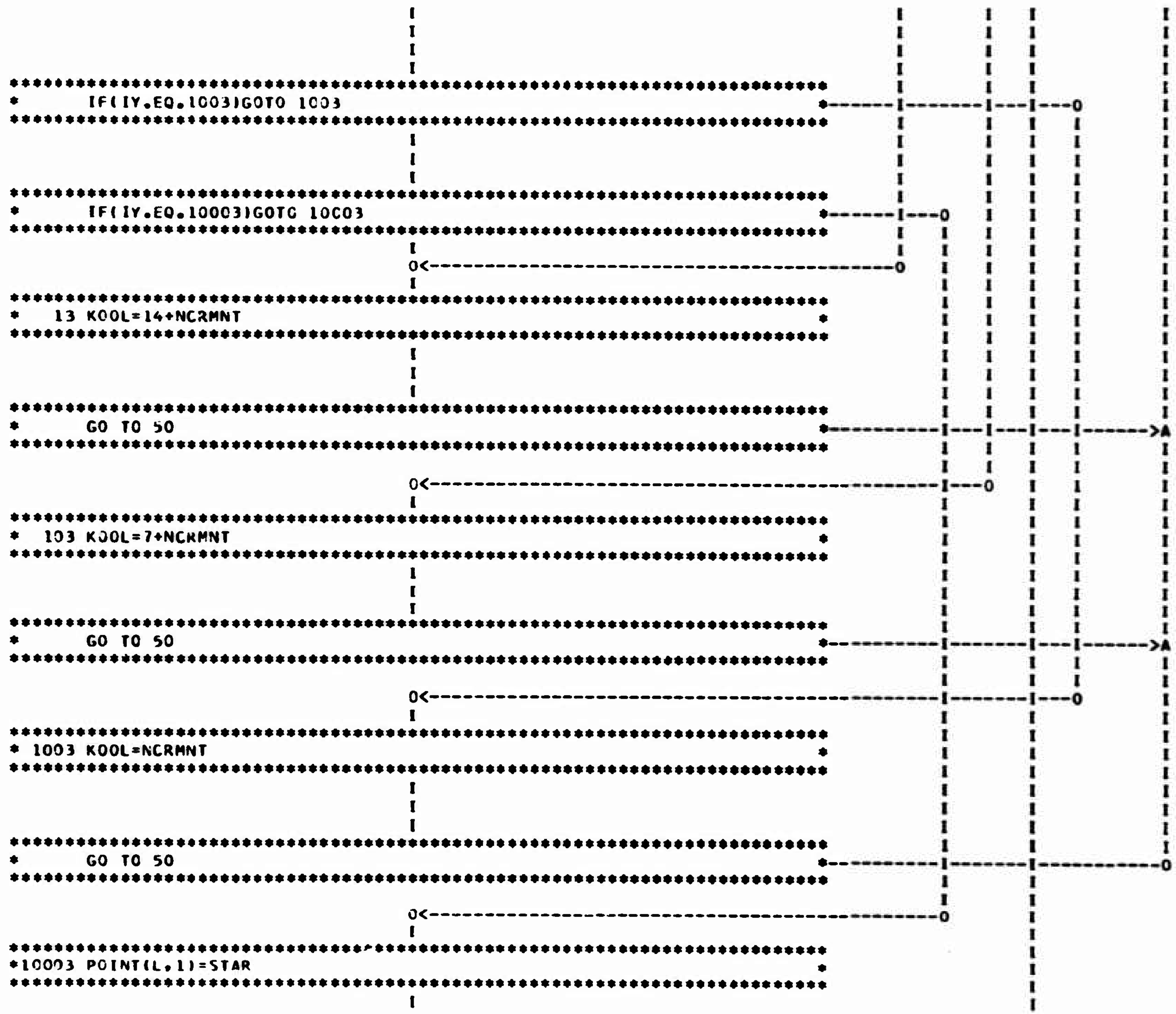

10 KOOL=64+KCOL
GO TO 50
100 KOOL=57+KCOL
GO TO 50
1000 KOOL=50+KCOL
GO TO 50
41 LCONS=ICONS+1
SKIP1=.FALSE.
IF(LCONS.EQ.11)GOTO 11
IF(LCONS.EQ.101)GOTO 101

```
      IF(LCONS.EQ.1001)GOTO 1001

   11 KOOL=78+KCOL

      GO TO 50

  101 KOOL=85+KCOL

      GO TO 50

 1001 KOOL=92+KCOL

   50 LESS=.FALSE.

      IF(PCINT(L,KCOL).EQ.DASH) GO TO 51

      IF(PCINT(L,KCOL).EQ.O    ) GO TO 51
```

```
14 IF(POINT(L,KCOL).EQ.STAR1) GO TO 15

   IF(POINT(L,KCOL).EQ.STAR2) GO TO 15

   IF(STAR.EQ.STAR1) GO TO 51

   IF(STAR.EQ.STAR2) GO TO 51

   IF(POINT(L,KCOL).NE.BLANK) GO TO 70

51 POINT(L,KCOL)=STAR
   ANUMBI=AKEEP

   GO TO 70

15 L=L+1

   IF(L.EQ.130) GO TO 51
```


GO TO 14
40 IF(SKIP1) GO TO 42
IZ=ICONS+2
IF(IZ.EQ.3)GOTO 3
IF(IZ.EQ.12)GOTO 12
IF(IZ.EQ.102)GOTO 102
IF(IZ.EQ.1002)GOTO 1002
3 KOOL=21+NCRMNT
GO TO 50
A

```
   12 KOOL=28+NCRMNT

      GO TO 50

  102 KOOL=35+NCRMNT

      GO TO 50

 1002 KOOL=42+NCRMNT

      GO TO 50

   42 IY=ICONS+3
      SKIP1=.FALSE.

      IF(IY.EQ.13)GOTO 13

      IF(IY.EQ.103)GOTO 103
```


IF(IY.EQ.1003)GOTO 1003
IF(IY.EQ.10003)GOTG 10C03
13 KOOL=14+NCRMNT
GO TO 50
103 KOOL=7+NCRMNT
GO TO 50
1003 KOOL=NCRMNT
GO TO 50
10003 POINT(L,1)=STAR

```
                                             I                                  I
                                             I                                  I
                                             I                                  I
                                             I                                  I
**************************************************************************      I
*      GO TO 70                                                          *      I
**************************************************************************      I
                                             I                                  I
                                             O<---------------------------------O
                                             I
**************************************************************************
*   70 RETURN                                                            *
**************************************************************************


**************************************************************************
*      END                                                               *
**************************************************************************
```

```
      IF(AXX.LT.10.0)GO TO 23
      AXX=AXX/10.0
      GO TO 30
   23 AXX=TENS(AXX)
      GO TO 30
   22 DO 24 I=1,19
   24 IF(AXX.GE.DIMS(I).AND.AXX.LT.DIMS(I+1)) GO TO 25
   25 LIMIT2=I
      LIMITW=LIMITX+LIMIT2
      RETURN
```

END

```
(ENTRANCE)
    |
C
C
      SUBROUTINE BLANKR (POINT,N1,N2,LIM1)
      INTEGER POINT(N1,N2),BLANK,DASH,Q
      COMMON /INFO4/ Q,BLANK,DASH
      DATA BLANK/1H /,DASH/1H-/,Q/1H1/
    |
      DO 51 M=1,N1
    |
      DO 51 N=1,N2
    |
   51 POINT(M,N)=BLANK
    |
      DO 52 M=1,N1
    |
      POINT(M,1)=DASH
    |
   52 POINT(M,N2)=DASH
    |
      DO 53 M=1,N2
    |
```

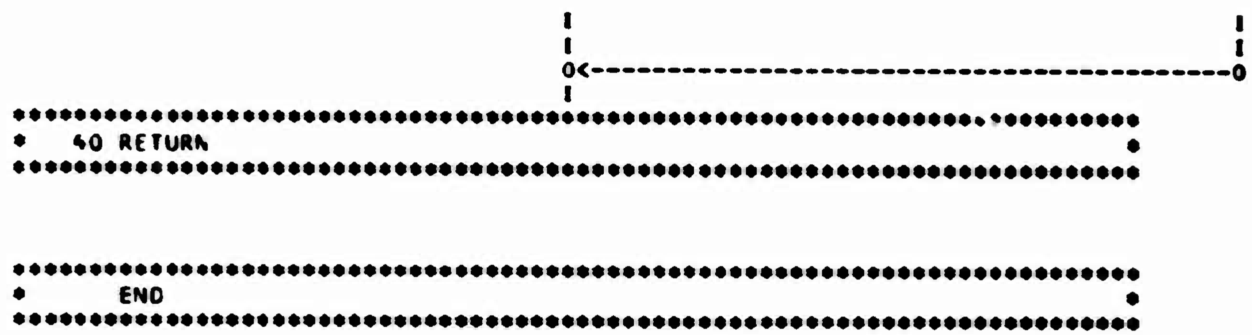
```
      POINT(1,M)=0
   53 POINT(LIM1,M)=0
      RETURN
      END
```

```
(ENTRANCE)

C
C
      SUBROUTINE NEWPLT(X,Y,Z,LET)
      INTEGER POINT(130,100),POINTF(120,58),STAR,Q
      DIMENSION ALIM(7)
      DIMENSION Y(3000),Z(3000)
      DIMENSION SAVE1(100,100),SAVE2(100,100)
      COMMON /INFO4/ Q,BLANK,DASH
      COMMON/INFO2/ALIM
      COMMON SAVE1,SAVE2,POINT
      EQUIVALENCE(POINT(1,1),POINTF(1,1))
      DATA STAR/1H*/,Q/1HI/
    1 FORMAT(1H1)
      CALL BLANKR(POINTF,120,58,120)

      DO 10 I=1,3000

      IF(Y(I).EQ.0.0) GO TO 11

      GO TO 12

   11 IF(Z(I).EQ.0.0) GO TO 10

   12 CALL LOGGER(Y(I),LIMITW)

      IF(LIMITW.EQ.0) GO TO 10
```

```
      CALL LINEAR(X,LIMITY,Z(I))

      IF(LIMITY.EQ.0) GO TO 10

      POINTF(LIMITW,LIMITY)=STAR

   10 CONTINUE

      DO 40 I=20,100,20

      DO 40 J=1,58

   40 POINTF(I,J)=Q

      WRITE(6,1)
   30 FORMAT(1X,6(1P1E10.1,1X,3HR/S,6X),1P1E10.1)
      J=X
      WRITE(6,51)J,LET,(POINTF(I,1),I=1,120)
   51 FORMAT(1X,I4, A4,1X,120A1)
      WRITE(6,20)((POINTF(I,J),I=1,120),J=2,57)
      J=-X
      WRITE(6,51)J,LET,(POINTF(I,58),I=1,120)
   20 FORMAT(10X,120A1)
      WRITE(6,30)ALIM
```

```
            |
*****************************************************
*     RETURN                                        *
*****************************************************

*****************************************************
*     END                                           *
*****************************************************
```

```
(ENTRANCE)
    I
C
C
      SUBROUTINE LINEAR(X,LIMITY,Y)
      DIMENSION ENCRMT(59)
      LIMITY=0
      FLIMIT=X*2.
      DELTA=FLIMIT/58.
      ENCRMT(1)=X
    I
      DO 10 I=2,58
    I
   10 ENCRMT(I)=ENCRMT(I-1)-DELTA
    I
      ENCRMT(59)=-X
    I
      DO 11 I=1,58
    I
   11 IF(Y.LE.ENCRMT(I).AND.Y.GE.ENCRMT(I+1))GO TO 20
    I
      GO TO 40 --------0
                       I
   20 LIMITY=I
    I
```


40 RETURN
END

```
                                        (ENTRANCE)
                                             I
                                             I
          *************************************************************************
          *C                                                                      *
          *C                                                                      *
          *      SUBROUTINE SWEEPR(XOMEGA)                                        *
          *      DIMENSION OMEGA(17),SWEPER(17)                                   *
          *      COMMON/INFO3/OMEGA                                               *
          *      DATA (SWEPER(I),I=1,17) /0.01,0.05,0.1,0.2,0.3,0.4,0.5,0.6,0.7,  *
          *     *0.8,0.9,1.0,2.0,4.0,6.0,8.0,10.0/                                *
          *************************************************************************
                                             I
                                             I
                                             I
          *************************************************************************
I---------*      DO 70 K=1,17                                                     *
I         *************************************************************************
I                                            I
I                                            I
I                                            I
I         *************************************************************************
----------*   70 OMEGA(K)= SWEPER(K)*XOMEGA                                       *
          *************************************************************************
                                             I
                                             I
                                             I
          *************************************************************************
          *      RETURN                                                           *
          *************************************************************************

          *************************************************************************
          *      END                                                              *
          *************************************************************************
```

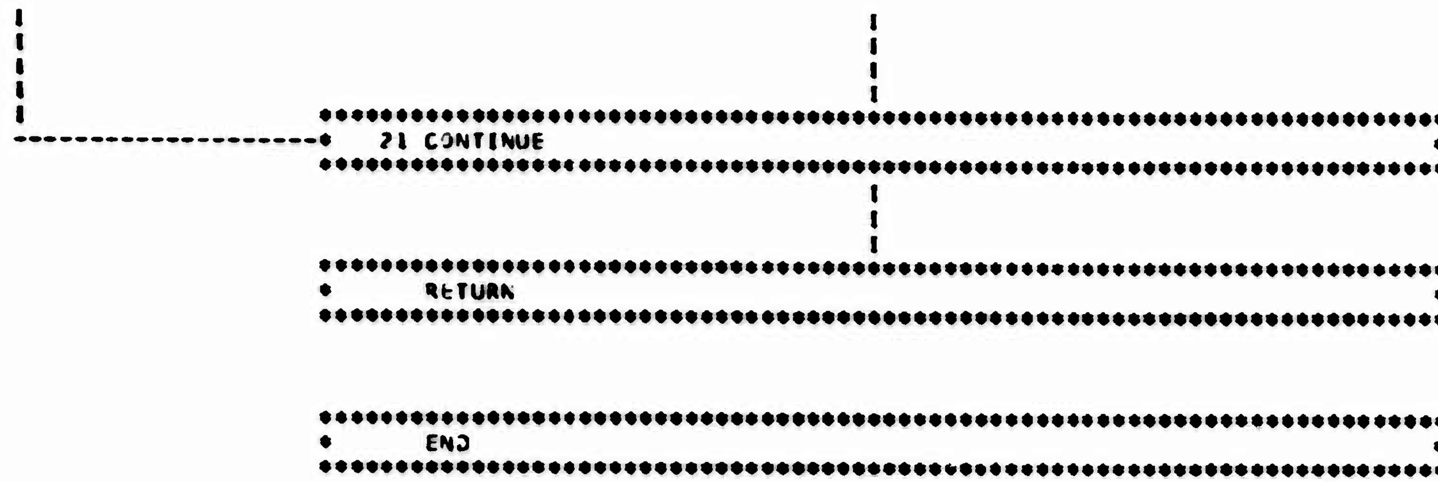
```
   11 CONTINUE

      GO TO 40

   80 IF(K.GE.ICOUNT) GO TO 40

      K=K+1

      GO TO 41

   40 DO 68 I=1,3000

      IF(WLCG(I).EQ.0.0.AND.PHILIN(I).EQ.0.0.AND.PHIPLN(I).EQ.0.0.AND.
     1MDBLIN(I).EQ.0.0.AND.MPDBLN(I).EQ.0.0.AND.BIGSEM(I).EQ.0.0.AND.
     2EMSUBS(I).EQ.0.0) GO TO 69

      JJ=MOD(I     ,57)
      IF(JJ.EQ.0) WRITE(6,51)

   68 WRITE(6,50) WLOG (I),BIGSEM(I),MDBLIN(I),PHILIN(I),EMSUBS(I),
     1MPDBLN(I),PHIPLN(I)
```

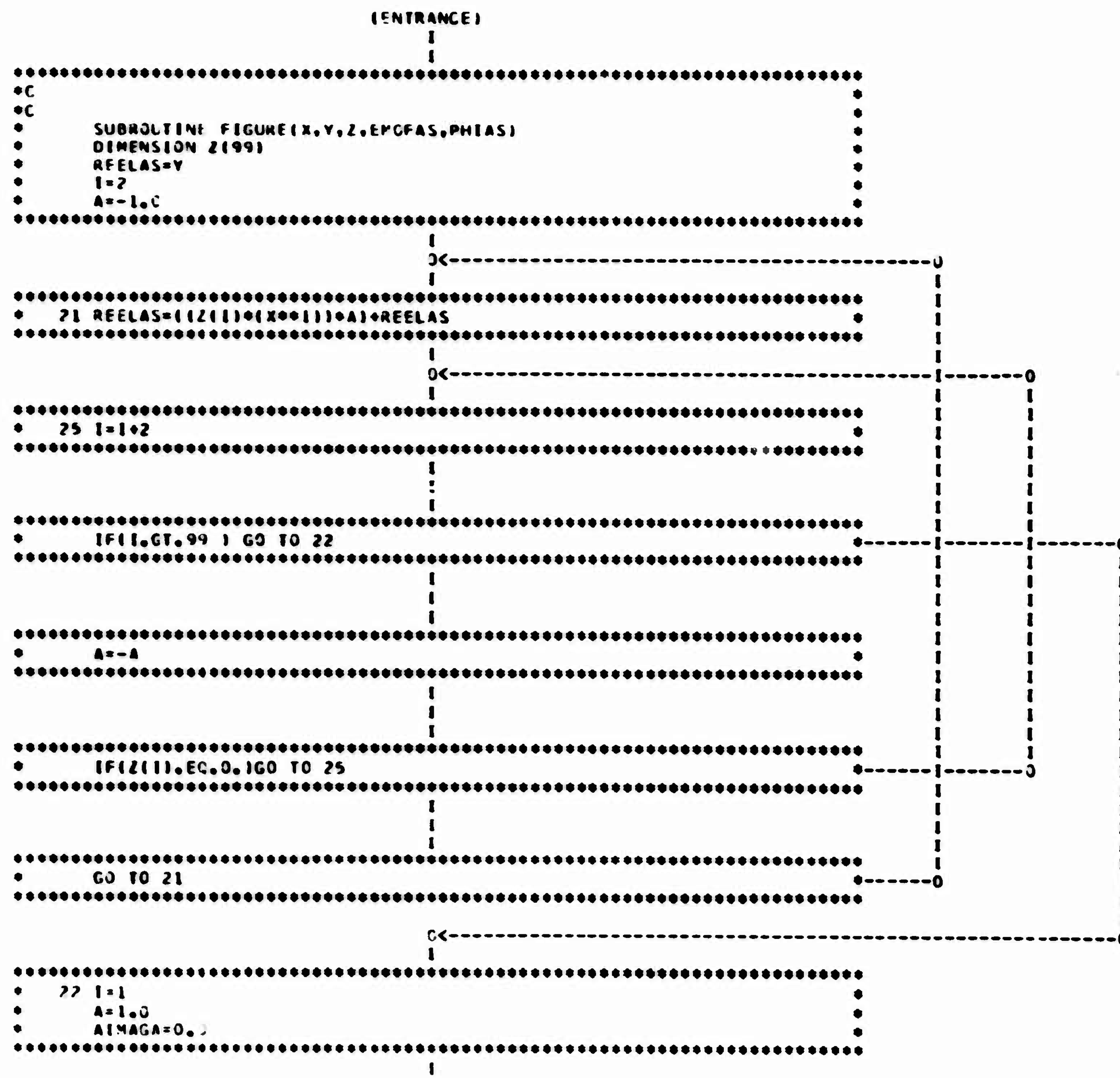
```
   69 WRITE(6,14)
      CALL NEWPLT(DEGREE,WLOG,PHILIN,LET2 )
      IF(IEXP(1).EQ.0) GC TO 30
      JJ=IEXP(1)
      DO 34 I=1,JJ
   34 CALL EXPLOT(WLOG,3000,PHILIN,3000,DEGREE,LET2)
   30 CALL NEWPLT(DBLIM,WLOG,MDBLIN,LET1)
      IF(IEXP(2).EQ.0) GC TO 31
      JJ=IEXP(2)
      DO 35 I=1,JJ
```

```
35 CALL EXPLOT(WLOG,3000,MDRLIN,3000,DRLIM,LET1)

31 WRITE(6,15)
   CALL NEWPLT(DEGREE,WLOG,PHIPLN,LET2)

   IF(IEXP(3).EQ.0) GO TO 32

   JJ=IEXP(3)

   DO 36 I=1,JJ

36 CALL EXPLOT(WLOG,3000,PHIPLN,3000,DEGREE,LET2)

32 CALL NEWPLT(DRLIM,WLOG,MPDBLN,LET1)
   JJ=IEXP(4)

   DO 37 I=1,JJ
```

```
      37 CALL EXPLOT(WLOG,3000,MPURLN,3000,DBLIM,LET1)

         IF(IEXP(4).EQ.0) GO TO 33

      51 FORMAT(1H1,5X,5HOMEGA,13X,3H/M/,16X,6H/M/-DB,10X,3HPHI,20X,4H/M/P,
        *10X,7H/M/P-DB,14X,5HPHI-P,/)
      50 FORMAT(2X,1P1E12.5,5X,1P1E12.5,6X,1P1E12.5,5X,1P1E12.5,11X,
        *1P1E12.5,3X,1P1E12.5,9X,1P1E12.5)
      14 FORMAT(1H1,////////////////////,25X,60HTHE FOLLOWING PLOTS ARE NOR
        *MAL FREQUENCY RESPONSE DATA                                      )
      15 FORMAT(1H1,////////////////////,25X,36HTHE FOLLOWING PLOTS ARE POP
        *OV CATA                                 )

      33 RETURN

         END
```

```
(ENTRANCE)
     I
     I
C
C
      SUBROUTINE FIGURR (KONT,OMEGA,KOUNT)
      DIMENSION SAVE1(100,100),SAVE2(100,100)
      DIMENSION ACOEFF(100),BCOEFF(100),AAAAAA(99),BBBBBB(99)
      DIMENSION WLOG(3000),PHILIN(3000),MDBLIN(3000),PHIPLN(3000),
     *MPDBLN(3000),BIGSEM(3000),EMSUBS(3000),OMEGA(KONT)
      REAL MDBLIN,MPDBLN
      INTEGER POINT(130,100)
      COMMON/FREEK/ACOEFF,BCOEFF
      EQUIVALENCE(AAAAAA(1),ACOEFF(2)),(BBBBBB(1),BCOEFF(2))
      COMMON/INFO6/EMSUBS
      COMMON SAVE1,SAVE2,POINT
      EQUIVALENCE (WLOG(1),SAVE1(1)),(PHILIN(1),SAVE2(1)),
     1            (SAVE1(3001),MDBLIN(1)),(SAVE2(3001),PHIPLN(1)),
     2            (SAVE1(6001),MPDBLN(1)),(SAVE2(6001),BIGSEM(1))
     I
     I
      DO 21 I=1,KONT
     I
     I
      CALL FIGURE(OMEGA(I),ACOEFF(1),AAAAAA,EMOFAS,PHIAS)
      CALL FIGURE(OMEGA(I),BCOEFF(1),BBBBBB,EMOFBS,PHIBS)
      BIGEMM=ABS(EMOFAS/EMOFBS    )
      BIGSEM(KOUNT)=BIGEMM
      WLOG(KOUNT)=OMEGA(I)
      PHIEND=PHIAS-PHIBS
      IF(PHIEND.LT.-180.) PHIEND=PHIEND+360.
      IF(PHIEND.GT. 180.) PHIEND=PHIEND-360.
      PHILIN(KOUNT)=PHIEND
      PHIEND = PHIEND * 0.0174533
      REEL = BIGEMM * COS ( PHIEND )
      EIMAG= BIGEMM * SIN ( PHIEND )
      EMSUBP = SQRT ( ( REEL** 2 ) + ( ( OMEGA ( I ) * EIMAG ) ** 2 ) )
      EMSUBS(KOUNT)=EMSUBP
      QUANZ=OMEGA(I) * EIMAG
      PHIPEE=ATAN2(QUANZ,REEL)
      PHIPEE = PHIPEE * 57.2957795131
      IF(PHIPEE . LT . 0.) PHIPEE = PHIPEE + 360.
      IF(PHIPEE.LT.-180.) PHIPEE=PHIPEE+360.
      IF(PHIPEE.GT. 180.) PHIPEE=PHIPEE-360.
      PHIPLN(KOUNT)=PHIPEE
      DEEBEE=20.0*(ALOG10(BIGEMM))
      MDBLIN(KOUNT)=DEEBEE
      DEEBEP=20.0*(ALOG10(EMSUBP))
      MPDBLN(KOUNT)=DEEBEP
      KOUNT=KOUNT+1
     I
```


21 CONTINUE
RETURN
END


(ENTRANCE)
C
C
SUBROUTINE FIGURE(X,Y,Z,EPOFAS,PHIAS)
DIMENSION Z(99)
REELAS=Y
I=2
A=-1.0
21 REELAS=((Z(I)*(X**I))*A)+REELAS
25 I=I+2
IF(I.GT.99) GO TO 22
A=-A
IF(Z(I).EQ.0.)GO TO 25
GO TO 21
22 I=1
A=1.0
AIMAGA=0.0

```
24 AIMAGA=(Z(I)*(X**I)*A)+AIMAGA

26 I=I+2

   IF(I.GT.99 ) GO TO 23

   A=-A

   IF(Z(I).EQ.0.)GO TO 26

   GO TO 24

23 EMOFAS=SQRT((REELAS**2)+(AIMAGA**2))
   PHIAS=ATAN2(AIMAGA,REELAS)
   PHIAS = PHIAS*57.2957795131
   IF(PHIAS.LT.0.) PHIAS=PHIAS+360.

   RETURN
```

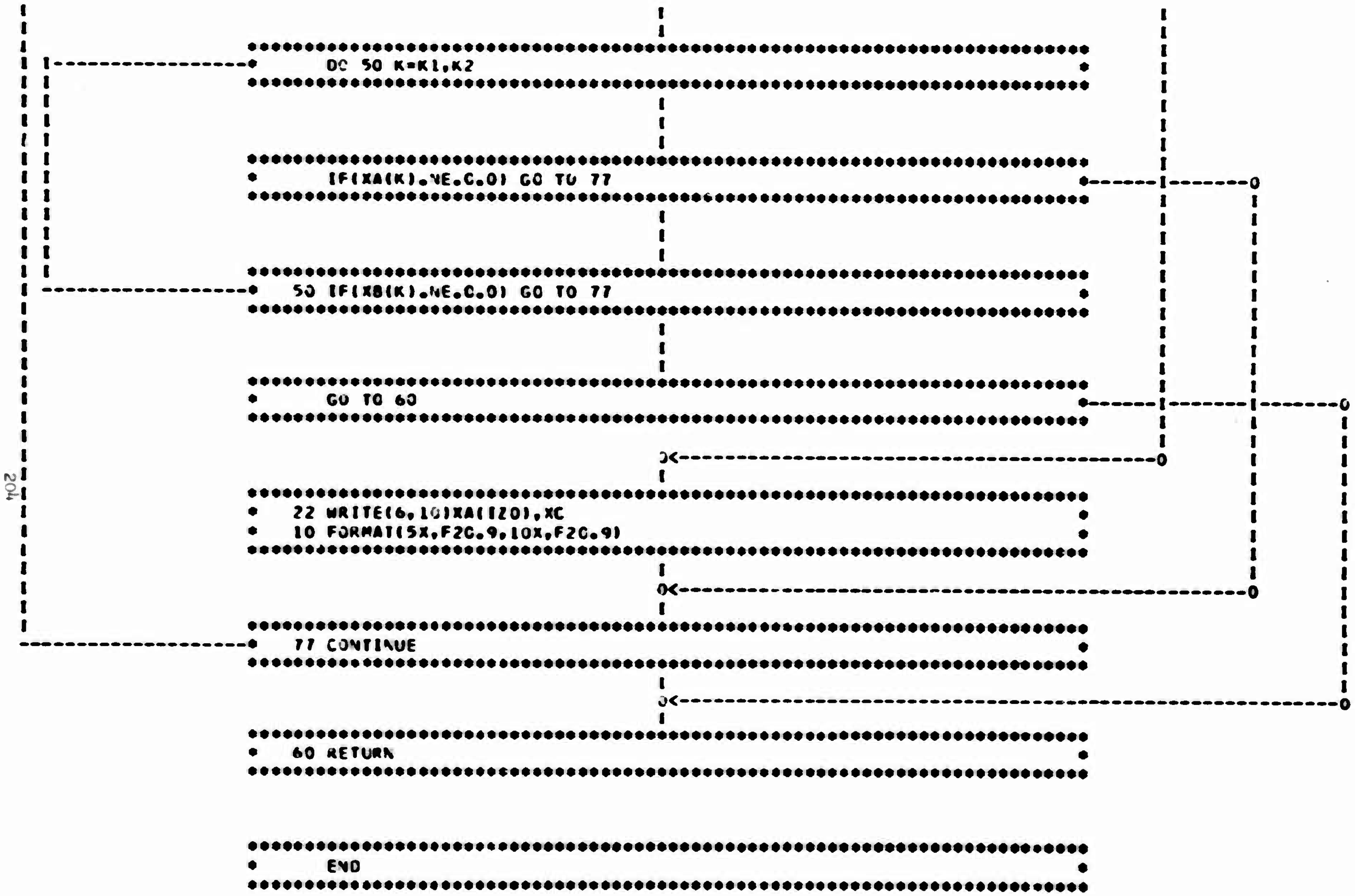
END

```
(ENTRANCE)

C
C
      SUBROUTINE LOGGER(XX,LIMITW)
      DIMENSION  ALIM(7),DIMS(20)
      EQUIVALENCE(WASTE,AXX)
      COMMON/INFO2/ALIM
      TENS(X)=X*10.0
      DATA (DIMS(I),I=1,20) /1.0,1.15,1.3,1.5,1.7,1.95,2.25,2.5,2.85,
     *3.15,3.5,3.95,4.5,4.85,5.4,6.05,6.8,7.7,8.8,9.999999/
      LIMITW=0
      LIMITX=0
      LIMIT2=0

      DO 20 I=1,6

      AA=2*(I-1)

   20 IF(XX.LT.ALIM(I+1).AND.XX.GE.ALIM(I)) GO TO 21

      RETURN

   21 LIMITX=TENS(AA)
      WASTE=ABS(XX)
      IF(WASTE.EQ.0.0) RETURN

   30 IF(AXX.GE.1.0.AND.AXX.LT.10.0) GO TO 22
```

```
                         (ENTRANCE)
                              I
                              I
*********************************************************************
*C                                                                  *
*C                                                                  *
*C                                                                  *
*C                                                                  *
*      SUBROUTINE PREPAR(POINT)                                     *
*      INTEGER POINT(130,100),PRT                                   *
*      DATA PRT /6/                                                 *
*      WRITE(6,1)                                                   *
*    1 FORMAT(1H1)                                                  *
*      WRITE(6,11)                                                  *
*   11 FORMAT(///)                                                  *
*      WRITE(PRT,14)                                                *
*   14 FORMAT(75X,8HLOG PLOT,/58X,42HCOMPLEX FREQUENCY PLANE,LEFT HAND QU*
*     *ADRANT,/75X,9H(RAD/SEC)  )                                   *
*      WRITE(6,12)                                                  *
*   12 FORMAT(32X,5H10000,9X,4H1000,9X,3H100,10X,2H10,12X,1H1,11X,2H.1,10*
*     *X,14H.01<---J-OMEGA,/ 18X,11HMINUS SIGMA)                    *
*   13 FORMAT(32X,5H10000,9X,4H1000,9X,3H100,10X,2H10,12X,1H1,11X,2H.1,10*
*     *X,14H.01<---J-OMEGA,/ 18X,11H PLUS SIGMA)                    *
*********************************************************************
                              I
                              I
                              I
*********************************************************************
I-------------------*      DO 50 I=1,130                            *
I                   *********************************************************************
I                             I
I                             I
I                             I
I                   *********************************************************************
--------------------*   50 POINT(I,50)=POINT(I,51)                  *
                    *********************************************************************
                              I
                              I
                              I
*********************************************************************
*      WRITE(PRT,10)POINT                                           *
*   10 FORMAT(1X,130A1)                                             *
*      WRITE(6,11)                                                  *
*      WRITE(PRT,15)                                                *
*   15 FORMAT(75X,8HLOG PLOT,/58X,43HCOMPLEX FREQUENCY PLANE,RIGHT HAND Q*
*     *UADRANT,/75X,9H(RAD/SEC)  )                                  *
*********************************************************************
                              I
                              I
                              I
*********************************************************************
*      RETURN                                                       *
*********************************************************************
```

END

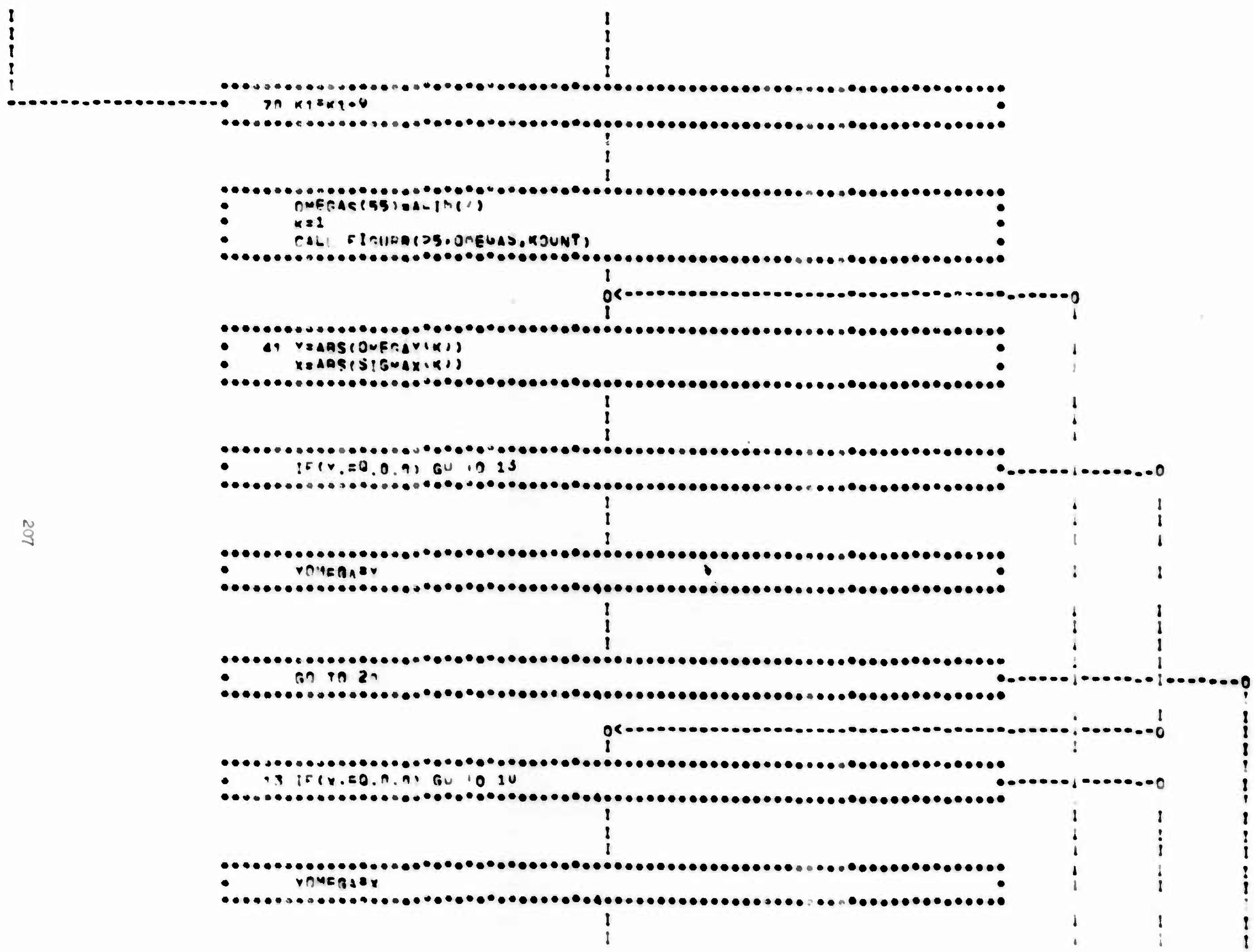
```
(ENTRANCE)

C
C
C
C
C
      SUBROUTINE WRITIT(XA,XB)
      DIMENSION XA(2C00),XR(2000)
      WRITE(6,1)
    1         FORMAT(1H1)
      WRITE(6,11)
   11 FORMAT(5X,48HTHE FOLLOWING  ROOTS ARE PLCTTED ON THE LOG PLOT,/
     15X,99HROOTS AT THE CRIGIN ARE NOT PRINTED CR PLOTTED, ROOTS ON THE
     * J-OMEGA AXIS ARE NOT PLOTTED.                                     ,
     2,//16X,5HSIGMA,25X,7HJ-OMEGA,//)

      DO 77 IZO=1,2C00

      XC=XB(IZC)
      IF(XC.GT.0.0) XC=-XC

      IF(XC.NE.0.0) GO TO 22

   20 IF(XA(IZO))22,40,22

   40 K1=IZC
      K2=IZC+12
```

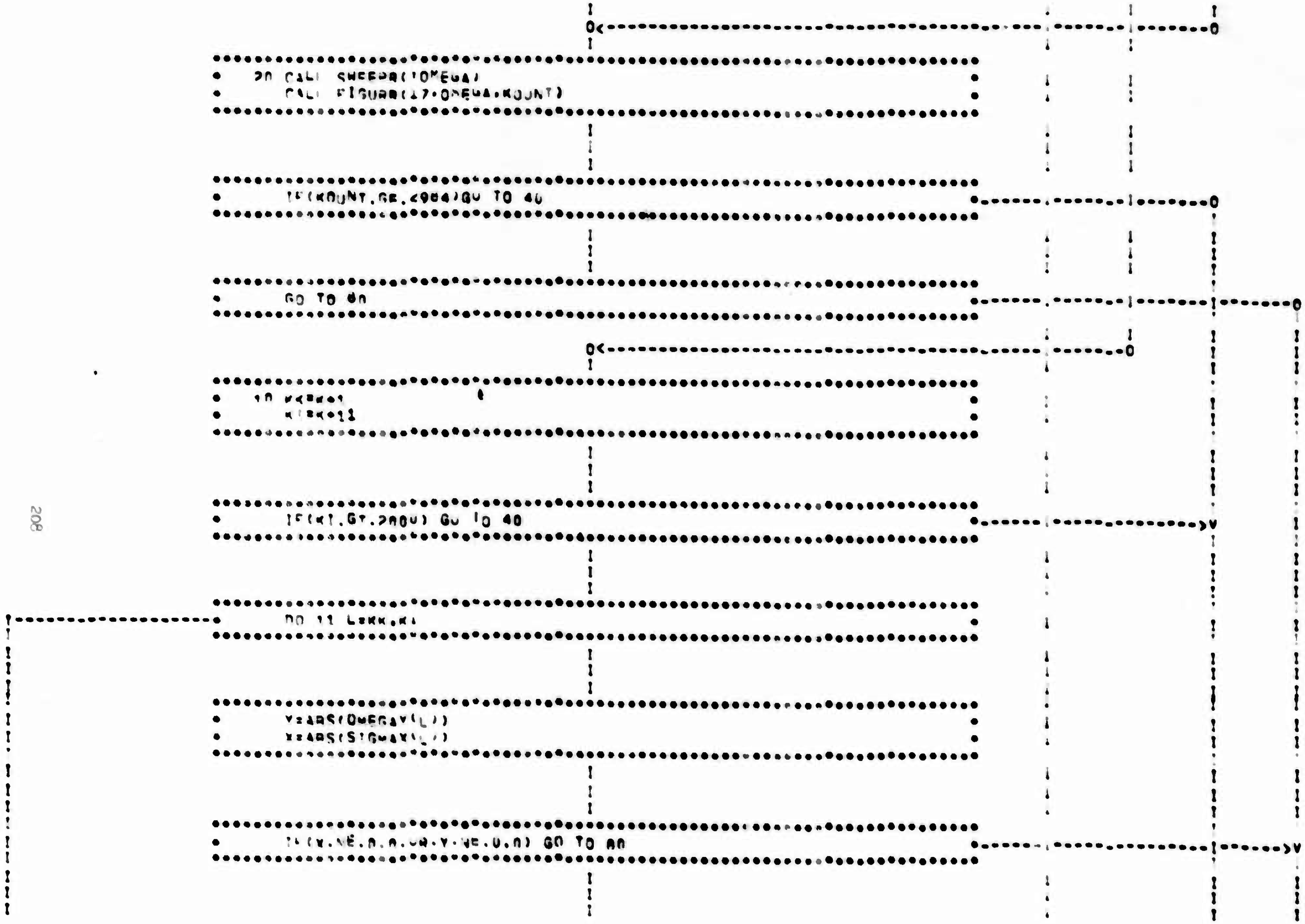

DO 50 K=K1,K2
IF(XA(K).NE.C.O) GO TO 77
50 IF(XB(K).NE.C.O) GO TO 77
GO TO 60
22 WRITE(6,10)XA(IZO),XC
10 FORMAT(5X,F2C.9,10X,F2C.9)
77 CONTINUE
60 RETURN
END

(ENTRANCE)

```
C
      SUBROUTINE FREQRS(SIGMAX,OMEGAY)
      DIMENSION SAVE1(-00,100),SAVE2(100,100),ACOEFF(100),BCOEFF(100),
     *          AAAAAA(99),BBBBBB(99),SIGMAX(2000),OMEGAY(2000),DEC(9),
     *          OMEGAS(52),OMEGA(17),W_OG(3000),PHILIN(3000),MDBLIN(3000
     *          ),PHIPLN(3000),MPDBLN(3000),ALIM(7),BIGSEM(3000),
     *          EMSUBS(3000)
      REAL MDBLIN,MPDBLN
      INTEGER FREAP,IEXP(4)
      INTEGER POINT(130,100)
      COMMON/FREEK/ACOEFF,BCOEFF
      COMMON/INFO2/ALIM
      COMMON/INFO3/OMEGA
      COMMON/INFO6/EMSUBS
      COMMON /INFO// ICOUNT
      COMMON SAVE1,SAVE2,POINT
      EQUIVALENCE (WLOG(1),SAVE1(1)),(PHILIN(1),SAVE2(1)),
     1            (SAVE1(3001),MDBLIN(1)),(SAVE2(3001),PHIPLN(1)),
     2            (SAVE1(6001),MPDBLN(1)),(SAVE2(6001),BIGSEM(1)),
     3            (ALIM(1),XSMALL),(AAAAAA(1),ACOEFF(2)),(BBBBBB(1),
     4            BCOEFF(2))
      TENS(X)=X*10.0
      DATA(DEC(I),I=1,7)/1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0/
      DATA LET1/4H DB/,LET2/4H DEG/
      K1=0
      KOUNT=1

      DO 61 I=1,4

   61 IEXP(I)=0

      DO 40 I=1,3000

   62 WLOG(I)=0.0
      PHILIN(I)=0.0
      BIGSEM(I)=0.0
      EMSUBS(I)=0.0
      MDBLIN(I)=0.0
      PHIPLN(I)=0.0
```

```
60 XPOWLN(I)=0.0

      WRITE(6,51)
      READ(5,19)XSMALL,DEGREE,DBLIM
      READ(5,12)FREXP
      IF(FREXP.NE.0)READ(5,12)(IEXP(K),K=1,4)
   12 FORMAT(4I10)
   19 FORMAT(3F10.0,50A)

      DO 90 I=2,7

   90 ALIM(I)=TEAS*ALIM(I-1)

      DO 70 J=1,6

      DO 71 I=1,9

      K=K+1

   71 OMEGAS(K)=REL(I)*ALIM(J)
```


70 K1=K1+9
OMEGAS(55)=ALIN(4)
K=1
CALL FIGURE(25,OMEGAS,KOUNT)
41 Y=ABS(OMEGAY(K))
X=ABS(SIGMAX(K))
IF(Y.EQ.0.0) GO TO 13
YOMEGA=Y
GO TO 20
13 IF(X.EQ.0.0) GO TO 10
YOMEGA=X


20 CALL SWEEPR(OMEGA)
CALL FIGURR(17,OMEGA,KOUNT)
IF(KOUNT.GE.2984)GO TO 40
GO TO 80
10 K=K+1
KI=K+11
IF(KI.GT.2800) GO TO 40
DO 11 L=KK,KI
Y=ABS(OMEGAY(L))
X=ABS(SIGMAX(L))
IF(X.NE.0.0.OR.Y.NE.0.0) GO TO 80

```
   11 CONTINUE

      GO TO 40

   30 IF(K.GE.ICOUNT) GO TO 40

      K=K+1

      GO TO 41

   40 DO 48 I=1,3000

      IF(WLOG(I).EQ.0.0.AND.PHILIN(I).EQ.0.0.AND.PHIPLN(I).EQ.0.0.AND.
     1MDHLIN(I).EQ.0.0.AND.MPDBLN(I).EQ.0.0.AND.BIGSEM(I).EQ.0.0.AND.
     2EMSUBS(I).EQ.0.0) GO TO 48

      JJ=MOD(I ,5/)
      IF(JJ.EQ.0) WRITE(6,21)

   48 WRITE(6,50) WLOG (I),BIGSEM(I),MDHLIN(I),PHILIN(I),EMSUBS(I),
     1MPDBLN(I),PHIPLN(I)
```

```
40 WRITE(6,14)
   CALL NEWPLT(DEGREE,ALOG,PHILIN,LFT2 )

   IF(IEXP(1).EQ.0) GO TO 30

   JJ=IEXP(1)

   DO 34 I=1,JJ

34 CALL EXPLOT(ALOG,3000,PHILIN,3000,DEGREE,LFT2)

30 CALL NEWPLT(OMLIM,ALOG,ADBLIN,LFT1)

   IF(IEXP(2).EQ.0) GO TO 31

   JJ=IEXP(2)

   DO 35 I=1,JJ
```

```
35 CALL EXPLOT(PLOG,3000,HDBLIN,3000,DBLIM,LET1)

31 WRITE(6,15)
   CALL NEWPLT(UEPR,B,PLOG,PHIPLN,LET2)

   IF(IEXP(3).EQ.0) GO TO 32

   JJ=IEXP(3)

   DO 34 I=1,JJ

34 CALL EXPLOT(PLOG,3000,PHIPLN,3000,DEGREE,LET2)

32 CALL NEWPLT(HBLIM,PLOG,HPDBLN,LET1)

   IF(IEXP(4).EQ.0) GO TO 33

   JJ=IEXP(4)
```

```
      DO 17 I=1,JJ

   17 CALL FRPLOT(PLOG,3000,MPOGLN,3000,DBLIM,LET)

   51 FORMAT(1H1,5X,7HOMEGA,13X,3HW/M/,16X,6HW/M/-DB,10X,3HPHI,20X,4HW/M/P,
     *19X,7HW/M/P-DB,14X,9HPHI-P,/)
   50 FORMAT(3X,1P1E12.5,5X,1P1E12.5,6X,1P1E12.5,9X,1P1E12.5,11X,
     *1P1E12.5,3X,1P1E12.5,9X,1P1E12.5)
   14 FORMAT(1H1,//////////////////////,25X,40HTHE FOLLOWING PLOTS ARE NOR
     *MAL FREQUENCY RESPONSE DATA                                          )
   15 FORMAT(1H1,//////////////////////,25X,30HTHE FOLLOWING PLOTS ARE POP
     *OV DATA                                )

   11 RETURN

      END
```

(ENTRANCE)

```
C
C
      SUBROUTINE FIGURE(KONT,OMEGA,KOUNT)
      DIMENSION SAVE1(100,100),SAVE2(100,100)
      DIMENSION ACOEFF(100),BCOEFF(100),AAAAAA(99),BBBBBB(99)
      DIMENSION WLOG(3000),PHILIN(3000),MODLIN(3000),PHIPLN(3000),
     *MODPLN(3000),BIGSEM(3000),EMSUBS(3000),OMEGA(KONT)
      REAL MODLIN,MODPLN
      INTEGER POINT(130,100)
      COMMON/FRED4/ACOEFF,BCOEFF
      EQUIVALENCE(AAAAAA(1),ACOEFF(2)),(BBBBBB(1),BCOEFF(2))
      COMMON/INFO/EMSUBP
      COMMON SAVE1,SAVE2,POINT
      EQUIVALENCE (WLOG(1),SAVE1(1)),(PHILIN(1),SAVE2(1)),
     1           (SAVE1(3001),MODLIN(1)),(SAVE2(3001),PHIPLN(1)),
     2           (SAVE1(6001),MODPLN(1)),(SAVE2(6001),BIGSEM(1))
```

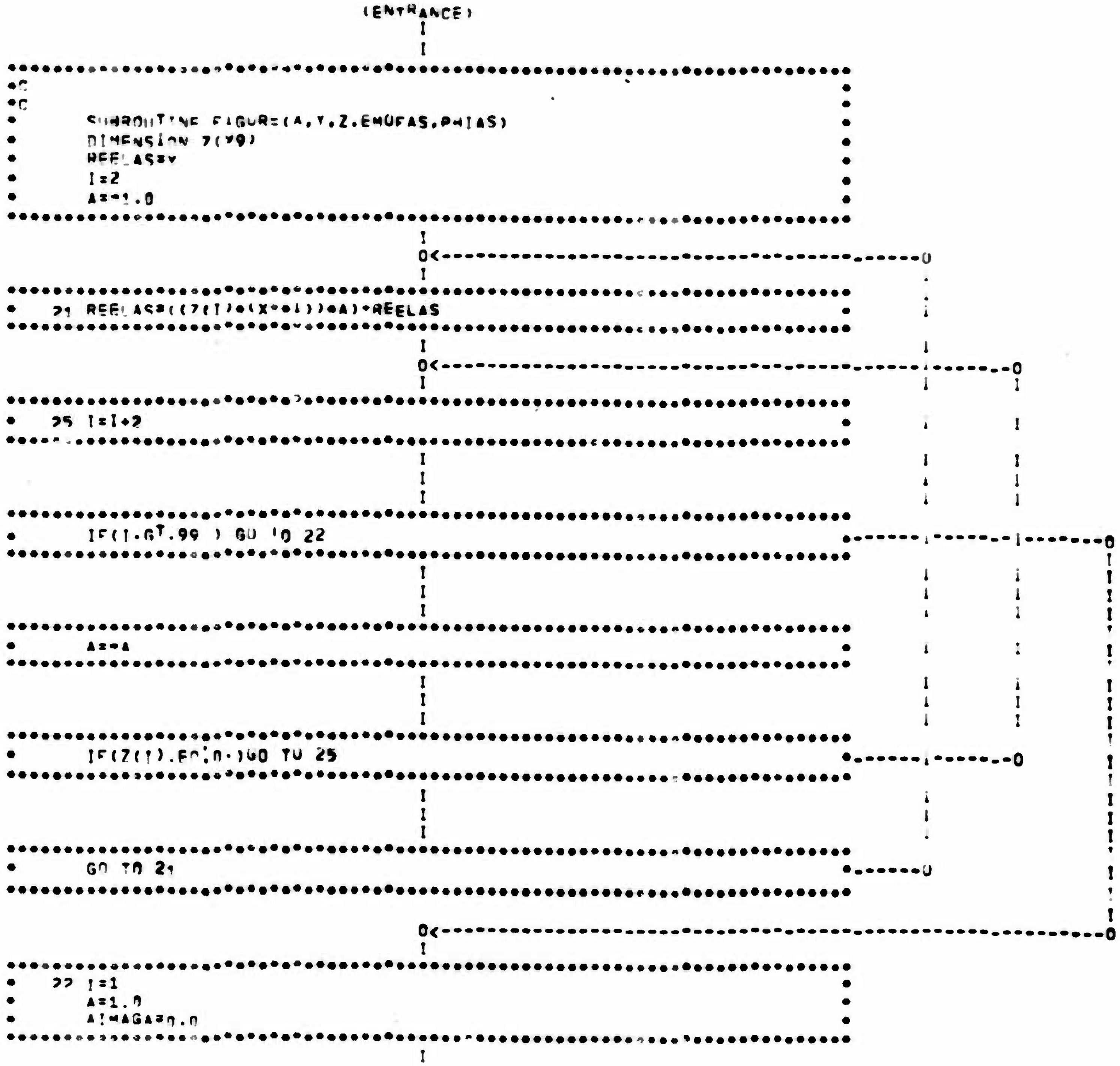
```
      DO 21 I=1,KONT
```

```
      CALL FIGURE(OMEGA(I),ACOEFF(1),AAAAAA,EMOFAS,PHIAS)
      CALL FIGURE(OMEGA(I),BCOEFF(1),BBBBBB,EMOFBS,PHIBS)
      BIGEMM=ABS(EMOFAS/EMOFBS)
      BIGSEM(KOUNT)=BIGEMM
      WLOG(KOUNT)=OMEGA(I)
      PHIEND=PHIAS-PHIBS
      IF(PHIEND.LT.-180.) PHIEND=PHIEND+360.
      IF(PHIEND.GT. 180.) PHIEND=PHIEND-360.
      PHILIN(KOUNT)=PHIEND
      PHIEND = PHIEND * 0.0174533
      REEL = BIGEMM * COS ( PHIEND )
      EIMAG= BIGEMM * SIN ( PHIEND )
      EMSUBP = SQRT ( ( REEL ** 2 ) + ( ( OMEGA ( I ) * EIMAG ) ** 2 ) )
      EMSUBS(KOUNT)=EMSUBP
      QUANZ=OMEGA(I) * EIMAG
      PHIPEE=ATAN2(QUANZ,REEL)
      PHIPEE = PHIPEE * 57.2957795131
      IF(PHIPEE .LT. 0.) PHIPEE = PHIPEE + 360.
      IF(PHIPEE.LT.-180.) PHIPEE=PHIPEE+360.
      IF(PHIPEE.GT. 180.) PHIPEE=PHIPEE-360.
      PHIPLN(KOUNT)=PHIPEE
      DEEREE=20.0*ALOG10(BIGEMM)
      MODLIN(KOUNT)=DEEREE
      DEEREP=20.0*ALOG10(EMSUBP)
      MODPLN(KOUNT)=DEEREP
      KOUNT=KOUNT+1
```

```
31 CONTINUE
```

```
RETURN
```

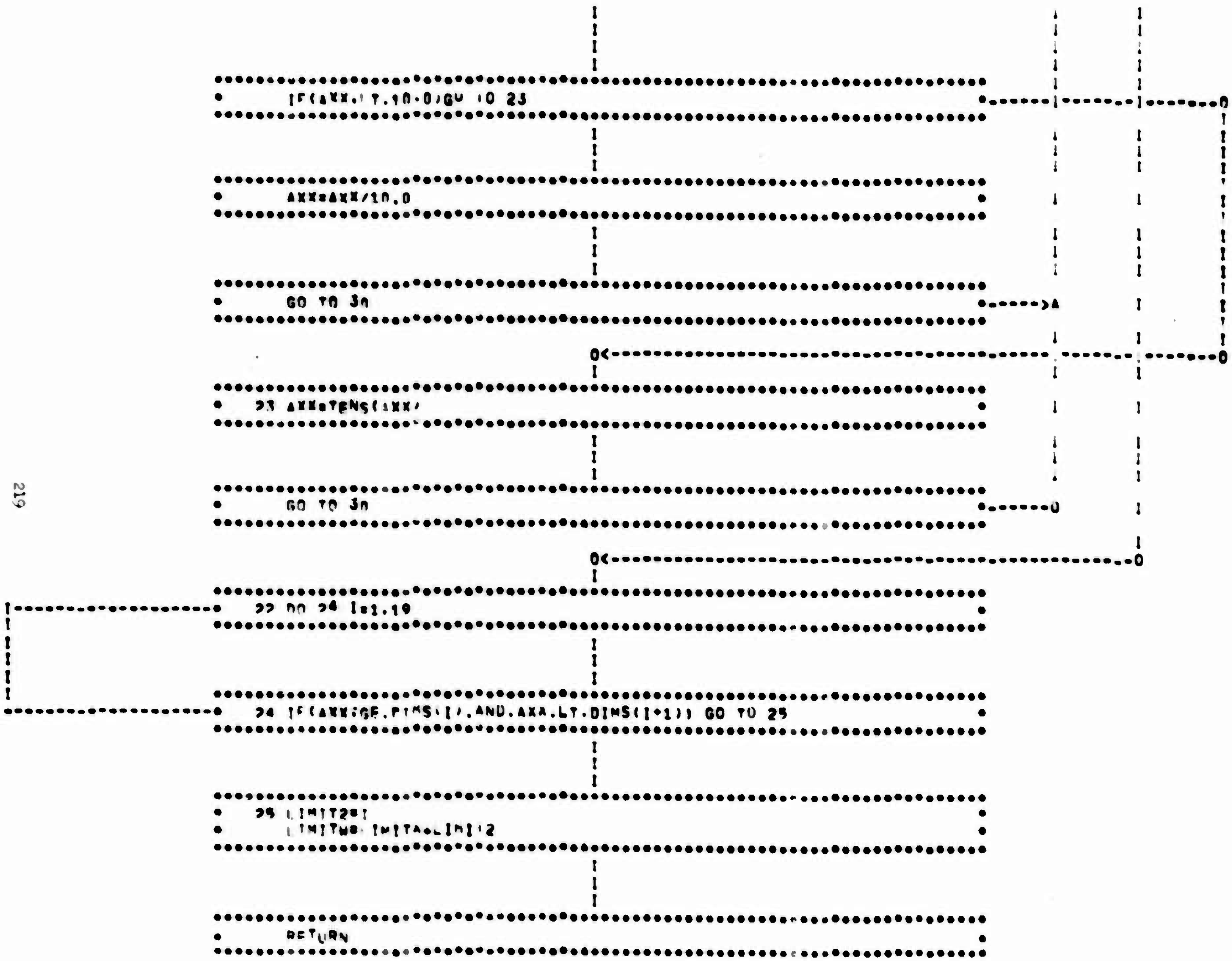
```
END
```


(ENTRANCE)
C
C
SUBROUTINE FIGURE=(A,Y,Z,EMUFAS,PHIAS)
DIMENSION Z(79)
REELAS=Y
I=2
A=-1.0
21 REELAS=((Z(I)*(X**I))*A)+REELAS
25 I=I+2
IF(I.GT.99) GO TO 22
A=-A
IF(Z(I).EQ.0.)GO TO 25
GO TO 21
22 I=1
A=1.0
AIMAGA=0.0

```
24 AIMAGA=(Z(I)*(X**I)*A)+AIMAGA
26 I=I+2
   IF(I.GT.99 ) GO TO 23
   A=-A
   IF(Z(I).EQ.0.)GO TO 26
   GO TO 24
23 EMOFAS=SQRT((REELAS**2)+(AIMAGA**2))
   PHIAS=ATAN2(AIMAGA,REELAS)
   PHIAS = PHIAS*57.2957795131
   IF(PHIAS.LT.0.) PHIAS=PHIAS+360.
   RETURN
```

END

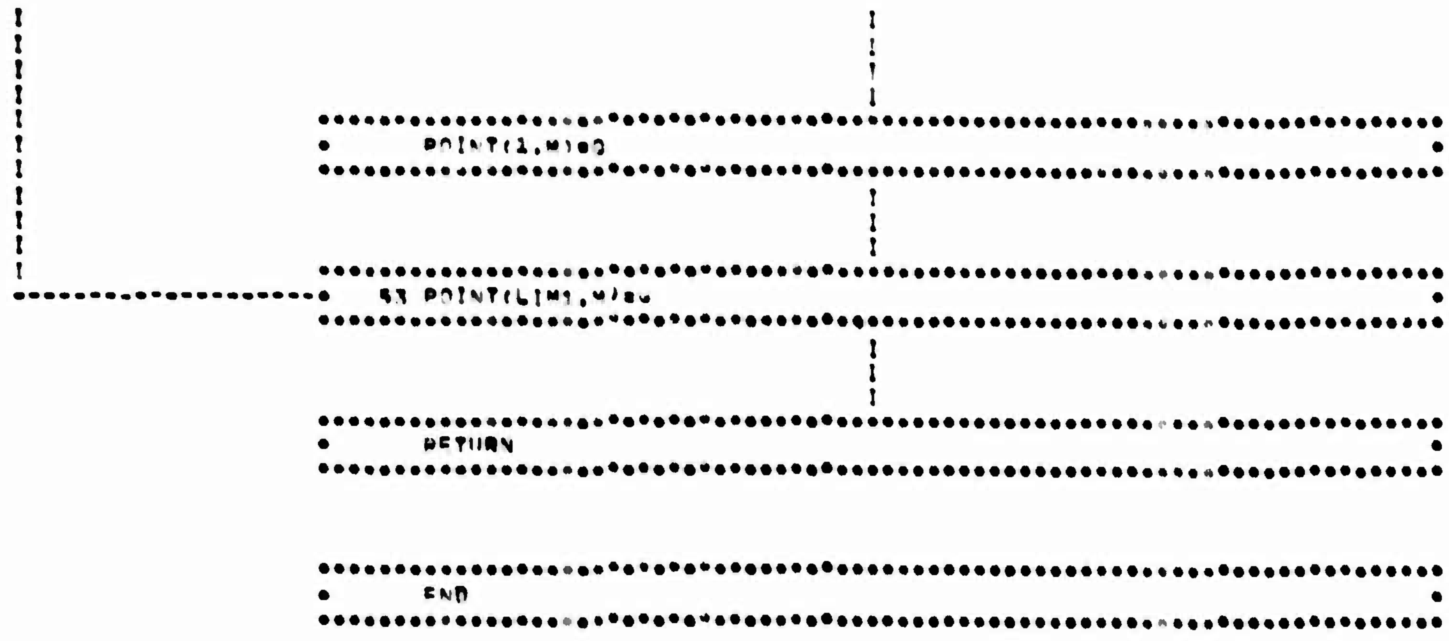
```
(ENTRANCE)
C
C
      SUBROUTINE LUGHER(AX,LIMITW)
      DIMENSION  ALIM(/),DIMS(20)
      EQUIVALENCE(HASTE,AXX)
      COMMON/!NF02/ALIM
      TENS(X)=X*10.0
      DATA (DIMS(I),I=1,20) /1.0,1.15,1.3,1.5,1.7,1.95,2.25,2.5,2.85,
     *3.15,3.5,3.95,4.5,4.85,5.4,6.05,6.8,7.7,8.8,9.999999/
      LIMITW=0
      LIMITX=0
      LIMITZ=0

      DO 20 I=1,6

      AA=2*(I-1)

   20 IF(XX.LT.ALIM(I+1).AND.XX.GE.ALIM(I)) GO TO 21

      RETURN

   21 LIMITX=TENS(AA)
      WASTE=ABS(XX)
      IF(WASTE.EQ.0.0) RETURN

   30 IF(AXX.GE.1.0.AND.AXX.LT.10.0) GO TO 22
```



IF(AXX.LT.10.0)GO TO 23
AXX=AXX/10.0
GO TO 30
23 AXX=TENS(AXX)
GO TO 30
22 DO 24 I=1,10
24 IF(AXX.GE.DIMS(I).AND.AXA.LT.DIMS(I+1)) GO TO 25
25 LIMIT2=I
LIMITW=LIMITA+LIMIT2
RETURN

END

```
(ENTRANCE)

C
C
      SUBROUTINE BLANKR (POINT,N1,N2,LIM1)
      INTEGER POINT(N1,N2),BLANK,DASH,Q
      COMMON /INFO/ Q,BLANK,DASH
      DATA BLANK/1H /,DASH/1H-/,Q/1HI/

      DO 51 M=1,N1

      DO 51 N=1,N2

   51 POINT(M,N)=BLANK

      DO 52 M=1,N1

      POINT(M,1)=DASH

   52 POINT(M,N2)=DASH

      DO 53 M=1,N2
```

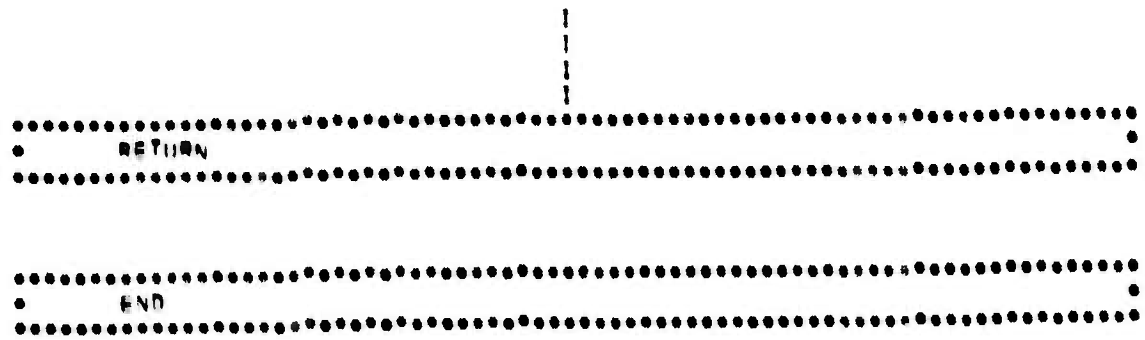


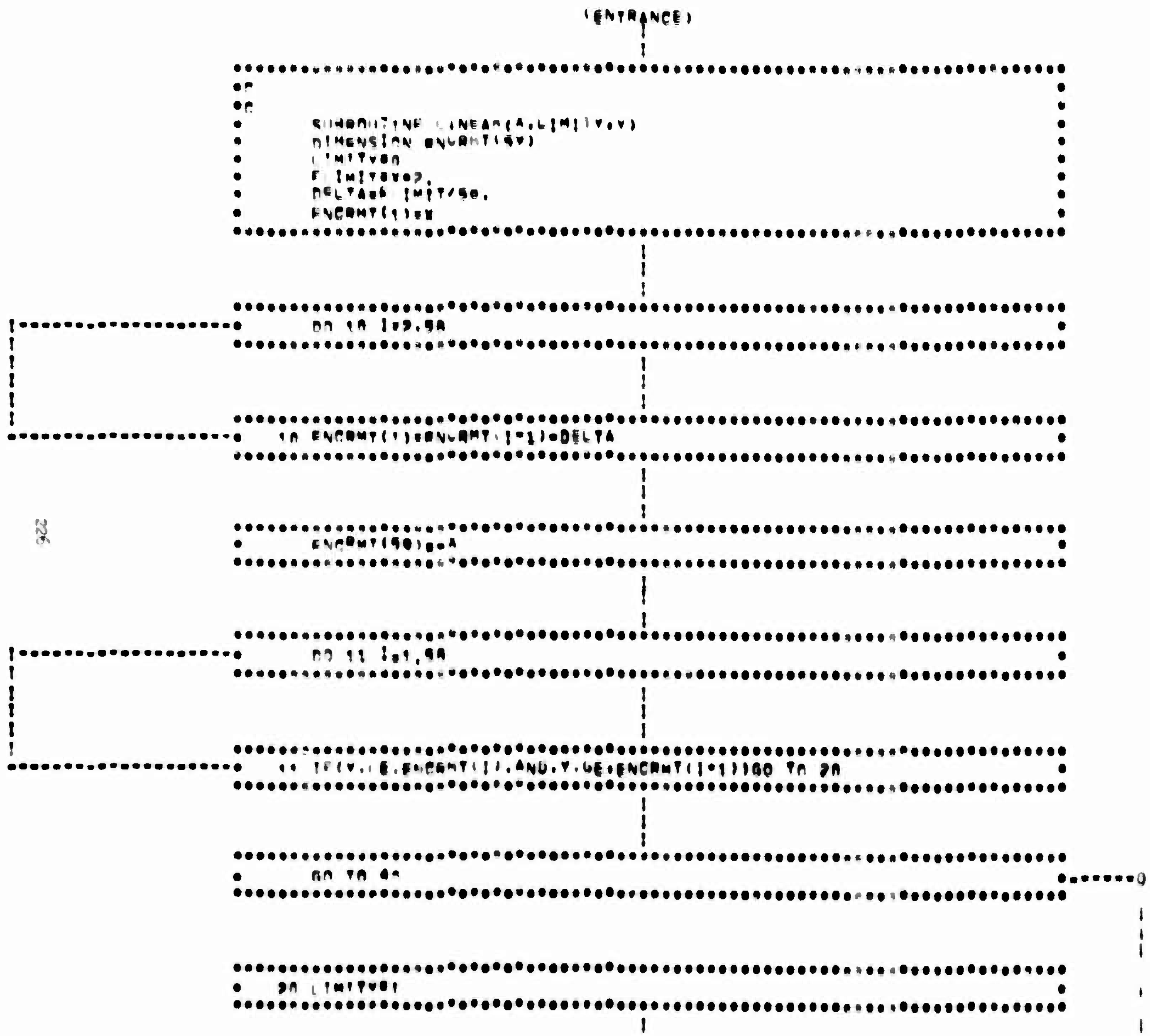
POINT(1,M)=0
53 POINT(LIM1,M)=0
RETURN
END

(ENTRANCE)

```
C
C
      SUBROUTINE NEWPLT(A,Y,Z,LET)
      INTEGER POINT(130,100),POINTF(120,58),STAR,O
      DIMENSION ALIM(7)
      DIMENSION Y(3000),Z(3000)
      DIMENSION SAVE1(100,100),SAVE2(100,100)
      COMMON /INFO1/ O,BLANK,DASH
      COMMON/INFO2/ALIM
      COMMON SAVE1,SAVE2,POINT
      EQUIVALENCE(POINT(1,1),POINTF(1,1))
      DATA STAR/1H*/,O/1HO/
    1 FORMAT(1H1)
      CALL BLANKR(POINTF,120,58,120)

      DO 10 I=1,3000

      IF(Y(I).EQ.0.0) GO TO 11

      GO TO 12

   11 IF(Z(I).EQ.0.0) GO TO 10

   12 CALL LOGGER(Y(I),LIMITW)

      IF(LIMITW.EQ.0) GO TO 10
```

```
CALL LINEAR(A,LIMITY,Z(I))

IF(LIMITY.EQ.0) GO TO 10

POINTE(LIMITX,LIMITY)=STAR

10 CONTINUE

DO 40 I=20,100,20

DO 40 J=1,58

40 POINTE(I,J)=W

      WRITE(6,1)
   30 FORMAT(1X,A(1P1E+0.1,1X,3HM/S,6X),1P1E10.1)
      J=X
      WRITE(6,51)J,LET,(POINTE(I,1),I=1,120)
   51 FORMAT(1X,I4, A4,1X,120A1)
      WRITE(6,20)((POINTE(I,J),I=1,120),J=2,57)
      J=Y
      WRITE(6,51)J,LET,(POINTE(I,58),I=1,120)
   20 FORMAT(10X,120A1)
      WRITE(6,30)ALIM
```


RETURN
END


(ENTRANCE)
C
C
SUBROUTINE LINEAR(A,LIMIT,V,Y)
DIMENSION ENCRMT(50)
LIMIT=0
ELIMIT=2.
DELTA=ELIMIT/50.
ENCRMT(1)=X
DO 10 I=2,50
10 ENCRMT(I)=ENCRMT(I-1)+DELTA
ENCRMT(50)=A
DO 11 I=1,50
11 IF(V.LE.ENCRMT(I).AND.V.GE.ENCRMT(I-1))GO TO 20
GO TO 40
20 LIMIT=I

```
                                                        I                            I
                                                        I                            I
                                                        O<---------------------------O
                                                        I
****************************************************************************************
*   40 RETURN                                                                          *
****************************************************************************************

****************************************************************************************
*      END                                                                             *
****************************************************************************************
```

```
(ENTRANCE)
    I
C
C
      SUBROUTINE SWEEPM(XOMEGA)
      DIMENSION OMEGA(17),SWEPER(17)
      COMMON/INFO3/OMEGA
      DATA (SWEPER(I),I=1,17) /0.01,0.05,0.1,0.2,0.3,0.4,0.5,0.6,0.7,
     *0.8,0.9,1.0,2.0,4.0,6.0,8.0,10.0/
    I
      DO 70 K=1,17
    I
   70 OMEGA(K)= SWEPER(K)*XOMEGA
    I
      RETURN

      END
```

AD________________ A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS

USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005

Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970

Trace Index Terms

Systems analyses, computer capability
Linear frequency response method
Nonlinear frequency response method
Open loop transfer function
Line printer plots, root loci
Phase-amplitude characteristic, nonlinear systems
Closed loop transfer function
Bode plots, system transfer
Popov plots, system transfer

1. Systems analyses, computer capability 2. Linear frequency response method
3. Nonlinear frequency response method 4. Line printer plots, root loci
A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.

AD________________ Burke, Harold H., Payne, Robert L., Jr.

USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005

Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970

Trace Index Terms

Systems analyses, computer capability
Linear frequency response method
Nonlinear frequency response method
Open loop transfer function
Line printer plots, root loci
Phase-amplitude characteristic, nonlinear systems
Closed loop transfer function
Bode plots, system transfer
Popov plots, system transfer

1. Systems analyses, computer capability 2. Linear frequency response method
3. Nonlinear frequency response method 4. Line printer plots, root loci
A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS Harold H. Burke, Robert L. Payne, Jr.

AD________________ Systems analyses, computer capability

USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005

Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970

Trace Index Terms

Systems analyses, computer capability
Linear frequency response method
Nonlinear frequency response method
Open loop transfer function
Line printer plots, root loci
Phase-amplitude characteristic, nonlinear systems
Closed loop transfer function
Bode plots, system transfer
Popov plots, system transfer

1. Systems analyses, computer capability 2. Linear frequency response method
3. Nonlinear frequency response method 4. Line printer plots, root loci
A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.

AD________________ Linear frequency response method

USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005

Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970

Trace Index Terms

Systems analyses, computer capability
Linear frequency response method
Nonlinear frequency response method
Open loop transfer function
Line printer plots, root loci
Phase-amplitude characteristic, nonlinear systems
Closed loop transfer function
Bode plots, system transfer
Popov plots, system transfer

1. Systems analyses, computer capability 2. Linear frequency response method
3. Nonlinear frequency response method 4. Line printer plots, root loci
A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS Harold H. Burke, Robert L. Payne, Jr.

AD________________ A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS

USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005

Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970

Trace Index Terms

Systems analyses, computer capability	Phase-amplitude characteristic, nonlinear systems
Linear frequency response method	Closed loop transfer function
Nonlinear frequency response method	Bode plots, system transfer
Open loop transfer function	Popov plots, system transfer
Line printer plots, root loci	

1. Systems analyses, computer capability 2. Linear frequency response method
3. Nonlinear frequency response method 4. Line printer plots, root loci
A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.

AD________________ Burke, Harold H., Payne, Robert L., Jr.

USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005

Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970

Trace Index Terms

Systems analyses, computer capability	Phase-amplitude characteristic, nonlinear systems
Linear frequency response method	Closed loop transfer function
Nonlinear frequency response method	Bode plots, system transfer
Open loop transfer function	Popov plots, system transfer
Line printer plots, root loci	

1. Systems analyses, computer capability 2. Linear frequency response method
3. Nonlinear frequency response method 4. Line printer plots, root loci
A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS Harold H. Burke, Robert L. Payne, Jr.

AD________________ Systems analyses, computer capability

USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005

Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970

Trace Index Terms

Systems analyses, computer capability	Phase-amplitude characteristic, nonlinear systems
Linear frequency response method	Closed loop transfer function
Nonlinear frequency response method	Bode plots, system transfer
Open loop transfer function	Popov plots, system transfer
Line printer plots, root loci	

1. Systems analyses, computer capability 2. Linear frequency response method
3. Nonlinear frequency response method 4. Line printer plots, root loci
A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.

AD________________ Linear frequency response method

USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005

Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970

Trace Index Terms

Systems analyses, computer capability	Phase-amplitude characteristic, nonlinear systems
Linear frequency response method	Closed loop transfer function
Nonlinear frequency response method	Bode plots, system transfer
Open loop transfer function	Popov plots, system transfer
Line printer plots, root loci	

1. Systems analyses, computer capability 2. Linear frequency response method
3. Nonlinear frequency response method 4. Line printer plots, root loci
A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS Harold H. Burke, Robert L. Payne, Jr.

AD_______________ Nonlinear frequency response method

USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005

Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970

Trace Index Terms

- Systems analyses, computer capability
- Linear frequency response method
- Nonlinear frequency response method
- Open loop transfer function
- Line printer plots, root loci
- Phase-amplitude characteristic, nonlinear systems
- Closed loop transfer function
- Bode plots, system transfer
- Popov plots, system transfer

1. Systems analyses, computer capability 2. Linear frequency response method 3. Nonlinear frequency response method 4. Line printer plots, root loci

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.

AD_______________ Line printer plots, root loci

USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005

Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970

Trace Index Terms

- Systems analyses, computer capability
- Linear frequency response method
- Nonlinear frequency response method
- Open loop transfer function
- Line printer plots, root loci
- Phase-amplitude characteristic, nonlinear systems
- Closed loop transfer function
- Bode plots, system transfer
- Popov plots, system transfer

1. Systems analyses, computer capability 2. Linear frequency response method 3. Nonlinear frequency response method 4. Line printer plots, root loci

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.

AD_______________

USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005

Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970

Trace Index Terms

- Systems analyses, computer capability
- Linear frequency response method
- Nonlinear frequency response method
- Open loop transfer function
- Line printer plots, root loci
- Phase-amplitude characteristic, nonlinear systems
- Closed loop transfer function
- Bode plots, system transfer
- Popov plots, system transfer

1. Systems analyses, computer capability 2. Linear frequency response method 3. Nonlinear frequency response method 4. Line printer plots, root loci

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.

AD_______________

USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005

Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970

Trace Index Terms

- Systems analyses, computer capability
- Linear frequency response method
- Nonlinear frequency response method
- Open loop transfer function
- Line printer plots, root loci
- Phase-amplitude characteristic, nonlinear systems
- Closed loop transfer function
- Bode plots, system transfer
- Popov plots, system transfer

1. Systems analyses, computer capability 2. Linear frequency response method 3. Nonlinear frequency response method 4. Line printer plots, root loci

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.

AD__________________ Nonlinear frequency response method

USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005

Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970

Trace Index Terms

Systems analyses, computer capability	Phase-amplitude characteristic, nonlinear systems
Linear frequency response method	Closed loop transfer function
Nonlinear frequency response method	Bode plots, system transfer
Open loop transfer function	Popov plots, system transfer
Line printer plots, root loci	

1. Systems analyses, computer capability 2. Linear frequency response method
3. Nonlinear frequency response method 4. Line printer plots, root loci

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.

AD__________________ Line printer plots, root loci

USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005

Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970

Trace Index Terms

Systems analyses, computer capability	Phase-amplitude characteristic, nonlinear systems
Linear frequency response method	Closed loop transfer function
Nonlinear frequency response method	Bode plots, system transfer
Open loop transfer function	Popov plots, system transfer
Line printer plots, root loci	

1. Systems analyses, computer capability 2. Linear frequency response method
3. Nonlinear frequency response method 4. Line printer plots, root loci

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.

AD__________________

USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005

Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970

Trace Index Terms

Systems analyses, computer capability	Phase-amplitude characteristic, nonlinear systems
Linear frequency response method	Closed loop transfer function
Nonlinear frequency response method	Bode plots, system transfer
Open loop transfer function	Popov plots, system transfer
Line printer plots, root loci	

1. Systems analyses, computer capability 2. Linear frequency response method
3. Nonlinear frequency response method 4. Line printer plots, root loci

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.

AD__________________

USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005

Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970

Trace Index Terms

Systems analyses, computer capability	Phase-amplitude characteristic, nonlinear systems
Linear frequency response method	Closed loop transfer function
Nonlinear frequency response method	Bode plots, system transfer
Open loop transfer function	Popov plots, system transfer
Line printer plots, root loci	

1. Systems analyses, computer capability 2. Linear frequency response method
3. Nonlinear frequency response method 4. Line printer plots, root loci

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

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

1. ORIGINATING ACTIVITY (Corporate author) USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Maryland 21005	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE
A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (First name, middle initial, last name)

Harold H. Burke
Robert L. Payne, Jr.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
March 1970	230	9

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO. RDT&E 1P765801M1102	AMSAA TM-69
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT This document may be further distributed by the holder only with specific prior approval of the Commanding General, U.S. Army Materiel Command, ATTN: AMCPA-SA, Washington, D.C. 20315.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY U.S. Army Materiel Command Washington, D.C.

13. ABSTRACT

A computer program to perform dynamic systems analyses and plot the results is presented. Linear systems and a large classification of nonlinear systems representing engineering, scientific, and economic disciplines can be modeled to permit application of the computer program. Two examples are given to demonstrate the capabilities of the analysis tool. The mathematical model of a missile guidance and control system is analyzed and a ratio of polynomials representing the closed loop transfer function of a high performance model follower aircraft is evaluated. Linear differential equations to the 100th order having real or complex roots can be studied. System characteristic equation root loci and system transfer functions are plotted.

DD FORM 1 NOV 65 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

Unclassified
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Systems analysis						
Systems synthesis						
Root locus						
Bode plots						
Steady state response						
Popov plots						
Modified phase-amplitude characteristic						