LINEAR SYSTEMS DYNAMICS

(LSD) PROGRAM

HC003C

VOLUME II - PROGRAMMER'S MANUAL

13 JANUARY 1969

Prepared by C. A. Argila
C.A. Argila
Electrical and
Electronics Group

Approved by W. P. Bennett
W. P. Bennett, Head
Subsystems Analysis Section

Approved by D. A. Schnebly
D.A. Schnebly
Group Leader
Electrical and
Electronics Group

Approved by D. L. Ball
D.L. Ball, Manager
Computational Simulations
Department

TRW
SYSTEMS GROUP

ACKNOWLEDGMENT

We wish to express our thanks to Mr. D. A. Schnebly without whose assistance it would have been impossible to isolate the various UNIVAC compiler errors. We wish to thank Mr. J. W. Pool, for his foresightedness in providing us with funds to completely restore and document the LSD Program; even though such an extensive effort was not immediately necessary, Mr. Pool realized how valuable a program as fundamental to systems analysis as the LSD Program would be to TRW's Houston Operations. We wish to express our gratitude, also, to the Messrs. R. Chan and L. Robinson for their assistance with the engineering aspects of this program and for providing us with many valuable suggestions on improving program usefulness. Finally, we wish to thank Mr. A. Paroczai of SCC for his efforts in providing us with current versions of the LSD Program at SCC and for keeping us informed of errors in SCC's version of the LSD Program as they were discovered.

ABSTRACT

The Linear Systems Dynamics (LSD) Program performs a root locus, frequency response, or time response analysis of a system when given either the poles and zeros of the transfer function of the system or a matrix array of equations that defines the system. For the time response analysis, a piece-wise quadratic driving function of up to one hundred segments may be specified. A data tape is generated by the LSD Program that may be used by the TRWPLT General Plotting Program to obtain root locus, frequency response, or time response plots.

The LSD Program is written in FORTRAN V for use on the SRU 1108 EXEC II computing system.

CONTENTS

CONTENTS (Continued)

CONTENTS (Continued)

TABLES

1. INTRODUCTION

The main program, LSD, and the individual subroutines are discussed in separate sections. Each section contains the applicable information for the following:

a) Identification (The main program identification data are applicable to each of the subroutines.)

b) Deck identification

c) Purpose

d) Storage (not including named COMMON storage)

e) Library usage

f) Subroutine usage (refer to Table 1.1 for subroutine cross reference)

g) Named COMMON usage (refer to Table 1.2 for named COMMON cross reference)

h) Arguments

i) Flowchart

Table 1.1. Subroutine Usage

Calling Routines	Called Routine: LSD ABC	ADDFRE ADJUST	ANGLE AUXSUB	BLOCK CDA	COMPUT CONIN	CONOUT C3CD	C3CM DB	DTVC EIGEN	ELIM EMU	EVAL FMASK	FREQR GAMMA	GCONJ ILT	ILTF MATVAR	MTRXPR ORDER	OUT ROOT	RTLC SCALE	SER
LSD CDA				X				X	X		X X	X X		X		X	
COMPUT CONIN			X				X									X	
CONOUT C3CD	X				X	X											
C3CM DTVC	X				X	X	X										
EIGEN ELIM	X				X	X X	X	X		X X			X	X	X		
EMU EVAL	X					X	X	X					X				X
FREQR ILT		X X			X								X				
ILTF ROOT	X		X						X	X						X	X
RTLC SCALE	X		X									X			X		

Table 1.2. Named Common Usage

Named Common Block	Subroutine																	
	LSD ABC	ADDFRE ADJUST	ANGLE AUXSUB	BLOCK CDA	COMPUT CONIN	CONOUT C3CD	C3CM DB	DTVC EIGEN	ELIM EMU	EVAL FMASK	FREQR GAMMA	GCONJ ILT	ILTF MATVAR	MTRXPR ORDER	OUT ROOT	RTLC SCALE	SER
ADJCOM		X									X						
AFCOM		X									X						
BASIC			X	X	X				X		X	X	X			X	
COMPLX	X		X	X X	X				X		X	X	X			X	
DCDA	X			X X													
DCMPLX	X			X X													
EXTRA			X													X	
FRQRSP	X			X							X						
ILTCOM												X	X				
INVLT	X			X X								X					
MATDAT	X			X X													
MTRX	X			X X													
OPTION	X			X X	X						X	X				X	
PARAM				X					X								
RLOCUS	X			X X													
ROOTS				X					X								
SKALE	X			X				X									
TITLES	X			X													

2. MAIN PROGRAM, LSD

IDENTIFICATION

Program number: HC003C

Title: LSD

Programming Language: FORTRAN V

Machine: SRU 1108 (EXEC II)

DECK IDENTIFICATION

LSD

PURPOSE

This routine serves as the overall driver, handling general input and output for the LSD Program.

DESCRIPTION

Each data case is read in and interrogated. A header page is printed out with the input options for each data case. Certain standard program initialization is performed, and control is then transferred to the routine that performs the actual LSD calculations. Upon completion of all the data cases, a summary page is printed out with the number of data cases processed, the run date, and the run time.

STORAGE

This routine requires 1316_8 locations.

LIBRARY USAGE

QQUFOF
CDATE
CTIME

SUBROUTINE USAGE

CDA

NAMED COMMON USAGE

COMPLX - contains poles, zeros, and closed-loop poles

DCDA - contains matrix data

DCMPLX - contains S-plane poles and zeros and additional poles and zeros

FRQRSP - contains frequency response table

INVLT - contains time response data

MATDAT - contains matrix data

MTRX - contains matrix

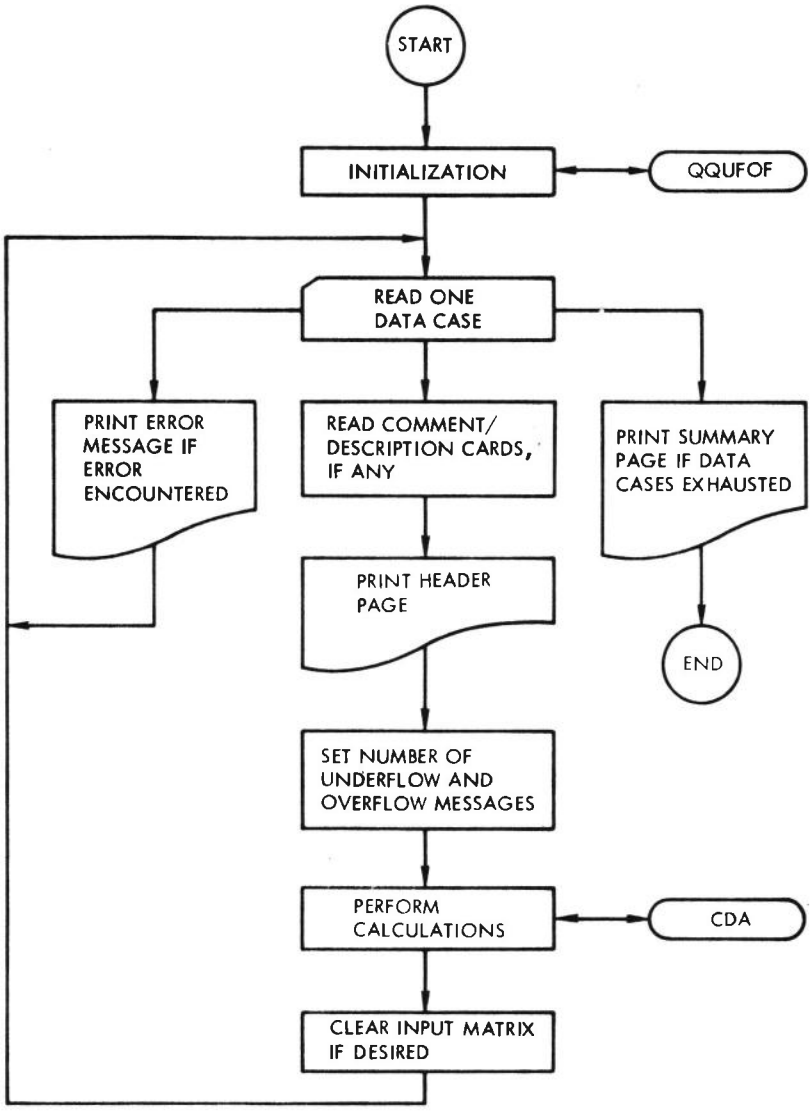
OPTION - contains input options

RLOCUS - contains root locus data

SKALE - contains scaling data

TITLES - contains comment/description cards

FLOWCHART


START
INITIALIZATION
QQUFOF
READ ONE DATA CASE
PRINT ERROR MESSAGE IF ERROR ENCOUNTERED
READ COMMENT/ DESCRIPTION CARDS, IF ANY
PRINT SUMMARY PAGE IF DATA CASES EXHAUSTED
PRINT HEADER PAGE
END
SET NUMBER OF UNDERFLOW AND OVERFLOW MESSAGES
PERFORM CALCULATIONS
CDA
CLEAR INPUT MATRIX IF DESIRED

3. SUBROUTINE ABC

DECK IDENTIFICATION

ABC

PURPOSE

Given a complex argument, Z, ABC(Z) is the maximum of the absolute value of the real and imaginary parts of Z.

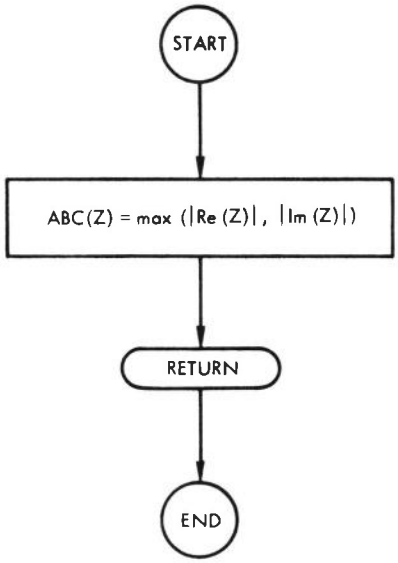
STORAGE

This routine requires 33_8 locations.

ARGUMENT

Z

FLOWCHART



4. SUBROUTINE ADDFRE

DECK IDENTIFICATION

ADDFRE

PURPOSE

This routine performs a calculation regarding additional frequencies for the LSD subroutine FREQR.

STORAGE

This routine requires 127_8 locations.

LIBRARY USAGE

CABS

NAMED COMMON USAGE

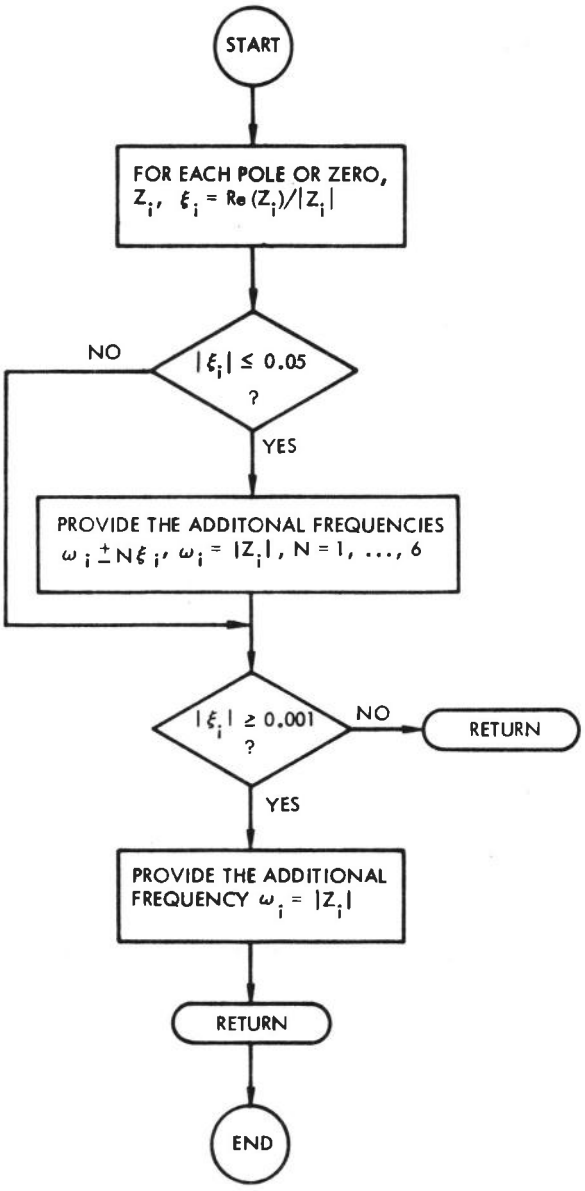
AFCOM - contains data for additional frequencies calculation.

ARGUMENTS

NPOINT - dimension of array

ARRAY - array of poles or zeros

FLOWCHART


START
FOR EACH POLE OR ZERO, Z_i, $\xi_i = Re(Z_i)/|Z_i|$
NO
$|\xi_i| \leq 0.05$?
YES
PROVIDE THE ADDITONAL FREQUENCIES $\omega_i \pm N\xi_i$, $\omega_i = |Z_i|$, N = 1, ..., 6
$|\xi_i| \geq 0.001$?
NO
RETURN
YES
PROVIDE THE ADDITIONAL FREQUENCY $\omega_i = |Z_i|$
RETURN
END

5. SUBROUTINE ADJUST

DECK IDENTIFICATION

ADJUST

PURPOSE

This routine adjusts the frequency for the LSD frequency response routine, FREQR.

STORAGE

This routine requires 67_8 locations.

NAMED COMMON USAGE

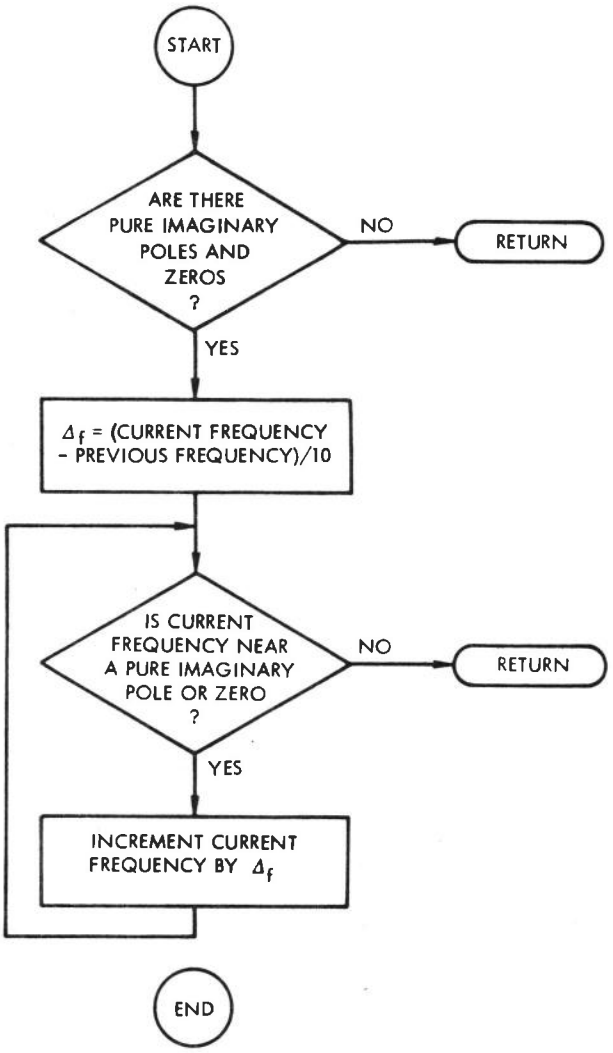
ADJCOM - contains frequency adjustment data

ARGUMENTS

OLDFRQ - prior frequency

FRQ - current frequency

FLOWCHART


START
ARE THERE PURE IMAGINARY POLES AND ZEROS ?
NO
RETURN
YES
Δ_f = (CURRENT FREQUENCY - PREVIOUS FREQUENCY)/10
IS CURRENT FREQUENCY NEAR A PURE IMAGINARY POLE OR ZERO ?
NO
RETURN
YES
INCREMENT CURRENT FREQUENCY BY Δ_f
END

6. SUBROUTINE ANGLE

DECK IDENTIFICATION

ANGLE

PURPOSE

This function returns the modulus of a complex argument. If the imaginary part of the argument is zero, the modulus is set to zero.

STORAGE

This routine requires 46_8 locations.

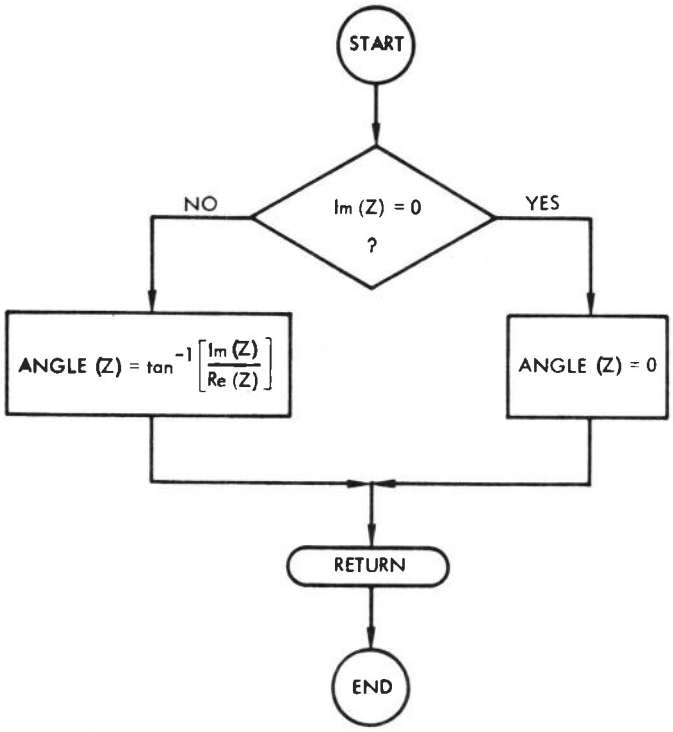
LIBRARY USAGE

ATAN2

ARGUMENT

Z

FLOWCHART


START
NO
Im (Z) = 0
?
YES
ANGLE (Z) = tan$^{-1}\left[\frac{Im\ (Z)}{Re\ (Z)}\right]$
ANGLE (Z) = 0
RETURN
END

7. SUBROUTINE AUXSUB

DECK IDENTIFICATION

AUXSUB

PURPOSE

This routine evaluates the root locus function for the LSD Program.

STORAGE

This routine requires 305_8 locations.

NAMED COMMON USAGE

BASIC - numbers of poles, zeros, and gains

COMPLX

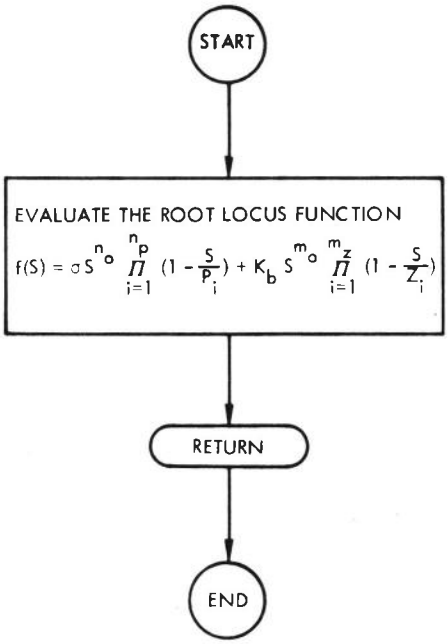
EXTRA - root locus phase, σ, and bode gain, K_b.

ARGUMENTS

S

FOFS

FLOWCHART


START
EVALUATE THE ROOT LOCUS FUNCTION
$f(S) = \sigma S^{n_o} \prod_{i=1}^{n_p} (1 - \frac{S}{P_i}) + K_b S^{m_o} \prod_{i=1}^{m_z} (1 - \frac{S}{Z_i})$
RETURN
END

8. SUBROUTINE BLOCK

DECK IDENTIFICATION

BLOCK

PURPOSE

This is a block data subroutine for the LSD Program.

NAMED COMMON USAGE

COMPLX

DCDA

DCMPLX

FRQRSP

INVLT

MATDAT

MTRX

OPTION

RLOCUS

SKALE

9. SUBROUTINE CDA

DECK IDENTIFICATION

CDA

PURPOSE

This subroutine interprets each data case for the LSD Program and calls the proper subroutines necessary to calculate a solution.

STORAGE

This routine requires 22106_8 locations.

SUBROUTINE USAGE

GCONJ

GAMMA

MTRXPR

EIGEN

ELIM

FREQR

RTLC

ILT

NAMED COMMON USAGE

BASIC

COMPLX

DCDA

DCMPLX

INVLT

MATDAT

MTRX

OPTION

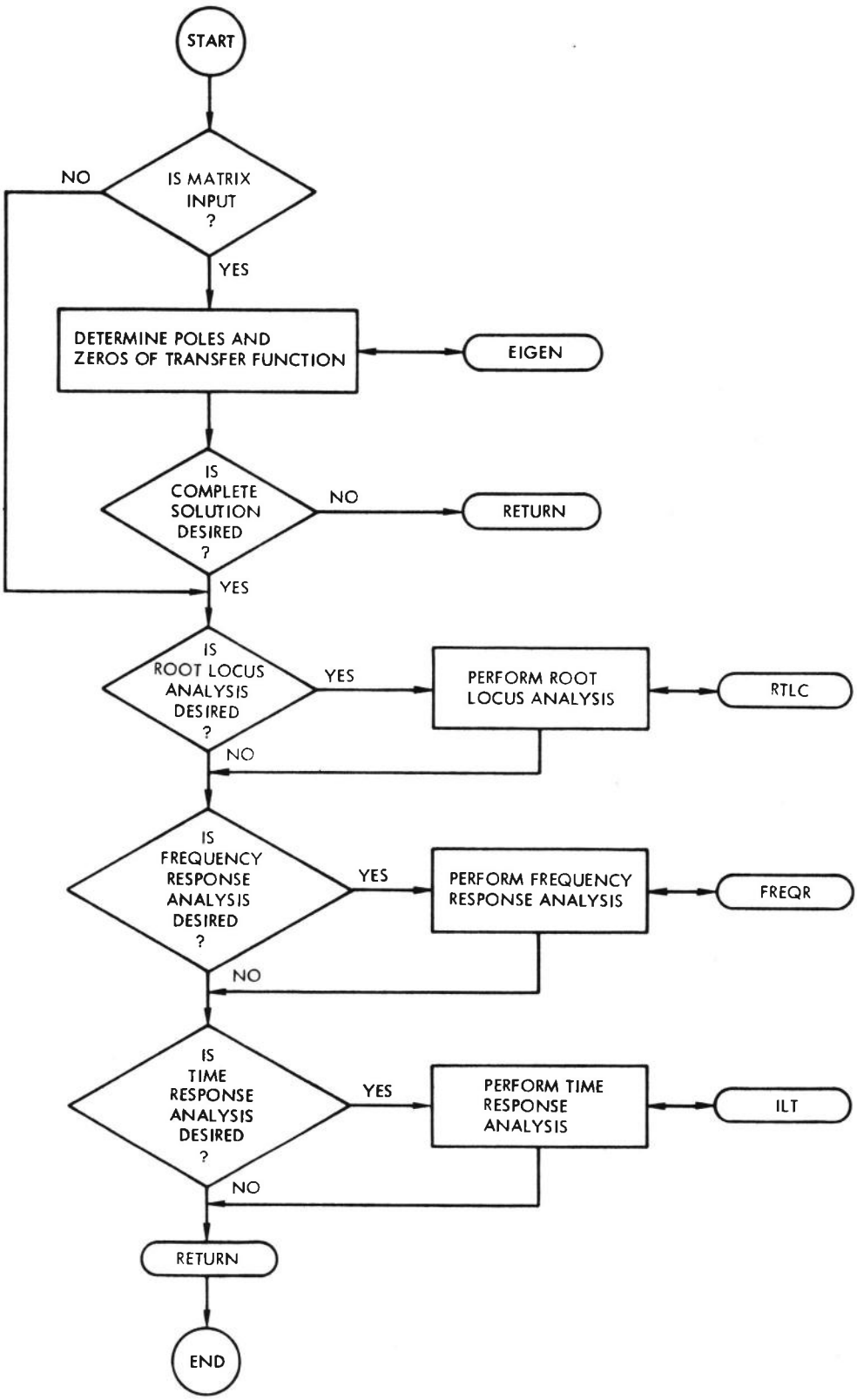
PARAM - numerator and denominator indices for transfer function

RLOCUS

ROOTS - roots of numerator and denominator of transfer function

TITLES

FLOWCHART


START
IS MATRIX INPUT ?
NO
YES
DETERMINE POLES AND ZEROS OF TRANSFER FUNCTION
EIGEN
IS COMPLETE SOLUTION DESIRED ?
NO
RETURN
YES
IS ROOT LOCUS ANALYSIS DESIRED ?
YES
PERFORM ROOT LOCUS ANALYSIS
RTLC
NO
IS FREQUENCY RESPONSE ANALYSIS DESIRED ?
YES
PERFORM FREQUENCY RESPONSE ANALYSIS
FREQR
NO
IS TIME RESPONSE ANALYSIS DESIRED ?
YES
PERFORM TIME RESPONSE ANALYSIS
ILT
NO
RETURN
END

10. SUBROUTINE COMPUT

DECK IDENTIFICATION

COMPUT

PURPOSE

This routine evaluates the system transfer function and prints one line of output for the frequency response portion of the LSD Program. One record of plot tape output is generated.

STORAGE

This routine requires 354_8 locations.

LIBRARY USAGE

CABS

SUBROUTINE USAGE

DB

ANGLE

NAMED COMMON USAGE

BASIC

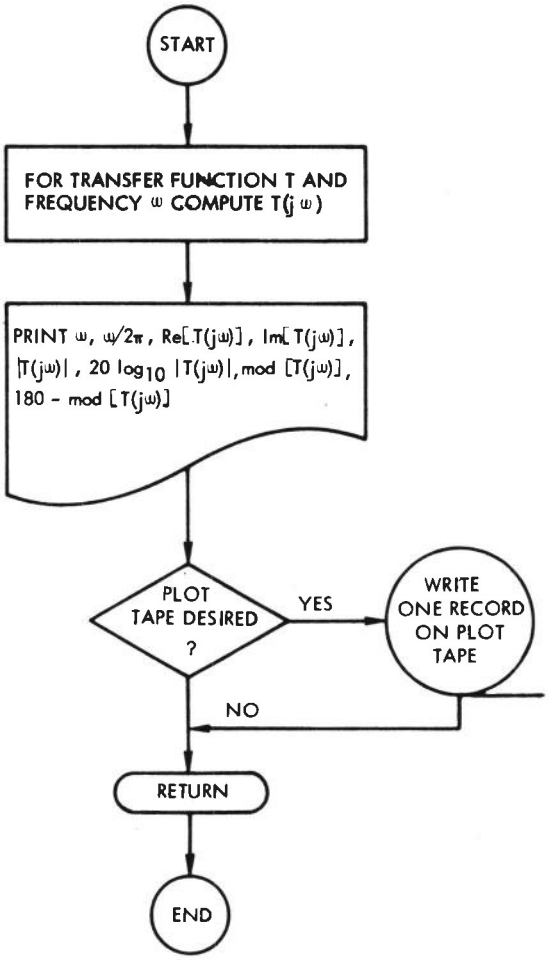
COMPLX

OPTION

ARGUMENT

OMEGA

FLOWCHART


START
FOR TRANSFER FUNCTION T AND FREQUENCY ω COMPUTE T(j ω)
PRINT ω, ω/2π, Re[T(jω)], Im[T(jω)], |T(jω)|, 20 log10 |T(jω)|, mod [T(jω)], 180 - mod [T(jω)]
PLOT TAPE DESIRED ?
YES
WRITE ONE RECORD ON PLOT TAPE
NO
RETURN
END

11. SUBROUTINE CONIN

DECK IDENTIFICATION

CONIN

PURPOSE

This routine accepts a complex argument, Z, and returns a complex argument, W, and integer, J, so that $Z = W \times 10^K$ where $1 \leq$ ABC(W) $\leq 10^{10}$.

STORAGE

This routine requires 41_8 locations.

SUBROUTINE USAGE

SCALE

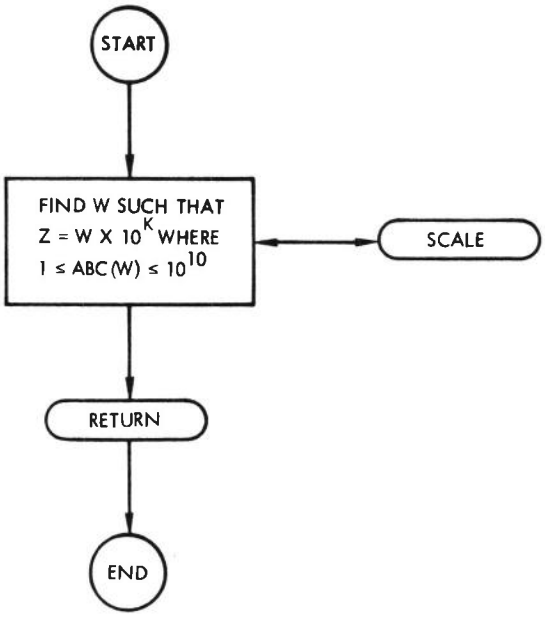
ARGUMENTS

Z

W

K

FLOWCHART


START
FIND W SUCH THAT
Z = W X 10^K WHERE
1 ≤ ABC(W) ≤ 10^10
SCALE
RETURN
END

12. SUBROUTINE CONOUT

DECK IDENTIFICATION

CONOUT

PURPOSE

Given a complex argument, Z, and integer, K, this routine attempts to set the complex argument $W = Z \times 10^K$. If |Z| or |K| are too large, W will be set to $Z \times 10^{K'}$ and K will be set to K - K' so that $W = Z \times 10^K$.

STORAGE

The routine requires 164_8 locations.

SUBROUTINE USAGE

ABC

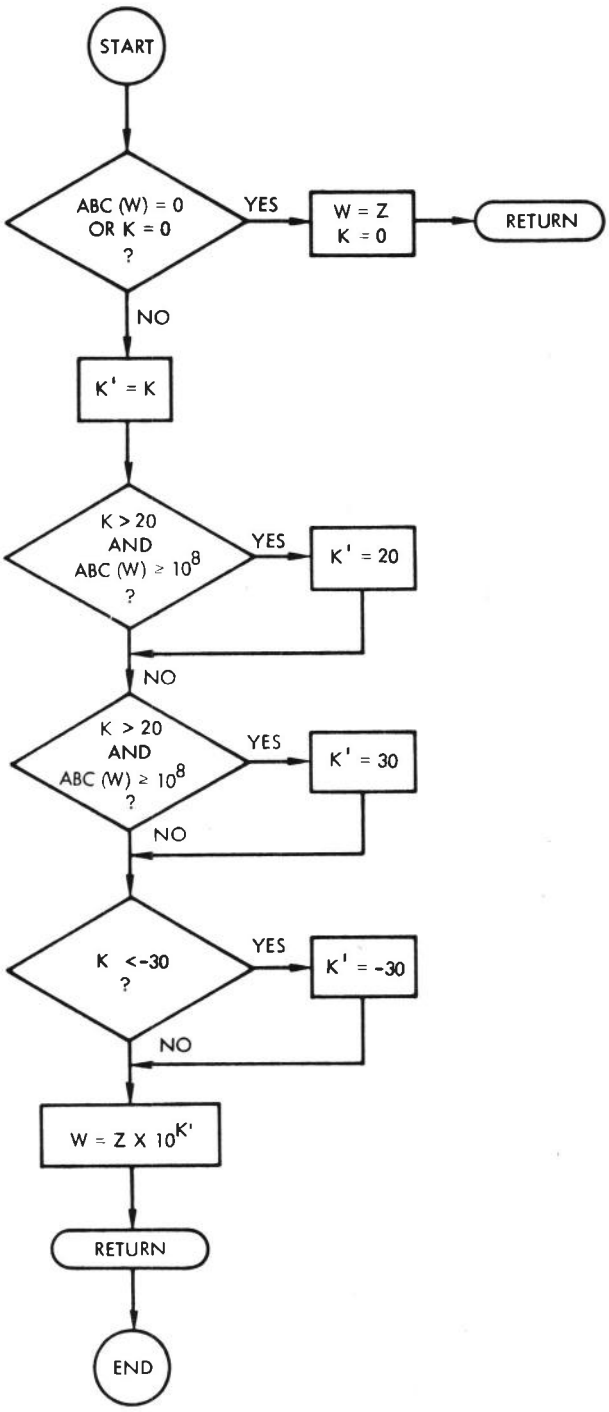
ARGUMENTS

Z

W

K

FLOWCHART


START
ABC (W) = 0 OR K = 0 ?
YES
W = Z
K = 0
RETURN
NO
K' = K
K > 20 AND ABC (W) ≥ 10^8 ?
YES
K' = 20
NO
K > 20 AND ABC (W) ≥ 10^8 ?
YES
K' = 30
NO
K <-30 ?
YES
K' = -30
NO
W = Z X 10^K'
RETURN
END

13. SUBROUTINE C3CD

DECK IDENTIFICATION

C3CD

PURPOSE

Given the three cell numbers (Z1, KZ1), (Z2, KZ2), this routine produces the three cell quotion, (Z, KZ).

STORAGE

This routine requires 114_8 locations.

SUBROUTINE USAGE

CONIN

CONOUT

ARGUMENTS

Z

KZ

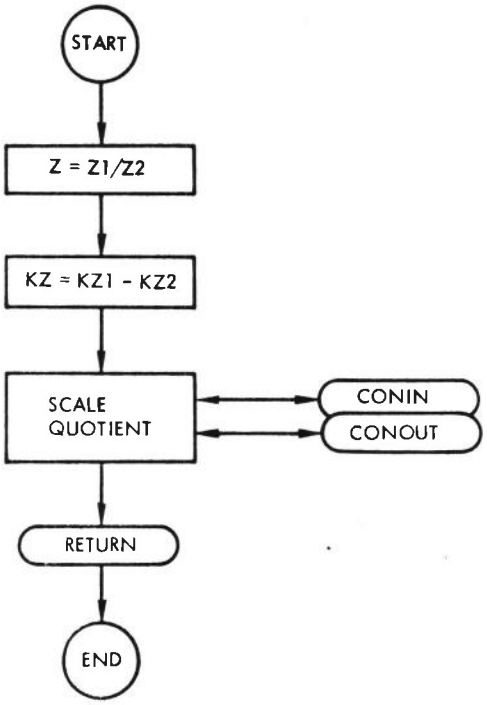
Z1

KZ1

Z2

KZ2

FLOWCHART


START
Z = Z1/Z2
KZ = KZ1 - KZ2
SCALE
QUOTIENT
CONIN
CONOUT
RETURN
END

14. SUBROUTINE C3CM

DECK IDENTIFICATION

C3CM

PURPOSE

This routine accepts an array of complex numbers, Z(1),...,Z(N) and a three cell number (Z0,K0) and returns the three cell product

$$(W,K) = Z(1) \times \ldots \times Z(N) \times (Z0,K0)$$

if IFLAG = 1; otherwise it returns the three cell product

$$(W,K) = Z(1) \times \ldots \times Z(N).$$

STORAGE

This routine requires 164_8 locations.

SUBROUTINE USAGE

CONIN

CONOUT

ARGUMENTS

IFLAG

Z

N

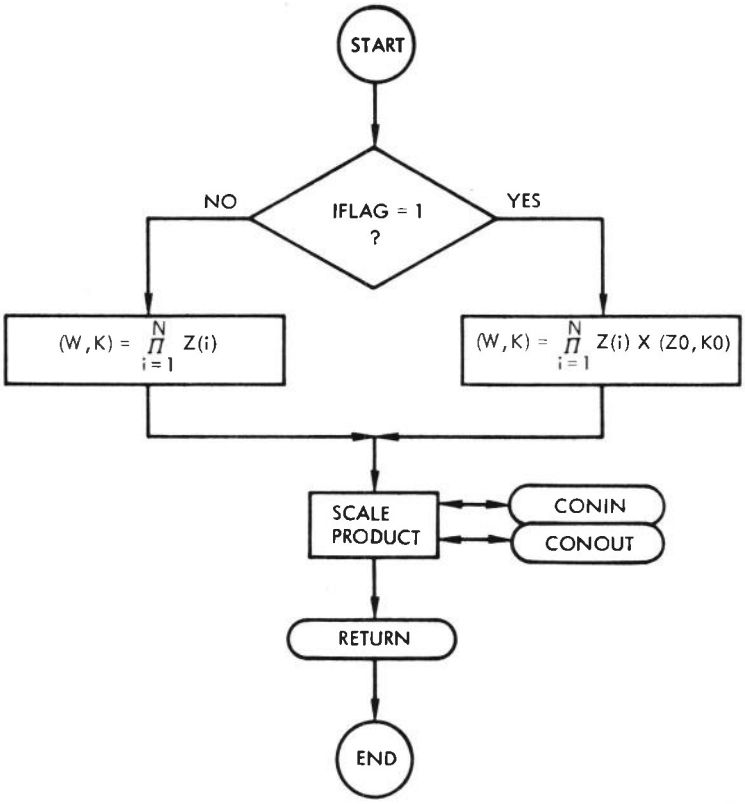
W

K

Z0

K0

FLOWCHART


START
IFLAG = 1
?
NO
YES
(W,K) = $\prod_{i=1}^{N}$ Z(i)
(W,K) = $\prod_{i=1}^{N}$ Z(i) X (Z0,K0)
SCALE
PRODUCT
CONIN
CONOUT
RETURN
END

15. SUBROUTINE DB

DECK IDENTIFICATION

DB

PURPOSE

This routine converts power ratio to decibels.

STORAGE

This routine requires 55_8 locations.

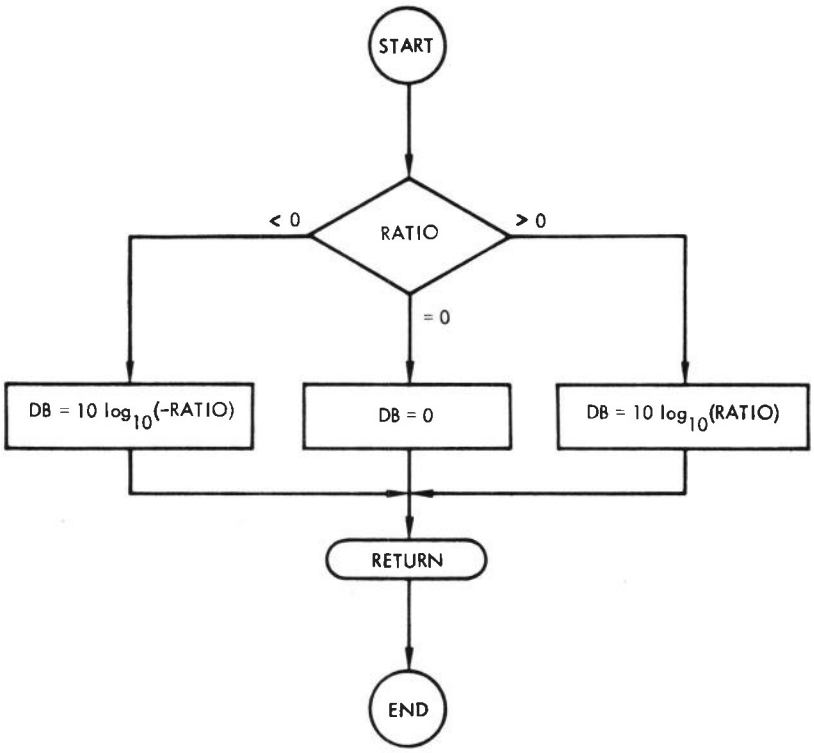
LIBRARY USAGE

ALOG10

ARGUMENT

RATIO - power ratio

FLOWCHART


START
RATIO
< 0
= 0
> 0
DB = 10 log10(-RATIO)
DB = 0
DB = 10 log10(RATIO)
RETURN
END

16. SUBROUTINE DTVC

DECK IDENTIFICATION

DTVC

PURPOSE

This routine performs a determinant manipulation and evaluation for the LSD Program.

STORAGE

This routine requires 1407_8 locations.

SUBROUTINE USAGE

ABC

C3CM

ARGUMENTS

A

N

DET

KP

Q

NSPEC

S

IP

D

NX

KAKE

17. SUBROUTINE EIGEN

DECK IDENTIFICATION

EIGEN

PURPOSE

Given a matrix representing a system of equations that describes a network or other system, this routine solves for the ratio of any two variables in the system as required for the LSD Program.

STORAGE

This routine requires 7153_8 locations.

LIBRARY USAGE

CSQRT

SUBROUTINE USAGE

ABC

C3CD

EVAL

MATVAR

DTVC

C3CM

OUT

CONIN

CONOUT

NAMED COMMON USAGE

SKALE

ARGUMENTS

R

SS

U

LAMBDA

A

C

B

STOR

IERR

18. SUBROUTINE ELIM

DECK IDENTIFICATION

ELIM

PURPOSE

This routine eliminates the common poles and zeros for the LSD Program. It also reorders the poles and zeros so that the zero elements occur first.

STORAGE

This routine requires 331_8 locations.

LIBRARY USAGE

CABS

SUBROUTINE USAGE

FMASK

ORDER

NAMED COMMON USAGE

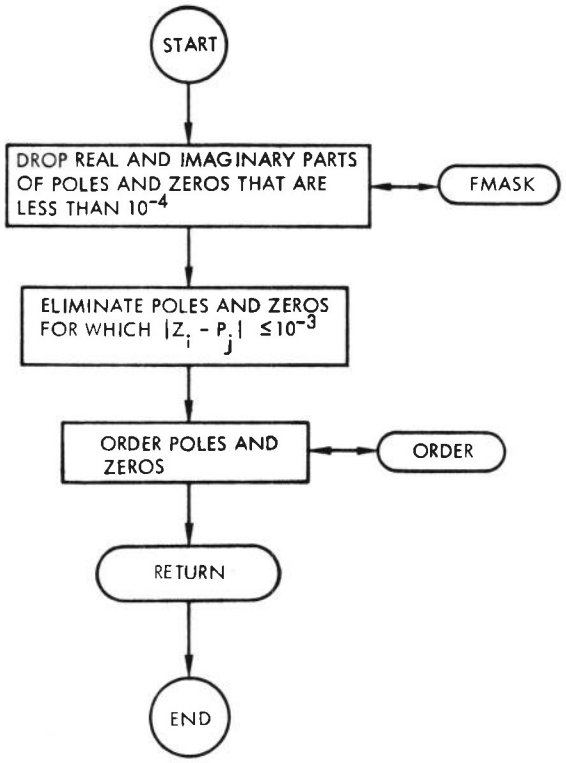
BASIC

COMPLX

ROOTS

PARAM

FLOWCHART


START
DROP REAL AND IMAGINARY PARTS OF POLES AND ZEROS THAT ARE LESS THAN 10^{-4}
FMASK
ELIMINATE POLES AND ZEROS FOR WHICH $|Z_i - P_j| \leq 10^{-3}$
ORDER POLES AND ZEROS
ORDER
RETURN
END

19. SUBROUTINE EMU

DECK IDENTIFICATION

EMU

PURPOSE

For a given argument, Z, and integer, KARG2, this routine returns

$$ANS = \begin{cases} e^{Z}, \text{ KARG2} = 0 \\ \text{SER (Z, KARG2, 15), KARG2} \neq 0 \text{ and } |Z| < 2 \\ e^{Z} - 1, \text{ KARG2} = 1, |Z| \geq 2 \\ e^{Z} - 1 - \text{SER (Z, 1, KARG2 - 1), otherwise} \end{cases}$$

STORAGE

This routine requires 140_8 locations.

LIBRARY USAGE

CEXP

CABS

SUBROUTINE USAGE

SER

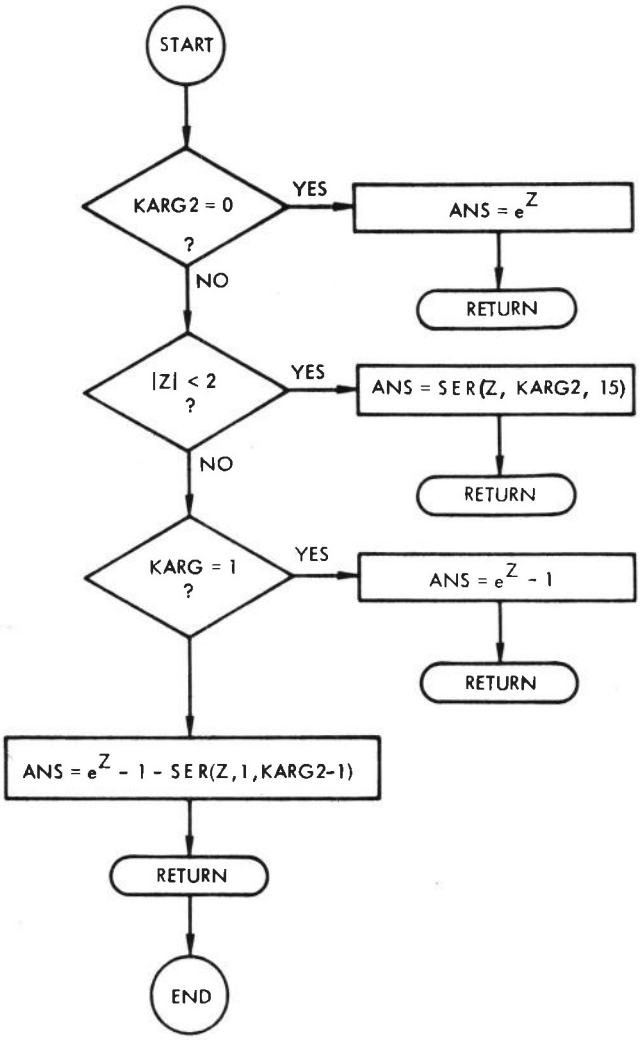
ARGUMENTS

Z

KARG2

ANS

FLOWCHART


START
KARG2 = 0 ?
YES
ANS = e^Z
RETURN
NO
|Z| < 2 ?
YES
ANS = SER(Z, KARG2, 15)
RETURN
NO
KARG = 1 ?
YES
ANS = e^Z - 1
RETURN
ANS = e^Z - 1 - SER(Z,1,KARG2-1)
RETURN
END

20. SUBROUTINE EVAL

DECK IDENTIFICATION

EVAL

PURPOSE

This routine evaluates the determinant equation (F, KF) = A * LAM for the LSD Program.

STORAGE

This routine requires 256_8 locations.

SUBROUTINE USAGE

MATVAR

DTVC

ABC

C3CM

C3CD

ARGUMENTS

A

B

N1X

F

KF

N1

N2

IP

S

DD

DIFF

LAM

LAMBDA

P

G

KG

KACC

21. SUBROUTINE FMASK

DECK IDENTIFICATION

FMASK

PURPOSE

This routine accepts one real argument, X, and returns this argument if its magnitude is greater than 10^{-4}; otherwise, it returns zero.

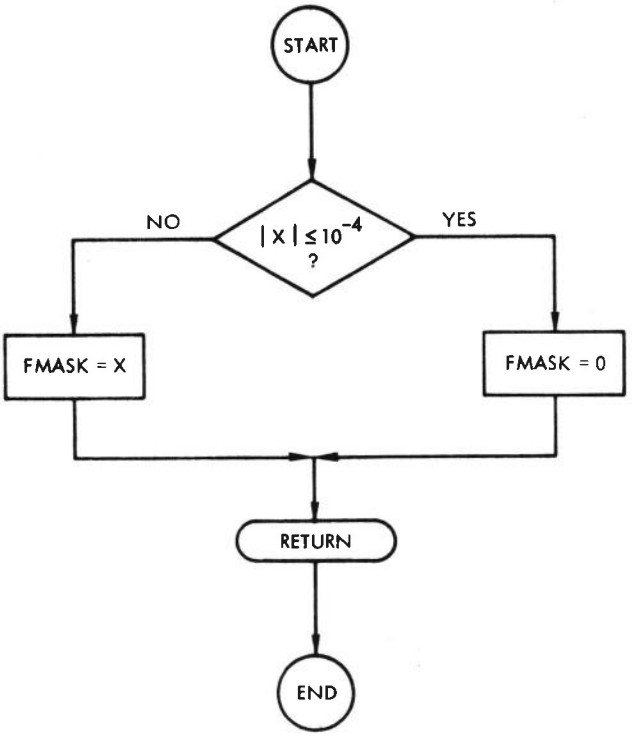
STORAGE

This routine requires 40_8 locations.

ARGUMENT

X

FLOWCHART



22. SUBROUTINE FREQR

DECK IDENTIFICATION

FREQR

PURPOSE

This routine computes the frequency response of a system given its transfer function and a table of frequencies. This routine will add frequencies to this table for portions of the frequency response curve that change most rapidly.

STORAGE

This routine requires 461_8 locations.

SUBROUTINE USAGE

ADDFRE

ADJUST

COMPUT

NAMED COMMON USAGE

ADJCOM

AFCOM

BASIC

COMPLX

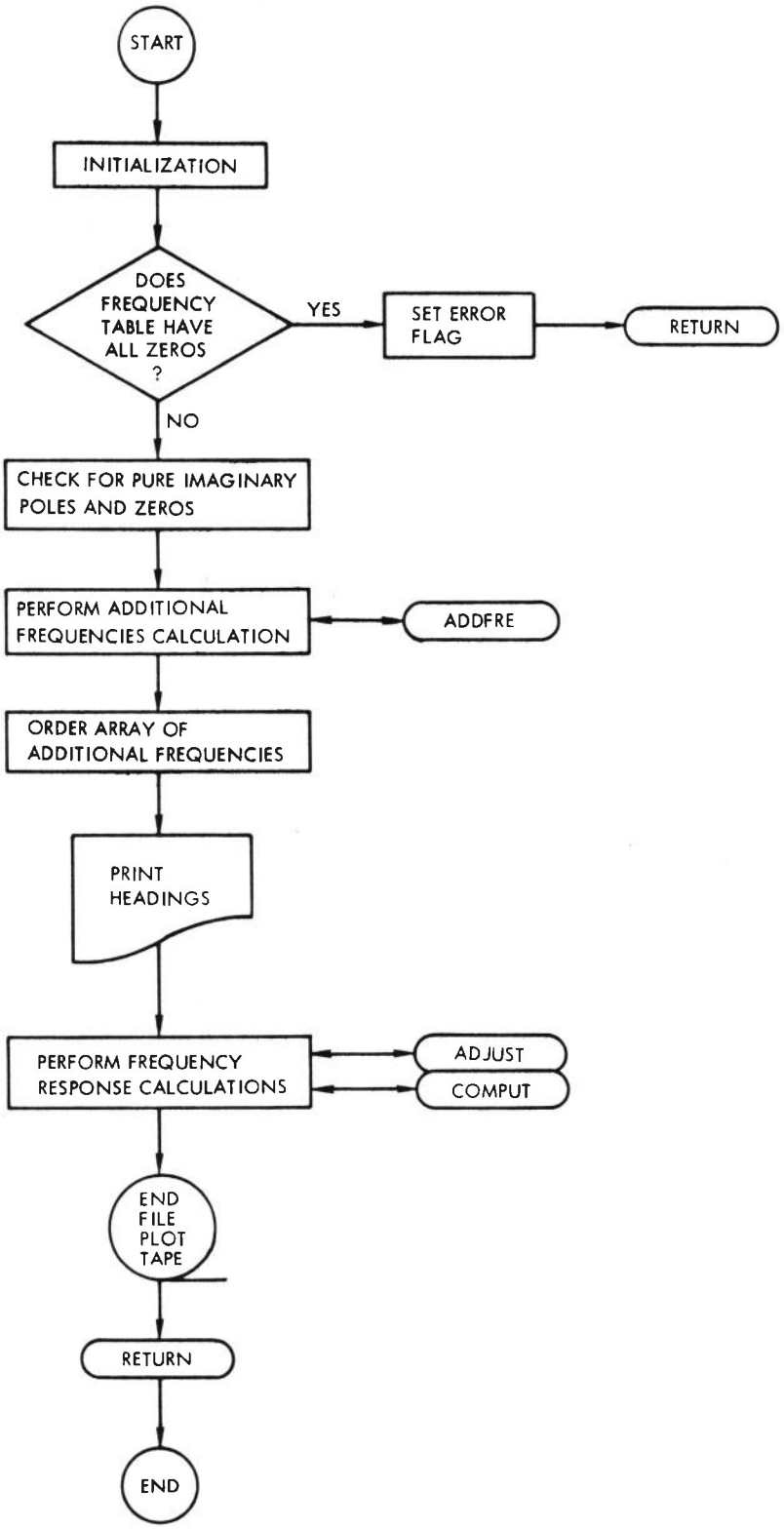
FRQRSP

OPTION

ARGUMENT

IERR - error flag

FLOWCHART


START
INITIALIZATION
DOES FREQUENCY TABLE HAVE ALL ZEROS ?
YES
SET ERROR FLAG
RETURN
NO
CHECK FOR PURE IMAGINARY POLES AND ZEROS
PERFORM ADDITIONAL FREQUENCIES CALCULATION
ADDFRE
ORDER ARRAY OF ADDITIONAL FREQUENCIES
PRINT HEADINGS
PERFORM FREQUENCY RESPONSE CALCULATIONS
ADJUST
COMPUT
END FILE PLOT TAPE
RETURN
END

23. SUBROUTINE GAMMA

DECK IDENTIFICATION

GAMMA

PURPOSE

This routine computes the ratio of the negative product of non-zero poles to zeros for the LSD Program.

STORAGE

This routine requires 220_8 locations.

ARGUMENTS

POLES - array of poles

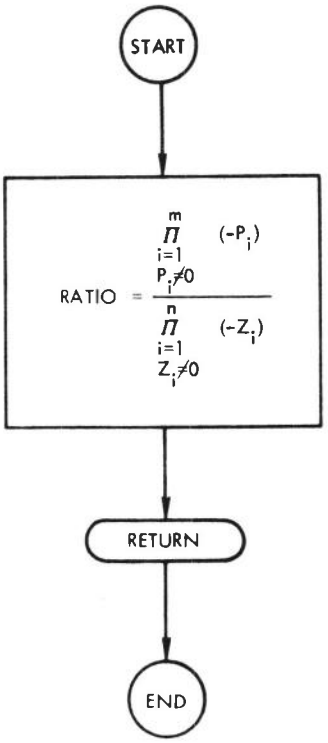
ZEROS - array of zeros

NPOLES - number of poles

NZEROS - number of zeros

RATIO - desired ratio

FLOW CHART


START
$$\text{RATIO} = \frac{\prod_{\substack{i=1 \\ P_i \neq 0}}^{m} (-P_i)}{\prod_{\substack{i=1 \\ Z_i \neq 0}}^{n} (-Z_i)}$$
RETURN
END

24. SUBROUTINE GCONJ

DECK IDENTIFICATION

GCONJ

PURPOSE

Given a table of complex numbers, this routine produces a second table of complex numbers whose entries are those of the first table and the complex conjugate of those entries of the first table whose imaginary parts are non-zero. These additional complex conjugates are entered into the second table immediately following the entries from the first table, to which they correspond. In general, the second table will be longer than the first table.

STORAGE

This routine requires 76_8 locations.

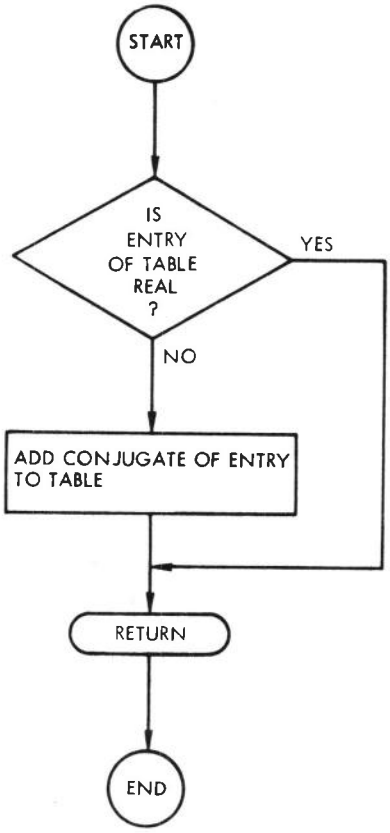
ARGUMENTS

TABIN - input table

TABOUT - output table

NPOINT - dimension of TABIN and TABOUT

FLOWCHART


START
IS ENTRY OF TABLE REAL ?
YES
NO
ADD CONJUGATE OF ENTRY TO TABLE
RETURN
END

25. SUBROUTINE ILT

DECK IDENTIFICATION

ILT

PURPOSE

This routine performs the time response calculations for the LSD Program.

STORAGE

This routine requires 1175_8 locations.

NAMED COMMON USAGE

BASIC

COMPLX

ILTCOM

INVLT

OPTION

ARGUMENTS

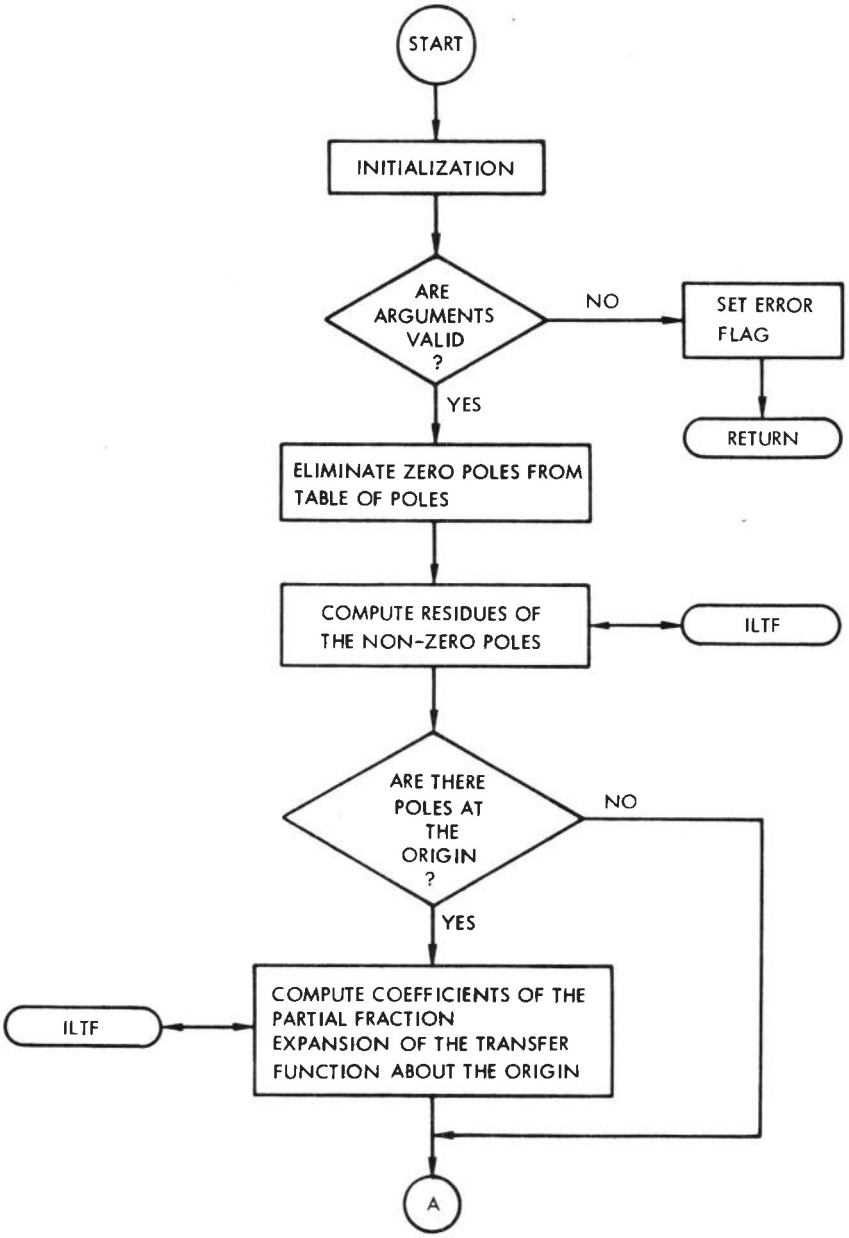
NPOLY - number of polynomial segments in driving function

TINF - array of driving function times

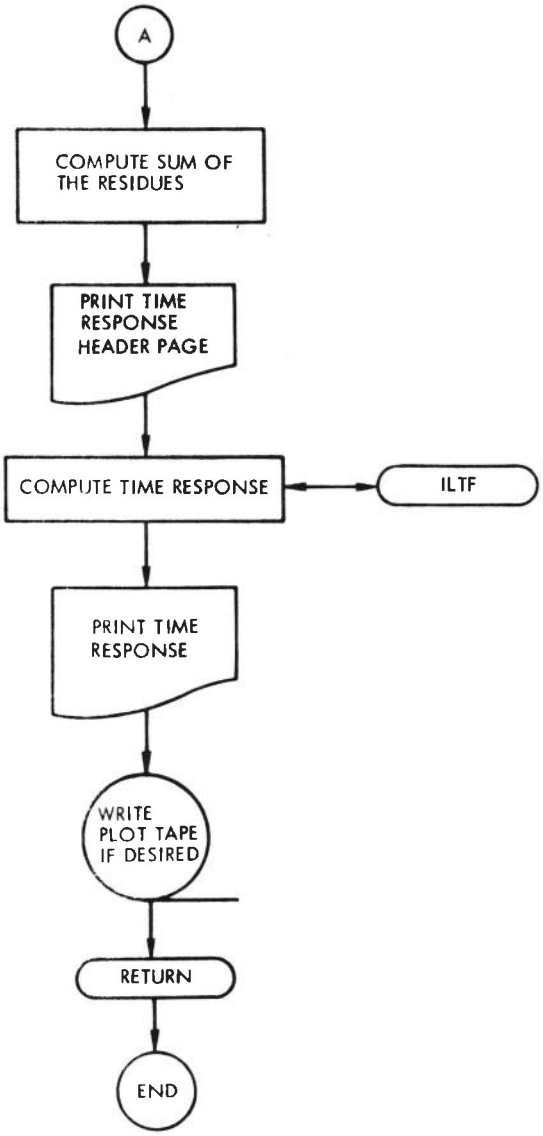
CINF - array of polynomial coefficients for driving function

IERR - error flag

FLOW CHART


START
INITIALIZATION
ARE ARGUMENTS VALID ?
NO
SET ERROR FLAG
RETURN
YES
ELIMINATE ZERO POLES FROM TABLE OF POLES
COMPUTE RESIDUES OF THE NON-ZERO POLES
ILTF
ARE THERE POLES AT THE ORIGIN ?
NO
YES
COMPUTE COEFFICIENTS OF THE PARTIAL FRACTION EXPANSION OF THE TRANSFER FUNCTION ABOUT THE ORIGIN
ILTF
A

FLOWCHART


A
COMPUTE SUM OF THE RESIDUES
PRINT TIME RESPONSE HEADER PAGE
COMPUTE TIME RESPONSE
ILTF
PRINT TIME RESPONSE
WRITE PLOT TAPE IF DESIRED
RETURN
END

26. SUBROUTINE ILTF

DECK IDENTIFICATION

ILTF

PURPOSE

Given the poles, zeros, and root locus gain that define a transfer function, this routine computes one point in the time response of the corresponding system. This routine also computes the residues of the poles of the transfer function and constructs a piece-wise polynomial driving function.

STORAGE

This routine requires 2151_8 locations.

SUBROUTINE USAGE

SER

SCALE

EMU

NAMED COMMON USAGE

BASIC

COMPLX

ILTCOM

ARGUMENTS

F - flag

ICG = 0, compute residues only
= 1, compute transfer function only
= 2, compute response only
= 3, compute driving function only

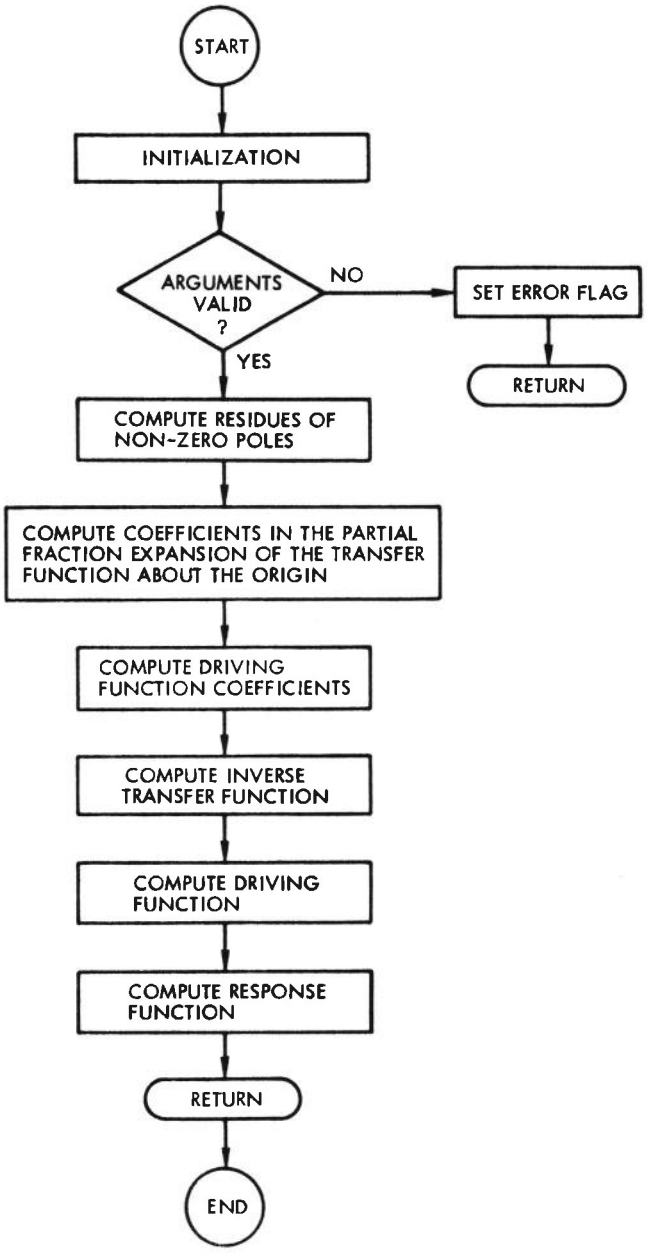
TS - array of driving function times

NN - number of polynomial segments

ALPHA - array of polynomial coefficients

IERR - error flag

FLOWCHART


START
INITIALIZATION
ARGUMENTS VALID ?
NO
SET ERROR FLAG
RETURN
YES
COMPUTE RESIDUES OF NON-ZERO POLES
COMPUTE COEFFICIENTS IN THE PARTIAL FRACTION EXPANSION OF THE TRANSFER FUNCTION ABOUT THE ORIGIN
COMPUTE DRIVING FUNCTION COEFFICIENTS
COMPUTE INVERSE TRANSFER FUNCTION
COMPUTE DRIVING FUNCTION
COMPUTE RESPONSE FUNCTION
RETURN
END

27. SUBROUTINE MATVAR

DECK IDENTIFICATION

MATVAR

PURPOSE

This routine accepts a real array, A, dimensioned (N1X, N1X, N2+1), and a complex constant, LAMBDA, and sets each element of a complex array, B, as follows:

$$B(I, J) = \sum_{K=1}^{N2+1} \lambda^{K-1} A(I, J, N2+2-K), \quad I, J = 1, \ldots, N1$$

STORAGE

This routine requires 344_8 locations.

ARGUMENTS

LAMBDA - λ

A

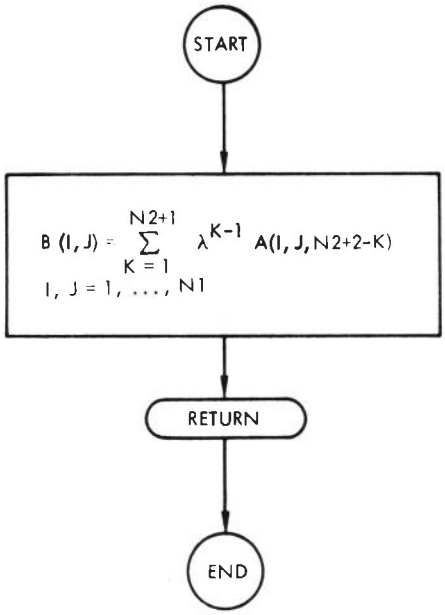
N1

N2

N1X

B

FLOWCHART



28. SUBROUTINE MTRXTR

DECK IDENTIFICATION

MTRXTR

PURPOSE

This routine performs the matrix printout for subroutine CDA of the LSD Program.

STORAGE

This routine requires 212_8 locations.

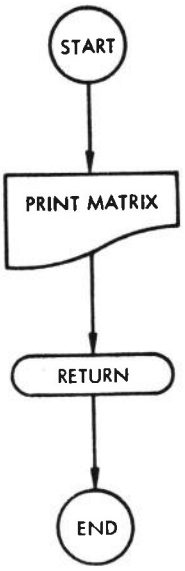
ARGUMENTS

A - matrix

M - number of rows

N - number of columns

FLOWCHART



29. SUBROUTINE ORDER

DECK IDENTIFICATION

ORDER

PURPOSE

This routine accepts an array of complex numbers and reorders it so that all zero elements occur first. Only a specified number of these leading zero elements will be retained.

STORAGE

This routine requires 207_8 locations.

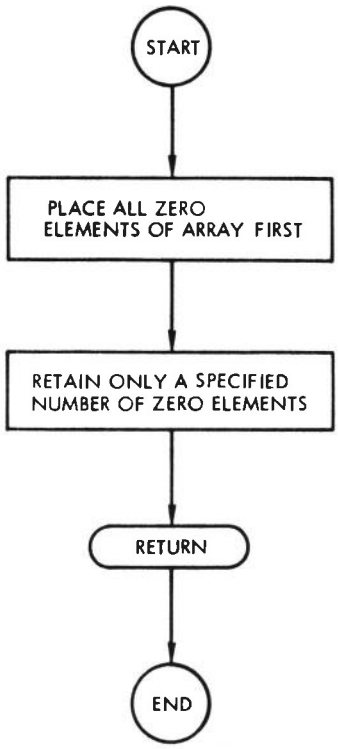
ARGUMENTS

ARRAY - array of complex numbers

NZER - number of leading zeros to be retained

NPOINT - dimension of array

FLOWCHART


START
PLACE ALL ZERO ELEMENTS OF ARRAY FIRST
RETAIN ONLY A SPECIFIED NUMBER OF ZERO ELEMENTS
RETURN
END

30. SUBROUTINE OUT

DECK IDENTIFICATION

OUT

PURPOSE

This routine performs the print output of eigenvalues and eigenvectors for the LSD subroutine EIGEN.

STORAGE

This routine requires 235_8 locations.

ARGUMENTS

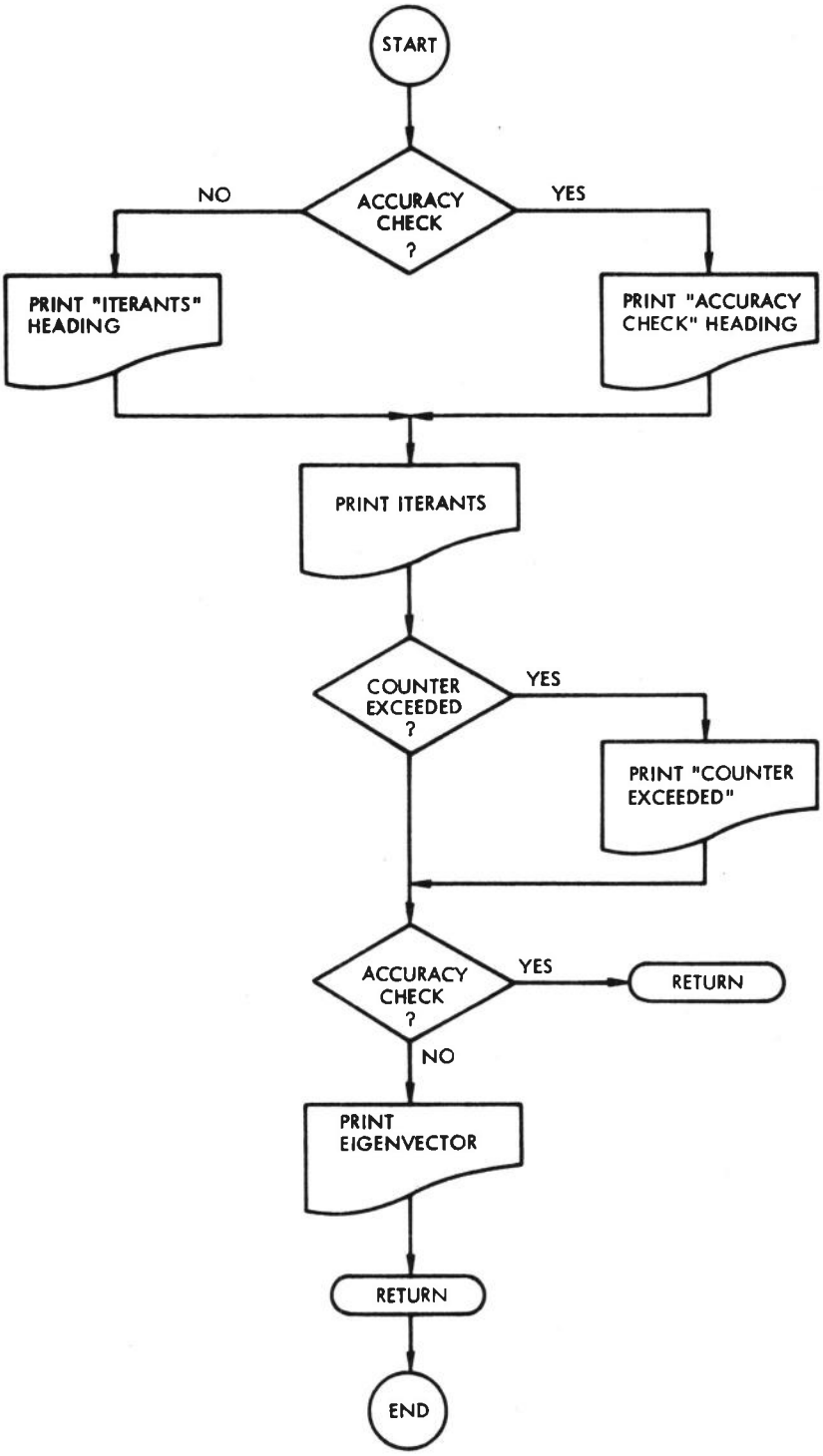
ITPAR - flag array

X - eigenvector

STOR - iterants and eigenvalues

KAP - dimension of X

FLOWCHART


START
ACCURACY CHECK ?
NO
YES
PRINT "ITERANTS" HEADING
PRINT "ACCURACY CHECK" HEADING
PRINT ITERANTS
COUNTER EXCEEDED ?
YES
PRINT "COUNTER EXCEEDED"
ACCURACY CHECK ?
YES
RETURN
NO
PRINT EIGENVECTOR
RETURN
END

31. SUBROUTINE ROOT

DECK IDENTIFICATION

ROOT

PURPOSE

This routine computes the roots of the root locus function for the LSD Program.

STORAGE

This routine requires 1540_8 locations.

LIBRARY USAGE

CSQRT

SUBROUTINE USAGE

ABC

AUXSUB

FMASK

ARGUMENTS

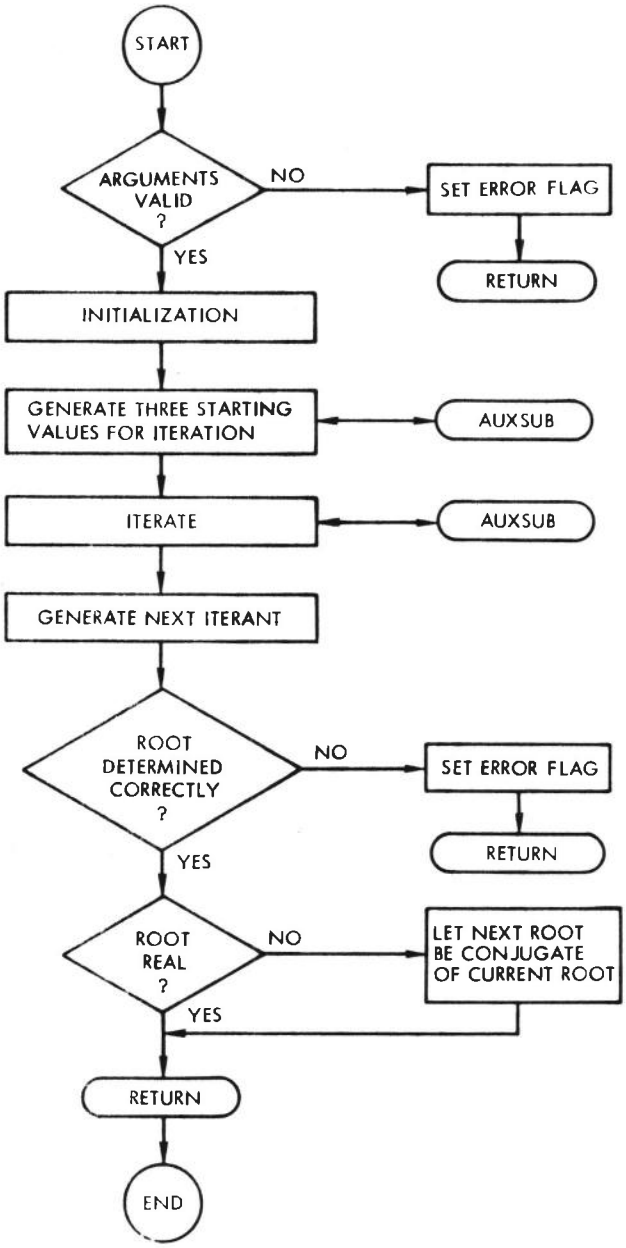
A - array of estimates for roots

Z - array of roots

NROOTS - number of roots

IERR - error flag

FLOWCHART


START
ARGUMENTS VALID ?
NO
SET ERROR FLAG
RETURN
YES
INITIALIZATION
GENERATE THREE STARTING VALUES FOR ITERATION
AUXSUB
ITERATE
AUXSUB
GENERATE NEXT ITERANT
ROOT DETERMINED CORRECTLY ?
NO
SET ERROR FLAG
RETURN
YES
ROOT REAL ?
NO
LET NEXT ROOT BE CONJUGATE OF CURRENT ROOT
YES
RETURN
END

32. SUBROUTINE RTLC

DECK IDENTIFICATION

RTLC

PURPOSE

This routine performs the root locus calculations for the LSD Program.

STORAGE

This routine requires 2254_8 locations.

LIBRARY USAGE

CSQRT

CABS

SUBROUTINE USAGE

GCONJ

ROOT

ANGLE

NAMED COMMON USAGE

BASIC

COMPLX

EXTRA

OPTION

ARGUMENTS

NCLPS - number of closed-loop poles

IPHASE - phase flag
= 0, 0° phase
= 1, 180° phase

TR1 - upper gain limit

TR2 - lower gain limit

TR3 - upper phase limit

TR4 - lower phase limit

ST1 - gain increment

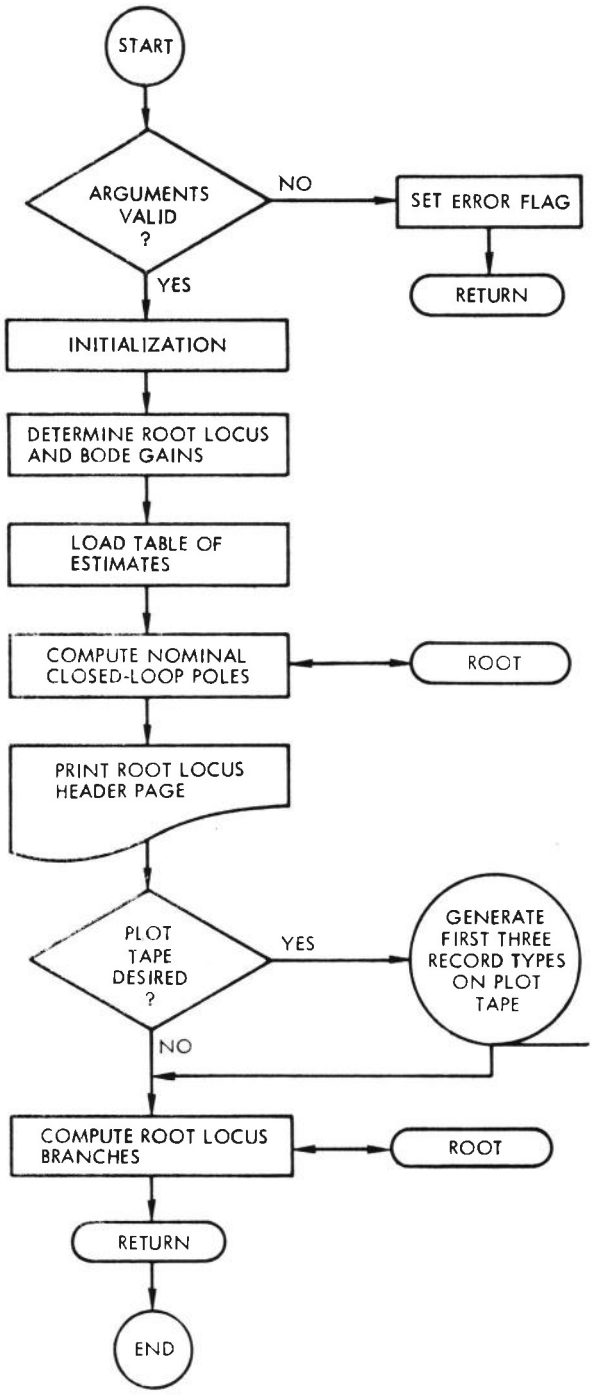
ST2 - gain decrement

ST3 - phase increment

ST4 - phase decrement

IERR - error flag

FLOWCHART


START
ARGUMENTS VALID ?
NO
SET ERROR FLAG
RETURN
YES
INITIALIZATION
DETERMINE ROOT LOCUS AND BODE GAINS
LOAD TABLE OF ESTIMATES
COMPUTE NOMINAL CLOSED-LOOP POLES
ROOT
PRINT ROOT LOCUS HEADER PAGE
PLOT TAPE DESIRED ?
YES
GENERATE FIRST THREE RECORD TYPES ON PLOT TAPE
NO
COMPUTE ROOT LOCUS BRANCHES
ROOT
RETURN
END

33. SUBROUTINE SCALE

DECK IDENTIFICATION

SCALE

PURPOSE

This routine accepts a complex argument, Z, and multiplies or divides it by 10^{10} so that the ABC(Z) is between 1 and 10^{10}. The argument J is incremented by 1 each time Z is divided by 10^{10} and is decremented by 1 each time Z is multiplied by 10^{10}.

STORAGE

This routine requires 155_8 locations.

SUBROUTINE USAGE

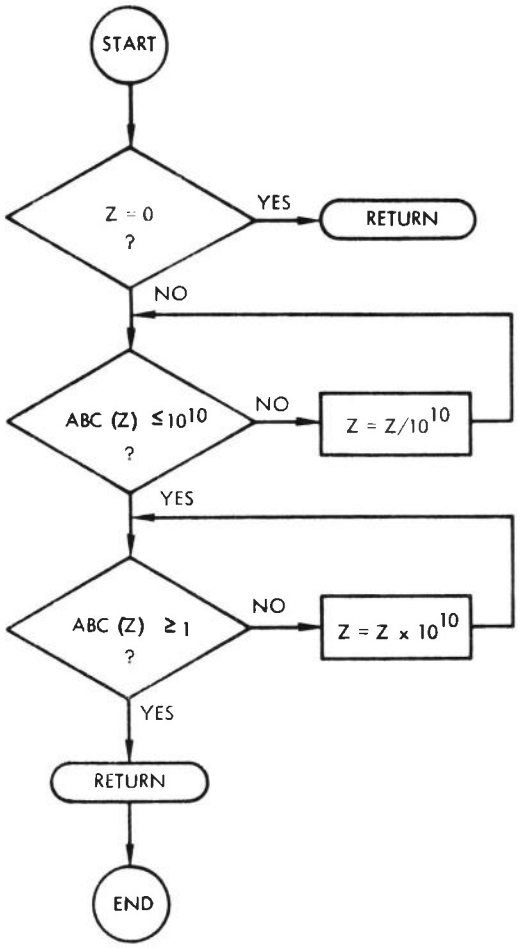
ABC

ARGUMENTS

Z

J

FLOWCHART


START
Z = 0 ?
YES
RETURN
NO
ABC (Z) ≤10^10 ?
NO
Z = Z/10^10
YES
ABC (Z) ≥1 ?
NO
Z = Z x 10^10
YES
RETURN
END

34. SUBROUTINE SER

DECK IDENTIFICATION

SER

PURPOSE

For a given complex argument, Z, this routine evaluates a certain polynomial function.

STORAGE

This routine requires 147_8 locations.

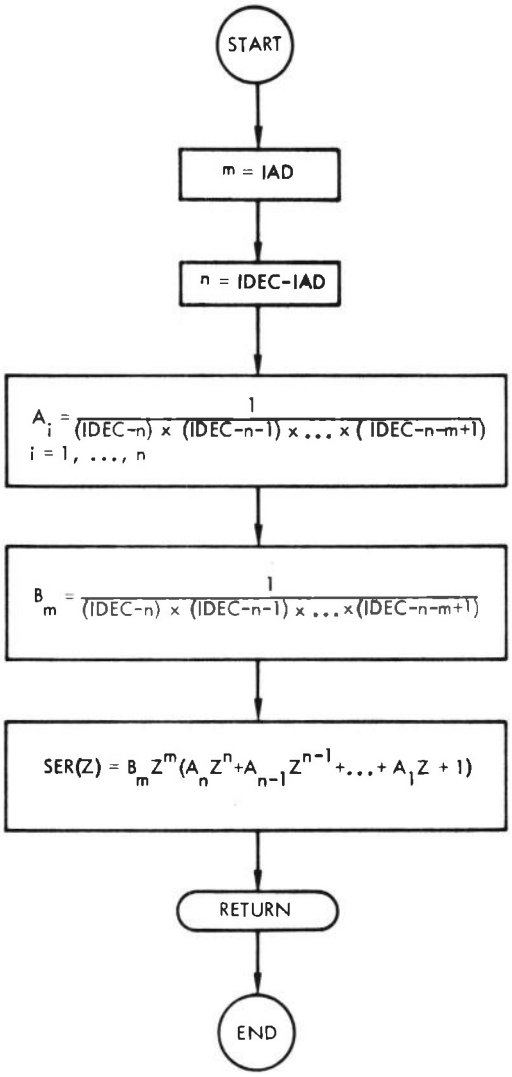
ARGUMENTS

Z

IAD

IDEC

FLOWCHART


START
m = IAD
n = IDEC-IAD
$A_i = \frac{1}{(IDEC-n) \times (IDEC-n-1) \times \ldots \times (IDEC-n-m+1)}$
i = 1, ..., n
$B_m = \frac{1}{(IDEC-n) \times (IDEC-n-1) \times \ldots \times (IDEC-n-m+1)}$
$SER(Z) = B_m Z^m (A_n Z^n + A_{n-1} Z^{n-1} + \ldots + A_1 Z + 1)$
RETURN
END

THE MAIN DRIVER, LSD DATE 101268 PAGE 1

03820843

```
& FOR LSD,LSD
UNIVAC 1108 FORTRAN V LEVEL   22C6 CC18 F5C18H
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03820843

   MAIN PROGRAM

   STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE   000354
        0000   *DATA   CC0742
        0002   *BLANK  CCCCCC
        0003   COMPLX  000764
        0004   DCDA    CC0335
        0005   DCMPLX  004374
        0006   FRQRSP  001037
        0007   INVLT   005372
        0010   MATDAT  000003
        0011   MTRX    C25060
        0012   OPTION  000030
        0013   RLOCUS  CCC106
        0014   SKALE   000002
        0015   TITLES  000106


   EXTERNAL REFERENCES (BLOCK, NAME)

        0016   QQUFOF
        0017   CDA
        0020   CDATE
        0021   CTIME
        0022   NREW$
        0023   NRNL$
        0024   NRDU$
        0025   NIO1$
        0026   NIO2$
        0027   NWDU$
        0030   NSTOP$
```

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001		000010	1L	0001		000266	10L	0001		000036	141G	0001		000044	143G	0001		000102	161G
0001		000110	163G	0000		000403	2F	0000		000521	20F	0001		000144	201G	0001		000153	205G
0001		000160	211G	0001		000213	226G	0001		000221	230G	0001		000241	241G	0001		000313	272G
0001		000315	275G	0000		000405	3F	0001		000275	30L	0001		000341	304G	0000		000421	4F
0000		000540	40F	0000		000423	5F	0000		000705	60F	0000		000464	8F	0000		000503	9F
0001		000246	98L	0001		000254	99L	0011	R	000000	AMAT	0004	R	000272	ASKB	0004	R	000263	ASKRL
0004	I	000245	ASNP	0004	I	000254	ASNZ	0005	C	001604	ASP	0005	C	003100	ASZ	0011	R	003100	BMAT
0000	I	000000	CLEAR	0003	C	000620	CLPOLE	0011	R	006200	CMAT	0011		000000	COMAT	0012	I	000025	COMENT
0004	R	000000	D	0013	R	000016	DBJOwN	0013	R	000007	DBUP	001C	I	000001	DEGREE	0007	R	000002	DELTAT
0011	R	011300	DMAT	0003	R	000000	DUMMY1	0011	R	000000	DUMMY2	0011	R	014400	EMAT	0005	C	000000	EST
0007	R	000001	FINALT	0011	R	017500	FMAT	0006	R	000000	FREQ	0015	R	000000	HEADER	0000	I	000025	I
0000	I	000023	ICNT	0012	I	000027	INPUT	0000	I	000027	I$	0000	I	000026	J	0000	I	000024	JCNT
0004	R	000310	LASTCO	0004	R	000301	LEADCO	0011	R	000000	M10	0011	R	000000	M20	0011	R	000000	M30
0011	R	000000	M40	0011	R	000000	M50	0011	R	000000	M60	000C	I	000021	N	0000	I	000001	NARRAY
0013	I	000077	NCLPOL	0004	I	000223	NFAKES	0000	I	000017	NOVFLS	0004	I	000232	NROOTS	0000	I	000015	NTEMP
0000	I	000020	NUNFLS	000C	I	000022	N6$	0010	I	000000	ORDER	0013	I	000000	PHASE	0013	R	000034	PHDOWN
0013	R	000025	PHUP	0012	I	000026	PLOT	0007	R	001306	POLYC	0007	I	000003	POLYN	0007	R	000012	POLYT
0004	I	000317	RATIO	0014	R	000000	SCALE1	0014	R	000001	SCALE2	0010	I	000002	SELECT	0012	I	000007	SFREQR
0004	R	000244	SKB	0004	R	000243	SKRL	0004	I	000241	SNP	0004	I	000242	SNZ	0005	C	001274	SP
0012	I	000000	SROOTL	0007	R	000000	STARTT	0013	R	000052	STEPDD	0013	R	000043	STEPDU	0013	R	000070	STEPPD
0013	R	000061	STEPPU	0012	I	000016	STIMER	0005	C	001440	SZ								


```
00100     1*   C
00100     2*   C ***************************************************************************
00100     3*   C .LINEAR SYSTEMS DYNAMICS (LSD) PROGRAM -- MODIFIED BY C. ARGILA, OCT. '68    .
00100     4*   C .                                                                            .
00100     5*   C .   DESCRIPTION-                                                            .
00100     6*   C .      THIS VERSION OF THE LINEAR SYSTEMS DYNAMICS PROGRAM COMPUTES THE      .
00100     7*   C .   FREQUENCY RESPONSE, TIME RESPONSE AND ROOT LOCUS OF A NETWORK GIVEN      .
00100     8*   C .   ITS S-PLANE POLES, ZEROS AND GAIN, OR A MATRIX WHICH DESCRIBES THE       .
00100     9*   C .   NETWORK.  THE LSD PROGRAM CAN GENERATE A PLOT TAPE TO BE USED BY THE     .
00100    10*   C .   TRWPLT GENERALIZED PLOT PROGRAM.                                         .
00100    11*   C .                                                                            .
00100    12*   C .   INPUT-                                                                  .
00100    13*   C .      INPUT IS VIA THE NAME LIST 'DATA.'  ALL INPUT PARAMETERS ARE          .
```

```
00100    14*   C .   INITIALIZED ZERO UNLESS OTHERWISE STATED.  ALL INPUT PARAMETERS RETAIN   .
00100    15*   C .   THEIR VALUE BETWEEN DATA CASES UNLESS CHANGED BY THE USER.               .
00100    16*   C .   GENERAL INPUT-                                                          .
00100    17*   C .      INPUT      = 1, A MATRIX IS INPUT                                    .
00100    18*   C .                 = 2, S-PLANE POLES, ZEROS AND GAIN ARE INPUT              .
00100    19*   C .                 = 4, A MATRIX IS INPUT, EIGEN VALUE SOLUTION ONLY         .
00100    20*   C .                   INPUT IS INITIALIZED TO 2                               .
00100    21*   C .      CLEAR      = 0, NO EFFECT                                            .
00100    22*   C .                 = 1, THE ENTIRE 10800 CELLS OF THE MATRIX ARE ZEROED      .
00100    23*   C .      COMENT     = 0, NO EFFECT                                            .
00100    24*   C .                 = N, NUMBER OF HEADER CARDS FOLLOWING THIS NAMELIST       .
00100    25*   C .                    (N=1,2,3,4,5)                                          .
00100    26*   C .      PLOT       = 0, NO EFFECT                                            .
00100    27*   C .                 = 1, PLOT TAPE WILL BE GENERATED ON UNIT 8                .
00100    28*   C .      SFREQR(J)  = 0, NO EFFECT                                            .
00100    29*   C .                 = 1, FREQUENCY RESPONSE WILL BE CALCULATED FOR THE J-TH   .
00100    30*   C .                    RATIO (J=1,2,...,7)                                    .
00100    31*   C .      SRCOTL(J)  = 0, NO EFFECT                                            .
00100    32*   C .                 = 1, ROOT LOCUS WILL BE CALCULATED FOR THE J-TH RATIO     .
00100    33*   C .      STIMER(J)  = 0, NO EFFECT                                            .
00100    34*   C .                 = 1, TIME RESPONSE WILL BE CALCULATED FOR THE J-TH RATIO  .
00100    35*   C .   S-PLANE INPUT-                                                          .
00100    36*   C .      SNP        NUMBER OF POLES                                           .
00100    37*   C .      SNZ        NUMBER OF ZEROS                                           .
00100    38*   C .      SP(I)      THE POLES, IN COMPLEX FORM (I=1,2,...,50)                 .
00100    39*   C .      SZ(I)      THE ZEROS                                                 .
00100    40*   C .      SKRL       ROOT LOCUS GAIN                                           .
00100    41*   C .      SKB        BODE GAIN                                                 .
00100    42*   C .      ASNP(J)    NUMBER OF ADDITIONAL POLES FOR THE J-TH RATIO             .
00100    43*   C .      ASNZ(J)    NUMBER OF ADDITIONAL ZEROS FOR THE J-TH RATIO             .
00100    44*   C .      ASP(I,J)   ADDITIONAL POLES FOR THE J-TH RATIO                       .
00100    45*   C .      ASZ(I,J)   ADDITIONAL ZEROS FOR THE J-TH RATIO                       .
00100    46*   C .      ASKRL(J)   ADDITIONAL ROOT LOCUS GAIN FOR THE J-TH RATIO             .
00100    47*   C .      ASKB(J)    ADDITIONAL BODE GAIN FOR THE J-TH RATIO                   .
00100    48*   C .   EIGENVALUE INPUT-                                                       .
00100    49*   C .      D(I,J)     CONTAIN THE I LAST ROW ELEMENTS FOR THE J-TH DETERMINANT  .
00100    50*   C .                 (I= 1,2,...,20)                                         .
00100    51*   C .      RATIO(1,J) NUMBER OF THE DETERMINANT TO BE USED FOR THE NUMERATOR   .
00100    52*   C .                 OF THE J-TH RATIO                                       .
00100    53*   C .      RATIO(2,J) NUMBER OF THE DETERMINANT TO BE USED FOR THE DENOMINATOR .
00100    54*   C .                 OF THE J-TH RATIO                                       .
```

```
00100    55*   C .      NROOTS(J) IS THE MAXIMUM NUMBER OF ROOTS TO BE FOUND FOR THE J-TH      .
00100    56*   C .                DETERMINANT                                                  .
00100    57*   C .      NFAKES(J) NUMBER OF ROOTS ALREADY KNOWN FOR THE J-TH DETERMINANT       .
00100    58*   C .      EST(I,J)  (COMPLEX) ESTIMATE FOR THE I-TH ROOT OF THE J-TH DETERMINANT.
00100    59*   C .      LEADCO(J) LEADING COEFFICIENT OF THE POLYNOMIAL EXPANSION OF THE       .
00100    60*   C .                J-TH DETERMINANT                                             .
00100    61*   C .      LASTCO(J) LAST NON-ZERO COEFFICIENT OF THE POLYNOMIAL EXPANSION OF     .
00100    62*   C .                THE J-TH DETERMINANT                                         .
00100    63*   C .      SCALE1    SCALE FACTOR TO PREVENT OVERFLOW/UNDERFLOW                   .
00100    64*   C .                SCALE1 IS INITIALIZED TO 1.0                                 .
00100    65*   C .      SCALE2    SCALE FACTOR TO PREVENT OVERFLOW/UNDERFLOW                   .
00100    66*   C .                SCALE2 IS INITIALIZED TO 1.0                                 .
00100    67*   C .   MATRIX INPUT-                                                            .
00100    68*   C .      ORDER     ORDER OF THE MATRIX                                          .
00100    69*   C .      DEGREE    HIGHEST DEGREE OF THE POLYNOMIAL TERMS                       .
00100    70*   C .      SELECT    'NOMINAL' ORDER OF THE MATRIX (10,20,...,60)                 .
00100    71*   C .      AMAT(L,M) MATRIX FOR K-TH DEGREE TERMS (L,M=1,2,...,40)                .
00100    72*   C .      BMAT(L,M) MATRIX FOR (K-1)-TH DEGREE TERMS                             .
00100    73*   C .      CMAT(L,M) MATRIX FOR (K-2)-TH DEGREE TERMS                             .
00100    74*   C .      DMAT(L,M) MATRIX FOR (K-3)-TH DEGREE TERMS                             .
00100    75*   C .      EMAT(L,M) MATRIX FOR (K-4)-TH DEGREE TERMS                             .
00100    76*   C .      FMAT(L,M) MATRIX FOR (K-5)-TH DEGREE TERMS                             .
00100    77*   C .      M10(L,M,N) MATRIX OF 'NOMINAL' ORDER 10 AND DEGREE 107                 .
00100    78*   C .      M20(L,M,N) MATRIX OF 'NOMINAL' ORDER 20 AND DEGREE 24                  .
00100    79*   C .      M30(L,M,N) MATRIX OF 'NOMINAL' ORDER 30 AND DEGREE 11                  .
00100    80*   C .      M40(L,M,N) MATRIX OF 'NOMINAL' ORDER 40 AND DEGREE 5                   .
00100    81*   C .      M50(L,M,N) MATRIX OF 'NOMINAL' ORDER 50 AND DEGREE 3                   .
00100    82*   C .      M60(L,M,N) MATRIX OF 'NOMINAL' ORDER 60 AND DEGREE 2                   .
00100    83*   C .   FREQUENCY RESPONSE INPUT-                                                .
00100    84*   C .      FREQ(I)   TABLE OF FREQUENCIES (I=1,2,...,543)                         .
00100    85*   C .                FREQ IS INITIALIZED TO A STANDARD TABLE                      .
00100    86*   C .   ROOT LOCUS INPUT-                                                        .
00100    87*   C .      PHASE(J)  = 0, ROOT LOCUS FOR THE J-TH RATIO WILL BE ZERO DEGREE PHASE.
00100    88*   C .                = 1, PHASE WILL BE 180 DEGREES                               .
00100    89*   C .      DBUP(J)   = 0, TRACE WILL NOT BE PERFORMED FOR THE J-TH RATIO          .
00100    90*   C .                .NE. 0, TRACE WILL BE PERFORMED FROM 0 UP TO DBUP            .
00100    91*   C .      DBDOWN(J) DECREASING GAIN ROOT LOCUS (SEE DBUP)                        .
00100    92*   C .      PHUP(J)   INCREASING PHASE ROOT LOCUS (SEE DBUP)                       .
00100    93*   C .      PHDOWN(J) DECREASING PHASE ROOT LOCUS (SEE DBUP)                       .
00100    94*   C .      STEPDU(J) MAXIMUM GAIN STEP SIZE FOR DBUP(J) (DB)                      .
00100    95*   C .      STEPDD(J) MAXIMUM GAIN STEP SIZE FOR DBDOWN(J) (DB)                    .
```

```
001C0    96*   C .      STEPPU(J) MAXIMUM PHASE STEP SIZE FOR PHUP(J) (DEGREES)                .
00100    97*   C .      STEPPD(J) MAXIMUM PHASE STEP SIZE FOR PHDOWN(J) (DEGREES)              .
00100    98*   C .      NCLPOL(J) NUMBER OF CLOSED LOOP POLES, FOR THE J-TH RATIO, FOR WHICH   .
00100    99*   C .                THE ROOT LOCUS IS DESIRED                                    .
00100   100*   C .      CLPOLE(I) (COMPLEX) ESTIMATES OF THE CLOSED LOOP POLES (I=1,2,...,50)  .
001C0   101*   C .   TIME RESPONSE INPUT-                                                      .
00100   102*   C .      STARTT    START TIME FOR OUTPUT RESPONSE                               .
00100   103*   C .      DELTAT    TIME INCREMENT                                               .
00100   104*   C .      FINALT    MAXIMUM TIME FOR OUTPUT RESPONSE                             .
00100   105*   C .      POLYN(J)  NUMBER OF POLYNOMIAL SEGMENTS IN THE INPUT FUNCTION FOR THE  .
00100   106*   C .                J-TH RATIO (POLYN(J)=1,2,...,99)                             .
00100   107*   C .      POLYT(I,J) IS THE I-TH DISCRETE TIME FOR THE INPUT FUNCTION FOR THE    .
00100   108*   C .                J-TH RATIO (I=1,2,...,100)                                   .
00100   109*   C .      POLYC(I,J) IS THE I-TH COEFFICIENT FOR THE INPUT POLYNOMIAL FOR THE    .
00100   110*   C .                J-TH RATIO (I=1,2,...,300)                                   .
00100   111*   C .                                                                             .
00100   112*   C .   OUTPUT-                                                                   .
00100   113*   C .      PRINTED OUTPUT IS BEGUN BY A HEADER PAGE, FOLLOWED BY A DESCRIPTION    .
00100   114*   C .   OF THE INPUT, FOLLOWED BY THE ROOT LOCUS, FREQUENCY RESPONSE AND TIME     .
00100   115*   C .   RESPONSE CALCULATIONS.  THE FREQUENCY RESPONSE AND TIME RESPONSE PORTIONS.
00100   116*   C .   OF THE PROGRAM EACH GENERATE PRECISELY ONE FILE ON THE PLOT TAPE.  EACH   .
00100   117*   C .   RECORD WITHIN EACH FILE CORRESPONDS TO A LINE OF OUTPUT.  FOR THE ROOT    .
00100   118*   C .   LOCUS PORTION OF THE PROGRAM, N+4 RECORD TYPES ARE GENERATED WITHIN ONE   .
00100   119*   C .   FILE OF THE PLOT TAPE.  RECORD TYPES 1 CONTAIN ALL ABSCISSA POINTS.       .
00100   120*   C .   RECORD TYPES 2-4 CONTAIN, RESPECTIVELY, THE ORDINATE POINTS OF THE ZEROS,.
00100   121*   C .   OPEN-LOOP POLES AND NOMINAL CLOSED-LOOP POLES.  RECORD TYPES 5,...,N+4   .
00100   122*   C .   CONTAIN THE ORDINATE POINTS OF THE N ROOT LOCUS BRANCHES, IN THE ORDER    .
00100   123*   C .   THEY ARE PRINTED OUT.                                                     .
00100   124*   C .                                                                             .
00100   125*   C .   NAMED COMMON USAGE-                                                       .
00100   126*   C .      COMPLX                                                                 .
00100   127*   C .      DCDA                                                                   .
00100   128*   C .      DCMPLX                                                                 .
00100   129*   C .      FRQRSP                                                                 .
00100   130*   C .      INVLT                                                                  .
00100   131*   C .      MATDAT                                                                 .
00100   132*   C .      MTRX                                                                   .
00100   133*   C .      OPTION                                                                 .
00100   134*   C .      RLOCUS                                                                 .
00100   135*   C .      SKALE                                                                  .
00100   136*   C .      TITLES                                                                 .
```

```
00100   137*   C .                                                                  .
00100   138*   C .   SUBROUTINE USAGE-                                              .
00100   139*   C .      CDA                                                         .
00100   140*   C .      QQUFOF                                                      .
00100   141*   C .      CDATE                                                       .
00100   142*   C .      CTIME                                                       .
00100   143*   C .                                                                  .
00100   144*   C ....................................................................
00100   145*   C
00101   146*         INTEGER  ASNP   , ASNZ   , CLEAR , COMENT, DEGREE, ORDER , PHASE ,
00101   147*        $ PLOT   , POLYN , RATIO , SELECT, SFREQR, SNP   , SNZ   , SROOTL,
00101   148*        $ STIMER
00103   149*         REAL     LASTCO, LEADCO, M10   , M2C   , M30   , M40   , M5C   ,
00103   150*        $ M60
00104   151*         COMPLEX  ASP   , ASZ   , CLPOLE, EST   , SP    , SZ
00105   152*         COMMON /COMPLX/  DUMMY1(400)   , CLPOLE(5C)
00106   153*         COMMON /DCDA   /  D(21,7)      , NFAKES(7)     , NROOTS(7)     ,
00106   154*        $ SNP   , SNZ   , SKRL  , SKB   , ASNP(7)      , ASNZ(7)       ,
00106   155*        $ ASKRL(7)      , ASKB(7)      , LEADCO(7)     , LASTCO(7)     ,
00106   156*        $ RATIO(2,7)
00107   157*         COMMON /DCMPLX/  EST(5C,7)     , SP(50), SZ(5C), ASP(50,7)     ,
00107   158*        $ ASZ(50,7)
00110   159*         COMMON /FRQRSP/  FREQ(543)
00111   160*         COMMON /INVLT /  STARTT, FINALT, DELTAT, POLYN(7)      , POLYT(100
00111   161*        $,7)    , POLYC(300,7)
00112   162*         COMMON /MATDAT/  ORDER , DEGREE, SELECT
00113   163*         COMMON /MTRX   /  DUMMY2(13800)
00114   164*         COMMON /OPTION/  SROOTL(7)     , SFREQR(7)     , STIMER(7)     ,
00114   165*        $ COMENT, PLOT   , INPUT
00115   166*         COMMON /RLOCUS/  PHASE(7)      , DBUP(7)       , DBDOWN(7)     ,
00115   167*        $ PHUP(7)       , PHDOWN(7)     , STEPDU(7)     , STEPDD(7)     ,
00115   168*        $ STEPPU(7)     , STEPPD(7)     , NCLPOL(7)
00116   169*         COMMON /SKALE /  SCALE1, SCALE2
00117   170*         COMMON /TITLES/  HEADER(14,5)
00120   171*         DIMENSION        AMAT(4C,40)   , BMAT(40,40)   , COMAT(40,40,6),
00120   172*        $ CMAT(40,40)   , DMAT(40,40)   , EMAT(40,40)   , FMAT(40,40)   ,
00120   173*        $ M10(10,1C,108), M2C(2C,2C,25) , M3C(30,3C,12) , M40(40,40,6)  ,
00120   174*        $ M50(50,5C,4)  , M60(6C,6C,3)  , NARRAY(6,2)   , NTEMP(2)
00121   175*         EQUIVALENCE      (AMAT,COMAT,DUMMY2,M1C,M2C,M3C,M40,M50,M6C)    ,
00121   176*        $ (BMAT,COMAT(1,1,2))           , (CMAT,COMAT(1,1,3))           ,
00121   177*        $ (DMAT,COMAT(1,1,4))           , (EMAT,COMAT(1,1,5))           ,
```

```
00121   178*        $ (FMAT,COMAT(1,1,6))
00122   179*         NAME LIST /DATA/ INPUT , CLEAR , COMENT, PLOT  , SFREQR, SROOTL,
00122   180*        $ STIMER, SNP   , SNZ   , SP    , SZ    , SKRL  , SKB   , ASNP  ,
00122   181*        $ ASNZ  , ASP   , ASZ   , ASKRL , ASKB  , D     , RATIO , NROOTS,
00122   182*        $ NFAKES, EST   , LEADCO, LASTCO, SCALE1, SCALE2, ORDER , DEGREE,
00122   183*        $ SELECT, AMAT  , BMAT  , CMAT  , DMAT  , EMAT  , FMAT  , M1C   ,
00122   184*        $ M2C   , M3C   , M4C   , M5C   , M6C   , FREQ  , PHASE , DBUP  ,
00122   185*        $ DBDOWN, PHUP  , PHDOWN, STEPDO, STEPDD, STEPPU, STEPPD, NCLPOL,
00122   186*        $ CLPOLE, STARTT, DELTAT, FINALT, POLYN , POLYT , POLYC , NOVFLS,
00122   187*        $ NUNFLS
00123   188*         DATA NOVFLS, NUNFLS, N, N6C /1CC, C, 0, C60/
00123   189*    C
00123   190*    C *** ESTABLISH OVERFLOW/UNDERFLOW MONITOR
00130   191*          CALL QQUFOF(ICNT,JCNT,C)
00130   192*    C
00130   193*    C *** REWIND PLOT TAPE
00131   194*          REWIND 8
00131   195*    C
00131   196*    C *** READ INPUT DATA
00132   197*        1 READ(5,DATA,ERR=98,END=99)
00132   198*    C
00132   199*    C *** READ COMMENT/DESCRIPTION CARDS
00135   200*          COMENT = MINO(5,COMENT)
00136   201*          IF(COMENT .NE. C)
00136   202*         $READ(5,2) ((HEADER(I,J),I=1,14),J=1,COMENT)
00150   203*        2 FORMAT(13A6,A2)
00150   204*    C
00150   205*    C *** PRINT PAGE HEADING
00151   206*          N = N + 1
00152   207*          WRITE(6,3) N
00155   208*        3 FORMAT(1H1,39X,31HLINEAR SYSTEMS DYNAMICS PROGRAM//47X,13HDATA CAS
00155   209*         $E NO.I3/)
00156   210*          IF (COMENT .NE. C)
00156   211*         $WRITE(6,4) ((HEADER(I,J),I=1,14),J=1,COMENT)
00170   212*        4 FORMAT(15X 13A6,A2)
00171   213*          WRITE(6,5) CLEAR, COMENT, INPUT, NUNFLS, NOVFLS, PLOT, SFREQR,
00171   214*         $ SROOTL, STIMER
00215   215*        5 FORMAT(/51X,7HOPTIONS
00215   216*         $/41X,8HCLEAR  =I3
00215   217*         $/41X,8HCOMENT =I3
00215   218*         $/41X,8HINPUT  =I3
```

```
00215   219*         $/41X,8HNUNFLS =I3
00215   220*         $/41X,8HNOVFLS =I3
00215   221*         $/41X,8HPLOT   =I3
00215   222*         $/41X,8HSFREQR =I3,6(1H,I2)
00215   223*         $/41X,8HSROUTL =I3,6(1H,I2)
00215   224*         $/41X,8HSTIMER =I3,6(1H,I2))
00215   225*    C
00215   226*    C *** SET NUMBER OF UNDERFLOW AND OVERFLOW MESSAGES
00216   227*          ICNT = NUNFLS - 1
00217   228*          JCNT = NOVFLS - 1
00217   229*    C
00217   230*    C *** PERFORM CALCULATIONS
00220   231*          WRITE(6,3) N
00223   232*          IF(COMENT .NE. 0)
00223   233*         $WRITE(6,4) ((HEADER(I,J),I=1,14),J=1,COMENT)
00235   234*          CALL CDA
00235   235*    C
00235   236*    C *** CLEAR MATRIX
00236   237*          IF(CLEAR .EQ. 0)
00236   238*         $GO TO 1
00240   239*          DO 6 I = 1, 10800
00243   240*        6 DUMMY2(I) = 0.0
00243   241*    C
00243   242*    C *** PROCESS ANOTHER DATA CASE
00245   243*          GO TO 1
00245   244*    C
00245   245*    C *** PRINT MESSAGE IF NAME LIST ERROR WAS ENCOUNTERED
00246   246*       98 WRITE(6,8)
00250   247*        8 FORMAT(84H1AN ERROR IN THE NAME LIST '$DATA' WAS ENCOUNTERED CAUSI
00250   248*         $NG ONE DATA CASE TO BE LOST.)
00251   249*          GO TO 1
00251   250*    C
00251   251*    C *** TERMINATE EXECUTION
00252   252*       99 IF(N .NE. 1)
00252   253*         $GO TO 10
00254   254*          WRITE(6,9)
00256   255*        9 FORMAT(79H1END OF LSD EXECUTION.  NORMAL TERMINATION.  ONE DATA CA
00256   256*         $SE PROCESSED THIS TRIP.)
00257   257*          GO TO 30
00260   258*       10 WRITE(6,20) N
00263   259*       20 FORMAT(43H1END OF LSD EXECUTION.  NORMAL TERMINATION.I4,32H DATA C
```

```
00263   260*        $ASES PROCESSED THIS TRIP.)
00264   261*       30 WRITE(6,40)
00266   262*       40 FORMAT(1HO,50X,7HX  X  X / 48X,1HX,5X,1HX,5X,1HX /46X,1HX,7X,1HX,
00266   263*         $7X,1HX / 44X,1HX,9X,1HX,9X,1HX / 43X,1HX,10X,1HX,10X,1HX / 42X,1HX
00266   264*         $11X,1HX,11X,1HX / 41X,1HX,12X,1HX,12X,1HX / 40X,1HX,13X,1HX,13X,1H
00266   265*         $X / 54X,1HX / 39X,1HX,14X,1HX,14X,1HX / 54X,1HX / 38X,1HX,15X,1HX,
00266   266*         $15X,1HX / 3(54X,1HX / 37X,1HX,16X,1HX,16X,1HX /) 53X,3HX X / 37X,
00266   267*         $1HX,14X,5HX   X,14X,1HX / 51X,7HX     X / 37X,1HX,12X,1HX,7X,1HX,
00266   268*         $12X,1HX / 49X,1HX,9X,1HX / 38X,1HX,9X,1HX,11X,1HX,9X,1HX / 47X,1HX
00266   269*         $13X,1HX / 39X,1HX,6X,1HX,15X,1HX,6X,1HX / 45X,1HX,17X,1HX / 40X,
00266   270*         $5HX   X,19X,5HX   X / 41X,3HX X,21X,3HX X / 42X,1HX,23X,1HX /43X,
00266   271*         $1HX,21X,1HX / 44X,1HX,19X,1HX / 46X,1HX,15X,1HX / 49X,1HX,9X,1HX /
00266   272*         $52X,5HX X X)
00267   273*          CALL CDATE(NTEMP(1))
00270   274*          CALL CTIME(NTEMP(2))
00271   275*          DO 50 J = 1, 2
00274   276*          DO 50 I = 1, 6
00277   277*       50 NARRAY(I,J) = FLD(6*I-6,6,NTEMP(J)) - N60
00302   278*          WRITE(6,60) NARRAY
00310   279*       60 FORMAT(24HOTHIS TRIP WAS TAKEN ON ,2X,1H/,2I1,1H/,T25,2I1,T31,2I1,
00310   280*         $4H AT ,2(2I1,1H&),2I1,1H.)
00311   281*          REWIND 8
00312   282*          STOP
00313   283*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
                SUBROUTINE ABC                                          DATE  68126   PAGE      1

& FOR ABC,ABC                                                           03620646
UNIVAC 1108 FORTRAN V LEVEL   2206 0018 F5018M
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03620646

   FUNCTION ABC          ENTRY POINT 000020

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001   *CODE    000023
         0000   *DATA    00001C
         0002   *BLANK   000000


   EXTERNAL REFERENCES (BLOCK, NAME)

         0003   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000 R 000000 ABC


00100      1*   C
00100      2*   C ..................................................................
00100      3*   C .LSD FUNCTION SUBROUTINE ABC                  C. ARGILA, OCT '68  .
00100      4*   C .                                                                 .
00100      5*   C .   DESCRIPTION-                                                 .
00100      6*   C .      GIVEN A COMPLEX ARGUMENT, Z, ABC(Z) IS THE MAXIMUM OF THE ABSOLUTE  .
00100      7*   C .   VALUE OF THE REAL AND IMAGINARY PARTS OF Z.                   .
00100      8*   C .                                                                 .
00100      9*   C ..................................................................
00100     10*   C
00101     11*         FUNCTION ABC(Z)
00103     12*         DIMENSION Z(2)
00104     13*         ABC = AMAX1(ABS(Z(1)),ABS(Z(2)))
```

```
00105     14*          RETURN
00106     15*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.      0 *DIAGNOSTIC* MESSAGE(S)
```

```
                SUBROUTINE ADDFRE                                   DATE  101268  PAGE      1

& FOR ADDFRE,ADDFRE                                                 03620647
UNIVAC 1108 FORTRAN V LEVEL   2206 0018 F5018H
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03620647


   SUBROUTINE ADDFRE     ENTRY POINT 000074

   STORAGE USED (BLOCK, NAME, LENGTH)

        0001    *CODE    000103
        0000    *DATA    000024
        0002    *BLANK   000000
        0003    AFCOM    001131


   EXTERNAL REFERENCES (BLOCK, NAME)

        0004    CABS
        0005    NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000060 10L        0001   000005 106G       0001   000033 3L        0001   000041 4L        0003 R 000000 ADFRQ
    0000 I 000000 I          0003 I 001130 KMAX       0003 I 000620 KPZ      0000 R 000001 W0        0000 R 000002 ZETA



00100     1*   C
00100     2*   C ..............................................................................
00100     3*   C .LSD SUBROUTINE ADDFRE--                          C. ARGILA, NOV. '68        .
00100     4*   C .                                                                            .
00100     5*   C .   DESCRIPTION-                                                            .
00100     6*   C .      THIS ROUTINE PERFORMS A CALCULATION REGARDING ADDITIONAL FREQUENCIES, .
00100     7*   C .   FOR THE LSD SUBROUTINE FREQR.                                           .
00100     8*   C .                                                                            .
00100     9*   C .   ARGUMENTS-                                                              .
00100    10*   C .      NPOINT                                                              .
```

```
00100    11*   C .      ARRAY  COMPLEX ARRAY                                    .
00100    12*   C .                                                              .
00100    13*   C .   NAMED COMMON USAGE-                                       .
00100    14*   C .      AFCOM                                                   .
00100    15*   C .                                                              .
00100    16*   C ................................................................
00100    17*   C
00101    18*         SUBROUTINE ADDFRE(NPOINT,ARRAY)
00103    19*         COMPLEX ARRAY(100)
00104    20*         COMMON /AFCOM / ADFRQ(200,2)  , KPZ(200)      , KMAX
00105    21*         DO 10 I = 1, NPOINT
00110    22*         IF(AIMAG(ARRAY(I)) .LT. 1.0E-04) GO TO 10
00112    23*         WO = CABS(ARRAY(I))
00113    24*         IF(REAL(ARRAY(I))) 2, 3, 2
00116    25*       2 ZETA = ABS(REAL(ARRAY(I))) / WO
00117    26*         KPZ(KMAX+1) = 0
00120    27*         GO TO 4
00121    28*       3 ZETA = ABS(AIMAG(ARRAY(I))) * 1.0E-03
00122    29*         KPZ(KMAX+1) = 1
00123    30*       4 IF(ZETA .GT. 0.05) GO TO 10
00125    31*         KMAX = KMAX + 1
00126    32*         ADFRQ(KMAX,1) = WO - 6.0 * ZETA
00127    33*         ADFRQ(KMAX,2) = ZETA
00130    34*      10 CONTINUE
00132    35*         RETURN
00133    36*         END
```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

```
                    SUBROUTINE ADJUST                                   DATE  101268  PAGE      1

& FOR ADJUST,ADJUST                                                      03820849
UNIVAC 1108 FORTRAN V LEVEL   2206 0018 F5018H
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03820849

   SUBROUTINE ADJUST    ENTRY POINT 000041

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001   *CODE   000052
         0000   *DATA   000015
         0002   *BLANK  000000
         0003   ADJCOM  000311


   EXTERNAL REFERENCES (BLOCK, NAME)

         0004   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000006 1L          0001   000012 110G        0001   000026 2L          0001   000030 3L          0000 R 000000 DELTAF
    0000 I 000001 K           0003 I 000310 MMK         0003 R 000000 PRBRT
```

```
00100     1*   C
00100     2*   C ..........................................................................
00100     3*   C .LSD SUBROUTINE ADJUST--                        C. ARGILA, NOV. '68      .
00100     4*   C .                                                                        .
00100     5*   C .   DESCRIPTION-                                                        .
00100     6*   C .      THIS ROUTINE ADJUSTS FREQUENCY (OMEGA) FOR THE LSD FREQUENCY RESPONSE .
00100     7*   C .   ROUTINE, FREQR.                                                     .
00100     8*   C .                                                                        .
00100     9*   C .   ARGUMENTS-                                                          .
00100    10*   C .      OLDFRQ                                                          .
00100    11*   C .      FRQ                                                             .
```

```
00100    12*   C .                                                              .
00100    13*   C .   NAMED COMMON USAGE-                                        .
00100    14*   C .      ADJCOM                                                  .
00100    15*   C .                                                              .
00100    16*   C ................................................................
00100    17*   C
00101    18*         SUBROUTINE ADJUST(CLOFRQ,FRQ)
00103    19*         COMMON /ADJCOM/ PRBRT(200)     , MMK
00104    20*         IF(MMK .EQ. 0) GO TO 3
00106    21*         DELTAF = 0.10 * (FRQ - CLOFRQ)
00107    22*       1 DO 2 K = 1, MMK
00112    23*         IF(ABS(FRQ - PRBRT(K)) .GT. 1.0E-04) GO TO 2
00114    24*         FRQ = FRQ + DELTAF
00115    25*         GO TO 1
00116    26*       2 CONTINUE
00120    27*       3 RETURN
00121    28*         END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
                SUBROUTINE ANGLE                                DATE  101268  PAGE      1

& FOR ANGLE,ANGLE                                                 03&20&5C
UNIVAC 1108 FORTRAN V LEVEL   2206 0018 F5018H
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03&20&5C

   FUNCTION ANGLE        ENTRY POINT 000032

   STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE   000035
        0000   *DATA   000011
        0002   *BLANK  000000


   EXTERNAL REFERENCES (BLOCK, NAME)

        0003   ATAN2
        0004   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000C1C 2L          000C R 000C0C ANGLE


00100     1*   C
00100     2*   C .....................................................................
00100     3*   C .   FUNCTION SUBROUTINE ANGLE                    C. ARGILA, OCT. '68   .
00100     4*   C .                                                                      .
00100     5*   C .   DESCRIPTION-                                                       .
00100     6*   C .      THIS FUNCTION RETURNS THE MODULUS OF A COMPLEX ARGUMENT.  IF THE .
00100     7*   C .   IMAGINARY PART OF THE ARGUMENT IS ZERO, THE MODULUS IS SET TO ZERO. .
00100     8*   C .                                                                      .
00100     9*   C .....................................................................
00100    10*   C
00101    11*         FUNCTION ANGLE(Z)
00103    12*         DIMENSION Z(2)
```

```
00104    13*          IF(Z(2)) 2, 1, 2
00107    14*        1 ANGLE = 0.0
00110    15*          RETURN
00111    16*        2 ANGLE = ATAN2(Z(2),Z(1))
00112    17*          RETURN
00113    18*          END
```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

```
                    SUBROUTINE AUXSUB                                   DATE  101268  PAGE      1

& FOR AUXSUB,AUXSUB                                                      03&20&51
UNIVAC 1108 FORTRAN V LEVEL   2206 0018 F5018H
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03&20&51


   SUBROUTINE AUXSUB     ENTRY POINT 000247

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001   *CODE    000256
         0000   *DATA    000027
         0002   *BLANK   000000
         0003   BASIC    000007
         0004   COMPLX   000764
         0005   EXTRA    000004


   EXTERNAL REFERENCES (BLOCK, NAME)

         0006   CDV$
         0007   NEXPD$
         0010   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000016 115G       0001   000111 127G       0001   000171 3L         0005 C 000002 BODE       0004 R 000620 DUMMY2
    0003 R 000002 DUMM1      0000 C 000000 FOFS       0000 I 000011 I          0000 I 000010 N          0003 I 000005 NPLOR
    0003 I 000001 NPOL       0003 I 000006 NZEOR      0003 I 000000 NZER       0000 C 000002 P          0004 C 000000 PLS
    0000 C 000004 Q          0000 C 000006 S          0005 C 000000 SIGMA      0004 C 000310 ZRS


00100      1*   C ***************************************************************************
00100      2*   C ***************************************************************************
00100      3*   C ************** THIS ROUTINE CONTAINS TEMPORARY MODIFICATIONS *****************
00100      4*   C ************ MADE TO AVOID UNIVAC 1108 FORTRAN V COMPILER ERRORS **************
00100      5*   C ***************************************************************************
```

```
00100     6*   C ***************************************************************************
00100     7*   C
00100     8*   C ...........................................................................
00100     9*   C .LSD SUBROUTINE AUXSUB -- MODIFIED BY C. ARGILA, OCT 68.                  .
00100    10*   C .                                                                         .
00100    11*   C .   DESCRIPTION-                                                         .
00100    12*   C .      THIS ROUTINE EVALUATES THE ROOT LOCUS FUNCTION FOR THE LSD PROGRAM. .
00100    13*   C .                                                                         .
00100    14*   C .   ARGUMENTS-                                                           .
00100    15*   C .      S       INDEPENDENT VARIABLE (COMPLEX)                             .
00100    16*   C .      FOFS    FUNCTION VALUE (COMPLEX)                                   .
00100    17*   C .                                                                         .
00100    18*   C .   NAMED COMMON USAGE-                                                  .
00100    19*   C .      BASIC  NZER    NUMBER OF ZEROS                                     .
00100    20*   C .             NPOL    NUMBER OF POLES                                    .
00100    21*   C .             DUMMY1(3)                                                  .
00100    22*   C .             NPLOR   NUMBER OF POLES AT ORIGIN                          .
00100    23*   C .             NZEOR   NUMBER OF ZEROS AT ORIGIN                          .
00100    24*   C .      COMPLX PLS(100) COMPLEX ARRAY OF POLES                            .
00100    25*   C .             ZRS(100) COMPLEX ARRAY OF ZEROS                            .
00100    26*   C .             DUMMY2(100)                                                .
00100    27*   C .      EXTRA  SIGMA   ROOT LOCUS PHASE CONSTANT (COMPLEX)                .
00100    28*   C .             BODE    BODE GAIN (COMPLEX)                                .
00100    29*   C .                                                                        .
00100    30*   C ...........................................................................
00100    31*   C
00100    32*   C     SUBROUTINE AUXSUB(S,FOFS)
00101    33*         SUBROUTINE AUXSUB(X,Y)                                & TEMPORARY CARD
00103    34*         COMPLEX X, Y                                          & TEMPORARY CARD
00104    35*         COMPLEX BODE, FOFS, P, PLS, Q, S, SIGMA, ZRS
00105    36*         COMMON /BASIC/ NZER,NPOL, DUMM1(3), NPLOR, NZEOR
00106    37*         COMMON /COMPLX/ PLS(100), ZRS(100), DUMMY2(100)
00107    38*         COMMON /EXTRA/ SIGMA, BODE
00110    39*         S = X                                                 & TEMPORARY CARD
00111    40*         P = (1.0,0.0)
00112    41*         Q = (1.0,0.0)
00113    42*         N = NPLOR + 1
00114    43*         DO 1 I = N, NPOL
00117    44*       1 P = P * ((1.0,0.0) - S / PLS(I))
00121    45*         IF(NPLOR .NE. 0)
00121    46*        $P = P * S**NPLOR
```

```
00123    47*          IF(NZER .EQ. 0) GO TO 3
00125    48*          N = NZEOR + 1
00126    49*          DO 2 I = N, NZER
00131    50*        2 Q = Q * ((1.0,0.0) - S / ZRS(I))
00133    51*          IF(NZEOR .NE. 0)
00133    52*         $Q = Q * S**NZEOR
00135    53*        3 FOFS = SIGMA * P + BOSE * Q
00136    54*          Y = FOFS                                    & TEMPORARY CARD
00137    55*          RETURN
00140    56*          END
```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

```
                SUBROUTINE BLOCK                                        DATE  101268  PAGE      1

& FOR BLOCK,BLOCK                                                        03&20&52
UNIVAC 1108 FORTRAN V LEVEL   2206 0018 F5018H
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03&20&52


   BLOCK DATA

   STORAGE USED (BLOCK, NAME, LENGTH)

        0003   COMPLX  000764
        0004   DCDA    000335
        0005   DCMPLX  004374
        0006   FRQRSP  001037
        0007   INVLT   005372
        0010   MATDAT  000003
        0011   MTRX    025C60
        0012   OPTION  000030
        0013   RLOCUS  000106
        0014   SKALE   000002


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0006 R 000000 FREQ       0003 I 000000 IBLK1      0004 I 000000 IBLK2      0005 I 000000 IBLK3      0007 I 000000 IBLK4
    0010 I 000000 IBLK5      0011 I 000000 IBLK6      0012 I 000000 IBLK7      0013 I 000000 IBLK8      0012 I 000027 INPUT
    000C I 000000 I$         0014 R 000000 SCALE1     0014 R 000001 SCALE2


00100     1*   C
00100     2*   C ..................................................................
00100     3*   C .LSD SUBROUTINE BLOCK -- MODIFIED BY C. ARGILA, OCT. '68         .
00100     4*   C .                                                                .
00100     5*   C .   DESCRIPTION-                                                .
00100     6*   C .      THIS IS A BLOCK DATA SUBROUTINE FOR THE LSD PROGRAM.      .
00100     7*   C .                                                                .
00100     8*   C ..................................................................
00100     9*   C
00101    10*         BLOCK DATA
```

```
00102    11*          COMMON /COMPLX/  IBLK1(500)
00103    12*          COMMON /DCOA   /  IBLK2(221)
00104    13*          COMMON /DCMPLX/  IBLK3(2300)
00105    14*          COMMON /FRQRSP/  FREQ(543)
00106    15*          COMMON /INVLT /  IBLK4(2810)
00107    16*          COMMON /MATDAT/  IBLK5(3)
00110    17*          COMMON /MTRX  /  IBLK6(10800)
00111    18*          COMMON /OPTION/  IBLK7(23), INPUT
00112    19*          COMMON /RLOCUS/  IBLK8(70)
00113    20*          COMMON /SKALE /  SCALE1, SCALE2
00114    21*          DATA IBLK1, IBLK2, IBLK3, IBLK4, IBLK5, IBLK6, IBLK7, IBLK8
00114    22*         $ /16727*0/
00125    23*          DATA FREQ /200*0.,.62831853E-2,.01,.12566371E-1,.015,.18849556E-1,
00125    24*         $.02,.025,.25132741E-1,.03,.31415926E-1,.37699112E-1,.04,
00125    25*         $.43982297E-1,.05,.50265482E-1,.06,.62831853E-1,.07,.08,.09,.1,
00125    26*         $.12566371,.15,.18849556,.2,.25,.25132741,.3,.31415926,.37699112,
00125    27*         $.4,.43982297,.5,.50265482,.56548667,.6,.62831853,.7,.8,.9,1.,
00125    28*         $1.2566371,1.5,1.8849556,2.,2.5,2.5132741,3.,3.1415926,3.5,3.769911
00125    29*         $,4.,4.3982297,4.5,5.,5.0265482,5.5,5.6548667,6.,6.2831853,6.5,7.,
00125    30*         $7.5,8.,8.5,9.,9.5,10.,11.,12.,12.566371,13.,14.,15.,16.,17.,18.,
00125    31*         $18.84956,19.,20.,21.,22.,23.,24.,25.132741,27.,30.,31.415926,32.,
00125    32*         $34.,36.,37.699112,38.,40.,42.,43.982297,46.,48.,50.,50.265482,54.,
00125    33*         $56.548667,58.,62.,62.831853,66.,70.,74.,78.,82.,86.,90.,95.,100.,
00125    34*         $105.,110.,120.,125.66371,130.,140.,150.,188.49556,200.,250.,
00125    35*         $251.32741,300.,314.15926,376.99112,400.,439.82297,500.,502.65482,
00125    36*         $565.48667,600.,628.31853,700.,800.,1000.,1256.6371,1500.,1884.9556
00125    37*         $,2000.,2513.2741,200*0./
00127    38*          DATA INPUT /2/
00131    39*          DATA SCALE1, SCALE2 /1.0, 1.0/
00134    40*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

SUBROUTINE CDA DATE 1C1268 PAGE 1

03&20&56

```
& FOR CDA,CDA
UNIVAC 1108 FORTRAN V LEVEL   22C6 0018 F5C18H
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03&20&56
```

SUBROUTINE CDA ENTRY POINT 002261

STORAGE USED (BLOCK, NAME, LENGTH)

0001	*CODE	002301
0000	*DATA	017605
0002	*BLANK	000000
0003	BASIC	0C0007
0004	COMPLX	000764
0005	DCDA	000335
0006	DCMPLX	004374
0007	INVLT	005372
0C10	MATDAT	000003
0011	MTRX	025060
0012	OPTION	000030
0013	PARAM	000002
0014	RLOCUS	000106
0015	ROOTS	002570
0016	TITLES	000106

EXTERNAL REFERENCES (BLOCK, NAME)

0017	GCONJ
0020	GAMMA
0021	MTRXPR
0022	EIGEN
0023	ELIM
0024	FREQR
0025	RTLC
0026	ILT
0027	NERR2$
0030	NWDU$
0031	NIO1$
0032	NIO2$

```
0033   CABS
0034   CDV$
0035   NEXP5$
0036   NERR3$
```

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

```
0000    016567 101F      0000    016573 102F      0000    016601 103F      0000    016612 104F      0000    016620 105F
0000    016625 106F      0000    016750 108F      0000    016757 109F      0000    016777 110F      0000    016774 111F
0000    017006 112F      0000    017014 113F      0000    017021 114F      0000    017031 115F      0000    017040 116F
0000    017053 119F      0000    017060 120F      0000    017101 121F      0000    017126 122F      0000    017134 123F
0000    016656 124F      0000    016521 125F      0001    000002 125G      0000    016722 126F      0000    016535 127F
0000    016552 128F      0000    016672 129F      0000    017110 130F      0000    017113 131F      0001    000110 160G
0001    000201 2000L     0001    000301 2060L     0001    000315 2080L     0001    000331 2100L     0001    000352 2150L
0001    000426 2170L     0001    000226 226G      0001    000552 2305L     0001    000776 2340L     0001    001007 2350L
0000    016634 2398F     0001    000625 2399L     0001    000663 2400L     0001    000270 242G      0001    000676 2500L
0001    001013 2900L     0001    001022 3000L     0001    001113 3020L     0001    001176 3110L     0001    001207 3111L
0001    000467 320G      0001    001255 3205L     0001    001310 3250L     0001    000513 327G      0001    001327 3300L
0001    001346 3350L     0001    001537 3400L     0001    001542 3420L     0001    001567 3450L     0001    001572 3460L
0001    001616 3500L     0001    000710 373G      0001    001707 3755L     0001    001735 3790L     0001    002016 3810L
0001    002135 3820L     0001    002227 4000L     0001    000730 404G      0001    001027 436G      0001    000044 500L
0001    000057 505L      0001    001232 516G      0001    001274 531G      0001    000160 600L      0001    001472 604G
0001    001552 623G      0001    001602 633G      0001    001701 664G      0001    001731 676G      0001    001756 712G
0001    001761 714G      0001    002037 743G      0001    002042 745G      0001    002156 771G      0001    002161 773G
0001    000165 800L      0001    000173 900L      0000    016507 99F       0000    017115 9998F     0001    002234 9999L
0005 R  000272 ASKB      0005 R  000263 ASKRL     0005 I  000245 ASNP      0005 I  000254 ASNZ      0006 C  001604 ASP
0006 C  003100 ASZ       0000 C  000000 C         0004 C  000620 CLPLS     0012 I  000025 COMENT    0000 R  016356 CTEMP
0014 R  000016 DBDOWN    0014 R  000007 DBUP      0007 R  000002 DELTAT    0005 R  000000 DETERM    0011 R  000000 DUMMY
0006 C  000000 EST       0000 C  000006 EST1      0007 R  000001 FINALT    0003 R  000004 GAM       0003 R  000002 GKB
0000 R  016500 GKB1      0003 R  000003 GKRL      0000 R  016477 GKRL1     0016 R  000000 HEADER    0000 I  016453 I
0000 I  016472 IERR      0000 I  016470 IR        0000 I  016471 IS        0000 I  016460 I1        0000 I  016461 I2
0000 I  016467 J         0000 I  016451 K         0000 I  016446 KC        0000 I  016403 KCTEMP    0013 I  000000 KDEN
0000 I  016466 KDENX     0010 I  000001 KMAX      0013 I  000001 KNUM      0000 I  016465 KNUMX     0000 I  016475 K1
0000 I  016476 K2        0005 R  000310 LASTCO    0005 R  000301 LEADCO    0000 I  016457 MMAX      0000 I  016503 MNIMUM
0010 I  000000 MSIZ      0000 I  016455 MTEMP     0000 I  016463 N         0014 I  000077 NCLPUL     0000 I  016456 NDEG
0005 I  000223 NFAKES    0000 I  016504 NP        0000 I  016505 NPA       0003 I  000005 NPLOR     0000 I  016474 NPLS
0003 I  000001 NPOL      0000 I  016462 NR        0000 I  016430 NROOT     0005 I  000232 NROOTS    0000 I  016437 NSLV
0000 I  016502 NT        0000 I  016454 NTEMP     0000 I  016464 NUM       0000 I  016452 NYRAT     0000 I  016506 NZ
0003 I  000006 NZEOR     0003 I  000000 NZER      0000 I  016501 NZP       0000 I  016473 NZRS      0014 I  000000 PHASE
```

```
0014 R 000034 PHDOWN     0014 R 000025 PHUP       0012 I 000026 PLOT       0004 C 000000 PLS        0007 R 001306 POLYC
0007 I 000003 POLYN      0007 R 000012 POLYT      0011 R 000000 QMAT       0005 I 000317 RATIL      0015 C 000000 RTS
0010 I 000002 SELECT     0012 I 000007 SFREQR     0005 R 000244 SKB        0005 R 000243 SKRL       0005 I 000241 SNP
0005 I 000242 SNZ        0006 C 001274 SP         0012 I 000000 SROOTL     0007 R 000000 STARTT     0014 R 000052 STEPDD
0014 R 000043 STEPDU     0014 R 000070 STEPPD     0014 R 000061 STEPPU     0012 I 000016 STIMER     0000 C 017143 STOR
0000 C 017145 STOR21     0000 C 017153 STOR22     0006 C 001440 SZ         0012 I 000027 TYPCAS     0000 C 000316 WMAT
0004 C 000310 ZRS
```

```
00100     1*   C
00100     2*   C ******************************************************************************
00100     3*   C .LSD SUBROUTINE CDA -- MODIFIED BY C. ARGILA, NOV. '68                       .
00100     4*   C .                                                                            .
00100     5*   C .   DESCRIPTION-                                                            .
00100     6*   C .      THIS ROUTINE INTERPRETS EACH DATA CASE FOR THE LSD PROGRAM AND CALLS  .
00100     7*   C .   THE PROPER SUBROUTINES TO EFFECT A SOLUTION.                            .
00100     8*   C .                                                                            .
00100     9*   C .   NAMED COMMON USAGE-                                                     .
00100    10*   C .      BASIC                                                               .
00100    11*   C .      COMPLX                                                              .
00100    12*   C .      DCDA                                                                .
00100    13*   C .      DCMPLX                                                              .
00100    14*   C .      INVLT                                                               .
00100    15*   C .      MATDAT                                                              .
00100    16*   C .      MTRX                                                                .
00100    17*   C .      OPTION                                                              .
00100    18*   C .      PARAM                                                               .
00100    19*   C .      RLOCUS                                                              .
00100    20*   C .      ROOTS                                                               .
00100    21*   C .      TITLES                                                              .
00100    22*   C .                                                                            .
00100    23*   C .   SUBROUTINE USAGE-                                                       .
00100    24*   C .      GCONJ                                                               .
00100    25*   C .      GAMMA                                                               .
00100    26*   C .      MTRXPR                                                              .
00100    27*   C .      EIGEN                                                               .
00100    28*   C .      ELIM                                                                .
00100    29*   C .      FREQR                                                               .
00100    30*   C .      RTLC                                                                .
```

```
00100    31*   C .      ILT                                                    .
00100    32*   C .                                                             .
00100    33*   C ..............................................................
00100    34*   C
00101    35*         SUBROUTINE CDA
00103    36*         INTEGER PHASE , TYPCAS, SNP   , SNZ   , ASNP  , ASNZ  , RATIO ,
00103    37*        $PLOT  , POLYN , COMENT, SROOTL, SFREQR, STIMER, SELECT
00104    38*         REAL    LEADCO, LASTCO
00105    39*         COMPLEX ASP   , ASZ   , C(3)  , CLPLS , EST   , EST1(100), PLS   ,
00105    40*        $RTS   , SP    , STOR(3,40)    , STOR21, STOR22, SZ    , WMAT(60,60
00105    41*        $)     , ZRS
00106    42*         COMMON /BASIC/ NZER, NPCL, GKB, GKRL, GAM  , NPLCR, NZECR
00107    43*         COMMON /COMPLX/ PLS(100), ZRS(100), CLPLS(50)
00110    44*         COMMON /DCDA/ DETERM(21,7), NFAKES(7), NROOTS(7), SNP, SNZ, SKRL,
00110    45*        1SKB, ASNP(7), ASNZ(7), ASKRL(7), ASKB(7), LEADCO(7), LASTCO(7),
00110    46*        2RATIO(2,7)
00111    47*         COMMON /DCMPLX/ EST(50,7), SP(50), SZ(50), ASP(50,7), ASZ(50,7)
00112    48*         COMMON /INVLT/ STARTT, FINALT, DELTAT, POLYN(7), POLYT(100,7),
00112    49*        $               POLYC(3,100,7)
00113    50*         COMMON /MATDAT/ MSIZ, KMAX, SELECT
00114    51*         COMMON /MTRX/ QMAT(60,60,3)
00115    52*         COMMON /OPTION/ SROOTL(7), SFREQR(7), STIMER(7), COMENT, PLOT  ,
00115    53*        $TYPCAS
00116    54*         COMMON /PARAM/ KDEN, KNUM
00117    55*         COMMON /RLOCUS/ PHASE(7), DBUP(7), DBDOWN(7), PHUP(7), PHDOWN(7),
00117    56*        1STEPDU(7), STEPDD(7), STEPPU(7), STEPPD(7), NCLPOL(7)
00120    57*         COMMON /ROOTS/ RTS(100,7)
00121    58*         COMMON /TITLES/ HEADER(14,5)
00122    59*         DIMENSION CTEMP(3,7), DUMMY(10800), KCTEMP(3,7), NROOT(7), NSLV(7)
00122    60*        $, KC(3)
00123    61*         EQUIVALENCE     (DUMMY,QMAT)  , (STOR21,STOR(2,1))    ,
00123    62*        $                (STOR22,STOR(2,2))
00123    63*   C
00123    64*   C *** INITIALIZATION
00124    65*         DO 501 K=1,7
00127    66*     501 NSLV(K) = 1
00127    67*   C
00127    68*   C *** TRANSFER DEPENDING ON THE TYPE OF CASE BEING PROCESSED
00131    69*         IF(TYPCAS .GE. 1 .AND. TYPCAS .LE. 4)
00131    70*        $GO TO (2000,500,9999,2000), TYPCAS
00133    71*         WRITE(6,99) TYPCAS
```

```
00136    72*       99 FORMAT(29H CASE ABORTED BECAUSE INPUT =I2,16H IS NOT ALLOWED.)
00137    73*          GO TO 9999
00137    74*    C
00137    75*    C *** CASE II -- S-PLANE POLES AND ZEROS ARE INPUT
00140    76*      500 IF(SNZ .LE. SNP) GO TO 505
00142    77*          WRITE(6,125)
00144    78*      125 FORMAT (62H CASE ABORTED BECAUSE NUMBER OF ZEROS EXCEEDS NUMBER OF
00144    79*         1 POLES.)
00145    80*          GO TO 9999
00146    81*      505 NYRAT = 1
00147    82*          CALL GCONJ(SP,RTS(1,2),SNP)
00150    83*          CALL GCONJ(SZ,RTS(1,1),SNZ)
00151    84*          RATIO(1,1)=1
00152    85*          RATIO(2,1)=2
00153    86*          KNUM=1
00154    87*          KDEN=2
00155    88*          NZER=SNZ
00156    89*          NPOL=SNP
00157    90*          DO 525 I=1,NPOL
00162    91*          PLS(I)=RTS(I,2)
00163    92*      525 ZRS(I) = RTS(I,1)
00165    93*          NROOT(1)=SNZ
00166    94*          NROOT(2)=SNP
00167    95*          CALL GAMMA(PLS,ZRS,NPOL,NZER,GAM)
00170 *DIAGNOSTIC*   THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00170    96*          IF (SKB.EQ.0..AND.SKRL.EQ.0.) GO TO 800
00172 *DIAGNOSTIC*   THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00172    97*          IF (SKRL.NE.0..AND.SKB.NE.0.) GO TO 900
00174 *DIAGNOSTIC*   THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00174    98*          IF (SKRL.EQ.0.) GO TO 600
00176    99*          GKRL=SKRL
00177   100*          GKB=SKRL/GAM
00200   101*          GO TO 3000
00201   102*      600 GKB = SKB
00202   103*          GKRL = SKB * GAM
00203   104*          GO TO 3000
00204   105*      800 WRITE(6,127)
00206   106*      127 FORMAT(71H CASE ABORTED BECAUSE NEITHER BODE GAIN NOR ROOT LOCUS G
00206   107*         $AIN WERE INPUT.)
00207   108*          GO TO 9999
00210   109*      900 WRITE(6,128)
```

```
00212   110*     128 FORMAT(68H CASE ABORTED BECAUSE BOTH BODE GAIN AND ROOT LOCUS GAIN
00212   111*        $ WERE INPUT.)
00213   112*         GO TO 9999
00213   113*    C
00213   114*    C *** CASE I -- MATRIX IS INPUT
00213   115*    C *** CASE IV -- MATRIX IS INPUT BUT ONLY EIGENVALUE SOLUTION IS FOUND
00214   116*     2000 NTEMP = KMAX
00215   117*          MTEMP = 1
00216   118*          NDEG=NTEMP+1
00217   119*          MMAX = SELECT
00220   120*          IF(MMAX .EQ. 0)
00220   121*         $MMAX = 40
00222   122*          WRITE(6,101)
00224   123*      101 FORMAT(1HC 48X 12HINPUT MATRIX)
00225   124*          DO 2050 I1=1,NDEG
00230   125*          WRITE(6,102) NTEMP
00233   126*      102 FORMAT(1HC 44X 18HDEGREE OF MATRIX = I3)
00234   127*          I2=1+(I1-1)*(MMAX**2)
00235   128*          CALL MTRXPR(DUMMY(I2),MMAX,MSIZ)
00236   129*          MTEMP = MTEMP + 1
00237   130*     2050 NTEMP = NTEMP - 1
00241   131*          DO 2900 NR = 1, 7
00244   132*          IF (TYPCAS.NE.4) GO TO 2060
00246   133*          N=1
00247   134*          NUM=1
00250   135*          GO TO 2150
00251   136*     2060 NUM=0
00252   137*          N=RATIO(2,NR)
00253   138*          IF (N.EQ.0) GO TO 2900
00255   139*          IF (NSLV(N).NE.2) GO TO 2100
00257   140*     2080 NUM=1
00260   141*          N=RATIO(1,NR)
00261   142*          IF (NSLV(N).EQ.2) GO TO 2900
00263   143*          GO TO 2150
00264   144*     2100 IF(NSLV(N) .NE. 3) GO TO 2150
00266   145*          WRITE(6,103) NR, N
00272   146*      103 FORMAT (6H RATIO,I3,1X,29HCANNOT BE SOLVED--DETERMINANT,I3)
00273   147*          GO TO 2900
00274   148*     2150 IF(NFAKES(N) .EQ. 0) GO TO 2170
00276   149*          CALL GCONJ(EST(1,N),ETS(1,N),NFAKES(N))
00277   150*          IF(NFAKES(N) .NE. NROOTS(N)) GO TO 2170
```

```
00301   151*          NSLV(N) = 2
00302   152*          KNUMX = RATIO(1,N)
00303   153*          KDENX = RATIO(2,N)
00304   154*          NROOT(KNUMX) = NROOTS(KNUMX)
00305   155*          NROOT(KDENX) = NROOTS(KDENX)
00306   156*          GO TO 2340
00307   157*     2170 CALL GCONJ(EST(1,N),EST1,100)
00310   158*          WRITE(6,104) N
00313   159*      104 FORMAT(1H1 43X 18HDETERMINANT NUMBER I3)
00314   160*          WRITE(6,105)
00316   161*      105 FORMAT(/46X 17HLAST ROW ELEMENTS)
00317   162*          DO 2200 J=1,MSIZ
00322   163*          I2=  (KMAX  )*(MMAX**2)+(J-1)*MMAX+MSIZ
00323   164*     2200 DUMMY(I2) = 0.0
00325   165*          NTEMP=2*MSIZ
00326   166*          DO 2300 J=1,19,2
00331   167*          MTEMP=DETERM(J,N)
00332   168*          IF (MTEMP.EQ.0) GO TO 2305
00334   169*          WRITE(6,106) MTEMP, DETERM(J+1,N)
00340   170*      106 FORMAT(37X 6HCOLUMN I3,15H HAS THE VALUE 1PE10.4)
00341   171*          I2=  (KMAX  )*(MMAX**2)+(MTEMP-1)*MMAX+MSIZ
00342   172*     2300 DUMMY(I2) = DETERM(J+1,N)
00344   173*     2305 NSLV(N) = 2
00345   174*          IR=100
00346   175*          IF (NROOTS(N).NE.0) IR=NROOTS(N)
00350   176*          IS=NFAKES(N)
00351   177*          CALL EIGEN(IR,IS,EST1,RTS(1,N),QMAT,C,WMAT,STOR,MMAX,MSIZ,KMAX,KC,
00351   178*         $IERR)
00352   179*          IF(IERR .EQ. 0) GO TO 2399
00354   180*          WRITE(6,2398)
00356   181*     2398 FORMAT(98HCAN ERROR OCCURRED IN DETERMINING THE EIGENVALUES AND/OR
00356   182*         $ EIGENVECTORS.  THIS CASE WILL BE ABORTED.)
00357   183*          GO TO 9999
00360 *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00360   184*     2399 IF(STOR22 .EQ. (0.0,0.0)) GO TO 2400
00362   185*          IF (ABS(REAL(CABS(STOR21-STOR22)/STOR22)).LT.1.E-3) GO TO 2500
00364   186*     2400 NSLV(N) = 3
00365   187*          WRITE(6,124) N
00370   188*      124 FORMAT(/58H UNSUCESSFUL DETERMINATION OF EIGENVALUES FOR DETERMINA
00370   189*         $NT ,I2, 1H.)
00371   190*          GO TO 2900
```

```
00372   191*     2500 DO 2501 J = 1, 3
00375   192*          CTEMP(J,N) = REAL(C(J))
00376   193*     2501 KCTEMP(J,N) = KC(J)
00400   194*          NROOT(N)=IR
00401   195*          WRITE(6,129) N, (CTEMP(J,N), J = 1, 3)
00410   196*      129 FORMAT(//31X44HCHARACTERISTIC POLYNOMIAL OF DETERMINANT NO.I3/T22,
00410   197*         $7HLEADINGT49,13HLAST NON-ZEROT83,4HLAST / 3(19X 11HCOEFFICIENT) /
00410   198*         $19X 3(1PE10.4,20X))
00411   199*          IF(NROOT(N) .EQ. 0 .OR. NROOT(N) .NE. NFAKES(N))
00411   200*         $GO TO 2350
00413  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00413   201*          IF (LEADCO(N).NE.0..AND.LASTCO(N).NE.C.) GO TO 2340
00415   202*          WRITE(6,126) N
00420   203*      126 FORMAT (/40H EITHER LEADCO OR LASTCO OF DETERMINANT ,I2,76H, A DET
00420   204*         1ERMINANT WITH KNOWN ROOTS, INPUT AS ZERO - DETERMINANT NOT PROCESS
00420   205*         2ED.)
00421   206*          NSLV(N)=3
00422   207*          GO TO 2900
00423   208*     2340 CTEMP(1,N) = LEADCO(N)
00424   209*          CTEMP(2,N)=LASTCO(N)
00425   210*          KCTEMP(1,N) = 0
00426   211*          KCTEMP(2,N) = C
00427   212*     2350 NYRAT = 2
00430   213*          IF (NUM.EQ.0) GO TO 2080
00432   214*     2900 IF(TYPCAS .EQ. 4) GO TO 9999
00432   215*    C
00432   216*    C *** COMPUTE OUTPUT
00435   217*     3000 DO 4000 N = 1, 7
00440   218*          KNUM = RATIO(1,N)
00441   219*          KDEN=RATIO(2,N)
00442   220*          IF(KNUM * KDEN .EQ. 0) GO TO 4000
00444   221*          IF(NSLV(KNUM) .EQ. 3 .OR. NSLV(KDEN) .EQ. 3) GO TO 4000
00446   222*          NZRS=NROOT(KNUM)
00447   223*          NPLS=NROOT(KDEN)
00450   224*          IF (TYPCAS.EQ.2) GO TO 3020
00452   225*          WRITE(6,108) N, KNUM, KDEN
00457   226*      108 FORMAT(1H1,42X 5HRATIO I2,5H (DETI3,4H/DETI3,1H))
00460   227*     3020 WRITE(6,109)
00462   228*      109 FORMAT(1H0 42X 23HINPUT TRANSFER FUNCTION /1H0 30X 9HBODE GAIN 28X
00462   229*         $15HROOT LOCUS GAIN )
00463   230*     3060 IF(TYPCAS .EQ. 2) GO TO 3110
```

```
00465   231*          K1 = KCTEMP(1,KNUM) - KCTEMP(1,KDEN)
00466   232*          K2 = KCTEMP(2,KNUM) - KCTEMP(2,KDEN)
00467   233*          GKRL1 = CTEMP(1,KNUM) / CTEMP(1,KDEN)
00470   234*          GKRL = GKRL1 * 10.0**K1
00471   235*          GKB1 = CTEMP(2,KNUM) / CTEMP(2,KDEN)
00472   236*          GKB = GKB1 * 10.0**K2
00473   237*          WRITE(6,111) GKB, GKRL
00477   238*      111 FORMAT(2(30X 1PE10.4))
00500   239*          GO TO 3111
00501   240*     3110 WRITE(6,111) GKB, GKRL
00505   241*     3111 WRITE(6,110) NZRS, NPLS
00511   242*      110 FORMAT(1HC27X7HZEROS (I2,1H) 34X 7HPOLES (I2,1H))
00512   243*          IF (NZRS.EQ.0) GO TO 3205
00514   244*          WRITE(6,112) (RTS(I,KNUM), RTS(I,KDEN), I = 1, NZRS)
00523   245*      112 FORMAT(2(21X 1H(1PE10.4,1H,E10.4,1H)))
00524   246*     3205 IF(NZRS .EQ. NPLS) GO TO 3250
00526   247*          NZP=NZRS + 1
00527   248*          WRITE(6,113)(RTS(NT,KDEN),NT=NZP,NPLS)
00535   249*      113 FORMAT(65X 1H(1PE10.4,1H,E10.4,1H))
00536  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00536   250*     3250 IF(ASKRL(N) .EQ. 0.)GO TO 3300
00540   251*          WRITE(6,114) ASKRL(N)
00543   252*      114 FORMAT(1HC 41X 26HADDITIONAL ROOT LOCUS GAIN /5CX 1PE10.4)
00544   253*          GKRL=GKRL*ASKRL(N)
00545   254*          GKB=GKRL/GAM
00546  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00546   255*     3300 IF(ASKB(N) .EQ. 0.0) GO TO 3350
00550   256*          WRITE(6,115) ASKB(N)
00553   257*      115 FORMAT(1HC 54X 20HADDITIONAL BODE GAIN /5CX 1PE10.4)
00554   258*          GKB=GKB*ASKB(N)
00555   259*          GKRL=GKB*GAM
00556   260*     3350 NZER = NROOT(KNUM)
00557   261*          NPOL=NROOT(KDEN)
00560   262*          IF (ASNP(N).EQ.0.AND.ASNZ(N).EQ.0) GO TO 3500
00562   263*          CALL GCONJ(ASP(1,N),RTS(NPLS+1,KDEN),ASNP(N))
00563   264*          CALL GCONJ(ASZ(1,N),RTS(NZRS+1,KNUM),ASNZ(N))
00564   265*          NZER=NZER + ASNZ(N)
00565   266*          NPOL=NPOL + ASNP(N)
00566   267*          WRITE(6,116) ASNZ(N), ASNP(N)
00572   268*      116 FORMAT(1HC 21X 18HADDITIONAL ZEROS (I2,1H) 23X 18HADDITIONAL POLES
00572   269*         $ (I2,1H))
```

```
00573   270*          IF (ASNZ(N).EQ.0) GO TO 3460
00575   271*          IF (ASNP(N).EQ.0) GO TO 3450
00577   272*          MNIMUM=MINO(ASNP(N),ASNZ(N))
00600   273*          NP=NPLS + 1
00601   274*          NPA=NP + MNIMUM - 1
00602   275*          NZ=NZRS + 1
00603   276*          DO 3370 I1=NP,NPA
00606   277*          WRITE(6,112) RTS(NZ,KNUM), RTS(I1,KDEN)
00612   278*     3370 NZ = NZ + 1
00614   279*          IF (MNIMUM.EQ.ASNP(N)) GO TO 3460
00616   280*          NP=NP + MNIMUM
00617   281*          GO TO 3420.
00620   282*     3400 NP = NPLS + 1
00621   283*     3420 WRITE(6,113) (RTS(I1,KDEN), I1 = NP, NPOL)
00627   284*          GO TO 3500
00630   285*     3450 NZ = NZRS + 1
00631   286*     3460 WRITE(6,119) (RTS(I1,KNUM), I1 = NZ, NZER)
00637   287*      119 FORMAT(21X 1H(1PE10.4,1H,E10.4,1H))
00640   288*     3500 CALL ELIM
00641   289*          CALL GAMMA(PLS,ZRS,NPOL,NZER,GAM)
00642  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00642   290*          IF (ASKB(N).NE.0.) GKRL=GKB*GAM
00644  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00644   291*          IF (ASKRL(N).NE.0.) GKB=GKRL/GAM
00646   292*          WRITE(6,120) GKB, GKRL, GAM
00653   293*      120 FORMAT(1H0,41X 27HRESULTANT TRANSFER FUNCTION /1H0 20X 9HBODE GAIN
00653   294*         $18X 15HROOT LOCUS GAIN 20X 5HGAMMA/3(20X 1PE10.4))
00654   295*          WRITE(6,110) NZER, NPOL
00660   296*          IF (NZER.EQ.0) GO TO 3755
00662   297*          WRITE(6,112) (ZRS(I), PLS(I), I = 1, NZER)
00671   298*     3755 IF(NZER .EQ. NPOL) GO TO 3790
00673   299*          NP=NZER+1
00674   300*          WRITE(6,113) (PLS(I), I = NP, NPOL)
00702   301*     3790 IF(SFREQR(N) .EQ. 0) GO TO 3810
00704   302*          WRITE(6,121)
00706   303*      121 FORMAT(1H1,40X,28H**** FREQUENCY RESPONSE ****/)
00707   304*          IF(COMENT .NE. 0)
00707   305*         $WRITE(6,130) ((HEADER(I,J),I=1,14),J=1,COMENT)
00721   306*      130 FORMAT(15X,13A6,1A2)
00722   307*          WRITE(6,131) N
00725   308*      131 FORMAT(6H RATIO(I2)
```

```
00726   309*          CALL FREQR(IERR)
00727   310*          IF(IERR .NE. 0)
00727   311*         $WRITE(6,9998)
00732   312*     9998 FORMAT(45HCAN ERROR OCCURRED IN THE ABOVE CALCULATIONS.)
00733   313*     3810 IF(SROOTL(N) .EQ. 0) GO TO 3820
00735   314*          WRITE(6,122)
00737   315*      122 FORMAT(1H1,44X,20H**** ROOT LOCUS ****/)
00740   316*          IF(COMENT .NE. 0)
00740   317*         $WRITE(6,130) ((HEADER(I,J),I=1,14),J=1,COMENT)
00752   318*          WRITE(6,131) N
00755   319*          CALL RTLC(NCLPOL(N),PHASE(N),DBUP(N),DBDOWN(N),PHUP(N),PHDOWN(N),
00755   320*         1STEPDU(N),STEPDD(N),STEPPU(N),STEPPD(N),IERR)
00756   321*          IF(IERR .NE. 0)
00756   322*         $WRITE(6,9998)
00761   323*     3820 IF(STIMER(N) .EQ. 0) GO TO 4000
00763   324*          WRITE(6,123)
00765   325*      123 FORMAT(1H1,40X,28H**** TRANSIENT RESPONSE ****/)
00766   326*          IF(COMENT .NE. 0)
00766   327*         $WRITE(6,130) ((HEADER(I,J),I=1,14),J=1,COMENT)
01000   328*          WRITE(6,131) N
01003   329*          CALL ILT(POLYN(N),POLYT(1,N),POLYC(1,1,N),IERR)
01004   330*          IF(IERR .NE. 0)
01004   331*         $WRITE(6,9998)
01007   332*     4000 CONTINUE
01011   333*     9999 RETURN
01012   334*          END

 END OF UNIVAC 1108 FORTRAN V COMPILATION.      9 *DIAGNOSTIC* MESSAGE(S)
```

```
& FOR COMPUT,COMPUT
UNIVAC 1108 FORTRAN V LEVEL   2206 0C14 F5018H
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03&21&02


   SUBROUTINE COMPUT     ENTRY POINT 000273

   STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE   000302
        0000   *DATA   000052
        0002   *BLANK  000000
        0003   BASIC   000007
        0004   COMPLX  000764
        0005   OPTION  000030


   EXTERNAL REFERENCES (BLOCK, NAME)

        0006   DB
        0007   ANGLE
        0010   CABS
        0011   NWDU$
        0012   NIO1$
        0013   NIO2$
        0014   NWBU$
        0015   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001   000024 120G      0001   000066 127G      0001   000053 2L      0000   000001 4F      0000 R 000004 A
 0000 D 000004 AA        0007 R 000000 ANGLE     0000 R 000006 B       0000 D 000006 BB      0006 R 0000C0 DB
 0004 R 000620 DUMMY1    0005 R 000000 DUMMY2    0000 C 000014 FOFS    0003 R 000004 GAM     0003 R 000002 GKB
 0003 R 000003 GKRL      0000 I 000000 I         0005 I 000027 INPUT   0005 I 000026 IPLOT   0003 I 000005 NPLOR
 0003 I 000001 NPOL      0003 I 000006 NZEOR     0003 I 000000 NZER    0000 R 000012 PHASEM  0004 C 000000 PLS
 0000 C 000010 TEMP      0000 D 000012 XX        0000 D 000014 YY      0004 C 000310 ZRS     0000 D 000010 ZZ
```

```
00100     1*   C
00100     2*   C ...............................................................................
00100     3*   C .LSD SUBROUTINE COMPUT--                          C. ARGILA, NOV. '68        .
00100     4*   C .                                                                            .
00100     5*   C .   DESCRIPTION-                                                            .
00100     6*   C .      THIS ROUTINE EVALUATES THE TRANSFER FUNCTION                          .
00100     7*   C .         F(S) = K*(S-ZRS(1))*...*(S-ZRS(M))/((S-PLS(1))*...*(S-PLS(N)))     .
00100     8*   C .   AND PRINTS ONE LINE OF OUTPUT FOR THE FREQUENCY RESPONSE PORTION OF THE  .
00100     9*   C .   LSC PROGRAM.  ONE RECORD OF PLOT TAPE OUTPUT IS GENERATED.               .
00100    10*   C .                                                                            .
00100    11*   C .   ARGUMENT-                                                               .
00100    12*   C .      OMEGA  FREQUENCY INPUT (S = (0.0,OMEGA))                              .
00100    13*   C .                                                                            .
00100    14*   C .   NAMED COMMON USAGE-                                                      .
00100    15*   C .      BASIC                                                                 .
00100    16*   C .      COMPLX                                                                .
00100    17*   C .      OPTION                                                                .
00100    18*   C .                                                                            .
00100    19*   C .   PRINT OUTPUT-                                                           .
00100    20*   C .      ONE LINE IS PRINTED CONTAINING THE SEVEN VALUES-                      .
00100    21*   C .         OMEGA                                                              .
00100    22*   C .         OMEGA / (2 * PI)                                                   )
00100    23*   C .         REAL(F((0.0,OMEGA)))                                               .
00100    24*   C .         AIMAG(F((0.0,OMEGA)))                                              .
00100    25*   C .         CABS(F((0.0,OMEGA)))                                               .
00100    26*   C .         20.0*ALOG10(CABS(F((0.0,OMEGA))))                                  .
00100    27*   C .         ANGLE(F((0.0,OMEGA)))                                              .
00100    28*   C .         ANGLE(F((0.0,OMEGA))) - 180                                        .
00100    29*   C .                                                                            .
00100    30*   C .   PLOT TAPE OUTPUT-                                                       .
00100    31*   C .      CORRESPONDING TO EACH LINE OF PRINT, ONE EIGHT WORD PLOT TAPE RECORD  .
00100    32*   C .   IS GENERATED.                                                           .
00100    33*   C .                                                                            .
00100    34*   C .   SUBROUTINE USAGE-                                                       .
00100    35*   C .      ANGLE                                                                 .
00100    36*   C .      DB                                                                    .
00100    37*   C .                                                                            .
00100    38*   C ...............................................................................
00100    39*   C
```

```
00101    40*          SUBROUTINE COMPUT(OMEGA)
00103    41*          COMPLEX FOFS, TEMP, PLS, ZRS
00104    42*          DOUBLE PRECISION AA, BB, XX, YY, ZZ
00105    43*          COMMON /BASIC /NZER, NPCL, GKB, GKRL, GAM, NPLER, NZEOR
00106    44*          COMMON /COMPLX/ PLS(100), ZRS(100), DUMMY1(100)
00107    45*          COMMON /OPTION/ DUMMY2(22), IPLOT, INPUT
00110    46*          EQUIVALENCE (A,AA), (B,BB), (TEMP,ZZ), (PHASEM,XX), (FOFS,YY)
00111    47*          AA = 1.0D+00
00112    48*          BB = 1.0D+00
00113    49*          XX = 0.0D+00
00114    50*          YY = 0.0D+00
00115    51*          IF(NZER .EQ. 0) GO TO 2
00117    52*          DO 1 I = 1, NZER
00122    53*          ZZ = AA
00123    54*          AA = -(ZZ*DBLE(REAL(ZRS(I)))+XX*DBLE(OMEGA-AIMAG(ZRS(I))))
00124    55*        1 XX = -(XX*DBLE(REAL(ZRS(I)))+ZZ*DBLE(AIMAG(ZRS(I))-OMEGA))
00126    56*        2 DO 3 I = 1, NPCL
00131    57*          ZZ = BB
00132    58*          BB = -(ZZ*DBLE(REAL(PLS(I)))+YY*DBLE(OMEGA-AIMAG(PLS(I))))
00133    59*        3 YY = -(YY*DBLE(REAL(PLS(I)))+ZZ*DBLE(AIMAG(PLS(I))-OMEGA))
00135    60*          ZZ = BB * BB + YY * YY
00136    61*          FOFS = GKRL * CMPLX(SNGL((AA*BB+XX*YY)/ZZ),SNGL((BB*XX-AA*YY)/ZZ))
00137    62*          A = OMEGA * .159154943
00140    63*          B = CABS(FOFS)
00141    64*          TEMP = CMPLX(2.0*DB(B),57.2957795*ANGLE(FOFS))
00142    65*          PHASEM = AIMAG(TEMP) - 180.0
00143    66*          IF(PHASEM .LT. -180.0) PHASEM = PHASEM + 360.0
00145    67*          WRITE(6,4) OMEGA, A, FOFS, B, TEMP, PHASEM
00155    68*        4 FORMAT(2X 8(3X 1PE10.4))
00156    69*          IF(IPLOT .NE. 0)
00156    70*         $WRITE(8) OMEGA, A, FOFS, B, TEMP, PHASEM
00167    71*          RETURN
00170    72*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
                    SUBROUTINE CONIN                                DATE  101268  PAGE      1

& FOR CONIN,CONIN                                                   03&21&03
UNIVAC 1108 FORTRAN V LEVEL   2206 0018 F5018H
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03&21&03


   SUBROUTINE CONIN     ENTRY POINT 000021

   STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE   000033
        0000   *DATA   000006
        0002   *BLANK  000000


   EXTERNAL REFERENCES (BLOCK, NAME)

        0003   SCALE
        0004   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)



00100     1*   C
00100     2*   C .....................................................................
00100     3*   C .LSD SUBROUTINE CONIN -- MODIFIED BY C. ARGILA, OCT. '68            .
00100     4*   C .                                                                   .
00100     5*   C .   DESCRIPTION-                                                   .
00100     6*   C .      THIS ROUTINE ACCEPTS A COMPLEX ARGUMENT, Z, AND RETURNS A COMPLEX  .
00100     7*   C .   ARGUMENT, W, AND INTEGER, K, SUCH THAT Z = W * 10**K AND 1 .LE. ABC(W)  .
00100     8*   C .   .LE. 1.0E+10.                                                   .
0C100     9*   C .                                                                   .
00100    10*   C .   ARGUMENTS-                                                      .
00100    11*   C .      Z      (COMPLEX)                                           .
00100    12*   C .      W      (COMPLEX)                                           .
```

```
00101    40*          SUBROUTINE COMPUT(OMEGA)
00103    41*          COMPLEX FOFS, TEMP, PLS, ZRS
00104    42*          DOUBLE PRECISION AA, BB, XX, YY, ZZ
00105    43*          COMMON /BASIC /NZER, NPOL, GKB, GKRL, GAM, NPLCR, NZEOR
00106    44*          COMMON /COMPLX/ PLS(100), ZRS(100), DUMMY1(100)
00107    45*          COMMON /OPTION/ DUMMY2(22), IPLOT, INPUT
00110    46*          EQUIVALENCE (A,AA), (B,BB), (TEMP,ZZ), (PHASEM,XX), (FOFS,YY)
00111    47*          AA = 1.0D+00
00112    48*          BB = 1.0D+00
00113    49*          XX = 0.0D+00
00114    50*          YY = 0.0D+00
00115    51*          IF(NZER .EQ. 0) GO TO 2
00117    52*          DO 1 I = 1, NZER
00122    53*          ZZ = AA
00123    54*          AA = -(ZZ*DBLE(REAL(ZRS(I)))+XX*DBLE(OMEGA-AIMAG(ZRS(I))))
00124    55*        1 XX = -(XX*DBLE(REAL(ZRS(I)))+ZZ*DBLE(AIMAG(ZRS(I))-OMEGA))
00126    56*        2 DO 3 I = 1, NPOL
00131    57*          ZZ = BB
00132    58*          BB = -(ZZ*DBLE(REAL(PLS(I)))+YY*DBLE(OMEGA-AIMAG(PLS(I))))
00133    59*        3 YY = -(YY*DBLE(REAL(PLS(I)))+ZZ*DBLE(AIMAG(PLS(I))-OMEGA))
00135    60*          ZZ = BB * BB + YY * YY
00136    61*          FOFS = GKRL * CMPLX(SNGL((AA*BB+XX*YY)/ZZ),SNGL((BB*XX-AA*YY)/ZZ))
00137    62*          A = OMEGA * .159154943
00140    63*          B = CABS(FOFS)
00141    64*          TEMP = CMPLX(2.0*DB(B),57.2957795*ANGLE(FOFS))
00142    65*          PHASEM = AIMAG(TEMP) - 180.0
00143    66*          IF(PHASEM .LT. -180.0) PHASEM = PHASEM + 360.0
00145    67*          WRITE(6,4) OMEGA, A, FOFS, B, TEMP, PHASEM
00155    68*        4 FORMAT(2X 8(3X 1PE10.4))
00156    69*          IF(IPLOT .NE. 0)
00156    70*         $WRITE(8) OMEGA, A, FOFS, B, TEMP, PHASEM
00167    71*          RETURN
00170    72*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
                SUBROUTINE CONOUT                                   DATE  101268  PAGE      1

& FOR CONOUT,CONOUT                                                    03&21&04
UNIVAC 1108 FORTRAN V LEVEL   2206 0018 F5018H
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03&21&05


   SUBROUTINE CONOUT    ENTRY POINT 000123

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001   *CODE   000151
         0000   *DATA   000013
         0002   *BLANK  000000


   EXTERNAL REFERENCES (BLOCK, NAME)

         0003   ABC
         0004   NEXP5$
         0005   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000015 1L          0001   000114 2L          0000 R 000000 A          0003 R 000000 ABC        0000 I 000001 I


00100      1*   C
00100      2*   C ..........................................................................
00100      3*   C .LSD SUBROUTINE CONOUT -- MODIFIED BY C. ARGILA, OCT. '68                .
00100      4*   C .                                                                        .
00100      5*   C .   DESCRIPTION-                                                        .
00100      6*   C .      GIVEN A COMPLEX ARGUMENT, Z, AND INTEGER, K, THIS ROUTINE ATTEMPTS TO .
00100      7*   C .   SET THE COMPLEX ARGUMENT W = Z * 10.0**K.  IF /Z/ OR /K/ ARE TOO LARGE,  .
00100      8*   C .   W WILL BE SET TO Z * 10.0**KPRIME AND K WILL BE SET TO K - KPRIME SO THAT.
00100      9*   C .   W = Z * 10.0**K.                                                     .
00100     10*   C .                                                                        .
00100     11*   C .   ARGUMENTS-                                                          .
```

```
00100    12*   C .       Z      (COMPLEX)                                    .
00100    13*   C .       W      (COMPLEX)                                    .
00100    14*   C .       K                                                   .
00100    15*   C .                                                           .
00100    16*   C .   SUBROUTINE USAGE-                                      .
00100    17*   C .       ABC                                                 .
00100    18*   C .                                                           .
00100    19*   C ...........................................................
00100    20*   C
00101    21*         SUBROUTINE CONCUT(Z,W,K)
00103    22*         COMPLEX W, Z
00104    23*         A = ABC(Z)
00105  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00105    24*         IF(A * K .NE. 0.0) GO TO 1
00107    25*         W = Z
00110    26*         K = 0
00111    27*         GO TO 2
00112    28*       1 I = K
00113    29*         IF(K .GT. 20 .AND. A .GE. 1.0E+08) I = 20
00115    30*         IF(K .GT. 20 .AND. A .LT. 1.0E+08) I = 30
00117    31*         IF(K .LT. -30) I = -30
00121    32*         K = K - I
00122    33*         W = Z * 10.0**I
00123    34*       2 RETURN
00124    35*         END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.      1 *DIAGNOSTIC* MESSAGE(S)
```

```
                SUBROUTINE C3CD                                    DATE  101268  PAGE      1

& FOR C3CD,C3CD                                                      03621606
UNIVAC 1108 FORTRAN V LEVEL   2206 0018 F5018M
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03621606


   SUBROUTINE C3CD      ENTRY POINT 000050

   STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE   000101
        0000   *DATA   000013
        0002   *BLANK  000000


   EXTERNAL REFERENCES (BLOCK, NAME)

        0003   CONIN
        0004   CONOUT
        0005   CDV$
        0006   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000 C 000000 CTEMP      0000 I 000002 ITEMP


00100      1*   C
00100      2*   C .....................................................................
00100      3*   C .LSD SUBROUTINE C3CD-- MODIFIED BY C. ARGILA, OCT. '68              .
00100      4*   C .                                                                   .
00100      5*   C .   DESCRIPTION-                                                   .
00100      6*   C .      GIVEN THE THREE CELL NUMBERS (Z1,KZ1), (Z2,KZ2) THIS ROUTINE PRODUCES .
00100      7*   C .   THE THREE CELL QUOTIENT (Z,KZ).  BY A THREE CELL NUMBER, (W,K), WE MEAN .
00100      8*   C .   THE COMPLEX NUMBER W*10**K, WHERE W IS COMPLEX AND K IS AN INTEGER.      .
00100      9*   C .                                                                   .
00100     10*   C .   ARGUMENTS-                                                      .
```

```
00100    11*   C .      Z      COMPLEX PART OF (Z,KZ), THE THREE CELL QUOTIENT                .
00100    12*   C .      KZ     INTEGER PART OF (Z,KZ), THE THREE CELL QUOTIENT                .
00100    13*   C .      Z1     COMPLEX PART OF (Z1,KZ1), THE THREE CELL NUMERATOR             .
00100    14*   C .      KZ1    INTEGER PART OF (Z1,KZ1), THE THREE CELL NUMERATOR             .
00100    15*   C .      Z2     COMPLEX PART OF (Z2,KZ2), THE THREE CELL DENOMINATOR           .
00100    16*   C .      KZ2    INTEGER PART OF (Z2,KZ2), THE THREE CELL DENOMINATOR           .
00100    17*   C .                                                                           .
00100    18*   C .   SUBROUTINE USAGE-                                                      .
00100    19*   C .      CONIN                                                              .
00100    20*   C .      CONOUT                                                             .
00100    21*   C .                                                                         .
00100    22*   C ...........................................................................
00100    23*   C
00101    24*         SUBROUTINE C3CD(Z,KZ,Z1,KZ1,Z2,KZ2)
00103    25*         COMPLEX CTEMP, Z, Z1, Z2
00104    26*         CALL CONIN(Z1,Z,KZ)
00105    27*         CALL CONIN(Z2,CTEMP,ITEMP)
00106    28*         KZ = KZ + KZ1 - ITEMP - KZ2
00107    29*         Z = Z / CTEMP
00110    30*         CALL CONIN(Z,Z,ITEMP)
00111    31*         KZ = KZ + ITEMP
00112    32*         CALL CONOUT(Z,Z,KZ)
00113    33*         RETURN
00114    34*         END
```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

```
                SUBROUTINE C3CM                                    DATE  101264  PAGE      1

& FOR C3CM,C3CM                                                    03621607
UNIVAC 1108 FORTRAN V LEVEL   2206 C018 F5018M
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03621607


   SUBROUTINE C3CM      ENTRY POINT 000105

   STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE   000142
        0000   *DATA   000022
        0002   *BLANK  000000


   EXTERNAL REFERENCES (BLOCK, NAME)

        0003   CONIN
        0004   CONOUT
        0005   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000016 1L          0001   000024 114G        0001   000021 2L          0000 I 000003 I          0000 I 000002 KTEMP
    0000 C 000000 TEMP


00100      1*   C
00100      2*   C ..............................................................................
00100      3*   C .LSD SUBROUTINE C3CM -- MODIFIED BY C. ARGILA, NOV. '68                       .
00100      4*   C .                                                                             .
00100      5*   C .   DESCRIPTION-                                                             .
00100      6*   C .      THIS ROUTINE ACCEPTS AN ARRAY OF COMPLEX NUMBERS, Z(1),...,Z(N) AND    .
00100      7*   C .   A THREE CELL NUMBER (ZC,KC) AND RETURNS THE THREE CELL PRODUCT             .
00100      8*   C .           (W,K) = Z(1) * Z(2) *...* Z(N) * (ZC,KC)                          .
00100      9*   C .   IF IFLAG=1, OTHERWISE IT RETURNS THE THREE CELL PRODUCT                   .
00100     10*   C .           (W,K) = Z(1) * Z(2) *...* Z(N)                                    .
```

```
00100    11*    C .                                                              .
00100    12*    C .   ARGUMENTS-                                                 .
00100    13*    C .      IFLAG                                                   .
00100    14*    C .      Z                                                       .
00100    15*    C .      N                                                       .
00100    16*    C .      W                                                       .
00100    17*    C .      K                                                       .
00100    18*    C .      ZC                                                      .
00100    19*    C .      KC                                                      .
00100    20*    C .                                                              .
00100    21*    C .   SUBROUTINE USAGE-                                          .
00100    22*    C .      CONIN                                                   .
00100    23*    C .      CONCUT                                                  .
00100    24*    C .                                                              .
00100    25*    C ................................................................
00100    26*    C
00101    27*          SUBROUTINE C3CM(IFLAG,Z,N,W,K,ZC,KC)
00103    28*          COMPLEX TEMP, W, Z(N), ZC
00104    29*          IF(IFLAG .EQ. 0)
00104    30*         $GO TO 1
00106    31*          CALL CONIN(ZC,TEMP,KTEMP)
00107    32*          K = KC + KTEMP
00110    33*          GO TO 2
00111    34*        1 TEMP = (1.0,0.0)
00112    35*          K = 0
00113    36*        2 DO 3 I = 1, N
00116    37*          CALL CONIN(Z(I),W,KTEMP)
00117    38*          TEMP = TEMP * W
00120    39*          K = K + KTEMP
00121    40*          CALL CONIN(TEMP,TEMP,KTEMP)
00122    41*        3 K = K + KTEMP
00124    42*          CALL CONOUT(TEMP,N,K)
00125    43*          RETURN
00126    44*          END
```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

```
                SUBROUTINE DB                                   DATE  101268  PAGE      1

& FOR DB,DB                                                          03&21&08
UNIVAC 1108 FORTRAN V LEVEL   2206 0018 F5018H
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03&21&08

   FUNCTION DB          ENTRY POINT 000036

   STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE   000043
        0000   *DATA   000012
        0002   *BLANK  000000


   EXTERNAL REFERENCES (BLOCK, NAME)

        0003   ALOG10
        0004   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000015 2L          0001   000021 3L          0003 R 000000 ALOG10      0000 R 000000 DB


00100     1*   C .....................................................................
00100     2*   C .   FUNCTION SUBROUTINE DB                        C. ARGILA, MAY '68          .
00100     3*   C .                                                                            .
00100     4*   C .   DESCRIPTION-                                                            .
00100     5*   C .      CHANGES POWER RATIO TO DECIBELS.                                      .
00100     6*   C .                                                                            .
00100     7*   C .   ARGUMENT-                                                               .
00100     8*   C .      RATIO  POWER RATIO                                                    .
00100     9*   C .                                                                            .
00100    10*   C .....................................................................
00100    11*   C
00101    12*         FUNCTION DB(RATIO)
```

```
00103    13*          IF(RATIO) 1, 2, 3
00106    14*        1 DB = 10.0 * ALOG10(-RATIO)
00107    15*          RETURN
00110    16*        2 DB = 0.0
00111    17*          RETURN
00112    18*        3 DB = 10.0 * ALOG10(RATIO)
00113    19*          RETURN
00114    20*          END
```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

```
& FOR DTVC,DTVC
UNIVAC 1108 FORTRAN V LEVEL   2206 0315 F5018F
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03621809
```

SUBROUTINE DTVC ENTRY POINT 001156

STORAGE USED (BLOCK, NAME, LENGTH)

0001	*CODE	001264
0000	*DATA	000123
0002	*BLANK	000000

EXTERNAL REFERENCES (BLOCK, NAME)

0003	ABC
0004	C3CM
0005	CDV$
0006	NERR3$

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	000704	100L	0001	001131	1000L	0001	000710	110L	0001	000052	111G	0001	000071	120G
0001	001034	120L	0001	000101	124G	0001	001037	140L	0001	000135	142G	0001	000152	146G
0001	000247	156G	0001	000257	162G	0001	000313	200G	0001	000330	204G	0001	000436	216G
0001	001075	220L	0001	000471	232G	0001	000532	240G	0001	001111	240L	0001	000273	25L
0001	001124	250L	0001	000566	251G	0001	000631	266G	0001	000663	273G	0001	000737	310G
0001	001060	332G	0001	001105	343G	0001	000410	45L	0001	000115	5L	0001	000455	50L
0001	000540	70L	0001	000603	90L	0003 R	000000	ABC	0000 C	000000	AS	0000 R	000011	DA
0000 R	000013	DB	0000 I	000010	I	0000 I	000015	IM	0000 I	000021	ISAVE	0000 I	000012	J
0000 I	000016	JM	0000 I	000017	L	0000 I	000023	LL	0000 I	000020	M	0000 I	000014	NP
0000 R	000022	PVL	0000 C	000002	S	0000 C	000004	SAVE	0000 C	000006	T			

```
00100     1*   C
```

```
00100     2*    C ***********************************************************************
00100     3*    C .LSD SUBROUTINE DTVC -- MODIFIED BY C. ARGIRIS, NOV. '68                .
00100     4*    C .   DESCRIPTION-                                                        .
00100     5*    C .      THIS ROUTINE PERFORMS A DETERMINANT MANIPULATION FOR THE LSD PROGRAM. .
00100     6*    C .                                                                       .
00100     7*    C .   ARGUMENTS-                                                          .
00100     8*    C .      A          COMPLEX ARRAY DIMENSIONED (NX,1)                      .
00100     9*    C .      N                                                                .
00100    10*    C .      DET        (COMPLEX)                                             .
00100    11*    C .      KP                                                               .
00100    12*    C .      Q          (COMPLEX)                                             .
00100    13*    C .      NSPEC                                                            .
00100    14*    C .      S          (COMPLEX)                                             .
00100    15*    C .      IP                                                               .
00100    16*    C .      D          (COMPLEX)                                             .
00100    17*    C .      NX                                                               .
00100    18*    C .      KAKE                                                             .
00100    19*    C .                                                                       .
00100    20*    C .   SUBROUTINE USAGE-                                                   .
00100    21*    C .      ABC                                                              .
00100    22*    C .      C3CM                                                             .
00100    23*    C .                                                                       .
00100    24*    C ***********************************************************************
00100    25*    C
00101    26*          SUBROUTINE DTVC(A,N,DET,KP,Q,NSPEC,S,IP,D,NX,KAKE)
00103    27*          COMPLEX A(NX,1), AS, D(1), DET, Q, R, S(1), SAVE, T
00104    28*          DIMENSION IP(1)
00105    29*          NSPEC = 0
00106    30*          IF(N.EQ.1) GO TO 240
00110    31*          DO 1 I=1,N
00113    32*        1 IP(I) = I
00115    33*          IF(KAKE .EQ. 1) GO TO 220
00117    34*          DO 10 I=1,N
00122    35*          DA = ABC(A(I,1))
00123    36*          DO 5 J=1,N
00126    37*          DB = ABC(A(I,J))
00127    38*          IF(DB .LE. DA) GO TO 5
00131    39*          DA = DB
00132    40*        5 CONTINUE
00134  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00134    41*          IF(DA.EQ.0.) GO TO 250
```

```
00136    42*          S(I) = CMPLX(DA,0.)
00137    43*     10   CONTINUE
00141    44*          DO 20 I=1,N
00144    45*          AS = 1./S(I)
00145    46*          DO 20 J=1,N
00150 *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00150    47*     20   IF((REAL(A(I,J)).NE.0.).OR.(AIMAG(A(I,J)).NE.0.)) A(I,J)=A(I,J)*AS
00154    48*          CALL C3CM(C,S,N,R,KP,CMPLX(0.,0.),C)
00155    49*          DO 30 J=1,N
00160    50*          DA = ABC(A(1,J))
00161    51*          DO 25 I=1,N
00164    52*          DB = ABC(A(I,J))
00165    53*          IF(DB .LE. DA) GO TO 25
00167    54*          DA = DB
00170    55*     25   CONTINUE
00172 *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00172    56*          IF(DA.EQ.0.) GO TO 250
00174    57*          S(J) = CMPLX(DA,0.)
00175    58*     30   CONTINUE
00177    59*          DO 40 J=1,N
00202    60*          AS = 1./S(J)
00203    61*          DO 40 I=1,N
00206 *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00206    62*     40   IF((REAL(A(I,J)).NE.0.).OR.(AIMAG(A(I,J)).NE.0.)) A(I,J)=A(I,J)*AS
00212    63*          CALL C3CM(1,S,N,T,NP,R,KP)
00213    64*     45   IM = 1
00214    65*          DA = ABC(A(1,1))
00215    66*          DO 50 I=1,N
00220    67*          DB = ABC(A(I,1))
00221    68*          IF(DB .LE. DA) GO TO 50
00223    69*          DA = DB
00224    70*          IM = I
00225    71*     50   CONTINUE
00227    72*          JM = 1
00230    73*          L = N-1
00231    74*          DO 150 M=1,L
00234    75*          IF(IM .EQ. M) GO TO 70
00236    76*          T = -T
00237    77*          DO 60 J=M,N
00242    78*          SAVE = A(M,J)
00243    79*          A(M,J) = A(IM,J)
```

```
00244    80*     60   A(IM,J) = SAVE
00246    81*     70   IF(JM .EQ. M) GO TO 90
00250    82*          DO 80 I=1,N
00253    83*          SAVE = A(I,M)
00254    84*          A(I,M) = A(I,JM)
00255    85*     80   A(I,JM) = SAVE
00257    86*          ISAVE = IP(M)
00260    87*          IP(M) = IP(JM)
00261    88*          IP(JM) = ISAVE
00262    89*          T = -T
00263    90*     90   PVL = 0.
00264    91*          LL = M+1
00265    92*          DO 140 I=LL,N
00270  *DIAGNOSTIC*   THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00270    93*          IF(ABC(A(I,M)) .NE. 0.) GO TO 110
00272    94*          DO 100 J=LL,N
00275    95*          DA = ABC(A(I,J))
00276    96*          IF(DA .LE. PVL) GO TO 100
00300    97*          IM = I
00301    98*          JM = J
00302    99*          PVL = DA
00303   100*     100  CONTINUE
00305   101*          GO TO 140
00306   102*     110  Q = -A(I,M)/A(M,M)
00307   103*          DO 120 J=LL,N
00312  *DIAGNOSTIC*   THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00312   104*          IF((REAL(A(M,J)).NE.0.).OR.(AIMAG(A(M,J)).NE.0.))
00312   105*         * A(I,J)=A(I,J)+Q*A(M,J)
00314   106*          DA = ABC(A(I,J))
00315   107*          IF(DA .LE. PVL) GO TO 120
00317   108*          IM = I
00320   109*          JM = J
00321   110*          PVL = DA
00322   111*     120  CONTINUE
00324   112*     140  CONTINUE
00326   113*     150  CONTINUE
00330   114*          Q = A(N,N)
00331   115*          DO 160 I=1,N
00334   116*     160  D(I) = A(I,I)
00336   117*          CALL C3CM(1,D,N,DET,KP,T,NP)
00337   118*          GO TO 1000
```

```
00340    119*    220    T = (1.0,0.0)
00341    120*           NP = 0
00342    121*           DO 230 I=1,N
00345    122*    230    S(I) = (1.0,0.0)
00347    123*           GO TO 45
00350   *DIAGNOSTIC*    THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00350    124*    240    IF(ABC(A(1,1)).EQ.0.) GO TO 250
00352    125*           DET=A(1,1)
00353    126*           KP=0
00354    127*           GO TO 1000
00355    128*    250    NSPEC=1
00356    129*           DET=(0.,0.)
00357    130*           KP=0
00360    131*    1000   RETURN
00361    132*           END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.      7 *DIAGNOSTIC* MESSAGE(S)
```

& FOR EIGEN,EIGEN
UNIVAC 1108 FORTRAN V LEVEL 2206 0018 F5218H
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03621613

SUBROUTINE EIGEN ENTRY POINT 004565

STORAGE USED (BLOCK, NAME, LENGTH)

0001	*CODE	005026
0000	*DATA	002125
0002	*BLANK	000000
0003	SKALE	000002

EXTERNAL REFERENCES (BLOCK, NAME)

0004	ABC
0005	C3CD
0006	EVAL
0007	MATVAR
0010	DTVC
0011	C3CM
0012	OUT
0013	CONIN
0014	CONOUT
0015	CDV$
0016	NEXP5$
0017	CSQRT
0020	NERR3$

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	000237 1L	0001	000672 1CL	0001	000060 1000L	0001	002642 101L	0001	003500 1012G
0001	000213 1015L	0001	000232 1017L	0001	003420 1034L	0001	003423 1036L	0001	003602 1042G
0001	002644 105L	0001	003651 1055G	0001	003763 1077G	0001	002663 110L	0001	003005 113L
0001	004141 1130G	0001	003011 114L	0001	004242 1145G	0001	003103 115L	0001	004350 1162G
0001	004427 1170G	0001	004433 1174G	0001	004434 1177G	0001	003465 119CL	0001	003470 1200L

```
0001   003137 122L    0001   004476 1223G   0001   003142 124L    0001   003261 126L    0001   003267 128L
0001   003345 131L    0001   003347 132L    0001   003527 133L    0001   003674 135L    0001   000152 136G
0001   003741 136L    0001   004006 138L    0001   000172 144G    0001   003574 145L    0001   004053 149L
0001   001012 15L     0001   000176 150G    0001   000177 153G    0001   004263 162L    0001   000227 165G
0001   004371 167L    0001   004451 175L    0001   004461 180L    0001   004532 190L    0001   004534 192L
0001   000363 223G    0001   001022 25L     0001   000507 253G    0001   001025 30L     0001   000575 303G
0001   001073 305L    0001   001217 32L     0001   000754 334G    0001   000773 340G    0001   001264 35L
0001   001123 360G    0001   001267 4CL     0001   001241 404G    0001   001413 432G    0001   001425 441G
0001   001470 461G    0001   000252 5L      0001   001416 5CL     0001   001442 51L     0001   001450 54L
0001   001512 55L     0001   002001 552G    0001   001515 60L     0001   001540 61L     0001   000276 620L
0001   002627 622G    0001   000303 630L    0001   000403 632L    0001   000445 634L    0001   000465 635L
0001   000475 640L    0001   001603 65L     0001   002753 650G    0001   000532 65CL    0001   000550 655L
0001   000572 665L    0001   001630 67L     0001   003064 672G    0001   000660 675L    0001   001660 68L
0001   001664 69L     0001   001674 7CL     0001   001650 71L     0001   003130 710G    0001   002470 710L
0001   002472 715L    0001   002502 716L    0001   001710 72L     0001   002530 720L    0001   003204 722G
0001   001723 73L     0001   003323 740G    0001   003354 747G    0001   001733 75L     0001   003366 752G
0001   001762 76L     0001   001776 77L     0001   003446 777G    0001   002104 80L     0004 R 000000 ABC
0000 I 000000 ACCUR   0000 C 000200 ALPHA   0000 R 001742 AMU3    0000 C 000202 APE     0000 R 001754 BETA
0000 R 002000 BLUB    0000 C 000204 BPE     0000 I 002006 CEFLG   0000 C 000206 CONS    0000 C 000216 CPE
0000 C 000220 CURENT  0000 C 000222 D       0000 C 000224 DD      0000 C 000416 DIFF    0000 C 000414 DP1
0000 C 001166 D1PE    0000 C 001170 D2PE    0000 R 001760 E       0000 R 001626 EPS     0000 R 002003 EP7S
0000 C 001172 F       0000 R 001751 FACTOR  0000 C 001200 FBAR    0000 C 001206 FFM     0000 C 001210 G
0000 C 001216 GGM     0000 C 001220 H       0000 I 001750 I       0000 I 001745 IEST    0000 I 001774 INDEX
0000 I 001635 IP      0000 I 002004 ITPAR   0000 I 001743 IS      0000 I 001777 II      0000 I 001753 J
0000 I 001761 JCONJ   0000 I 001731 JI      0000 I 001762 JTOR    0000 I 001763 JTRAN   0000 I 001752 K
0000 I 001746 KA      0000 I 001734 KF      0000 I 001766 KFM     0000 I 001737 KG      0000 I 001767 KGM
0000 I 001770 KH      0000 I 002002 KKKC    0000 I 001757 KKRD    0000 I 001775 KL      0000 I 001771 KLM
0000 I 002004 KRD     0000 I 001756 KRD1    0000 I 001764 KRT     0000 I 001765 KRT1    0000 I 001744 KTR1
0000 C 001222 LAM     0003 R 000001 M       0000 C 001230 MULT    0000 C 001232 MUU1    0000 C 001234 MUU2
0000 R 000175 MU1     0000 R 000176 MU2     0000 R 000177 MU3     0000 I 001755 NEF     0000 I 001773 NSPEC
0000 I 001747 N4      0000 I 002005 P       0003 R 000000 Q       0000 I 000004 RDFLG   0000 C 001236 S
0000 R 002001 SIGMA   0000 R 001772 SPOT    0000 R 001776 SPOT1   0000 I 000005 STOR1   0000 C 001426 TM
0000 C 001430 TOR     0000 C 001432 UP      0000 C 001434 X       0000 C 001624 ZETA
```

```
00100      1*   C
00100      2*   C .....................................................................
00100      3*   C .LSD SUBROUTINE EIGEN -- MODIFIED BY C. ARGILA, NOV. '68            .
00100      4*   C .                                                                   .
```

```
00100     5*   C .   DESCRIPTION-                                                    .
00100     6*   C .      GIVEN A MATRIX REPRESENTING A SYSTEM OF EQUATIONS WHICH DESCRIBE A  .
00100     7*   C .   NETWORK OR OTHER SYSTEM, THIS ROUTINE SOLVES FOR THE RATIO OF ANY TWO  .
00100     8*   C .   VARIABLES IN THE SYSTEM AS REQUIRED FOR THE LSD PROGRAM.       .
00100     9*   C .                                                                   .
00100    10*   C .   ARGUMENTS-                                                      .
00100    11*   C .      R         (INTEGER)                                          .
00100    12*   C .      SS        (INTEGER)                                          .
00100    13*   C .      J         COMPLEX, ONE-DIMENSIONAL, ARRAY                    .
00100    14*   C .      LAMBDA    COMPLEX, ONE-DIMENSIONAL, ARRAY                    .
00100    15*   C .      A         ARRAY DIMENSIONED (N1X,N1X,1)                      .
00100    16*   C .      C         COMPLEX, ONE-DIMENSIONAL, ARRAY                    .
00100    17*   C .      B         COMPLEX ARRAY DIMENSIONED (N1X,1)                  .
00100    18*   C .      STOR      COMPLEX ARRAY DIMENSIONED (3,40)                   .
00100    19*   C .      IERR      ERROR FLAG                                         .
00100    20*   C .                                                                   .
00100    21*   C .   SUBROUTINE USAGE-                                              .
00100    22*   C .      ABC                                                         .
00100    23*   C .      C3CD                                                        .
00100    24*   C .      EVAL                                                        .
00100    25*   C .      MATVAR                                                      .
00100    26*   C .      DTVC                                                        .
00100    27*   C .      C3CM                                                        .
00100    28*   C .      OUT                                                         .
00100    29*   C .      CONIN                                                       .
00100    30*   C .      CONOUT                                                      .
00100    31*   C .                                                                   .
00100    32*   C .   NAMED COMMON USAGE-                                           .
00100    33*   C .      SKALE                                                       .
00100    34*   C .                                                                   .
00100    35*   C .   REMARKS-                                                       .
00100    36*   C .      WE COPY A DESCRIPTION, HERE, OF SOME OF THE MNEMONICS APPEARING IN  .
00100    37*   C .   THIS ROUTINE, AS PROVIDED WITH THE ORIGINAL VERSION OF LSD.     .
00100    38*   C .      B         IS THE COMPLEX MATRIX EQUATION OBTAINED BY MULTIPLYING OUT  .
00100    39*   C .              THE MATRIX POLYNOMIAL EQUATION.                      .
00100    40*   C .      LAM(1),...,LAM(3) ARE THE CURRENT ITERANTS.                 .
00100    41*   C .      LAM(4)  IS THE NEW ITERANT.                                 .
00100    42*   C .      LAMBDA(1),...,LAMBDA(P-1) CONTAIN THE EIGENVALUES ALREADY FOUND.  .
00100    43*   C .      LAMBDA(P) IS THE NEW EIGENVALUE.                            .
00100    44*   C .      F         CONTAINS THREE VALUES OF DET(B).                  .
00100    45*   C .      KF        ARE THE EXTENDED RANGE EXPONENTS FOR F.           .
```

```
00100    46*   C .      W, S AND DD ARE REQUIRED BY GTVC.                                .
00100    47*   C .      DIFF     CONTAINS DIFFERENCES BETWEEN CURRENT ITERANTS AND ALREADY .
00100    48*   C .               FOUND EIGENVALUES.                                       .
00100    49*   C .      PROD     IS THE PRODUCT OF THESE DIFFERENCES.                     .
00100    50*   C .      FBAR     IS F / MAX(F(1),F(2),F(3)).                              .
00100    51*   C .      D        IS A MULTIPLIER.                                         .
00100    52*   C .      A        IS THE SET OF MATRICES A1, A2,..., A(K+1).               .
00100    53*   C .      NFLAG    IS THE 'COUNTER EXCEEDED' FLAG.                          .
00100    54*   C .      MFLAG    IS USED TO CONTROL SETUP FOR OBTAINING F.                .
00100    55*   C .      STOR1(1, ) IS THE THIRD CELL OF DET F.                            .
00100    56*   C .      STOR1(2, ) IS THE THIRD CELL OF (DET F) / PRODUCT.                .
00100    57*   C .      STOR1(3, ) IS ACCURACY.                                           .
00100    58*   C .      STOR(1, ) IS DET F.                                               .
00100    59*   C .      STOR(2, ) IS (DET F) / PRODUCT.                                   .
00100    60*   C .      STOR(3, ) IS LAMBDA.                                              .
00100    61*   C .      RDFLG    IS THE RAPID DECLINE FLAG.                               .
00100    62*   C .      CEFLG    IS THE COUNTER EXCEEDED FLAG.                            .
00100    63*   C .      I        INDEX COUNTING NUMBER OF ITERATIONS.                     .
00100    64*   C .      KTR1     IS THE MAXIMUM NUMBER OF ITERATIONS ALLOWED              .
00100    65*   C .                                                                       .
00100    66*   C ........................................................................
00100    67*   C
00101    68*         SUBROUTINE EIGEN(R,SS,D,LAMBDA,A,C,B,STOR,N1X,N1,N2,KC,IERR)
00103    69*         INTEGER ACCUR(4)       , CEFLG , P      , R      , RDFLG , SS      ,
00103    70*        $        STOR1(3,40)
00104    71*         REAL    M       , MU1    , MU2    , MU3
00105    72*         COMPLEX ALPHA , APE    , B(N1X,1)       , BPE    , C(3) , CONS(4),
00105    73*        $        CPE    , CURENT, D      , DD(60), DP1    , DIFF(180)       ,
00105    74*        $        D1PE   , D2PE   , F(3)  , FBAR(3)        , FFM    , G(3)    ,
00105    75*        $        GGM    , H      , LAM(3), LAMBDA(1)      , MULT   , MUU1    ,
00105    76*        $        MUU2   , S(60)  , STOR(3,40)    , TM     , TOR    , U(1)    ,
00105    77*        $        UP     , X(60)  , ZETA
00106    78*         COMMON /SKALE / S      , Y
00107    79*         DIMENSION       A(N1X,N1X,1)  , EPS(7), IP(60), ITPAR(3)       ,
00107    80*        $        JI(3) , KC(3) , KF(3) , KG(3)
00110    81*         EQUIVALENCE     (ITPAR(1),KRD), (ITPAR(2),P)   , (ITPAR(3),CEFLG)
00111    82*         DATA AMU3, CONS, EPS, KTR1, MU3 /1.0E+04, (1.0,0.0), (-0.5,0.0),
00111    83*        $(0.5,0.0), (0.0,0.0), 1.0E-07, 1.0E-02, 1.0E-01, 1.0E+05, 1.0E-01,
00111    84*        $1.0E-03, 1.0E-04, 40, 1.0E+04/
00111    85*   C
00111    86*   C *** INITIALIZATION
```

```
00117    87*          IERR = 0
00120    88*          IEST=1
00121    89*          KA = N1*N2
00121    90*    C
00121    91*    C *** ERROR RETURN
00122    92*          IF(R*KA .NE. 0) GO TO 1002
00124    93*          IERR = 1
00125    94*          GO TO 192
00126    95*     1000 IF(R .GT. KA) R = KA
00130    96*          IF(SS .GE. R) GO TO 145
00132    97*          N4 = N2+1
00133 *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00133    98*          IF(M .EQ. 1. .AND. Q .EQ. 1.) GO TO 1015
00135    99*          DO 1005 I=1,R
00140   100*     1005 U(I) = U(I)/M
00142   101*          FACTOR =M/Q
00143   102*          DO 1010 K=1,N4
00146   103*          FACTOR = FACTOR/M
00147   104*          DO 1010 I=1,N1
00152   105*          DO 1010 J=1,N1
00155   106*     1010 A(I,J,K) = FACTOR*A(I,J,K)
00155   107*    C
00155   108*    C *** MATRICES HAVE BEEN SCALED
00161   109*     1015 P = SS+1
00162   110*          IF(SS .EQ. 0) GO TO 1017
00164   111*          DO 1016 I=1,SS
00167   112*     1016 LAMBDA(I) = U(I)
00171   113*     1017 MU2 = MU3/M
00172   114*          MUU2 = CMPLX(MU2,0.)
00172   115*    C
00172   116*    C *** START ROOT FINDING
00173   117*     1    RDFLG = 0
00174   118*          CEFLG = 0
00175   119*          UP = U(P)
00176   120*          KRD = 1
00177   121*          BETA = EPS(3)
00200   122*     5    D = CONS(2)
00201   123*          DP1 = CONS(3)
00202   124*          NEF=P-1
00203   125*          IF(RDFLG .NE. 0) GO TO 10
00203   126*    C
```

```
00203   127*   C *** GENERATE THREE STARTING VALUES
00205   128*         IF(IEST.EQ.0) GO TO 630
00207   129*         IF(ABC(UP).LE.EPS(1)) GO TO 620
00207   130*   C
00207   131*   C *** ESTIMATE PROVIDED
00211   132*    615  IEST=0
00212   133*         GO TO 10
00213   134*    620  IEST=0
00214   135*         UP=CMPLX(-BETA,0.)
00214   136*   C
00214   137*   C *** USE NOMINAL INITIAL ITERANTS
00215   138*         GO TO 10
00216   139*    630  IF(ABC(UP).GT.EPS(1)) GO TO 10
00216   140*   C
00216   141*   C *** OBTAIN NEW ITERANTS FROM PREVIOUS ITERANTS
00220   142*         KRD1=KKRD-1
00221   143*         E=.04*AMAX1(10.*EPS(1),ABC(LAMBDA(P-1)))
00222   144*         DO 635 I=1,KRD1
00225   145*         D1PE=STOR(3,I)-LAMBDA(P-1)
00226   146*         IF(ABC(D1PE).GT.E) GO TO 632
00230   147*         STOR1(2,I)=1111111
00231   148*         GO TO 635
00232   149*    632  CALL C3CD(STOR(2,I),STOR1(2,I),STOR(2,I),
00232   150*        1 STOR1(2,I),D1PE,0)
00233   151*         IF(JCONJ.EQ.0) GO TO 635
00235   152*         D1PE=STOR(3,I)-LAMBDA(P-2)
00236   153*         IF(ABC(D1PE).GT.E) GO TO 634
00240   154*         STOR1(2,I)=1111111
00241   155*         GO TO 635
00242   156*    634  CALL C3CD(STOR(2,I),STOR1(2,I),STOR(2,I),
00242   157*        1 STOR1(2,I),D1PE,0)
00243   158*    635  CONTINUE
00243   159*   C
00243   160*   C *** FIND INDICES OF 3 SMALLEST FP.  JI(3) IS INDEX OF SMALLEST.
00245   161*         JI(3)=0
00246   162*         JI(2)=0
00247   163*         K=3
00250   164*    640  JTOR=1000000
00251   165*         I=0
00252   166*         DO 655 J=1,KRD1
00255   167*         IF(STOR1(2,J)-JTOR) 650,645,655
```

```
00260   168*    645  IF(ABC(STOR(2,J)).GE.ABC(TOR)) GO TO 655
00262   169*    650  IF(J.EQ.JI(3)) GO TO 655
00264   170*         IF(J.EQ.JI(2)) GO TO 655
00266   171*         I=J
00267   172*         JTOR=STOR1(2,J)
00270   173*         TOR=STOR(2,J)
00271   174*    655  CONTINUE
00273   175*         IF(I.EQ.0) GO TO 675
00275   176*         JI(K)=I
00276   177*         IF(K.EQ.1) GO TO 665
00300   178*         K=K-1
00301   179*         GO TO 640
00301   180*    C
00301   181*    C *** PUT SMALLEST VALUES IN WORKING LOCATIONS
00302   182*     665 DO 670 J=1,3
00305   183*         K=JI(J)
00306   184*         G(J)=STOR(1,K)
00307   185*         F(J)=STOR(2,K)
00310   186*         LAM(J)=STOR(3,K)
00311   187*         KG(J)=STOR1(1,K)
00312   188*         KF(J)=STOR1(2,K)
00313   189*     670 ACCUR(J)=STOR1(3,K)
00315   190*         JTRAN=1
00316   191*         D=(LAM(3)-LAM(2))/(LAM(2)-LAM(1))
00317   192*         DP1=D+(1.,0.)
00320   193*         GO TO 305
00320   194*    C
00320   195*    C *** 3 USABLE PREVIOUS ITERANTS NOT AVAILABLE
00321   196*     675 IF(K.EQ.3) GO TO 620
00323   197*         K=JI(3)
00324   198*         UP=STOR(3,K)
00325   199*      10 JTRAN=0
00326   200*         LAM(1) = (1.+BETA)*UP
00327   201*         LAM(2) = (1.-BETA)*UP
00330   202*         LAM(3) = UP
00331   203*         IF(P .EQ. 1) GO TO 30
00331   204*    C
00331   205*    C *** CHECK TO SEE IF ANY STARTING VALUE IS NEAR A PREVIOUSLY FOUND EIGENVALUE
00331   206*    C *** IF SO CHANGE STARTING VALUES
00333   207*         DO 20 J=1,3
00336   208*         E = EPS(1)*AMAX1(1.,ABC(LAM(J)))
```

```
00337   209*          DO 15 I=1,NEF
00342   210*          IF(ABC(LAM(J)-LAMBDA(I)) .GT. E) GO TO 15
00344   211*          GO TO 25
00345   212*    15    CONTINUE
00347   213*    20    CONTINUE
00351   214*          GO TO 30
00351   215*    C
00351   216*    C *** TOO CLOSE A STARTING VALUE
00352   217*    25    UP = LAM(1)
00353   218*          GO TO 10
00353   219*    C
00353   220*    C *** 3 GOOD STARTING VALUES EXIST IN LAM(1),...,LAM(3)
00353   221*    C *** START GETTING FUNCTIONAL VALUES
00354   222*    30    CALL EVAL(A,B,N1X,F(1),KF(1),N1,N2,IP,S,DD,DIFF,
00354   223*         1LAM(1),LAMBDA,P,G(1),KG(1),ACCUR(1))
00355   224*          CALL EVAL(A,B,N1X,F(2),KF(2),N1,N2,IP,S,DD,DIFF,
00355   225*         1LAM(2),LAMBDA,P,G(2),KG(2),ACCUR(2))
00356   226*    305   KRD1=KRD+1
00357   227*          DO 31 KRT=KRD,KRD1
00362   228*          KRT1 = KRT-KRD1+2
00363   229*          STOR(1,KRT) = G(KRT1)
00364   230*          STOR(2,KRT) = F(KRT1)
00365   231*          STOR(3,KRT) = LAM(KRT1)
00366   232*          STOR1(1,KRT) = KG(KRT1)
00367   233*          STOR1(2,KRT) = KF(KRT1)
00370   234*    31    STOR1(3,KRT) = ACCUR(KRT1)
00372   235*          KRD = KRD+1
00372   236*    C
00372   237*    C *** MAKE END TESTS
00373   238*          IF(KF(1) .NE. KF(2)) GO TO 5
00375   239*          IF(ABC(F(1)-F(2)).GT.EPS(7)*AMAX1(ABC(F(1)),
00375   240*         1 ABC(F(2)))) GO TO 5
00377   241*          IF(P .EQ. 1) GO TO 4
00401   242*    32    E = EPS(1)*AMAX1(1.,ABS(MU2))
00402   243*          MU2 = CMPLX(MU2,0.)
00403   244*          DO 35 J=1,NEF
00406   245*          IF(ABC(MU2-LAMBDA(J)) .GT. E) GO TO 35
00410   246*          MU2 = (1.+BETA)*MU2
00411   247*          GO TO 32
00412   248*    35    CONTINUE
00414   249*    40    CALL EVAL(A,B,N1X,FFM,KFM,N1,N2,IP,S,DD,DIFF,
```

```
00414   250*        1MUU2,LAMBDA,P,GGM,KGM,ACCUR(4))
00415   251*         KRD = KRD+1
00416   252*         STOR(1,KRD) = GGM
00417   253*         STOR(2,KRD) = FFM
00420   254*         STOR(3,KRD) = MUU2
00421   255*         STOR1(1,KRD) = KGM
00422   256*         STOR1(2,KRD) = KFM
00423   257*         STOR1(3,KRD) = ACCUR(4)
00424   258*         IF(KF(1) .NE. KFM) GO TO 50
00426   259*         IF(ABC(FFM-F(1)).GT.EPS(7)*AMAX1(ABC(F(1)),
00426   260*        1 ABC(FFM))) GO TO 50
00426   261*    C
00426   262*    C *** USE END TEST VALUES FOR ACCURACY CHECK
00430   263*         R = P-1
00431   264*         DO 49 I=1,3
00434   265*         STOR(I,2)=STOR(I,3)
00435   266*     49  STOR1(I,2)=STOR1(I,3)
00437   267*         GO TO 149
00437   268*    C
00437   269*    C *** READY TO START ITERATING -- THERE ARE MORE ROOTS TO FIND
00437   270*    C *** THIS IS THE MAJOR ITERATION LOOP
00440   271*     50  DO 100 I=1,KTR1
00443   272*         KRD = KRD+1
00444   273*         IF(KRD+3 .LT. KTR1) GO TO 51
00446   274*         KRD = KRD-1
00447   275*         GO TO 101
00450   276*     51  CURENT = LAM(3)
00451   277*         IF(JTRAN.EQ.0) GO TO 54
00453   278*         JTRAN=0
00454   279*         GO TO 61
00455   280*     54  IF(P.EQ.1) GO TO 60
00457   281*         E = EPS(1)*AMAX1(1.,ABC(CURENT))
00460   282*         DO 55 J=1,NEF
00463   283*         IF(ABC(LAMBDA(J)-CURENT) .GT. E) GO TO 55
00465   284*         KRD = KRD-1
00466   285*         GO TO 105
00467   286*     55  CONTINUE
00467   287*    C
00467   288*    C *** GET THIRD FUNCTIONAL VALUE
00471   289*     60  CALL EVAL(A,B,N1X,F(3),KF(3),N1,N2,IP,S,JD,DIFF,
00471   290*        1CURENT,LAMBDA,P,G(3),KG(3),ACCUR(3))
```

```
00472   291*     61   STOR(1,KRD) = C(3)
00473   292*          STOR(2,KRD) = F(3)
00474   293*          STOR(3,KRD) = CURENT
00475   294*          STOR1(1,KRD) = KG(3)
00476   295*          STOR1(2,KRD) = KF(3)
00477   296*          STOR1(3,KRD) = ACCUR(3)
00500  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00500   297*          IF(ABC(F(3)) .NE. 0.) GO TO 65
00502   298*          LAMBDA(P) = CURENT
00503   299*          GO TO 110
00503   300*    C
00503   301*    C *** DETERMINE H AND THEN GET FBAR
00504   302*     65   IF(KF(1) .EQ. KF(2) .AND. KF(2) .EQ. KF(3)) GO TO 75
00506   303*          IF(KF(1)-KF(2)) 66,67,68
00511   304*     66   IF(KF(2)-KF(3)) 69,70,71
00514   305*     67   IF(KF(1) .LT. KF(3)) GO TO 69
00516   306*          IF(ABC(F(1)) .GT. ABC(F(2))) GO TO 73
00520   307*     71   H = ABC(F(2))
00521   308*          KH = KF(2)
00522   309*          GO TO 77
00523   310*     68   IF(KF(1)-KF(3)) 69,72,73
00526   311*     69   H = ABC(F(3))
00527   312*          KH = KF(3)
00530   313*          GO TO 77
00531   314*     70   IF(ABC(F(2)) .GT. ABC(F(3))) GO TO 71
00533   315*          GO TO 69
00534   316*     72   IF(ABC(F(1)) .LT. ABC(F(3))) GO TO 69
00536   317*     73   H = ABC(F(1))
00537   318*          KH = KF(1)
00540   319*          GO TO 77
00541   320*     75   IF(ABC(F(1)) .GT. ABC(F(2))) GO TO 76
00543   321*          IF(ABC(F(2)) .GT. ABC(F(3))) GO TO 71
00545   322*          GO TO 69
00546   323*     76   IF(ABC(F(1)) .GT. ABC(F(3))) GO TO 73
00550   324*          GO TO 69
00551   325*     77   DO 78 KLM =1,3
00554   326*          CALL C3CD(FBAR(KLM),KA,F(KLM),KF(KLM),H,KH)
00555   327*          IF(KA.LT.0) FBAR(KLM)=FBAR(KLM)*10.**KA
00557   328*     78   CONTINUE
00557   329*    C
00557   330*    C *** TEST CLOSENESS OF BACK VALUES
```

```
00561   331*          IF(ABC(FBAR(1)) .LT. EPS(4)*ABC(FBAR(2))) GO TO 50
00561   332*    C
00561   333*    C *** SITUATION IS RAPID DECLINE
00563   334*          UP = CURENT
00564   335*          KDFLG = 1
00565   336*          BETA = BETA*EPS(2)
00566   337*          IF(BETA .LE. EPS(1)) GO TO 110
00570   338*          KRD = KRD+1
00571   339*          GO TO 5
00571   340*    C
00571   341*    C *** GET NEXT ITERANT
00572   342*     80   CONTINUE
00573   343*          APE = D*(D*FBAR(1)-DP1*FBAR(2)+FBAR(3))
00574   344*          BPE = D*D*FBAR(1)-DP1*DP1*FBAR(2)+(1.+2.*D)*FBAR(3)
00575   345*          CPE = DP1*FBAR(3)
00576   346*          D1PE = CSQRT(BPE*BPE-4.*APE*CPE)
00577   347*          D2PE = BPE-D1PE
00600   348*          D1PE = BPE+D1PE
00601   349*          IF(ABC(D1PE) .GE. ABC(D2PE)) GO TO 710
00603   350*          D = D2PE
00604   351*          GO TO 715
00605   352*     710  D = D1PE
00606  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00606   353*     715  IF(ABC(D).NE.0.) GO TO 716
00610   354*          D=(1.,0.)
00611   355*          GO TO 720
00612   356*     716  D=-2.*CPE/D
00613   357*     720  CURENT = LAM(3)+D*(LAM(3)-LAM(2))
00613   358*    C
00613   359*    C *** TEST CURENT AS ROOT
00614   360*     740  IF(ABC(CURENT) .LE. EPS(1)) GO TO 105
00616   361*          IF(ABC(CURENT-LAM(3))/AMAX1(1.,ABC(CURENT)) .LE. EPS(1)) GO TO 105
00620   362*          DP1 = D+CONS(1)
00620   363*    C
00620   364*    C *** MOVE UP ITERATES
00621   365*          DO 85 K=1,2
00624   366*          LAM(K) = LAM(K+1)
00625   367*          F(K) = F(K+1)
00626   368*     85   KF(K) = KF(K+1)
00630   369*          LAM(3) = CURENT
00631   370*     100  CONTINUE
```

```
00631   371*   C
00631   372*   C *** END OF MAJOR LOOP
00633   373*    101  CEFLG = 1
00634   374*    105  LAMBDA(P) = CURENT
00635   375*         KRD = KRD+1
00636   376*         STOR(3,KRD) = CURENT
00637   377*         STOR1(3,KRD)=0
00640   378*    110  SPOT = 100.*ABS(LAMBDA(P))
00641   379*         IF(SPOT .GE. MU2) MU2 = SPOT
00643   380*         CALL MATVAR(LAMBDA(P),A,N1,N2,N1X,B)
00644   381*         CALL DTVC(B,N1,G(2),KG(2),ZETA,NSPEC,S,IP,DD,N1X,1)
00645   382*         IF(P.EQ.1) GO TO 113
00647   383*         DO 112 J=1,NEF
00652   384*    112  DIFF(J)=LAMBDA(P)-LAMBDA(J)
00654   385*         CALL CBCM(0,DIFF,NEF,G(1),KG(1),C,D)
00655   386*         CALL CBCD(F(3),KF(3),G(2),KG(2),G(1),KG(1))
00656   387*         GO TO 114
00657   388*    113  F(3)=G(2)
00660   389*         KF(3)=KG(2)
00661   390*    114  STOR(2,KRD)=F(3)
00662   391*         STOR1(2,KRD)=KF(3)
00663   392*         STOR(1,KRD)=G(2)
00664   393*         STOR1(1,KRD)=KG(2)
00665   394*         IF(N1 .EQ. 1) GO TO 131
00667   395*         SPOT = ABS(B(1,1))
00670   396*         INDEX = 1
00671   397*         DO 115 KL=2,N1
00674   398*         SPOT1 = ABS(B(KL,KL))
00675   399*         IF(SPOT .LE. SPOT1) GO TO 115
00677   400*         SPOT = SPOT1
00700   401*         INDEX = KL
00701   402*    115  CONTINUE
00701   403*   C
00701   404*   C *** DEVELOP EIGENVECTOR
00703   405*         X(INDEX) = CONS(1)
00704   406*         IF(INDEX .GE. N1) GO TO 122
00706   407*         I1 = INDEX+1
00707   408*         DO 120 I=I1,N1
00712   409*    120  X(I) = CONS(4)
00714   410*         IF(INDEX .EQ. 1) GO TO 132
00716   411*    122  I = INDEX-1
```

```
00717   412*     124  ALPHA = CONS(4)
00720   413*          K = I+1
00721   414*          DO 125 J=K,N1
00724   415*     125  ALPHA = ALPHA+B(I,J)*X(J)
00726   416*          IF(ABC(ALPHA) .GT. ABC(B(I,I))) GO TO 128
00730   417*          X(I) = -ALPHA/B(I,I)
00731   418*     126  I = I-1
00732   419*          IF(I .NE. 0) GO TO 124
00734   420*          GO TO 132
00735   421*     128  X(I) = CONS(1)
00736   422*          MULT = -B(I,I)/ALPHA
00737   423*          DO 130 J=K,N1
00742   424*     130  X(J) = MULT*X(J)
00744   425*          GO TO 126
00745   426*     131  X(1) = CONS(1)
00746   427*     132  DO 1036 I=1,N1
00751   428*          DO 1034 J=I,N1
00754   429*          K = IP(J)
00755   430*          IF(K .NE. I) GO TO 1034
00757   431*          DIFF(I) = X(I)
00760   432*          X(I) = X(J)/S(I)
00761   433*          IF(J .EQ. I) GO TO 1036
00763   434*          X(J) = DIFF(I)
00764   435*          IP(J) = IP(I)
00765   436*          GO TO 1036
00766   437*     1034 CONTINUE
00770   438*     1036 CONTINUE
00772   439*          IF(N1 .EQ. 1) GO TO 1200
00774   440*          I = 1
00775   441*          SPOT = ABC(X(1))
00776   442*          DO 1190 J = 2,N1
01001   443*          SPOT1 = ABC(X(J))
01002   444*          IF(SPOT .GT. SPOT1) GO TO 1190
01004   445*          SPOT = SPOT1
01005   446*          I = J
01006   447*     1190 CONTINUE
01010   448*     1200 ZETA = X(I)
01011   449*          DO 1201 J=1,N1
01014   450*     1201 X(J) = X(J)/ZETA
01014   451*     C
01014   452*     C *** CALL AUXILIARY OUTPUT ROUTINE
```

```
01016   453*          KKRD=KRD
01017   454*          CALL OUT(ITPAR,X,STOR,N1)
01017   455*    C
01017   456*    C *** INCREASE P
01020   457*          P = P+1
01021   458*          JCONJ=0
01022   459*          IF(P .GT. R) GO TO 145
01024   460*          IF(CEFLG .NE. 0) GO TO 1
01026   461*          BLUB = ABS(AIMAG(LAMBDA(P-1)))
01027   462*          IF(BLUB .LT. EPS(5)) GO TO 1
01031   463*          IF(BLUB .LT. EPS(6)*ABS(REAL(LAMBDA(P-1)))) GO TO 1
01033   464*          LAMBDA(P) = CONJG(LAMBDA(P-1))
01034   465*          JCONJ=1
01035   466*          P = P+1
01036   467*          IF(P .LE. R) GO TO 1
01036   468*    C
01036   469*    C *** ACCURACY CHECK
01040   470*     145  SIGMA=0.
01041   471*          DO 146 J=1,R
01044   472*     146  SIGMA=AMAX1(SIGMA,ABS(LAMBDA(J)))
01046  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
01046   473*          IF(SIGMA.EQ.0.) SIGMA=1.
01050   474*          MU1=.01*SIGMA
01051   475*          MU2=100.*SIGMA
01052   476*     133  E = EPS(1)*AMAX1(1.,ABS(MU1))
01053   477*          MUU1 = CMPLX(MU1,0.)
01054   478*          DO 135 J=1,R
01057   479*          IF(ABS(MUU1-LAMBDA(J)) .GT. E) GO TO 135
01061   480*          MU1 = (1.+BETA)*MU1
01062   481*          GO TO 133
01063   482*     135  CONTINUE
01065   483*          CALL EVAL(A,B,N1X,F(1),KF(1),N1,N2,IP,S,DD,DIFF,
01065   484*         1 MUU1,LAMBDA,R+1,G(1),KG(1),ACCUR(1))
01066   485*          STOR(1,1) = G(1)
01067   486*          STOR(2,1) = F(1)
01070   487*          STOR(3,1) = MUU1
01071   488*          STOR1(1,1) = KG(1)
01072   489*          STOR1(2,1) = KF(1)
01073   490*          STOR1(3,1) = ACCUR(1)
01074   491*     136  E = EPS(1)*AMAX1(1.,ABS(MU2))
01075   492*          MUU2 = CMPLX(MU2,0.)
```

```
01076   493*          DO 138 J=1,R
01101   494*          IF(ABS(MU2-LAMBDA(J)) .GT. E) GO TO 138
01103   495*          MU2 = (1.+BETA)*MU2
01104   496*          GO TO 136
01105   497*     138  CONTINUE
01107   498*          CALL EVAL(A,B,N1X,FFM,KFM,N1,N2,IP,S,DD,DIFF,
01107   499*         1 MU2,LAMBDA,R+1,GGM,KGM,ACCUR(2))
01110   500*          STOR(1,2) = GGM
01111   501*          STOR(2,2) = FFM
01112   502*          STOR(3,2) = MU2
01113   503*          STOR1(1,2) = KGM
01114   504*          STOR1(2,2) = KFM
01115   505*          STOR1(3,2) = ACCUR(2)
01116   506*     149  KRD=2
01117   507*          P = 0
01120   508*          CALL OUT(ITPAR,X,STOR,C)
01121   509*     150  CONTINUE
01122   510*          CALL MATVAR(CONS(4),A,N1,N2,N1X,B)
01123   511*          CALL DTVC(B,N1,G(3),KG(3),CURENT,NSPEC,S,IP,DD,N1X,C)
01124  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
01124   512*          IF(Q .EQ. 1. .AND. M .EQ. 1.) GO TO 175
01126   513*          TM = CMPLX(Q*M**N2,0.)
01127   514*          DO 155 I=1,N1
01132   515*     155  DD(I) = TM
01134   516*          CALL C3CM(C,DD,N1,C(1),KC(1),C,C)
01135   517*          CALL C3CM(1,G(3),1,C(3),KC(3),C(1),KC(1))
01136   518*          CALL CONIN(C(3),C(3),KKKC)
01137   519*          KC(3)=KC(3)+KG(3)+KKKC
01140   520*          CALL CONOUT(C(3),C(3),KC(3))
01140   521*     C
01140   522*     C *** (C(3),KC(3)) CONTAINS Q**N1 * M**KA * DET(A(0))
01141   523*          MULT = CMPLX(M,0.)
01142   524*          IF(R.EQ.0) GO TO 162
01144   525*          DO 160 I=1,R
01147   526*          CALL C3CD(C(1),KC(1),C(1),KC(1),MULT,C)
01150   527*     160  CONTINUE
01152   528*     162  DD(1)=FFM
01153   529*          CALL C3CM(1,DD,1,C(1),KC(1),C(1),KC(1))
01154   530*          CALL CONIN(C(1),C(1),KKKC)
01155   531*          KC(1)=KC(1)+KFM+KKKC
01156   532*          CALL CONOUT(C(1),C(1),KC(1))
```

```
01156   533*    C
01156   534*    C *** CC = Q**N * M**(KK-K) * FFM
01157   535*          IF(R.EQ.C) GO TO 167
01161   536*          DO 165 I=1,R
01164   537*     165  LAMBDA(I) = M*LAMBDA(I)
01166   538*     167  FACTOR=Q/M
01167   539*          DO 170 K=1,N4
01172   540*          FACTOR = FACTOR*M
01173   541*          DO 170 I=1,N1
01176   542*          DO 170 J=1,N1
01201   543*     170  A(I,J,K) = FACTOR*A(I,J,K)
01205   544*          GO TO 180
01206   545*     175  C(3) = G(3)
01207   546*          KC(3) = KG(3)
01210   547*          C(1) = FFM
01211   548*          KC(1) = KFM
01211   549*    C
01211   550*    C *** HAVE CO AND CR, NOW GET CQ
01212   551*     180  C(2)=C(1)
01213   552*          KC(2) = KC(1)
01214   553*          IF(R.EQ.C) GO TO 192
01216   554*          EPTS=1C.*EPS(1)
01217   555*          DO 19C I=1,R
01222   556*          IF(ABC(LAMBDA(I)).LE.EPTS) GO TO 190
01224   557*          DD(1) = -LAMBDA(I)
01225   558*          CALL C3CM(1,DD,1,C(2),KC(2),C(2),KC(2))
01226   559*     190  CONTINUE
01230   560*      192 RETURN
01231   561*          END
```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 5 *DIAGNOSTIC* MESSAGE(S)

```
& FOR ELIM,ELIM
UNIVAC 1108 FORTRAN V LEVEL   2206 0018 F5018H
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03621621


   SUBROUTINE ELIM      ENTRY POINT 000266

   STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE   000276
        0000   *DATA   000033
        0002   *BLANK  000000
        0003   BASIC   000007
        0004   COMPLX  000764
        0005   ROOTS   00257C
        0006   PARAM   000002


   EXTERNAL REFERENCES (BLOCK, NAME)

        0007   FMASK
        0010   ORDER
        0011   CABS
        0012   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000011 115G       0001   000061 124G       0001   000115 132G       0001   000053 2L         0001   000157 3L
    0001   000162 4L         0001   000177 5L         0001   000222 7L         0001   000225 8L         0001   000231 9L
    0000 C 000000 A          0003 R 000002 DUMMY1     0004 R 000620 DUMMY2     0007 R 000000 FMASK      0000 I 000002 I
    0000 I 000003 J          0006 I 000000 KDEN       0006 I 000001 KNUM       0000 I 000004 NIN        0003 I 000005 NPLOR
    0003 I 000001 NPOL       0003 I 000006 NZEOR      0003 I 000000 NZER       0004 C 000000 PLS        0005 C 000000 RTS
    0004 C 000310 ZRS


00100     1*   C
```

```
00100     2*    C .....................................................................
00100     3*    C .LSD SUBROUTINE ELIM -- MODIFIED BY C. ARGILA, OCT. '68             .
00100     4*    C .                                                                   .
00100     5*    C .   DESCRIPTION-                                                    .
00100     6*    C .      THIS ROUTINE ELIMINATES THE COMMON POLES AND ZEROS FOR THE LSD .
00100     7*    C .   PROGRAM.  THIS ROUTINE ALSO REORDERS THE POLES AND ZEROS SO THAT ZERO .
00100     8*    C .   ELEMENTS OCCUR FIRST.                                           .
00100     9*    C .                                                                   .
00100    10*    C .   NAMED COMMON USAGE-                                             .
00100    11*    C .     BASIC  NZER   NUMBER OF ZEROS (INPUT)                         .
00100    12*    C .            NPOL   NUMBER OF POLES (INPUT)                         .
00100    13*    C .            DUMMY1(3)                                              .
00100    14*    C .            NPLOR  NUMBER OF POLES AT THE ORIGIN (RETURNED)        .
00100    15*    C .            NZEOR  NUMBER OF ZEROS AT THE ORIGIN (RETURNED)        .
00100    16*    C .     COMPLX PLS    ARRAY OF POLES (RETURNED)                       .
00100    17*    C .            ZRS    ARRAY OF ZEROS (RETURNED)                       .
00100    18*    C .            DUMMY2(100)                                            .
00100    19*    C .     ROOTS  RTS    ARRAY OF POLES AND ZEROS (INPUT)                .
00100    20*    C .     PARAM                                                        .
00100    21*    C .                                                                  .
00100    22*    C .   SUBROUTINE USAGE-                                              .
00100    23*    C .      FMASK                                                       .
00100    24*    C .      ORDER                                                       .
00100    25*    C .                                                                  .
00100    26*    C .....................................................................
00100    27*    C
00101    28*          SUBROUTINE ELIM
00103    29*          COMPLEX A, PLS, RTS, ZRS
00104    30*          COMMON /BASIC/ NZER, NPOL, DUMMY1(3), NPLOR, NZEOR
00105    31*          COMMON /COMPLX/ PLS(100), ZRS(100), DUMMY2(100)
00106    32*          COMMON /ROOTS/ RTS(100,7)
00107    33*          COMMON /PARAM/ KDEN, KNUM
00110    34*          NPLOR = 0
00111    35*          NZEOR = 0
00112    36*          IF(NZER .EQ. 0) GO TO 2
00114    37*          DO 1 I = 1, NZER
00117    38*          ZRS(I) = CMPLX(FMASK(REAL(RTS(I,KNUM))),FMASK(AIMAG(RTS(I,KNUM))))
00120  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00120    39*        1 IF(ZRS(I) .EQ. (0.0,0.0)) NZEOR = NZEOR + 1
00123    40*        2 DO 5 J = 1, NPOL
00126    41*          A = CMPLX(FMASK(REAL(RTS(J,KDEN))),FMASK(AIMAG(RTS(J,KDEN))))
```

```
00127    42*          IF(NZER .EQ. 0) GO TO 4
00131    43*          DO 3 I = 1, NZER
00134    44*          IF(CABS(A - ZRS(I)) .GT. 1.0E-03) GO TO 3
00136  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00136    45*          IF(ZRS(I) .NE. (0.0,0.0)) NZEOR = NZEOR + 1
00140    46*          ZRS(I) = (0.0,0.0)
00141    47*          PLS(J) = (0.0,0.0)
00142    48*          NPLOR = NPLOR + 1
00143    49*          GO TO 5
00144    50*        3 CONTINUE
00146  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00146    51*        4 IF(A .EQ. (0.0,0.0)) NPLOR = NPLOR + 1
00150    52*          PLS(J) = A
00151    53*        5 CONTINUE
00153    54*          NIN = MIN0(NPLOR,NZEOR)
00154    55*          IF(NPLOR - NZEOR) 6, 7, 8
00157    56*        6 NZEOR = NZEOR - NPLOR
00160    57*          NPLOR = 0
00161    58*          GO TO 9
00162    59*        7 NPLOR = 0
00163    60*          NZEOR = 0
00164    61*          GO TO 9
00165    62*        8 NPLOR = NPLOR - NZEOR
00166    63*          NZEOR = 0
00167    64*        9 CALL ORDER(PLS,NPLOR,NPOL)
00170    65*          CALL ORDER(ZRS,NZEOR,NZER)
00171    66*          NZER = NZER - NIN
00172    67*          NPOL = NPOL - NIN
00173    68*          RETURN
00174    69*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     3 *DIAGNOSTIC* MESSAGE(S)
```

```
                SUBROUTINE EMU                                          DATE  10/26/68  PAGE      1

& FOR EMU,EMU                                                           03:21:22
UNIVAC 1108 FORTRAN V LEVEL   2206.0018 F5.18H
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03:21:23


   SUBROUTINE EMU       ENTRY POINT 000070

   STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE   000125
        0000   *DATA   000013
        0002   *BLANK  000000


   EXTERNAL REFERENCES (BLOCK, NAME)

        0003   SER
        0004   CEXP
        0005   CABS
        0006   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000051 1L          0001   000061 1CL         0000 I 000000 KARG1       0003 C 000000 SER


00100     1*   C
00100     2*   C ..............................................................................
00100     3*   C .LSD SUBROUTINE EMU -- MODIFIED BY C. ARGILA, OCT. '68                        .
00100     4*   C .                                                                             .
00100     5*   C .   DESCRIPTION-                                                             .
00100     6*   C .      FOR A GIVEN COMPLEX ARGUMENT, Z, AND INTEGER, KARG2, THIS ROUTINE      .
00100     7*   C .   RETURNS-                                                                 .
00100     8*   C .            ANS = EXP(Z), IF KARG2 = 0,                                     .
00100     9*   C .            ANS = SER(Z,KARG2,15), IF KARG2 .NE. 0 AND ABS(Z) .LT. 2,       .
00100    10*   C .            ANS = EXP(Z) - 1, IF ABS(Z) .GE. 2 AND KARG2 = 1,               .
```

```
00100    11*   C .            ANS = EXP(Z) - 1 - SER(Z,1,KARG2-1), OTHERWISE.          .
00100    12*   C .                                                                     .
00100    13*   C .   SUBROUTINE USAGE-                                                .
00100    14*   C .      SER                                                           .
00100    15*   C .                                                                     .
00100    16*   C .....................................................................
00100    17*   C
00101    18*         SUBROUTINE EMU(Z,KARG2,ANS)
00103    19*         COMPLEX ANS, SER, Z
00104    20*         ANS = CEXP(Z)
00105    21*         IF(KARG2 .EQ. 0) GO TO 10
00107    22*         IF(CABS(Z) .LT. 2.0) GO TO 1
00111    23*         ANS = ANS - (1.0,0.0)
00112    24*         IF(KARG2 .EQ. 1) GO TO 10
00114    25*         KARG1 = KARG2 - 1
00115    26*         KARG2 = 1
00116    27*         ANS = ANS - SER(Z,KARG2,KARG1)
00117    28*         GO TO 10
00120    29*       1 KARG1 = 15
00121    30*         ANS = SER(Z,KARG2,KARG1)
00122    31*      10 RETURN
00123    32*         END
```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

```
                    SUBROUTINE EVAL                                    DATE  101268  PAGE      1

& FOR EVAL,EVAL                                                          03&21&24
UNIVAC 1108 FORTRAN V LEVEL   2206 0018 F5018M
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03&21&24


   SUBROUTINE EVAL      ENTRY POINT 000136

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001   *CODE   000225
         0000   *DATA   000031
         0002   *BLANK  000000


   EXTERNAL REFERENCES (BLOCK, NAME)

         0003   MATVAR
         0004   DTVC
         0005   ABC
         0006   C3CM
         0007   C3CD
         0010   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000037 1L          0001   000124 100L        0001   000066 125G        0001   000120 20L         0001   000053 5L
    0005 R 000000 ABC         0000 I 000007 J           0000 I 000010 KPROD       0000 I 000004 NP          0000 I 000005 NSPEC
    0000 C 000002 PROD        0000 C 000000 Q           0000 R 000036 Z


00100      1*   C
00100      2*   C ..............................................................................
00100      3*   C .LSD SUBROUTINE EVAL -- MODIFIED BY C. ARGILA, NOV. '68                      .
00100      4*   C .                                                                            .
00100      5*   C .   DESCRIPTION-                                                            .
00100      6*   C .      THIS ROUTINE EVALUATES THE DETERMINANT EQUATION (F,KF) = A * LAM      .
```

```
00100     7*    C .   FOR THE LSD PROGRAM.                                          .
00100     8*    C .                                                                 .
00100     9*    C .   ARGUMENTS-                                                    .
00100    10*    C .      A          REAL ARRAY DIMENSIONED (N1X,N1X,1)              .
00100    11*    C .      B          COMPLEX ARRAY DIMENSIONED (N1X,1)               .
00100    12*    C .      N1X                                                        .
00100    13*    C .      F          (COMPLEX)                                       .
00100    14*    C .      KF                                                         .
00100    15*    C .      N1                                                         .
00100    16*    C .      N2                                                         .
00100    17*    C .      IP                                                         .
00100    18*    C .      S          (COMPLEX)                                       .
00100    19*    C .      DD         (COMPLEX)                                       .
00100    20*    C .      DIFF       (COMPLEX)                                       .
00100    21*    C .      LAM        (COMPLEX)                                       .
00100    22*    C .      LAMBDA     (COMPLEX)                                       .
00100    23*    C .      P          (INTEGER)                                       .
00100    24*    C .      G          (COMPLEX)                                       .
00100    25*    C .      KG                                                         .
00100    26*    C .      KACC                                                       .
00100    27*    C .                                                                 .
00100    28*    C .   SUBROUTINE USAGE-                                            .
00100    29*    C .      MATVAR                                                     .
00100    30*    C .      DTVC                                                       .
00100    31*    C .      C3CM                                                       .
00100    32*    C .      C3CD                                                       .
00100    33*    C .      ABC                                                        .
00100    34*    C .                                                                 .
00100    35*    C ..................................................................
00100    36*    C
00101    37*          SUBROUTINE EVAL(A,B,N1X,F,KF,N1,N2,IP,S,DD,DIFF,
00101    38*         1LAM,LAMBDA,P,G,KG,KACC)
00103    39*          INTEGER P
00104    40*          COMPLEX LAM,B(N1X,1),F,...,S(1),DD(1),DIFF(1),LAMBDA(1),G,PROD
00105    41*          DIMENSION A(N1X,N1X,1),IP(1)
00106    42*          NP = P-1
00107    43*          CALL MATVAR(LAM,A,N1,N2,N1X,B)
00110    44*          CALL DTVC(B,N1,G,KG,Q,NSPEC,S,IP,DD,N1X,C)
00111    45*          Z = ABC(Q)
00112    46*          KACC = C
00113  *DIAGNOSTIC*   THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
```

```
00113    47*          IF(Z .EQ. 0.) GO TO 5
00115    48*    1     IF(Z .GE. 1.) GO TO 5
00117    49*          KACC = KACC+1
00120    50*          Z = 10.*Z
00121    51*          GO TO 1
00122    52*    5     IF(P .EQ. 1) GO TO 20
00124    53*          DO 10 J=1,NP
00127    54*    10    DIFF(J) = LAM-LAMBDA(J)
00131    55*          CALL C3DM(G,DIFF,NP,PROD,KPROD,...)
00132    56*          CALL C3CD(F,KF,G,KG,PROD,KPROD)
00133    57*          GO TO 1000
00134    58*    20    F = G
00135    59*          KF = KG
00136    60*    1000  RETURN
00137    61*          END
```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 1 *DIAGNOSTIC* MESSAGE(S)

```
                SUBROUTINE FMASK                                        DATE  101268  PAGE      1

& FOR FMASK,FMASK                                                       03&21&25
UNIVAC 1108 FORTRAN V LEVEL   2206 C018 F5018H
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03&21&25

   FUNCTION FMASK       ENTRY POINT 000022

   STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE   000027
        0000   *DATA   000011
        0002   *BLANK  000000


   EXTERNAL REFERENCES (BLOCK, NAME)

        0003   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000010 2L          0000 R 000000 FMASK


00100     1*   C
00100     2*   C ..................................................................
00100     3*   C .   FUNCTION SUBROUTINE FMASK                  C. ARGILA, OCT. '68  .
00100     4*   C .                                                                   .
00100     5*   C .   DESCRIPTION-                                                   .
00100     6*   C .      THIS ROUTINE ACCEPTS ONE REAL ARGUMENT AND RETURNS THIS ARGUMENT  .
00100     7*   C .   IF ITS MAGNITUDE IS GREATER THAN 1.0E-04, OTHERWISE IT RETURNS ZERO.  .
00100     8*   C .                                                                   .
00100     9*   C ..................................................................
00100    10*   C
00101    11*         FUNCTION FMASK(X)
00103    12*         IF(ABS(X) - 1.0E-04) 1, 1, 2
00106    13*       1 FMASK = 0.0
```

```
00107    14*          RETURN
00110    15*        2 FMASK = X
00111    16*          RETURN
00112    17*          END
```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

& FOR FREQR,FREQR
UNIVAC 1108 FORTRAN V LEVEL 2206 0015 F5218P
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 13&21&26

SUBROUTINE FREQR ENTRY POINT 000356

STORAGE USED (BLOCK, NAME, LENGTH)

0001	*CODE	000372
0000	*DATA	000067
0002	*BLANK	000000
0003	ADJCOM	000311
0004	AFCOM	001131
0005	BASIC	000007
0006	COMPLX	000764
0007	FRQRSP	001037
0010	OPTION	000030

EXTERNAL REFERENCES (BLOCK, NAME)

0011	ADDFRE
0012	ADJUST
0013	COMPUT
0014	NWDU$
0015	NIO2$
0016	NWEF$
0017	NERR3$

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	000202	13L	0001	000327	100L	0001	000007	117G	0001	000026	131G
0001	000116	160G	0001	000126	164G	0001	000320	2L	0001	000265	20L
0001	000230	221G	0001	000046	3L	0001	000276	30L	0001	000050	4L
0001	000074	5L	0001	000311	50L	0001	000165	8L	0000	000010	9F
0001	000340	999L	0004 R	000000	ADFRQ	0010 R	000000	DUMMY	0005 R	000002	DUMMY1
0007 R	000000	FREQ	0000 I	000002	I	0000 I	000001	IL	0010 I	000027	INPUT

0001	000054	142G	0001	000176	207G
0001	000300	40L	0001	000333	998L
0006 R	000620	DUMMY2			
0010 I	000026	IPLOT			

```
0000 I 000006 ITEMP     0002 I 000003 J         0001 I 000000 K         0004 I 001130 KMAX      0004 I 000620 KPZ
0000 I 000004 KSMAL     0003 I 000310 KMK       0005 I 000001 NPOL      0005 I 000000 NZER      0000 R 000007 OMEGA
0006 C 000000 PLS       0003 R 000000 PRBRT     0000 R 000305 TEMP      0006 C 000310 ZRS
```

```
00100     1*   C
00100     2*   C .....................................................................
00100     3*   C .LSD SUBROUTINE FREQR-- MODIFIED BY C. ARGILA, NOV. '68             .
00100     4*   C .                                                                   .
00100     5*   C .   DESCRIPTION-                                                   .
00100     6*   C .      THIS ROUTINE COMPUTES THE FREQUENCY RESPONSE OF A NETWORK GIVEN ITS  .
00100     7*   C .   TRANSFER FUNCTION AND A TABLE OF FREQUENCIES.  THIS ROUTINE WILL ADD   .
00100     8*   C .   FREQUENCIES TO THIS TABLE FOR PORTIONS OF THE FREQUENCY RESPONSE CURVE  .
00100     9*   C .   WHICH CHANGE MOST RAPIDLY.                                      .
00100    10*   C .                                                                   .
00100    11*   C .   ARGUMENT-                                                      .
00100    12*   C .      IERR  ERROR FLAG                                           .
00100    13*   C .                                                                   .
00100    14*   C .   NAMED COMMON USAGE-                                           .
00100    15*   C .      ADJCOM                                                     .
00100    16*   C .      AFCOM                                                      .
00100    17*   C .      BASIC                                                      .
00100    18*   C .      COMPLX                                                     .
00100    19*   C .      FRQRSP                                                     .
00100    20*   C .      OPTION                                                     .
00100    21*   C .                                                                   .
00100    22*   C .   PRINT OUTPUT-                                                  .
00100    23*   C .      ONE LINE IS PRINTED FOR EACH FREQUENCY, EACH LINE CONTAINS THE VALUES .
00100    24*   C .   FREQUENCY (RAD/SEC AND HZ), TRANSFER FUNCTION (REAL AND IMAGINARY PARTS),.
00100    25*   C .   GAIN (RATIO AND DB), PHASE(DEGREES), PHASE MARGIN(DEGREES).     .
00100    26*   C .                                                                   .
00100    27*   C .   PLOT TAPE OUTPUT-                                              .
00100    28*   C .      CORRESPONDING TO EACH LINE OF PRINT, ONE EIGHT WORD PLOT TAPE RECORD  .
00100    29*   C .   IS GENERATED.                                                  .
00100    30*   C .                                                                   .
00100    31*   C .   SUBROUTINE USAGE-                                             .
00100    32*   C .      ADDFRE                                                     .
00100    33*   C .      ADJUST                                                     .
00100    34*   C .      COMPUT                                                     .
```

```
00100    35*    C .                                                                  .
00100    36*    C ....................................................................
00100    37*    C
00101    38*          SUBROUTINE FREQR(IERR)
00103    39*          COMPLEX PLS(100)       , ZRS(100)
00104    40*          COMMON /ADJCOM/ PRBRT(200)     , MMK
00105    41*          COMMON /AFCOM / ADFRQ(200,2)   , KPZ(200)       , KMAX
00106    42*          COMMON /BASIC / NZER   , NPOL   , DUMMY1(5)
00107    43*          COMMON /COMPLX/ PLS    , ZRS    , DUMMY2(100)
00110    44*          COMMON /FRQRSP/ FREQ(543)
00111    45*          COMMON /OPTION/ DUMMY(22)      , IPLOT , INPUT
00111    46*    C
00111    47*    C *** INITIALIZATION
00112    48*          IERR = 0
00113    49*          K    = 1
00114    50*          KMAX = 0
00115    51*          MMK  = 0
00115    52*    C
00115    53*    C *** CHECK FREQUENCY TABLE
00116    54*          DO 1 IL = 1, 543
00121  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00121    55*        1 IF(FREQ(IL) .NE. 0.0) GO TO 2
00124    56*          IERR = 1
00125    57*          GO TO 999
00125    58*    C
00125    59*    C *** CHECK FOR PURE IMAGINARY POLES AND ZEROS
00126    60*        2 IF(NZER .EQ. 0) GO TO 4
00130    61*          DO 3 I = 1, NZER
00133  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00133    62*          IF(REAL(ZRS(I)) .NE. 0.0 .OR. AIMAG(ZRS(I)) .EQ. 0.0) GO TO 3
00135    63*          MMK = MMK + 1
00136    64*          PRBRT(MMK) = AIMAG(ZRS(I))
00137    65*        3 CONTINUE
00141    66*        4 DO 5 I = 1, NPOL
00144  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00144    67*          IF(REAL(PLS(I)) .NE. 0.0 .OR. AIMAG(PLS(I)) .EQ. 0.0) GO TO 5
00146    68*          MMK = MMK + 1
00147    69*          PRBRT(MMK) = AIMAG(PLS(I))
00150    70*        5 CONTINUE
00150    71*    C
00150    72*    C *** PERFORM ADDITIONAL FREQUENCIES CALCULATIONS
```

```
00152    73*          IF(NZER .NE. 0)
00152    74*         $CALL ADDFRE(NZER,ZRS)
00154    75*          CALL ADDFRE(NPOL,PLS)
00154    76*    C
00154    77*    C *** ORDER ARRAYS
00155    78*          IF(KMAX .EQ. 0) GO TO 8
00157    79*          DO 7 J = 1, KMAX
00162    80*          KSMAL = J
00163    81*          DO 6 K = J, KMAX
00166    82*        6 IF(ADFRQ(KSMAL,1) .GT. ADFRQ(K,1))
00166    83*         $KSMAL = K
00171    84*          TEMP = ADFRQ(J,1)
00172    85*          ADFRQ(J,1) = ADFRQ(KSMAL,1)
00173    86*          ADFRQ(KSMAL,1) = TEMP
00174    87*          ITEMP = KPZ(J)
00175    88*          KPZ(J) = KPZ(KSMAL)
00176    89*          KPZ(KSMAL) = ITEMP
00177    90*          TEMP = ADFRQ(J,2)
00200    91*          ADFRQ(J,2) = ADFRQ(KSMAL,2)
00201    92*        7 ADFRQ(KSMAL,2) = TEMP
00201    93*    C
00201    94*    C *** PRINT HEADINGS
00203    95*        8 WRITE(6,9)
00205    96*        9 FORMAT(1H0 11X 9HFREQUENCY 14X 17HTRANSFER FUNCTION 15X 4HGAIN 7X
00205    97*         $2(8X 5HPHASE) /6X 9H(RAD/SEC) 6X 4H(HZ) 8X 6H(REAL) 5X 11H(IMAGINA
00205    98*         $RY)4X 7H(RATIO) 7X 4H(DB) 9X 5H(DEG) 8X 6HMARGIN)
00205    99*    C
00205   100*    C *** PERFORM FREQUENCY RESPONSE CALCULATIONS
00206   101*          DO 100 I = IL, 543
00211   102*          IF(FREQ(I)) 10, 998, 10
00214   103*       10 IF(K .GT. KMAX) GO TO 50
00216   104*          IF(FREQ(I) .LE. ADFRQ(K,1)) GO TO 50
00220   105*          DO 30 J = 1, 13
00223   106*          IF(KPZ(K) * J .EQ. 7) GO TO 30
00225   107*          OMEGA = ADFRQ(K,1) + (J - 1) * ADFRQ(K,2)
00226   108*          IF(K .EQ. KMAX) GO TO 20
00230   109*          IF(OMEGA .GE. ADFRQ(K+1,1)) GO TO 40
00232   110*       20 IF(I .GT. IL)
00232   111*         $CALL ADJUST(TEMP,OMEGA)
00234   112*          CALL COMPUT(OMEGA)
00235   113*       30 CONTINUE
```

```
00237   114*       40 K = K + 1
00240   115*          IF(FREQ(I) - OMEGA)100, 10, 10
00243   116*       50 OMEGA = FREQ(I)
00244   117*          IF(I .GT. IL)
00244   118*         $CALL ADJUST(TEMP,OMEGA)
00246   119*          CALL COMPUT(OMEGA)
00247   120*      100 TEMP = OMEGA
00247   121*    C
00247   122*    C *** END FILE PLOT TAPE AND RETURN
00251   123*      998 IF(IPLOT .NE. 0)
00251   124*         $END FILE 8
00253   125*      999 RETURN
00254   126*          END
```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 3 *DIAGNOSTIC* MESSAGE(S)

```
                SUBROUTINE GAMMA                                        DATE  101268  PAGE      1

& FOR GAMMA,GAMMA                                                       03&21&28
UNIVAC 1108 FORTRAN V LEVEL   2206 C018 F5018M
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03&21&28


   SUBROUTINE GAMMA     ENTRY POINT 000145

   STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE   000163
        0000   *DATA   000035
        0002   *BLANK  000000


   EXTERNAL REFERENCES (BLOCK, NAME)

        0003   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000051 1L          0001   000011 1106        0001   000065 1306        0001   000125 2L          0001   000127 3L
    0000 D 000000 A           0000 R 000012 AI          0000 D 000002 B           0000 I 000010 I           0000 R 000011 RE
    0000 D 000004 X           0000 D 000006 Y


00100      1*   C
00100      2*   C ..........................................................................
00100      3*   C .LSD SUBROUTINE GAMMA -- MODIFIED BY C. ARGILA, OCT 68.                  .
00100      4*   C .                                                                        .
00100      5*   C .   DESCRIPTION-                                                        .
00100      6*   C .      THIS ROUTINE COMPUTES THE RATIO OF THE NEGATIVE PRODUCT OF NON-ZERO .
00100      7*   C .   POLES TO ZEROS FOR THE LSD PROGRAM.                                  .
00100      8*   C .                                                                        .
00100      9*   C .   ARGUMENTS-                                                          .
00100     10*   C .     POLES  COMPLEX ARRAY OF POLES                                      .
00100     11*   C .     ZEROS  COMPLEX ARRAY OF ZEROS                                      .
```

```
00100    12*    C .      NPOLES DIMENSION OF POLES                                    .
00100    13*    C .      NZEROS DIMENSION OF ZEROS                                    .
00100    14*    C .      RATIO  BODE TO ROOT LOCUS GAIN CONVERSION FACTOR             .
00100    15*    C .                                                                   .
00100    16*    C .....................................................................
00100    17*    C
00101    18*           SUBROUTINE GAMMA(POLES,ZEROS,NPOLES,NZEROS,RATIO)
00103    19*           COMPLEX POLES(NPOLES), ZEROS(NZEROS)
00104    20*           DOUBLE PRECISION A, B, X, Y
00104    21*    C
00104    22*    C *** COMPUTE NUMERATOR
00105    23*           A = 1.0D+00
00106    24*           X = 0.0D+00
00107    25*           DO 1 I = 1, NPOLES
00112    26*           RE = REAL(POLES(I))
00113    27*           AI = AIMAG(POLES(I))
00114  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00114    28*           IF(RE .EQ. 0.0 .AND. AI .EQ. 0.0) GO TO 1
00116    29*           Y = A
00117    30*           A = X * AI - Y * RE
00120    31*           X = -(X * RE + Y * AI)
00121    32*         1 CONTINUE
00121    33*    C
00121    34*    C *** COMPUTE DENOMINATOR
00123    35*           B = 1.0D+00
00124    36*           X = 0.0D+00
00125    37*           IF(NZEROS .EQ. 0) GO TO 3
00127    38*           DO 2 I = 1, NZEROS
00132    39*           RE = REAL(ZEROS(I))
00133    40*           AI = AIMAG(ZEROS(I))
00134  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00134    41*           IF(RE .EQ. 0.0 .AND. AI .EQ. 0.0) GO TO 2
00136    42*           Y = B
00137    43*           B = X * AI - Y * RE
00140    44*           X = -(X * RE + Y * AI)
00141    45*         2 CONTINUE
00141    46*    C
00141    47*    C *** COMPUTE RATIO
00143    48*         3 RATIO = A / B
00144    49*           RETURN
00145    50*           END
```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 2 *DIAGNOSTIC* MESSAGE(S)

```
                SUBROUTINE GCONJ                                        DATE  101268  PAGE      1

& FOR GCONJ,GCONJ                                                        03621630
UNIVAC 1108 FORTRAN V LEVEL   2206 0018 F5018F
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03621630


   SUBROUTINE GCONJ     ENTRY POINT 000054

   STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE   000061
        0000   *DATA   000015
        0002   *BLANK  000000


   EXTERNAL REFERENCES (BLOCK, NAME)

        0003   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000010 IL          0001   000037 3L          0000 I 000000 I          0000 I 000001 J


001C0      1*   C
00100      2*   C ..........................................................................
00100      3*   C .LSD SUBROUTINE GCONJ - MODIFIED BY C. ARGILA, OCT 68.                   .
00100      4*   C .                                                                        .
001C0      5*   C .   DESCRIPTION-                                                        .
001C0      6*   C .      GIVEN A TABLE OF COMPLEX NUMBERS, THIS ROUTINE PRODUCES A SECOND  .
00100      7*   C .   TABLE OF COMPLEX NUMBERS WHOSE ENTRIES ARE THOSE OF THE FIRST TABLE  .
00100      8*   C .   PLUS THE COMPLEX CONJUGATE OF THOSE ENTRIES OF THE FIRST TABLE WHOSE .
001C0      9*   C .   IMAGINARY PARTS ARE NON-ZERO.  THESE ADDITIONAL COMPLEX CONJUGATES ARE .
001C0     10*   C .   ENTERED INTO THE SECOND TABLE IMMEDIATELY FOLLOWING THE ENTRIES FROM THE .
00100     11*   C .   FIRST TABLE TO WHICH THE CORRESPOND.  IN GENERAL THE SECOND TABLE WILL .
001C0     12*   C .   BE LONGER THAN THE FIRST TABLE.                                      .
001C0     13*   C .                                                                        .
```

```
00100    14*    C .   ARGUMENTS-                                                    .
00100    15*    C .      TABIN   INPUT TABLE                                       .
00100    16*    C .      TABOUT  OUTPUT TABLE                                      .
00100    17*    C .      NPOINT  DIMENSION OF TABIN AND TABOUT                     .
00100    18*    C .                                                                .
00100    19*    C .   RESTRICTIONS-                                                .
00100    20*    C .      EXECUTION OF THIS ROUTINE WILL BE TERMINATED WHEN THE NUMBER OF .
00100    21*    C .   ENTRIES IN TABOUT EXCEEDS NPOINT, EVEN IF NOT EVERY ENTRY OF TABIN HAS .
00100    22*    C .   BEEN INTERROGATED.                                           .
00100    23*    C .                                                                .
00100    24*    C ..................................................................
00100    25*    C
00101    26*          SUBROUTINE GCONJ(TABIN,TABOUT,NPOINT)
00103    27*          COMPLEX TABIN(NPOINT), TABOUT(NPOINT)
00104    28*          I = 0
00105    29*          J = 0
00106    30*        1 I = I + 1
00107    31*          J = J + 1
00110    32*          TABOUT(J) = TABIN(I)
00111    33*          IF(AIMAG(TABIN(I))) 2, 3, 2
00114    34*        2 J = J + 1
00115    35*          TABOUT(J) = CONJG(TABIN(I))
00116    36*        3 IF(J .LT. NPOINT) GO TO 1
00120    37*          RETURN
00121    38*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
& FOR ILT,ILT
UNIVAC 1108 FORTRAN V LEVEL  2206 0618 F5618M
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03621631

   SUBRCUTINE ILT        ENTRY POINT 000647

   STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE    000704
        0000   *DATA    000271
        0002   *BLANK   000000
        0003   BASIC    000007
        0004   COMPLX   000764
        0005   ILTCOM   000317
        0006   INVLT    005372
        0007   OPTION   000030

   EXTERNAL REFERENCES (BLOCK, NAME)

        0010   ILTF
        0011   NEXPD$
        0012   NWDU$
        0013   NIO1$
        0014   NIO2$
        0015   NWBU$
        0016   NWEF$
        0017   NERR3$

   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000056 10L       0001   000330 100L      0001   000631 1000L     0001   000034 127G      0001   000067 140G
    0000   000061 150F      0001   000130 154G      0001   000144 161G      0001   000232 174G      0001   000337 200L
    0001   000311 224G      0000   000066 250F      0001   000356 250G      0001   000410 264G      0001   000072 30L
    0000   000076 300F      0001   000437 301G      0001   000442 304G      0001   000477 327G      0001   000366 350L
    0000   000104 400F      0001   000416 450L      0001   000203 50L       0000   000113 500F      0000   000144 550F
    0001   000450 600L      0000   000147 650F      0000   000010 70F       0001   000460 700L      0000   000174 750F
```

```
0000   000046 80F      0001   000473 80CL     0002   000221 90DF     0000   000054 95F      0001   000557 950L
0001   000612 975L     0001   000627 999L     0005 C 000310 BLOCK    0000 C 000000 CKSUM    0006 R 000002 DELT
0004 R 000620 DUMMY1   0006 R 000003 DUMMY2   0007 R 000000 DUMMY3   0000 R 000002 DUMMY4   0003 R 000004 GAM
0003 R 000002 GKB      0003 R 000003 GKKL      0005 R 000310 HOFT     0000 I 000004 I        0007 I 000027 INPUT
0000 I 000003 J        0000 I 000005 JERR      0000 I 000006 K        0000 I 000007 L        0003 I 000005 NPLOR
0003 I 000001 NPOL     0003 I 000006 NZEOR     0003 I 000000 NZER     0005 I 000316 NZP      0007 I 000026 PLOT
0004 C 000000 PLS      0005 R 000315 T         0005 C 000000 TABLE    0006 R 000001 TOFF     0006 R 000000 TON
0005 R 000314 XOFT     0005 R 000312 YOFT      0004 C 000310 ZRS
```

```
00100     1*   C
00100     2*   C ...........................................................................
00100     3*   C .LSD SUBROUTINE ILT-- MODIFIED BY C. ARGILA, NOV. '68                    .
00100     4*   C .                                                                        .
00100     5*   C .   DESCRIPTION-                                                        .
00100     6*   C .      THIS ROUTINE PERFORMS THE TIME RESPONSE CALCULATIONS FOR THE LSD  .
00100     7*   C .   PROGRAM.                                                            .
00100     8*   C .                                                                        .
00100     9*   C .   ARGUMENTS-                                                          .
00100    10*   C .      NPOLY NUMBER OF POLYNOMIAL SEGMENTS IN DRIVING FUNCTION         .
00100    11*   C .      TINF  BREAK-POINT TIMES FOR DRIVING FUNCTION                    .
00100    12*   C .      CINF  ARRAY OF POLYNOMIAL SEGMENT COEFFICIENTS FOR DRIVING FUNCTION .
00100    13*   C .      IERR  ERROR FLAG                                                .
00100    14*   C .                                                                        .
00100    15*   C .   PRINT OUTPUT-                                                       .
00100    16*   C .      THE VARIOUS PARAMETERS ARE PRINTED ON THE TIME RESPONSE HEADER PAGE .
00100    17*   C .   FOLLOWED BY A LISTING OF THE COMPUTED TIME RESPONSE, INVERSE TRANSFORM .
00100    18*   C .   AND DRIVING FUNCTION.                                               .
00100    19*   C .                                                                        .
00100    20*   C .   PLOT TAPE OUTPUT-                                                    .
00100    21*   C .      WHEN PLOT = 1, ONE FILE WILL BE GENERATED ON A PLOT TAPE (UNIT 8). .
00100    22*   C .   EACH RECORD IS FOUR WORDS LONG AND CORRESPONDS PRECISELY TO ONE TIME  .
00100    23*   C .   POINT.  THE FOUR WORDS IN EACH RECORD CORRESPOND, RESPECTIVELY, TO TIME, .
00100    24*   C .   RESPONS FUNCTION, INVERSE TRANSFORM AND DRIVING FUNCTION.           .
00100    25*   C .                                                                        .
00100    26*   C .   NAMED COMMON USAGE-                                                 .
00100    27*   C .      BASIC                                                           .
00100    28*   C .      COMPLX                                                          .
00100    29*   C .      ILTCOM                                                          .
```

```
00100    30*   C .      INVLT                                                .
00100    31*   C .      OPTION                                               .
00100    32*   C .                                                           .
00100    33*   C .   SUBROUTINE USAGE-                                       .
00100    34*   C .      ILTF                                                 .
00100    35*   C .                                                           .
00100    36*   C ...........................................................
00100    37*   C
00101    38*         SUBROUTINE ILT(NPOLY,TINF,CINF,IERR)
00103    39*         INTEGER PLOT
00104    40*         COMPLEX BLOCK , CKSUM , PLS   , TABLE , ZRS
00105    41*         COMMON /BASIC / NZER  , NPOL  , GKB   , GKRL  , GAM   , NPLCR ,
00105    42*        $NZEOR
00106    43*         COMMON /COMPLX/ PLS(100)      , ZRS(100)      , DUMMY1(100)
00107    44*         COMMON /ILTCOM/ TABLE(100)    , BLOCK(2)      , XOFT  , T     ,
00107    45*        $NZP
00110    46*         COMMON /INVLT / TON   , TOFF  , DELT  , DUMMY2(2807)
00111    47*         COMMON /OPTION/ DUMMY3(22)    , PLOT  , INPUT
00112    48*         DIMENSION       CINF(3,100)   , TINF(100)
00113    49*         EQUIVALENCE     (BLOCK(1),XOFT),        (BLOCK(2),YOFT)
00114    50*         DATA DUMMY4 /0.0/
00114    51*   C
00114    52*   C *** INITIALIZATION
00116    53*         IERR = 0
00117    54*         T = TON
00120    55*         NZP = NPOL - NPLCR
00120    56*   C
00120    57*   C *** CHECK FOR VALID ARGUMENTS
00121    58*         IF(NZER .GT. NPOL .OR. NPOLY .GT. 100)
00121    59*        $GO TO 999
00121    60*   C
00121    61*   C *** REPACK TABLE OF POLES
00123    62*         IF(NPLCR .EQ. 0)
00123    63*        $GO TO 30
00125    64*         J = 1
00126    65*         DO 10 I = 1, NPOL
00131 *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00131    66*         IF(PLS(I) .EQ. (0.0,0.0))
00131    67*        $GO TO 10
00133    68*         PLS(J) = PLS(I)
00134    69*         J = J + 1
```

```
00139     73*       10 CONTINUE
00137     71*          DO 20 I = 3, NPLE
00142     72*       20 PLS(I) = (0.0,0.0)
00142     73*    C
00142     74*    C *** COMPUTE RESIDUES, SUM OF RESIDUES AND LAMBDA'S
00144     75*          J = 3
00145     76*          IF(NPOLY .EQ. 0)
00145     77*         $I = 1
00147     78*          CALL ILTF(0,I,TINF,NPOLY,CINF,JERR)
00150     79*          K = NPLOR - 1
00151     80*          IF(K .LE. 0)
00151     81*         $GO TO 50
00153     82*          DO 40 I = 1, K
00156     83*          L = NZP + I
00157     84*          TABLE(L) = (0.0,0.0)
00160     85*          DO 40 J = 1, NZP
00163     86*       40 TABLE(K) = TABLE(K) - PLS(J)**(I-1) * TABLE(J)
00166     87*       50 IF(K .EQ. 0)
00166     88*         $TABLE(NPOL) = GKRL / GAM
00170     89*          CKSUM = (0.0,0.0)
00171     90*          IF(NPLOR .NE. 0)
00171     91*         $CKSUM = TABLE(NZP + 1)
00173     92*          DO 60 I = 1, NZP
00176     93*       60 CKSUM = CKSUM + TABLE(I)
00176     94*    C
00176     95*    C *** PRINT TIME RESPONSE HEADER PAGE
00200     96*          WRITE(6,70) GKB, GKRL, GAM, TCN, TOFF, DELT
00210     97*       70 FORMAT(///21X,9HBODE GAIN,18X,15HROOT LOCUS GAIN,20X,5HGAMMA/3(20X
00210     98*         $,1PE10.4)///20X,10HSTART TIME,21X,8HEND TIME,22X,9HINCREMENT/3(20X
00210     99*         $,E10.4)///24X,19HSUM OF THE RESIDUES,29X,8HLAMBDA'S)
00211    100*          IF(NPLOR .EQ. 0)
00211    101*         $GO TO 100
00213    102*          L = NZP + 1
00214    103*          WRITE(6,80) CKSUM, TABLE(L)
00220    104*       80 FORMAT(2X,2(20X,1H(1PE10.4,1H,E10.4,1H)))
00221    105*          IF(K .EQ. 0)
00221    106*         $GO TO 200
00223    107*          DO 90 J = 2, NPLOR
00226    108*          L = L + 1
00227    109*       90 WRITE(6,95) TABLE(L)
00233    110*       95 FORMAT(65X,1H(1PE10.4,1H,E10.4,1H))
```

```
00234   111*          GO TO 200
00235   112*       100 WRITE(6,150) CRSCV
00240   113*       150 FORMAT(22X,1H(1PE10.4,1H,E10.4,1H))
00241   114*       200 WRITE(6,250)
00243   115*       250 FORMAT(///19X 5HZEROS 23X 14HNON-ZERO POLES 22X 8HRESIDUES)
00244   116*           IF(NZER .EQ. 0)
00244   117*          $GO TO 350
00246   118*           WRITE(6,300) (ZRS(I), PLS(I), TABLE(I), I = 1, NZER)
00256   119*       300 FORMAT(3(10X,1H(,1PE10.4,1H,E10.4,1H)))
00257   120*       350 IF(NZP .EQ. NZER)
00257   121*          $GO TO 450
00261   122*           J = NZER + 1
00262   123*           WRITE(6,400) (PLS(I), TABLE(I), I = J, NZP)
00271   124*       400 FORMAT((43X,2(1H(,1PE10.4,1H,E10.4,1H),10X)))
00272   125*       450 IF(NPOLY .EQ. 0)
00272   126*          $GO TO 600
00274   127*           WRITE(6,500)
00276   128*       500 FORMAT(///37X,36HPIECEWISE QUADRATIC DRIVING FUNCTION/14X,11HBREAK
00276   129*          $-POINT,14X,8HCONSTANT,17X,6HLINEAR,17X,9HQUADRATIC/17X,4HTIME,17X,
00276   130*          $3(11HCOEFFICIENT,13X))
00277   131*           WRITE(6,550) (TINF(J), (CINF(I,J), I = 1, 3), J = 1, NPOLY)
00311   132*       550 FORMAT(4(14X,1PE10.4))
00312   133*       600 IF(JERR .EQ. 0)
00312   134*          $GO TO 700
00314   135*           WRITE(6,650)
00316   136*       650 FORMAT(63HOAN ERROR OCCURRED IN CALCULATING THE RESIDUES AND/OR LA
00316   137*          $MBDA'S./5CH CALCULATION OF THE TIME RESPONSE WILL BE ABORTED.)
00317   138*           GO TO 1000
00317   139*     C
00317   140*     C *** CALCULATE AND PRINT TIME RESPONSE
00320   141*       700 WRITE(6,750)
00322   142*       750 FORMAT(1H1,16X,4HTIME,18X,8HRESPONSE,17X,7HINVERSE,17X,7HDRIVING/
00322   143*          $15X,9H(SECONDS),15X,8HFUNCTION,16X,9HTRANSFORM,15X,8HFUNCTION)
00323   144*           J = 3
00324   145*           IF(NPOLY .EQ. 0)
00324   146*          $J = 1
00326   147*       800 DO 850 I = 1, J
00331   148*           CALL ILTF(I,C,TINF,NPOLY,CINF,JERR)
00332   149*       850 IF(JERR .NE. 0)
00332   150*          $WRITE(6,900)
00336   151*       900 FORMAT(3X,9(1H*)3X,43HERROR OCCURRED IN TIME RESPONSE CALCULATION)
```

```
00337   152*          IF(NPOLY .EQ. 0)
00337   153*         $GO TO 950
00341   154*          WRITE(6,550) T, YOFT, HOFT, XOFT
00347   155*          IF(PLOT .NE. 0)
00347   156*         $WRITE(8) T, YOFT, HOFT, XOFT
00356   157*          GO TO 975
00357   158*      950 WRITE(6,550) T, HOFT, HOFT, DUMMY4
00365   159*          IF(PLOT .NE. 0)
00365   160*         $WRITE(8) T, HOFT, HOFT, DUMMY4
00374   161*      975 T = T + DELT
00375   162*          IF(T .LE. TOFF)
00375   163*         $GO TO 800
00377   164*          IF(PLOT .NE. 0)
00377   165*         $END FILE 8
00401   166*          GO TO 1000
00402   167*      999 IERR = 1
00403   168*     1000 RETURN
00404   169*          END
```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 1 *DIAGNOSTIC* MESSAGE(S)

& FOR ILTF,ILTF
UNIVAC 1108 FORTRAN V LEVEL 22C6 3C13 F[illegible]
THIS COMPILATION WAS DONE ON 1[illegible] DEC 65 AT [illegible]3621834

[illegible]3621834

SUBROUTINE ILTF ENTRY POINT 001304

STORAGE USED (BLOCK, NAME, LENGTH)

0001	*CODE	001357
0000	*DATA	000572
0002	*BLANK	000000
0003	BASIC	000007
0004	COMPLX	000764
0005	ILTCOM	000317

EXTERNAL REFERENCES (BLOCK, NAME)

0006	SER
0007	SCALE
0010	EMU
0011	CDV$
0012	NEXP5$
0013	NERR2$
0014	NEXPD$
0015	NERR3$

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	001255	100L	0001	000250	121L	0001	000130	125G	0001	000141	134G	0001	000176	142G
0001	000234	156G	0001	000317	173L	0001	000332	171G	0001	000410	195L	0001	000344	200G
0001	000427	200L	0001	000350	204G	0001	000522	210L	0001	000450	222G	0001	000507	226G
0001	000564	230L	0001	000534	242G	0001	000573	252L	0001	000640	263G	0001	000172	30L
0001	000611	300L	0001	000677	301G	0001	000716	306G	0001	000625	310L	0001	000225	35L
0001	000630	350L	0001	000631	361L	0001	001235	363G	0001	000660	370L	0001	000663	380L
0001	000665	390L	0001	001036	423L	0001	001135	470L	0001	001143	480L	0001	001151	490L
0001	001163	500L	0001	001174	513L	0001	001176	520L	0001	001217	530L	0001	001253	999L

```
0004 C 000310 A          0000 C 000001 ARG1       0000 I 000000 ARG2       0004 C 000000 B          0000 R 000015 C
0000 R 000500 DTS        0004 R 000620 DUMMY      0005 C 000000 G          0003 R 000004 GAM        0003 R 000002 GKB
0003 R 000003 GKRL       0005 C 000310 H          0000 I 000471 I          0000 I 000474 J          0000 I 000475 K
0000 I 000472 KWS2       0000 I 000476 L          0000 I 000477 M          0003 I 000005 MU         0000 I 000473 N
0003 I 000001 NPOL       0003 I 000006 NZERO      0005 I 000316 P          0003 I 000000 Q          0006 C 000000 SER
0005 R 000315 T          0000 C 000003 WS2        0000 C 000013 WS200      0000 C 000005 WS3        0000 R 000501 WS4
0000 C 000007 WS5        0000 R 000502 WS6        0000 C 000011 WS7        0005 R 000314 X          0005 C 000312 Y
```

```
00100     1*   C
00100     2*   C ******************************************************************************
00100     3*   C .LSD SUBROUTINE ILTF-- MODIFIED BY C. ARGILA, NOV. '68                       .
00100     4*   C .                                                                            .
00100     5*   C .   DESCRIPTION-                                                            .
00100     6*   C .      GIVEN THE POLES, ZEROS AND ROOT LOCUS GAIN WHICH DEFINE A TRANSFER    .
00100     7*   C .   FUNCTION, THIS ROUTINE COMPUTES ONE POINT IN THE TIME RESPONSE OF THE    .
00100     8*   C .   CORRESPONDING SYSTEM.  THIS ROUTINE ALSO COMPUTES THE RESIDUES OF THE    .
00100     9*   C .   POLES OF THE TRANSFER FUNCTION AND CONSTRUCTS A PIECE-WISE POLYNOMIAL    .
00100    10*   C .   DRIVING FUNCTION.                                                        .
00100    11*   C .                                                                            .
00100    12*   C .   ARGUMENTS-                                                              .
00100    13*   C .      F = 0, COMPUTE RESIDUES ONLY                                        .
00100    14*   C .        = 1, COMPUTE TRANSFER FUNCTION ONLY                               .
00100    15*   C .        = 2, COMPUTE RESPONSE ONLY                                        .
00100    16*   C .        = 3, COMPUTE DRIVING FUNCTION ONLY                                .
00100    17*   C .      ICG = 0, DO NOT COMPUTE RESIDUES                                    .
00100    18*   C .          = 1, DO NOT COMPUTE DRIVING FUNCTION COEFFICIENTS               .
00100    19*   C .          = 3, COMPUTE AS INDICATED BY F                                  .
00100    20*   C .      TS     ARRAY OF DRIVING FUNCTION TIMES                             .
00100    21*   C .      NN     NUMBER OF POLYNOMIAL SEGMENTS IN DRIVING FUNCTION           .
00100    22*   C .      ALPHA  ARRAY OF POLYNOMIAL COEFFICIENTS                            .
00100    23*   C .      IERR   ERROR FLAG                                                  .
00100    24*   C .                                                                        .
00100    25*   C .   NAMED COMMON USAGE-                                                  .
00100    26*   C .      BASIC                                                             .
00100    27*   C .      COMPLX                                                            .
00100    28*   C .      ILTCOM                                                            .
00100    29*   C .                                                                        .
00100    30*   C .   SUBROUTINE USAGE-                                                   .
```

```
00100    31*   C .      SER                                                  .
00100    32*   C .      SCALE                                                .
00100    33*   C .      EMU                                                  .
00100    34*   C .                                                           .
00100    35*   C ...........................................................
00100    36*   C
00101    37*         SUBROUTINE ILTF(F,ICG,TS,NN,ALPHA,IERR)
00103    38*         INTEGER ARG2   , F      , P      , Q
00104    39*         COMPLEX A      , ARG1   , B      , G      , H      , SER   , WS2    ,
00104    40*        $WS3    , WS5    , WS7    , WS200 , Y
00105    41*         COMMON /BASIC / Q      , NPOL   , GKB    , GKRL   , GAM    , MU     ,
00105    42*        $NZEOR
00106    43*         COMMON /CCMPLX/ B(100), A(100), DUMMY(100)
00107    44*         COMMON /ILTCOM/ G(100), H      , Y      , X      , T      , P
00110    45*         DIMENSION       ALPHA(3,100)  , C(3,100)       , TS(100)
00110    46*   C
00110    47*   C *** INITIALIZATION
00111    48*         IERR = 0
00111    49*   C
00111    50*   C *** CHECK FOR VALID ARGUMENTS
00112    51*         IF(NN .LT. 0 .OR. (NN .EQ. 0 .AND. (F .GE. 2 .OR. ICG .GE. 2)))
00112    52*        $GO TO 999
00114    53*         IF((ICG .EQ. 0 .AND. (F .EQ. 2 .OR. F .EQ. 3))
00114    54*        $GO TO 200
00116    55*         IF(P .LT. 0 .OR. (P .EQ. 0 .AND. MU .LE. 0))
00116    56*        $GO TO 999
00120    57*         IF((ICG .EQ. 0)
00120    58*        $GO TO 200
00120    59*   C
00120    60*   C *** COMPUTE RESIDUES OF THE NON-ZERO POLES
00122    61*         IF(P .EQ. 0)
00122    62*        $GO TO 170
00124    63*         DO 130 I = 1, P
00127    64*         WS2 = (1.0,0.0)
00130    65*         KWS2 = 0
00131    66*         IF(Q .EQ. 0)
00131    67*        $GO TO 30
00133    68*         DO 25 N = 1, Q
00136    69*         CALL SCALE(WS2,KWS2)
00137    70*      25 WS2 = WS2 * (B(I) - A(N))
00141    71*      30 DO 35 J = 1, P
```

```
00144    72*          WS3 = B(I) - B(J)
00145  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00145    73*          IF(WS3 .EQ. (0.0,0.0))
00145    74*         $GO TO 35
00147    75*          CALL SCALE(WS2,KWS2)
00150    76*          WS2 = WS2 / WS3
00151    77*       35 CONTINUE
00153    78*          IF(MU .EQ. 0)
00153    79*         $GO TO 120
00155    80*          DO 115 K = 1, MU
00160    81*          CALL SCALE(WS2,KWS2)
00161    82*      115 WS2 = WS2 / B(I)
00163    83*      120 WS2 = WS2 * 1.0E+10**KWS2
00164    84*      130 G(I) = WS2 * GKRL
00164    85*    C
00164    86*    C *** COMPUTE THE COEFFICIENTS IN THE PARTIAL FRACTION EXPANSION ABOUT (0,0)
00166    87*      170 IF(MU .EQ. 0)
00166    88*         $GO TO 200
00170    89*          DO 180 L = 1, MU
00173    90*      180 G(P + L) = (0.0,0.0)
00175    91*          IF(P .EQ. 0)
00175    92*         $GO TO 195
00177    93*          DO 190 I = 1, P
00202    94*          WS2 = -G(I)
00203    95*          DO 190 J = 1, MU
00206    96*          K = P + J
00207    97*          G(K) = G(K) + WS2
00210    98*      190 WS2 = WS2 * B(I)
00213    99*      195 IF(P .NE. 0)
00213   100*         $GO TO 200
00215   101*          K = P + MU
00216   102*          G(K) = G(K) + GKRL
00216   103*    C
00216   104*    C *** COMPUTE DRIVING FUNCTION COEFFICIENTS
00217   105*      200 IF(ICG .LE. 1 .OR. NN .EQ. 0)
00217   106*         $GO TO 300
00221   107*          DO 250 M = 1, NN
00224   108*          I = 1
00225   109*          DO 250 L = 1, 3
00230   110*          C(L,M) = 0.0
00231   111*          IF(M .NE. 1)
```

```
00231   112*          $GO TO 210
00233   113*           C(L,1) = (ALPHA(L,1) - C(L,1)) * I
00234   114*           GO TO 250
00235   115*       210 K = 3 - L
00236   116*           IF(K .EQ. 0)
00236   117*          $GO TO 230
00240   118*           DTS = TS(M) - TS(M-1)
00241   119*           DO 220 J = 1, K
00244   120*           N = 4 - J
00245   121*       220 C(L,M) = (C(L,M) + ALPHA(N,M-1)) * DTS * (3 - J) / (4 - J - L)
00247   122*       230 C(L,M) = (ALPHA(L,M) - ALPHA(L,M-1) - C(L,M)) * I
00250   123*       250 I = I * L
00250   124*     C
00250   125*     C *** COMPUTE TRANSFER FUNCTION AND RESPONSE FUNCTION
00253   126*       300 IF(F .NE. 0)
00253   127*          $GO TO (310,350,500), F
00255   128*           GO TO 1000
00256   129*       310 WS4 = 1.0
00257   130*           GO TO 360
00260   131*       350 WS4 = 0.0
00261   132*       360 WS5 = (0.0,0.0)
00262   133*           DO 490 J = 1, NN
00265  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00265   134*           IF(WS4 .EQ. 1.0)
00265   135*          $GO TO 380
00267   136*           WS6 = T - TS(J)
00270   137*           IF(WS6)
00270   138*          $ ,370,390
00273   139*           IF(J .LE. 1)
00273   140*          $GO TO 999
00275   141*       370 Y = WS5
00276   142*           GO TO 1000
00277   143*       380 WS6 = T
00300   144*       390 DO 490 L = 1, 3
00303   145*           IF(P .EQ. 0)
00303   146*          $GO TO 420
00305   147*           DO 410 K = 1, P
00310   148*           ARG1 = B(K) * WS6
00311   149*           ARG2 = MU
00312  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00312   150*           IF(WS4 .EQ. 0.0)
```

```
00312   151*          $ARG2 = ARG2 + L
00314   152*           CALL EMU(ARG1,ARG2,wS7)
00315  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00315   153*           IF(wS4 .NE. 1.0)
00315   154*          $wS7 = wS7 * C(L,J) / G(K)**L
00317   155*      410 wS5 = wS5 + wS7 * G(K)
00321   156*      420 IF(P .NE. 0)
00321   157*          $GO TO 480
00323   158*           wS200 = WS6
00324   159*           K = MU - 1
00325  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00325   160*           IF(wS4 .EQ. 0.0)
00325   161*          $GO TO 470
00327   162*           wS5 = wS5 + GKRL * SER(wS200,K,K)
00330   163*           GO TO 480
00331   164*      470 K = K + L
00332   165*           wS5 = wS5 + C(L,J) * GKRL * SER(WS200,K,K)
00333  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00333   166*      480 IF(WS4 .EQ. 0.0)
00333   167*          $GO TO 490
00335   168*           H = wS5
00336   169*           GO TO 1000
00337   170*      490 CONTINUE
00342   171*           Y = WS5
00343   172*           GO TO 1000
00343   173*    C
00343   174*    C *** COMPUTE DRIVING FUNCTION
00344   175*      500 IF(T - TS(1))
00344   176*          $999, ,510
00347   177*           X = ALPHA(1,1)
00350   178*           GO TO 1000
00351   179*      510 I = 1
00352   180*      520 IF(TS(I) .GE. T)
00352   181*          $GO TO 530
00354   182*           I = I + 1
00355   183*           IF(I - NN -1)
00355   184*          $520,520,999
00360   185*      530 X = ALPHA(3,I-1)
00361   186*           DTS = T - TS(I - 1)
00362   187*           DO 540 J = 1, 2
00365   188*           L = 3 - J
```

```
00366   189*      540 X = X * DTS + ALPHA(L,I-1)
00370   190*          GO TO 1000
00371   191*      999 IERR = 1
00372   192*     1000 RETURN
00373   193*          END
```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 6 *DIAGNOSTIC* MESSAGE(S)

```
                SUBROUTINE MATVAR                                   DATE  101268  PAGE      1

& FOR MATVAR,MATVAR                                                     13621637
UNIVAC 1108 FORTRAN V LEVEL   22.0 601.0 FC.14M
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 13621637

   SUBROUTINE MATVAR     ENTRY POINT 000252

   STORAGE USED (BLOCK, NAME, LENGTH)

        0001    *CODE    000271
        0000    *DATA    000053
        0002    *BLANK   000000


   EXTERNAL REFERENCES (BLOCK, NAME)

        0003    FMASK
        0004    NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000061 107G       0001   000062 112G       0001   000140 123G       0001   000141 126G       0001   000077 2L
    0003 R 000000 FMASK      0000 I 000003 I          0000 I 000004 J          0000 I 000002 K          0000 C 000000 Z



000100     1*   C
000100     2*   C ******************************************************************
000100     3*   C .LSD SUBROUTINE MATVAR -- MODIFIED BY C. ARGILA, NOV. '68        .
000100     4*   C .                                                                .
000100     5*   C .   DESCRIPTION-                                                 .
000100     6*   C .      THIS ROUTINE ACCEPTS A REAL ARRAY, A, DIMENSIONED (N1X,N1X,N2+1) AND  .
000100     7*   C .   A COMPLEX CONSTANT, LAMBDA, AND SETS EACH ELEMENT OF A COMPLEX ARRAY, B, .
000100     8*   C .   AS FOLLOWS-                                                  .
000100     9*   C .      B(I,J) = A(I,J,N2+1) + LAMBDA*A(I,J,N2) + LAMBDA**2 * A(I,J,N2-1)     .
000100    10*   C .             +...+ LAMBDA**N2 * A(I,J,1),  I, J = 1,...,N1      .
000100    11*   C .                                                                .
```

```
00100    12*   C .   ARGUMENTS-                                                .
00100    13*   C .      LAMBDA                                                 .
00100    14*   C .      A                                                      .
00100    15*   C .      N1                                                     .
00100    16*   C .      N2                                                     .
00100    17*   C .      N1X                                                    .
00100    18*   C .      B                                                      .
00100    19*   C .                                                             .
00100    20*   C .............................................................
00100    21*   C
00101    22*         SUBROUTINE MATVAR(LAMBDA,A,N1,N2,N1X,B)
00103    23*         COMPLEX LAMBDA, B(N1X,1), Z
00104    24*         DIMENSION A(N1X,N1X,1)
00105    25*         K = N2 + 1
00106    26*         DO 1 I = 1, N1
00111    27*         DO 1 J = 1, N1
00114    28*       1 B(I,J) = A(I,J,K)
00117    29*         Z = (1.0,0.0)
00120    30*       2 K = K - 1
00121    31*         Z = Z * LAMBDA
00122    32*         DO 3 I = 1, N1
00125    33*         DO 3 J = 1, N1
00130    34*       3 B(I,J) = B(I,J) + Z * A(I,J,K)
00133    35*         B(I,J) = CMPLX(FMASK(REAL(B(I,J))),FMASK(AIMAG(B(I,J))))
00134    36*         IF(K .GT. 1)
00134    37*        $GO TO 2
00136    38*         RETURN
00137    39*         END
```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

```
                SUBROUTINE MTRXPR                                   DATE  101268  PAGE      1

& FOR MTRXPR,MTRXPR                                                 03621638
UNIVAC 1108 FORTRAN V LEVEL   2206 0018 F501RH
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03621639


   SUBROUTINE MTRXPR    ENTRY POINT 000130

   STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE   000145
        0000   *DATA   000045
        0002   *BLANK  000000


   EXTERNAL REFERENCES (BLOCK, NAME)

        0003   NWDU$
        0004   NIO1$
        0005   NIO2$
        0006   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000   000005 1F         0001   000017 106G       0001   000053 113G       0001   000067 121G       0001   000077 126G
    0000   000007 3F         0000 I 000004 I          0000 I 000003 J          0000 I 000001 K          0000 I 000002 L
    0000 I 000000 NLAST


00100      1*   C
00100      2*   C ..........................................................................
00100      3*   C .LSD SUBROUTINE MTRXPR -- MODIFIED BY C. ARGILA, NOV. '68                .
00100      4*   C .                                                                        .
00100      5*   C .   DESCRIPTION-                                                        .
00100      6*   C .      THIS ROUTINE PERFORMS THE MATRIX PRINT OUT FOR SUBROUTINE COA OF THE  .
00100      7*   C .   LSD PROGRAM.                                                        .
00100      8*   C .                                                                        .
```

```
00100     9*    C .   ARGUMENTS-                                                .
00100    10*    C .      A      ARRAY DIMENSIONED (M,M)                         .
00100    11*    C .      M                                                      .
00100    12*    C .      N                                                      .
00100    13*    C .                                                             .
00100    14*    C ..............................................................
00100    15*    C
00101    16*          SUBROUTINE MTRXPR(A,M,N)
00103    17*          DIMENSION A(M,M)
00104    18*          NLAST = N - 1
00105    19*          DO 2 K = 1, N, 9
00110    20*          L = MIN0(N-K,8) + K
00111    21*          WRITE(6,1) (J, J = K, L)
00117    22*        1 FORMAT(1H0 9(8X I3))
00120    23*          DO 2 I = 1, NLAST
00123    24*        2 WRITE(6,3) I, (A(I,J), J = K, L)
00134    25*        3 FORMAT(I4,1X 9(1X 1PE10.4))
00135    26*          RETURN
00136    27*          END
```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

```
                SUBROUTINE ORDER                                        DATE  101268  PAGE      1

& FCR ORDER,ORDER                                                       03621640
UNIVAC 1108 FORTRAN V LEVEL   2206 3018 F5318H
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03621640

   SUBROUTINE ORDER      ENTRY POINT 000140

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001   *CODE    000160
         0000   *DATA    000027
         0002   *BLANK   000000


   EXTERNAL REFERENCES (BLOCK, NAME)

         0003   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000007 1L         0001   000124 1CL        0001   000042 115G       0001   000111 132G       0001   000116 135G
    0001   000066 3L         0001   000070 4L         0000 I 000000 I          0000 I 000002 J          0000 I 000001 M


001CC      1*   C
001CC      2*   C .....................................................................
0010C      3*   C .   SUBROUTINE ORDER                          C. ARGILA, OCT. '68    .
001CC      4*   C .                                                                   .
001CC      5*   C .   DESCRIPTION-                                                    .
00100      6*   C .      THIS ROUTINE ACCEPTS AN ARRAY OF COMPLEX NUMBERS AND REORDERS IT .
00100      7*   C .   SO THAT ALL ZERO ELEMENTS OCCUR FIRST.  ONLY A SPECIFIED NUMBER OF THESE .
00100      8*   C .   LEADING ZERO ELEMENTS WILL BE RETAINED.                         .
C01CC      9*   C .                                                                   .
0010C     10*   C .   ARGUMENTS-                                                      .
C0100     11*   C .      ARRAY ARRAY OF COMPLEX NUMBERS                             .
001CC     12*   C .      NZER  NUMBER OF LEADING ZEROS TO BE RETAINED               .
```

```
00100    13*    C .      NPOINT DIMENSION OF A                                  .
00100    14*    C .                                                             .
00100    15*    C ..............................................................
00100    16*    C
00101    17*          SUBROUTINE ORDER(ARRAY,NZER,NPOINT)
00103    18*          COMPLEX ARRAY(NPOINT)
00104    19*          I = 0
00105    20*        1 I = I + 1
00106    21*          IF(I - NPOINT) 2, 4, 4
00111  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00111    22*        2 IF(ARRAY(I) .EQ. (0.0,0.0)) GO TO 1
00113    23*          M = I + 1
00114    24*          DO 3 J = M, NPOINT
00117  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00117    25*          IF(ARRAY(J) .NE. (0.0,0.0)) GO TO 3
00121    26*          ARRAY(J) = ARRAY(I)
00122    27*          ARRAY(I) = (0.0,0.0)
00123    28*          I = I + 1
00124    29*        3 CONTINUE
00126    30*        4 M = I - NZER - 1
00127    31*          IF(M .LE. 0) GO TO 10
00131    32*          DO 6 I = 1, M
00134    33*          DO 5 J = 2, NPOINT
00137    34*        5 ARRAY(J-1) = ARRAY(J)
00141    35*        6 ARRAY(NPOINT) = (0.0,0.0)
00143    36*       10 RETURN
00144    37*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     2 *DIAGNOSTIC* MESSAGE(S)
```

```
                    SUBROUTINE OUT                                  DATE  101268  PAGE      1

& FOR OUT,OUT                                                          03621641
UNIVAC 1108 FORTRAN V LEVEL   2206 (018 P5018H
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03621641


   SUBROUTINE OUT        ENTRY POINT 000120

   STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE   000133
        0000   *DATA   000102
        0002   *BLANK  000000


   EXTERNAL REFERENCES (BLOCK, NAME)

        0003   NWDU$
        0004   NIO1$
        0005   NIO2$
        0006   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000   000004 1F        0001   000076 10L       0000   000052 11F       0001   000034 116G      0001   000035 120G
    0001   000067 136G      0001   000023 2L        0000   000024 3F        0000   000032 4F        0000   000041 5F
    0001   000073 9L        0000 I 000001 I         0000 I 000003 I$        0000 I 000002 J         0000 I 000000 N



00100     1*   C
00100     2*   C .....................................................................
00100     3*   C .LSD SUBROUTINE OUT -- MODIFIED BY C. ARGILA, NOV. '68              .
00100     4*   C .                                                                   .
00100     5*   C .   DESCRIPTION-                                                   .
00100     6*   C .      THIS ROUTINE PERFORMS THE PRINT OUTPUT OF EIGENVALUES AND EIGENVECTORS.
00100     7*   C .   FOR THE LSD SUBROUTINE EIGEN.                                  .
00100     8*   C .                                                                   .
```

```
00100     9*    C .   ARGUMENTS-                                                .
00100    10*    C .      ITPAR                                                  .
00100    11*    C .      X                                                      .
00100    12*    C .      STOR                                                   .
00100    13*    C .      KAP                                                    .
00100    14*    C .                                                             .
00100    15*    C ..............................................................
00100    16*    C
00101    17*          SUBROUTINE OUT(ITPAR,X,STOR,KAP)
00103    18*          COMPLEX STOR(3,1), X(KAP)
00104    19*          DIMENSION ITPAR(3)
00105    20*          N = ITPAR(1)
00106    21*          IF(N .EQ. 2)
00106    22*         $GO TO 10
00110    23*          WRITE(6,1) ITPAR(2)
00113    24*        1 FORMAT(//41X   23HITERANTS FOR EIGENVALUE,I3//19X,6HLAMBDA,23X,16H
00113    25*         $F SUB P (LAMBDA),18X,13HDET A(LAMBDA))
00114    26*        2 WRITE(6,3) ((STOR(4-I,J),I=1,3),J=1,N)
00125    27*        3 FORMAT(3(10X1H(1PE10.4,1H,E10.4,1H)))
00126    28*          IF(ITPAR(3) .EQ. 1)
00126    29*         $WRITE(6,4)
00131    30*        4 FORMAT(3X 9(1H*) 3X 25HCOUNTER HAS BEEN EXCEEDED)
00132    31*          IF(N .EQ. 2)
00132    32*         $GO TO 9
00134    33*          WRITE(6,5) X
00142    34*        5 FORMAT(1HO 46X 11HEIGENVECTOR/(43X 1H(1PE10.4,1H,E10.4,1H)))
00143    35*        9 RETURN
00144    36*       10 WRITE(6,11)
00146    37*       11 FORMAT(1HO,47X,14HACCURACY CHECK)
00147    38*          GO TO 2
0015C    39*          END
```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

SUBROUTINE ROOT DATE 101268 PAGE 1

& FOR ROOT,ROOT 3621642
UNIVAC 1108 FORTRAN V LEVEL 22C6 0C1B F3C1F4
THIS COMPILATION WAS DONE ON 1B DEC 68 AT 13621643

SUBROUTINE ROOT ENTRY POINT 001361

STORAGE USED (BLOCK, NAME, LENGTH)

0001	*CODE	001404
0000	*DATA	000134
0002	*BLANK	000000

EXTERNAL REFERENCES (BLOCK, NAME)

0003	ABC
0004	AUXSUB
0005	FMASK
0006	DDV$
0007	DSQRT
0010	NERR3$

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	000016	10L	0001	000471	153L	0001	001064	1C8L	0001	001112	11CL	0001	001230	135L
0001	000140	137G	0001	001262	137L	0001	000157	143G	0001	000231	161G	0001	000265	167G
0001	000310	176G	0001	000021	20L	0001	000344	207G	0001	000420	217G	0001	000034	22L
0001	000057	25L	0001	001215	255G	0001	001245	271G	0001	000200	35L	0001	000207	45L
0001	000260	47L	0001	000331	60L	0001	000361	65L	0001	001343	999L	0003 R	000000	ABC
0000 C	000002	AI	0000 R	000761	BETA	0000 C	000004	BI	0000 C	000006	CI	0000 C	000010	CONS
0000 C	000016	D	0000 C	000020	DP1	0000 C	000022	DI	0000 R	000063	E	0000 R	000052	EPS
0000 C	000024	F	0000 C	000032	FBAR	0005 R	000000	FMASK	0000 R	000065	H	0000 I	000064	I
0000 I	000057	I$	0000 I	000062	J	0000 C	000040	L	0000 C	000046	NEW	0000 I	000060	NP
0000 I	000000	P	0000 I	000001	RDELG	0000 C	000050	U						

```
00100     1*   C
00100     2*   C .....................................................................
00100     3*   C .LSD SUBROUTINE ROOT --  MODIFIED BY C. ARGILA, OCT. '68           .
00100     4*   C .                                                                  .
00100     5*   C .   DESCRIPTION-                                                  .
00100     6*   C .      THIS ROUTINE COMPUTS THE ROOTS OF THE ROOT LOCUS FUNCTION FOR THE .
00100     7*   C .   LSD PROGRAM.                                                  .
00100     8*   C .                                                                  .
00100     9*   C .   ARGUMENTS-                                                    .
00100    10*   C .      A       ARRAY OF ESTIMATES FOR THE ROOTS (COMPLEX)        .
00100    11*   C .      Z       ARRAY OF ROOTS (COMPLEX)                          .
00100    12*   C .      NROOTS  NUMBER OF ROOTS                                   .
00100    13*   C .      IERR    ERROR FLAG                                        .
00100    14*   C .                                                                  .
00100    15*   C .   SUBROUTINE USAGE-                                            .
00100    16*   C .      ABC                                                       .
00100    17*   C .      AUXSUB                                                    .
00100    18*   C .      FMASK                                                     .
00100    19*   C .                                                                  .
00100    20*   C .   REMARKS-                                                      .
00100    21*   C .      THIS ROUTINE IS A VERSION OF TRW'S GENERAL ROOT FINDING PROGRAM .
00100    22*   C .   SPECIFIC TO THE LSD PROGRAM.                                  .
00100    23*   C .                                                                  .
00100    24*   C .....................................................................
00100    25*   C
00101    26*         SUBROUTINE ROOT(A,Z,NROOTS,IERR)
00103    27*         INTEGER P, RDFLG
00104    28*         COMPLEX A(NROOTS), AI, BI, CI, CONS(3), D, DP1, D1, F(3), FBAR(3),
00104    29*        $ L(3), NEW, U, Z(NROOTS)
00105    30*         DIMENSION EPS(5)
00106    31*         DATA CONS, EPS /(0.0,0.0), (1.0,0.0), (-0.5,0.0), 1.0E-07, 1.0E-02
00106    32*        $ , 1.0E-01, 1.0E+05, 1.0E-03/
00106    33*   C
00106    34*   C *** CHECK FOR VALID ARGUMENTS
00111    35*         IF(NROOTS .NE. 0) GO TO 10
00113    36*         IERR = 1
00114    37*         GO TO 999
00114    38*   C
00114    39*   C *** INITIALIZATION
00115    40*      10 IERR = 0
```

```
00116    41*          P = 1
00117    42*       20 RDFLG = 0
00120    43*          NP = P - 1
00121    44*          U = A(P)
00122    45*          BETA = EPS(3)
00123    46*       22 D = CONS(3)
00124    47*          DP1 = CONS(2) + CONS(3)
00125    48*          IF(RDFLG .NE. 0) GO TO 25
00127    49*          IF(ABS(U) .LT. EPS(1)) U = -BETA
00127    50*    C
00127    51*    C *** GENERATE THREE STARTING VALUES FOR ITERATION
00131    52*       25 L(1) = (1.0 + BETA) * U
00132    53*          L(2) = (1.0 - BETA) * U
00133    54*          L(3) = U
00134    55*          IF(P .EQ. 1) GO TO 45
00136    56*          DO 35 J = 1, 3
00141    57*          E = EPS(1) * AMAX1(1.0,ABS(L(J)))
00142    58*          DO 35 I = 1, NP
00145    59*          IF(ABS(L(J)-Z(I)) .GT. E) GO TO 35
00147    60*          U = L(1)
00150    61*          GO TO 25
00151    62*       35 CONTINUE
00154    63*       45 CALL AUXSUB(L(1),F(1))
00155    64*          CALL AUXSUB(L(2),F(2))
00156    65*          IF(P .EQ. 1) GO TO 47
00160    66*          DO 46 J = 1, NP
00163    67*          F(1) = F(1) / (L(1) -  Z(J))
00164    68*       46 F(2) = F(2) / (L(2) -  Z(J))
00164    69*    C
00164    70*    C *** ITERATE
00166    71*       47 DO 130 I = 1, 100
00171    72*          NEW = L(3)
00172    73*          IF(P .EQ. 1) GO TO 60
00174    74*          E = EPS(1) * AMAX1(1.0,ABS(NEW))
00175    75*          DO 55 J = 1, NP
00200    76*       55 IF(ABS(Z(J) - NEW) .LE. E) GO TO 135
00203    77*       60 CALL AUXSUB(NEW,F(3))
00204    78*          IF(P .EQ. 1) GO TO 65
00206    79*          DO 63 J = 1, NP
00211    80*       63 F(3) = F(3) / (NEW - Z(J))
00213  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
```

```
00213    81*       65 IF(ABC(F(3)) .EQ. 0.0) GO TO 137
00215    82*          H = 1.0 / AMAX1(ABC(F(1)),ABC(F(2)),ABC(F(3)))
00216    83*          DO 95 J = 1, 3
00221    84*       95 FBAR(J) = F(J) * H
00223    85*          IF(ABC(FBAR(1)) .LT. EPS(4) * ABC(FBAR(2))) GO TO 100
00225    86*          D = NEW
00226    87*          RDFLG = 1
00227    88*          BETA = BETA * EPS(2)
00230    89*          IF(BETA - EPS(1)) 135, 135, 22
00230    90*    C
00230    91*    C *** GET NEXT ITERANT
00233    92*      100 AI = D * (D * FBAR(1) - DP1 * FBAR(2) + FBAR(3))
00234    93*          BI = D**2 * FBAR(1) - DP1**2 * FBAR(2) + (1.0 + 2.0 * D) * FBAR(3)
00235    94*          CI = DP1 * FBAR(3)
00236    95*          D1 = CSQRT(BI**2 - 4.0 * AI * CI)
00237    96*          D = BI - D1
00240    97*          D1 = BI + D1
00241    98*          IF(ABC(D1) .GE. ABC(D)) D = D1
00243 *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00243    99*          IF(ABC(D) .NE. 0.0) GO TO 109
00245   100*          D = CONS(1)
00246   101*          GO TO 110
00247   102*      109 D = -2.0 * CI / D
00250   103*      110 NEW = L(3) + D * (L(3) - L(2))
00251   104*          IF(ABC(NEW) .LE. EPS(1) .OR. ABC(NEW-L(3))/AMAX1(1.0,ABC(NEW))
00251   105*         $ .LE. EPS(1)) GO TO 135
00253   106*          DP1 = D + CONS(2)
00254   107*          DO 125 J = 1, 2
00257   108*          L(J) = L(J+1)
00260   109*      125 F(J) = F(J+1)
00262   110*      130 L(3) = NEW
00262   111*    C
00262   112*    C *** SET ERROR FLAG IF ROOT WAS NOT DETERMINED CORRECTLY
00264   113*          IERR = 1
00265   114*      135 CALL AUXSUB(NEW,F(3))
00266   115*          IF(P .EQ. 1) GO TO 137
00270   116*          DO 136 J = 1, NP
00273   117*      136 F(3) = F(3) / (NEW - Z(J))
00275   118*      137 Z(P) = CMPLX(FMASK(REAL(NEW)),FMASK(AIMAG(NEW)))
00276   119*          P = P + 1
00277   120*          IF(P .GT. NROOTS) GO TO 999
```

```
00277   121*    C
00277   122*    C *** SET Z(P) = Z(P-1)*, IF Z(P-1) IS NOT REAL
00301  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00301   123*          IF(AIMAG(Z(P-1)) .EQ. 0.0) GO TO 20
00303   124*          Z(P) = CONJG(Z(P-1))
00304   125*          P = P + 1
00305   126*          IF(P .LE. NPOTS) GO TO 20
00307   127*      999 RETURN
00310   128*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     3 *DIAGNOSTIC* MESSAGE(S)
```

03&21&45

& FOR RTLC,RTLC
UNIVAC 1108 FORTRAN V LEVEL 22c6 CC18 F5C18H
THIS COMPILATION WAS DONE ON 1C DEC 68 AT 03&21&45

SUBROUTINE RTLC ENTRY POINT CC123C

STORAGE USED (BLOCK, NAME, LENGTH)

0001	*CODE	CC13C2
0000	*DATA	000752
0002	*BLANK	000000
0003	BASIC	000007
0004	COMPLX	000764
0005	EXTRA	000004
0006	OPTICN	000030

EXTERNAL REFERENCES (BLOCK, NAME)

0007	GCONJ
0010	RCOT
0011	ANGLE
0012	NEXP1$
0013	NWDU$
0014	NIO1$
0015	NIO2$
0016	NWBU$
0017	CSQRT
0020	CABS
0021	CDV$
0022	NWEF$
0023	NERR3$

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	000024 1L	0000	000417 100F	0001	000122 157G	0001	000140 165G	0001	000155 174G
0001	000164 20L	0000	000514 200F	0001	000230 212G	0001	000235 216G	0001	000253 226G

```
0001   000310 241G      0001   000316 25L       0001   000333 253G      0001   000363 274G      0001   000100 3L
0001   000351 30L       0000   000526 300F      0001   000425 315G      0001   000414 32L       0001   000462 336G
0001   000513 35L       0001   000520 357G      0001   000533 36L       0001   000540 370G      0001   000546 40L
0000   000533 400F      0001   000612 407G      0001   000622 414G      0001   000632 421G      0001   000636 45L
0001   000110 5L        0001   000653 50L       0000   000553 500F      0001   000745 54L       0001   000752 55L
0001   001007 56L       0001   000113 6L        0001   001064 60L       0000   000576 600F      0001   001101 65L
0001   001170 70L       0000   000627 700F      0000   000641 800F      0000   000655 900F      0001   001177 99L
0001   001202 999L      0011 R 000000 ANGLE     0000 C 000000 ASTER     0005 C 000002 BODEG     0000 C 000002 C
0000 C 000004 CC        0000 R 000415 CDELTA    0004 C 000620 CLP       0000 R 000332 DEN       0000 R 000342 DIPG
0006 R 000000 DUMMY     0000 R 000410 G         0000 R 000413 GAIN      0003 R 000004 GAM       0003 R 000002 GKB
0003 R 000003 GKRL      0000 R 000377 GMU       0000 R 000412 GPRIME    0000 R 000407 H         0000 I 000376 I
0006 I 000027 INPUT     0000 I 000402 IREC      0000 I 000372 IS        0000 I 000374 II        0000 I 000411 J
0000 I 000400 JERR      0000 I 000406 K         0000 I 000401 M         0000 I 000404 NCASE     0003 I 000005 NPLOR
0003 I 000001 NPOL      0000 I 000405 NU        0003 I 000006 NZEOR     0003 I 000000 NZER      0000 R 000414 PHI
0006 I 000026 PLOT      0004 C 000000 PLS       0001 R 000373 RADIAN    0000 C 000014 S         0005 C 000000 SIGMA
0000 C 000160 SPOLS     0000 R 000362 STEP      0000 C 000162 SZERS     0000 R 000403 TEMP      0000 R 000375 THETA
0000 R 000366 TRACE     0000 R 000416 XH        0000 C 000164 XS        0004 C 000310 ZRS       0000 C 000166 ZS
```



```
00100     1*   C
00100     2*   C ***********************************************************************
00100     3*   C .LSD SUBROUTINE RTLC -- MODIFIED BY C. ARGILA, OCT. '68              .
00100     4*   C .                                                                    .
00100     5*   C .   DESCRIPTION-                                                    .
00100     6*   C .      THIS ROUTINE PERFORMS THE ROOT LOCUS CALCULATIONS FOR THE LSD .
00100     7*   C .   PROGRAM.                                                        .
00100     8*   C .                                                                    .
00100     9*   C .   ARGUMENTS-                                                      .
00100    10*   C .      NCLPS  NUMBER OF CLOSED-LOOP POLES                          .
00100    11*   C .      IPHASE PHASE FLAG                                           .
00100    12*   C .             = 0, INDICATES ZERO DEGREE PHASE                     .
00100    13*   C .             = 1, INDICATES 180 DEGREE PHASE                      .
00100    14*   C .      TR1    UPPER GAIN LIMIT                                     .
00100    15*   C .      TR2    LOWER GAIN LIMIT                                     .
00100    16*   C .      TR3    UPPER PHASE LIMIT                                    .
00100    17*   C .      TR4    LOWER PHASE LIMIT                                    .
00100    18*   C .      ST1    UPWARD GAIN INCREMENT                                .
00100    19*   C .      ST2    DOWNWARD GAIN INCREMENT                              .
00100    20*   C .      ST3    UPWARD PHASE INCREMENT                               .
```

```
00100    21*    C .      ST4    DOWNWARD PHASE INCREMENT                          .
00100    22*    C .      IERR   ERROR FLAG                                        .
00100    23*    C .                                                               .
00100    24*    C .   PRINT OUTPUT-                                               .
00100    25*    C .      THE OPEN-LOOP POLES AND ZEROS, ESTIMATES FOR CLOSED-LOOP POLES,  .
00100    26*    C .   NOMINAL CLOSED-LOOP POLES AND VARIOUS PARAMETERS ARE PRINTED ON THE  .
00100    27*    C .   ROOT LOCUS HEADER PAGE.  FOLLOWING THE ROOT LOCUS HEADER PAGE, 4*N NUMBER.
00100    28*    C .   OF TRACES ARE PRINTED, THE FIRST N FOR INCREASING GAIN, THE NEXT N FOR  .
00100    29*    C .   DECREASING GAIN, NEXT INCREASING PHASE AND FINALLY DECREASING PHASE.    .
00100    30*    C .   N IS THE NUMBER OF NOMINAL CLOSED LOOP POLES OCCURING IN THE UPPER HALF  .
00100    31*    C .   OF THE COMPLEX PLANE.                                       .
00100    32*    C .                                                               .
00100    33*    C .   PLOT TAPE OUTPUT-                                           .
00100    34*    C .      WHEN PLOT = 1, N+4 RECORD TYPES WILL BE GENERATED WITHIN A FILE ON  .
00100    35*    C .   THE PLOT TAPE (UNIT 8).  ALL RECORDS ARE ONE DATA WORD LONG.        .
00100    36*    C .      RECORD TYPES 1-  CONTAIN ALL ABSCISSA POINTS TO BE PLOTTED     .
00100    37*    C .      RECORD TYPES 2-  CONTAIN ALL ORDINATE POINTS OF THE ZEROS      .
00100    38*    C .      RECORD TYPES 3-  CONTAIN ALL ORDINATE POINTS OF THE OPEN-LOOP POLES  .
00100    39*    C .      RECORD TYPES 4-  CONTAIN ALL ORDINATE POINTS OF THE NOMINAL CLOSED-  .
00100    40*    C .   -LOOP POLES                                                 .
00100    41*    C .      RECORD TYPES 5, 6,..., N+4 CONTAIN ALL ORDINATE POINTS OF THE N ROOT  .
00100    42*    C .   LOCUS BRANCHES.  THEY CORRESPOND, RESPECTIVELY, TO THE BRANCHES AS THEY  .
00100    43*    C .   ARE PRINTED OUT.                                            .
00100    44*    C .                                                               .
00100    45*    C .   NAMED COMMON USAGE-                                         .
00100    46*    C .      BASIC                                                    .
00100    47*    C .      COMPLX                                                   .
00100    48*    C .      EXTRA                                                    .
00100    49*    C .      OPTION                                                   .
00100    50*    C .                                                               .
00100    51*    C .   SUBROUTINE USAGE-                                          .
00100    52*    C .      ANGLE                                                    .
00100    53*    C .      GCONJ                                                    .
00100    54*    C .      ROOT                                                     .
00100    55*    C .                                                               .
00100    56*    C ...............................................................................
00100    57*    C
00101    58*          SUBROUTINE RTLC(NCLPS,IPHASE,TR1,TR2,TR3,TR4,ST1,ST2,ST3,ST4,IERR)
00103    59*          INTEGER PLOT
00104    60*          COMPLEX ASTER , BODES , C     , CC(4) , CLP   , PLS   , S(50) ,
00104    61*         $SIGMA , SPOLS , SZERS , XS    , ZRS   , ZS(50)
```

```
00105    62*          COMMON /BASIC / NZER  , NPOL  , GKB   , GKRL  , GAM   , NPLUS ,
00105    63*         $NZERO
00106    64*          COMMON /COMPLX/ PLS(100)      , ZRS(100)      , CLP(50)
00107    65*          COMMON /EXTRA / SIGMA         , PCSEC
00110    66*          COMMON /OPTION/ DUMMY(22)     , PLOT  , INPUT
00111    67*          DIMENSION       DEN(4,2)      , DIPS(4,4)     , STEP(4)       ,
00111    68*         $TRACE(4)
00112    69*          DATA DEN /48HDECIBELS FROM NOMINAL   DEGREES FROM NOMINAL    /
00114    70*          DATA DIPS /96HINCREASING GAIN TO      DECREASING GAIN TO  -   INCR
00114    71*         $EASING PHASE TO     DECREASING PHASE TO -   /
00116    72*          DATA CC, RADIAN, II /(1.122018,0.0), (.8912512,0.0), (.99939083,
00116    73*         $-.034899497), (.99939083,.034899497), 57.2957795, 1/
00116    74*    C
00116    75*    C *** CHECK FOR VALID PARAMETERS
00122 *DIAGNOSTIC* THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00122    76*          IF((GKB .NE. 0.0 .OR. GKRL .NE. 0.0) .AND. NPOL .GE. NZER)
00122    77*         $GO TO 1
00124    78*          IERR = 1
00125    79*          GO TO 999
00125    80*    C
00125    81*    C *** INITIALIZATION
00126    82*        1 IERR = 0
00127    83*          TRACE(1) = TR1
00130    84*          TRACE(2) = TR2
00131    85*          TRACE(3) = TR3
00132    86*          TRACE(4) = TR4
00133    87*          STEP(1) = ST1
00134    88*          STEP(2) = ST2
00135    89*          STEP(3) = ST3
00136    90*          STEP(4) = ST4
00137    91*          SIGMA = -(-1)**IPHASE
00140    92*          SPOLS = (0.0,0.0)
00141    93*          SZERS = (0.0,0.0)
00142    94*          THETA = 180 * IPHASE
00142    95*    C
00142    96*    C *** DETERMINE ROOT LOCUS AND BODE GAINS
00143    97*          IF(GKB) 2, 3, 2
00146    98*        2 GKB = SIGN(GKB,GAM)
00147    99*          GKRL = GAM * GKB
00147   100*    C
00147   101*    C *** LOAD TABLE OF ESTIMATES
```

```
00150   102*        3 IF(NCLPS) 4, 5, 4
00153   103*        4 CALL GCONJ(CLP,S,NCLPS)
00154   104*          GO TO 6
00155   105*        5 NCLPS = NPOL - NPLOR
00156   106*        6 DO 9 I = 1, NCLPS
00161  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00161   107*        9 IF(S(I) .EQ. (0.0,0.0))
00161   108*         $S(I) = PLS(I)
00161   109*   C
00161   110*   C *** COMPUTE NOMINAL CLOSED LOOP POLES
00164   111*          DO 10 I = 1, NPOL
00167   112*       10 SPOLS = SPOLS + PLS(I)
00171   113*          IF(NZER .EQ. 0) GO TO 20
00173   114*          DO 15 I = 1, NZER
00176   115*       15 SZERS = SZERS + ZRS(I)
00200   116*       20 GMU = REAL(SPOLS - SZERS) / (NPOL - NZER)
00201   117*          BODEG = GKB
00202   118*          CALL ROOT(S,ZS,NCLPS,JERR)
00202   119*   C
00202   120*   C *** PRINT ROOT LOCUS HEADER PAGE
00203   121*          WRITE(6,100) THETA, GKB, GKRL, GAM, GMU, (TRACE(I), I = 1, 4),
00203   122*         $ (STEP(I), I = 1, 4)
00222   123*      100 FORMAT(///12X,5HPHASE,14X,9HBODE GAIN,8X,15HROOT LOCUS GAIN,10X,5H
00222   124*         $GAMMA,13X,8HCENTROID/12X,F5.1,13X,1PE10.4,3(10X,E10.4)
00222   125*         $      ///31X,27HGAIN LOCUS (CONSTANT PHASE),13X,27HPHASE LOCUS (CON
00222   126*         $STANT GAIN)/33X,2HOP,17X,4HOPEN,14X,8HINCREASE,12X,8HDECREASE/10X,
00222   127*         $15HEXTENT OF TRACE,4X,4(E10.4,10X)/10X,17HMAXIMUM STEP SIZE,2X,
00222   128*         $4(E10.4,10X)///30X,15HOPEN-LOOP ZEROS,26X,15HOPEN-LOOP POLES)
00223   129*          IF(NZER .NE. 0)
00223   130*         $WRITE(6,200) (ZRS(I), PLS(I), I = 1, NZER)
00233   131*      200 FORMAT(26X,1H(,1PE10.4,1H,E10.4,1H),20X,1H(,E10.4,1H,E10.4,1H))
00234   132*          IF(NZER .EQ. NPOL)
00234   133*         $GO TO 25
00236   134*          M = NZER + 1
00237   135*          WRITE(6,300) (PLS(I), I = M, NPOL)
00245   136*      300 FORMAT(69X,1H(,1PE10.4,1H,E10.4,1H))
00246   137*       25 WRITE(6,400)
00250   138*      400 FORMAT(///23X,31HESTIMATES FOR CLOSED-LOOP POLES,13X,25HNOMINAL CL
00250   139*         $OSED-LOOP POLES)
00251   140*          WRITE(6,200) (S(I), ZS(I), I = 1, NCLPS)
00260   141*          IF(JERR .EQ. 0) GO TO 30
```

```
00262   142*          WRITE(6,500)
00264   143*      500 FORMAT(64HCAN ERROR OCCURRED IN CALCULATING THE NOMINAL CLOSED-LOO
00264   144*         $P POLES./56H CALCULATION OF THE ROOT LOCUS BRANCHES WILL BE ABORTE
00264   145*         $D.)
00265   146*          GO TO 999
00265   147*    C
00265   148*    C *** GENERATE FIRST THREE RECORD TYPES ON PLOT TAPE
00266   149*       30 IF(PLOT .EQ. 0) GO TO 35
00270   150*          IF(NZER .EQ. 0) GO TO 32
00272   151*          IREC = 2
00273   152*          DO 31 I = 1, NZER
00276   153*          TEMP = REAL(ZRS(I))
00277   154*          WRITE(8) I1, I1, TEMP
00304   155*          TEMP = ABS(AIMAG(ZRS(I)))
00305   156*       31 WRITE(8) IREC, I1, TEMP
00313   157*       32 IREC = 3
00314   158*          DO 33 I = 1, NPOL
00317   159*          TEMP = REAL(PLS(I))
00320   160*          WRITE(8) I1, I1, TEMP
00325   161*          TEMP = ABS(AIMAG(PLS(I)))
00326   162*       33 WRITE(8) IREC, I1, TEMP
00334   163*          IREC = 4
00335   164*          DO 34 I = 1, NCLPS
00340   165*          TEMP = REAL(ZS(I))
00341   166*          WRITE(8) I1, I1, TEMP
00346   167*          TEMP = ABS(AIMAG(ZS(I)))
00347   168*       34 WRITE(8) IREC, I1, TEMP
00347   169*    C
00347   170*    C *** COMPUTE ROOT LOCUS BRANCHES
00355   171*       35 M = 0
00356   172*          DO 36 I = 1, NCLPS
00361   173*          IF(AIMAG(ZS(I)) .LT. -1.0E-04) GO TO 36
00363   174*          M = M + 1
00364   175*          S(M) = ZS(I)
00365   176*       36 CONTINUE
00367   177*          DO 99 NCASE = 1, 4
00372  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00372   178*          IF(TRACE(NCASE) .EQ. 0.0) GO TO 99
00374   179*          NU = 1
00375   180*       40 C = CC(NCASE)
00376   181*          K = (NCASE + 1) / 2
```

```
00377   182*          H = X
00400   183*          BODEG = GKB
00401   184*          G = 0.0
00402   185*          XS = S(NO)
00403   186*          TRACE(NCASE) = ABS(TRACE(NCASE))
00404   187*          IREC = IREC + 1
00404   188*    C
00404   189*    C ****** PRINT CASE HEADER
00405   190*          WRITE(6,600) (DIPS(J,NCASE),J=1,4), TRACE(NCASE), (DEN(J,K),J=1,4)
00405   191*         $, XS, (DEN(J,K),J=1,4)
00425   192*       600 FORMAT(1H1,16X,3A6,1A3,1PE9.4,1X,3A6,1A3/17X,18HCLOSED-LOOP POLE (
00425   193*          $E10.4,1H,E10.4,1H)///13X,9HBODE GAIN,9X,5HPHASE,15X,17HCLOSED-LOOP
00425   194*          $ POLES,15X,3A6,1A3)
00426   195*        45 IF(H .LE. STEP(NCASE)) GO TO 50
00430   196*          H = H / 2.0
00431   197*          G = CSQRT(G)
00432   198*          GO TO 45
00432   199*    C
00432   200*    C ****** PRINT ONE POINT ON ROOT LOCUS
00433   201*        50 GPRIME = -(-1)**NCASE * G
00434   202*          GAIN = CABS(BODEG)
00435   203*          PHI = THETA - RADIAN * ANGLE(BODEG)
00436   204*          WRITE(6,700) GAIN, PHI, XS, GPRIME
00444   205*       700 FORMAT(12X,1PE10.4,6X,E10.4,10X,1H(,E10.4,1H,,E10.4,1H),16X,E10.4)
00444   206*    C
00444   207*    C ****** WRITE ONE POINT ON PLOT TAPE
00445   208*          IF(PLOT .EQ. 0) GO TO 54
00447   209*          TEMP = REAL(XS)
00450   210*          WRITE(8) I1, I1, TEMP
00455   211*          TEMP = AIMAG(XS)
00456   212*          WRITE(8) IREC, I1, TEMP
00463   213*        54 IF(G .GE. TRACE(NCASE)) GO TO 70
00465   214*        55 BODEG = BODEG * G
00466  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00466   215*          IF(BODEG .NE. (0.0,0.0)) GO TO 56
00470   216*          WRITE(6,800)
00472   217*       800 FORMAT(64HOBODE GAIN ATTAINED A VALUE OF ZERO.  THIS CASE WILL BE
00472   218*         $ABORTED.)
00473   219*          GO TO 99
00474   220*        56 CALL ROOT(XS,ASTER,1,JERR)
00475   221*          IF(JERR .NE. 0)
```

```
00475   222*        $WRITE(6,900)
00500   223*    900 FORMAT(3X 9(1H*) 3X 43HERROR OCCURRED IN POLE LOCUS CALCULATION)
00501   224*        ASTER = CMPLX(REAL(ASTER),ABS(AIMAG(ASTER)))
00502   225*        IF(CABS(ASTER) .GE. 0.1) GO TO 65
00504   226*        CDELTA = CABS(ASTER - XS)
00505   227*        IF(CDELTA .GT. 0.1) GO TO 60
00507   228*        G = G + H
00510   229*        XS = ASTER
00511   230*        GO TO 50
00512   231*     60 BODEG = BODEG / C
00513   232*        H = H /2.0
00514   233*        C = CSQRT(C)
00515   234*        GO TO 55
00516   235*     65 CDELTA = CABS((ASTER - XS) / ASTER)
00517   236*        IF(CDELTA .GT. 1.0) GO TO 60
00521   237*        G = G + H
00522   238*        XS = ASTER
00523   239*        IF(CDELTA .GE. 0.1) GO TO 50
00525   240*        XH = 2.0 * H
00526   241*        IF(XH .GT. STEP(NCASE)) GO TO 50
00530   242*        H = XH
00531   243*        C = C**2
00532   244*        GO TO 50
00533   245*     70 NU = NU + 1
00534   246*        IF(NU .LE. M) GO TO 40
00536   247*     99 CONTINUE
00540   248*    999 IF(PLOT .NE. 0)
00540   249*       $END FILE 8
00542   250*        RETURN
00543   251*        END
```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 4 *DIAGNOSTIC* MESSAGE(S)

```
                SUBROUTINE SCALE                                        DATE  101268   PAGE      1

& FOR SCALE,SCALE                                                       03621649
UNIVAC 1108 FORTRAN V LEVEL   2296 (2018.5)
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03621649


   SUBROUTINE SCALE      ENTRY POINT 000114

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001   *CODE    000134
         0000   *DATA    000021
         0002   *BLANK   000000


   EXTERNAL REFERENCES (BLOCK, NAME)

         0003   ABC
         0004   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000015 1L          0001   000105 1CL         0001   000051 2L          0000 R 000000 A          0003 R 000000 ABC


00100      1*   C
00100      2*   C ..............................................................................
00100      3*   C .LSD SUBROUTINE SCALE -- MODIFIED BY C. ARGILA, OCT. '68                      .
00100      4*   C .                                                                             .
00100      5*   C .   DESCRIPTION-                                                             .
00100      6*   C .      THIS ROUTINE ACCEPTS A COMPLEX ARGUMENT, Z, AND MULTIPLIES OR DIVIDES .
00100      7*   C .   IT BY 1.0E+10 SO THAT   1.0 .LE. ABS(Z) .LE 1.0E+10.  THE ARGUMENT, K,  .
00100      8*   C .   IS INCREMENTED BY ONE FOR EACH TIME Z IS DIVIDED BY 1.0E+10 AND IS       .
00100      9*   C .   DECREMENTED BY ONE FOR EACH TIME Z IS MULTIPLIED BY 1.0E+10.             .
00100     10*   C .                                                                             .
00100     11*   C .   ARGUMENTS-                                                               .
00100     12*   C .      Z      (COMPLEX)                                                       .
```

```
00100    13*   C .      K                                                    .
00100    14*   C .                                                           .
00100    15*   C .   SUBROUTINE USAGE-                                       .
00100    16*   C .      ABC                                                  .
00100    17*   C .                                                           .
00100    18*   C ...........................................................
00100    19*   C
00101    20*         SUBROUTINE SCALE(Z,K)
00103    21*         COMPLEX Z
00104  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00104    22*         IF(Z .EQ. (0.0,0.0)) GO TO 10
00106    23*         A = ABC(Z)
00107    24*       1 IF(A .LE. 1.0E+10) GO TO 2
00111    25*         A = A * 1.0E-10
00112    26*         Z = Z * 1.0E-10
00113    27*         K = K + 1
00114    28*         GO TO 1
00115    29*       2 IF(A .GE. 1.0) GO TO 10
00117    30*         A = A * 1.0E+10
00120    31*         Z = Z * 1.0E+10
00121    32*         K = K - 1
00122    33*         GO TO 2
00123    34*      10 RETURN
00124    35*         END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     1 *DIAGNOSTIC* MESSAGE(S)
```

```
                SUBROUTINE SER                                          DATE  101268  PAGE      1

& FOR SER,SER                                                            03621650
UNIVAC 1108 FORTRAN V LEVEL   2206 0018 F5018F
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03621650

   FUNCTION SER          ENTRY POINT 000117

   STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE   000124
        0000   *DATA   000023
        0002   *BLANK  000000


   EXTERNAL REFERENCES (BLOCK, NAME)

        0003   CDV$
        0004   NERR2$
        0005   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000006 1L          0001   000016 2L          0001   000025 3L          0000 I 000003 IFLAG      0000 C 000000 SER
    0000 R 000002 X
```

```
00100      1*   C
00100      2*   C ..............................................................................
00100      3*   C .LSD FUNCTION SER -- MODIFIED BY C. ARGILA, OCT. '68                         .
00100      4*   C .                                                                            .
00100      5*   C .   DESCRIPTION-                                                            .
00100      6*   C .      FOR A GIVEN COMPLEX ARGUMENT, Z, THIS ROUTINE EVALUATES THE           .
00100      7*   C .   POLYNOMIAL-                                                             .
00100      8*   C .              B(M)*Z**M + (A(N)*Z**N + A(N-1)*Z**(N-1) +...+ A(1)*Z + 1)   .
00100      9*   C .   WHERE      M = IAD                                                       .
00100     10*   C .              N = IDEC - IAD                                                .
```

```
00100    11*   C .   AND       A(I) = 1 /((IDEC-N+1)*(IDEC-N+2)*...*(IDEC-N+I)), I=1,...,N      .
00100    12*   C .             B(M) = 1 /((IDEC-N)*(IDEC-N-1)*...*(IDEC-N-M+1))                 .
00100    13*   C .                                                                              .
00100    14*   C .   ARGUMENTS-                                                                .
00100    15*   C .      Z      (COMPLEX)                                                        .
00100    16*   C .      IAD                                                                     .
00100    17*   C .      IDEC                                                                    .
00100    18*   C .                                                                              .
00100    19*   C ..............................................................................
00100    20*   C
00101    21*         COMPLEX FUNCTION SER(Z,IAD,IDEC)
00103    22*         COMPLEX Z
00104    23*         X = 1.0
00105    24*         SER = 1.0
00106    25*         IFLAG = 1
00107    26*       1 IF(IDEC .GT. IAD) GO TO 3
00111    27*         X = 0.0
00112    28*         IFLAG = 2
00113    29*       2 IF(IDEC .LE. 0) RETURN
00115    30*       3 SER = SER * Z / FLOAT(IDEC) + X
00116    31*         IDEC = IDEC - 1
00117    32*         GO TO (1,2), IFLAG
00120    33*         END
```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

& FOR UTILT1,UTILT1
UNIVAC 1108 FORTRAN V LEVEL 2206 0116 R5118-
THIS COMPILATION WAS DONE ON 10 DEC 68 AT 03621651

03621651

MAIN PROGRAM

STORAGE USED (BLOCK, NAME, LENGTH)

0001	*CODE	000036
0000	*DATA	000401
0002	*BLANK	000000
0003	COMPLX	000764
0004	DCDA	000335
0005	DCMPLX	004374
0006	FRQRSP	001037
0007	INVLT	005372
0010	MATDAT	000003
0011	MTRX	02506C
0012	OPTION	000030
0013	RLOCUS	000106
0014	SKALE	000002
0015	TITLES	000106

EXTERNAL REFERENCES (BLOCK, NAME)

0016	NWDU$
0017	NIO1$
0020	NIO2$
0021	NRNL$
0022	NSTOP$

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	000000	1L	0000	000357	2F	0000	000363	3F	0001	000025	4L
0000	000371	5F	0011	R 000000	AMAT	0004	R 000272	ASKB	0004	R 000263	ASKRL
0004	I 000245	ASNP	0004	I 000254	ASNZ	0005	C 001604	ASP	0005	C 003100	ASZ
0011	R 003100	BMAT	0000	I 000000	CLEAR	0003	C 000620	CLPOLE			

```
0011 R 006200 CMAT      0011 R [illegible] COMMN1   0012 I 000125 COMENT   0004 R 000000 D         0013 R 000016 DBDOWN
0013 R 000007 DBUP      0010 I 000001 DEGREE    0007 R 000002 DELTAT    0011 R 011300 DMAT      0003 R 000000 DUMMY1
0011 R 000000 DUMMY2    0011 R 014400 EMAT      0005 C 000000 EST       0007 R 000001 FINALT    0011 R 017500 FMAT
0006 R 000000 FREQ      0015 R 000017 HEADER    0012 I 000127 INPUT     0004 R 000310 LASTCO    0004 R 000301 LEADCO
0011 R 000000 M1C       0011 R 000000 M2C       0011 R 000000 M3C       0011 R 000000 M4C       0011 R 000000 M5O
0011 R 000000 M6C       0000 I 000003 \         0013 I 000077 NCLPOL    0004 I 000223 \FAKES    0000 1 000001 NOVELS
0004 I 000232 NROOTS    0000 I 000002 NOVELS    0010 I 000000 ORDER     0013 I 000000 PHASE     0013 R 000034 PHDOWN
0013 R 000025 PHUP      0012 I 000026 PLOT      0007 R 001306 POLYC     0007 I 000003 POLYN     0007 R 000012 POLYT
0004 I 000317 RATIC     0014 R 000000 SCALE1    0014 R 000001 SCALE2    0010 I 000002 SELECT    0012 I 000007 SFREQR
0004 R 000244 SKB       0004 R 000243 SKRL      0004 I 000241 SNP       0004 I 000242 SNZ       0005 C 001274 SP
0012 I 000000 SROOTL    0007 R 000000 START     0013 R 000052 STEPDD    0013 R 000043 STEPDU    0013 R 000070 STEPPD
0013 R 000061 STEPPU    0012 I 000016 STIMER    0005 C 001440 SZ
```

```
00000  *DIAGNOSTIC*  THE NAME NARRAY APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00000  *DIAGNOSTIC*  THE NAME NTEMP APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00100     1*    C
00100     2*    C .....................................................................
00100     3*    C .   UTILT1                                        C. ARGILA, DEC. '68   .
00100     4*    C .                                                                       .
00100     5*    C .   DESCRIPTION-                                                        .
00100     6*    C .      UTILT1 IS A UTILITY SUPPORT PROGRAM FOR USE WITH THE LINEAR SYSTEMS  .
00100     7*    C .   DYNAMICS PROGRAM (LSD).  UTILT1 INTERROGATES THE NAMELIST INPUT FOR THE  .
00100     8*    C .   LSD PROGRAM AND PROVIDES DIAGNOSTIC INFORMATION ON NAMELIST ERRORS.      .
00100     9*    C .                                                                       .
00100    10*    C .   INPUT-                                                              .
00100    11*    C .      DATA CASES ARE SET UP AS INPUT TO UTILT1 EXACTLY AS THEY WOULD BE SET .
00100    12*    C .   UP AS INPUT TO THE LSD PROGRAM.                                     .
00100    13*    C .                                                                       .
00100    14*    C .   OUTPUT-                                                             .
00100    15*    C .      A MESSAGE IS PRINTED OUT FOR EACH DATA CASE WHICH IS SUCCESSFULLY     .
00100    16*    C .   INTERROGATED.  IF A NAMELIST ERROR IS ENCOUNTERED THE PROPER DIAGNOSTIC  .
00100    17*    C .   INFORMATION IS PRINTED OUT AND UTILT1 IS TERMINATED.                 .
00100    18*    C .                                                                       .
00100    19*    C .....................................................................
00100    20*    C
00101    21*          INTEGER  ASNP  , ASNZ  , CLEAR , COMENT, DEGREE, ORDER , PHASE ,
00101    22*         $ PLOT  , POLYN , RATIC , SELECT, SFREQR, SNP   , SNZ   , SROOTL,
00101    23*         $ STIMER
```

```
00103    24*       REAL     LASTCO, LEADCO, M1C   , M2    , M3C   , M4C   , M5C   ,
00103    25*      $ M6C
00104    26*       COMPLEX  ASP   , ASZ   , CLPOLE, EST   , SP    , SZ
00105    27*       COMMON /COMPLX/  DUMMY1(400)   , CLPOLE(50)
00106    28*       COMMON /CODA   /  D(21,7)      , NEAKES(7)     , NROOTS(7)     ,
00106    29*      $ SNP   , SNZ   , SKRL  , SKB   , ASNP(7)      , ASNZ(7)       ,
00106    30*      $ ASKRL(7)      , ASKB(7)      , LEADCO(7)     , LASTCO(7)     ,
00106    31*      $ RATIO(2,7)
00107    32*       COMMON /DCMPLX/  EST(50,7)     , SP(50), SZ(50), ASP(50,7)     ,
00107    33*      $ ASZ(50,7)
00110    34*       COMMON /FRQRSP/  FREQ(543)
00111    35*       COMMON /INVLT /  STARTT, FINALT, DELTAT, POLYN(7)      , POLYT(100
00111    36*      $,7)    , POLYC(300,7)
00112    37*       COMMON /MATDAT/  ORDER , DEGREE, SELECT
00113    38*       COMMON /MTRX   /  DUMMY2(13600)
00114    39*       COMMON /OPTION/  SROOTL(7)     , SFREQR(7)     , STIMER(7)     ,
00114    40*      $ COMENT, PLOT  , INPUT
00115    41*       COMMON /RLOCUS/  PHASE(7)      , DBUP(7)       , DBDOWN(7)     ,
00115    42*      $ PHUP(7)       , PHDOWN(7)     , STEPDU(7)     , STEPDD(7)     ,
00115    43*      $ STEPPU(7)     , STEPPD(7)     , NCLPOL(7)
00116    44*       COMMON /SKALE /  SCALE1, SCALE2
00117    45*       COMMON /TITLES/  HEADER(14,5)
00120    46*       DIMENSION        AMAT(40,40)   , BMAT(40,40)   , COMAT(40,40,6),
00120    47*      $ CMAT(40,40)   , DMAT(40,40)   , EMAT(40,40)   , FMAT(40,40)   ,
00120    48*      $ M1C(10,10,100), M2C(20,20,25) , M3C(30,30,12) , M4C(40,40,6)  ,
00120    49*      $ M5C(50,50,4)  , M6C(60,60,3)  , NARRAY(6,2)   , NTEMP(2)
00121    50*       EQUIVALENCE      (AMAT,COMAT,DUMMY2,M1C,M2C,M3C,M4C,M5C,M6C)   ,
00121    51*      $ (BMAT,COMAT(1,1,2))           , (CMAT,COMAT(1,1,3))          ,
00121    52*      $ (DMAT,COMAT(1,1,4))           , (EMAT,COMAT(1,1,5))          ,
00121    53*      $ (FMAT,COMAT(1,1,6))
00122    54*       NAME LIST /DATA/ INPUT , CLEAR , COMENT, PLOT  , SFREQR, SROOTL,
00122    55*      $ STIMER, SNP   , SNZ   , SP    , SZ    , SKRL  , SKB   , ASNP  ,
00122    56*      $ ASNZ  , ASP   , ASZ   , ASKRL , ASKB  , D     , RATIO , NROOTS,
00122    57*      $ NEAKES, EST   , LEADCO, LASTCO, SCALE1, SCALE2, ORDER , DEGREE,
00122    58*      $ SELECT, AMAT  , BMAT  , CMAT  , DMAT  , EMAT  , FMAT  , M1C   ,
00122    59*      $ M2C   , M3C   , M4C   , M5C   , M6C   , FREQ  , PHASE , DBUP  ,
00122    60*      $ DBDOWN, PHUP  , PHDOWN, STEPDU, STEPDD, STEPPU, STEPPD, NCLPOL,
00122    61*      $ CLPOLE, STARTT, DELTAT, FINALT, POLYN , POLYT , POLYC , NOVELS,
00122    62*      $ NUNELS
00123    63*       DATA N/1/
00125    64*      1 WRITE(6,2) N
```

```
00130     65*       2 FORMAT(14HCDATA CASE NO. I3)
00131     66*         READ(5,DATA,END=4)
00134     67*         WRITE(6,3)
00136     68*       3 FORMAT(26H INTERROGATED SUCCESSFULLY)
00137     69*         N = N + 1
00140     70*         GO TO 1
00141     71*       4 WRITE(6,5)
00143     72*       5 FORMAT(18H IS A '& EOF' CARD)
00144     73*         STOP
00145     74*         END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     2 *DIAGNOSTIC* MESSAGE(S)
```

36. PROGRAM ALLOCATION

```
STARTING ADDRESS  C14000

CORE LIMITS 014000 050275      063645 163771      163772 163777


LSD    /CODE
0    063645-064606
1    014000-014353

NSTOP$/CODE
1    014354-014371

NOUT$ /CODE
0    064607-064610
1    014372-015434
2    064611-064651

NTAB$ /CODE
0    064652-064773

NFTV$ /CODE
1    015435-015457

NIOIN$/CODE
1    015460-015523
2    064774-065024

NOTIN$/CODE
```

```
0    065025-065025
1    015524-016047
2    065026-065066

FPACK$/CODE
1    016050-016113

DEPTH /******
0    065067-065074

NFMT$ /CODE
1    016114-017076
2    065075-065215

NIER$ /CODE
0    065216-065216
1    017077-017235
2    065217-065302

NERR$ /CODE
0    065303-065432
1    017236-017634

NINPT$/CODE
0    065433-065433
1    017635-021026
2    065434-065466

FLOATX/CODE
0    065467-065602

NEXP$X/CODE
0    065603-065710

CONVTX/CODE
0    065711-065763

NINI$/CODE
1    021027-021172
2    065764-065767
```

```
NLINP /CODE
0    065770-065776
1    021173-023104
2    065777-066146

NRWND$/CODE
1    023105-023171

MSC$01/CODE
0    066147-066203

ETOD$ /CODE
0    066204-066204
1    023172-023225

CDA   /CODE
0    066205-106011
1    023226-025526

NEXP5$/CODE
1    025527-025624
2    106012-106014

NXPAF$/CODE
1    025625-026003
2    106015-106021

EXP   /CODE
1    026004-026074
2    106022-106041

ALOG  /CODE
1    026075-026162
2    106042-106116

NXPAX$/CODE
1    026163-026205
2    106117-106117

CDV$  /CODE
0    106120-106121
```

```
1    026206-026255

CABS  /CODE
1    026256-026302
2    106122-106125

SQRT  /CODE
0    106126-106162
2    106163-106224

ILT   /CODE
0    106225-106515
1    026303-027206

NFOUT$/CODE
1    027207-027413
2    106516-106516

NBUFF$/CODE
1    027414-027450
2    106517-107525

NEXPD$/CODE
1    027451-027546
2    107526-107527

NXPCX$/CODE
1    027547-027656
2    107530-107540

NXPCF$/CODE
1    027657-027777
2    107541-107550

CEXP  /CODE
1    030000-030061
2    107551-107611

SINCOS/CODE
1    030062-030151
2    107612-107645
```

```
CLOG  /CODE
1    030152-030252
2    107646-107660

ATAN  /CODE
1    030253-030352
2    107661-107732

ILTF  /CODE
0    107733-110524
1    030353-031731

EMU   /CODE
0    110525-110537
1    031732-032056

SCALE /CODE
0    110540-110560
1    032057-032212

ABC   /CODE
0    110561-110570
1    032213-032235

SER   /CODE
0    110571-110613
1    032236-032361

ILTCOM/******
0    110614-111132

RTLC  /CODE
0    111133-112104
1    032362-033663

CSQRT /CODE
1    033664-034021
2    112105-112114

NEXP1$/CODE
```

```
1    034022-034121
2    112115-112116

ANGLE /CODE
0    112117-112127
1    034122-034156

ROOT  /CODE
0    112130-112263
1    034157-035562

FMASK /CODE
0    112264-112274
1    035563-035611

AUXSUB/CODE
0    112275-112323
1    035612-036067

EXTRA /******
0    112324-112327

FREQR /CODE
0    112330-112416
1    036070-036461

COMPUT/CODE
0    112417-112470
1    036462-036763

DB    /CODE
0    112471-112502
1    036764-037026

ALOG10/CODE
1    037027-037075
2    112503-112536

ADJUST/CODE
0    112537-112553
1    037076-037147
```

```
ADDFRE/CODE
0    112554-112577
1    037150-037252

AFCOM /******
0    112600-113730

ADJCOM/******
0    113731-114241

ELIM  /CODE
0    114242-114274
1    037253-037550

ORDER /CODE
0    114275-114323
1    037551-037730

EIGEN /CODE
0    114324-116450
1    037731-044756

CONOUT/CODE
0    116451-116463
1    044757-045127

CONIN /CODE
0    116464-116471
1    045130-045162

OUT   /CODE
0    116472-116573
1    045163-045315

C3CM  /CODE
0    116574-116615
1    045316-045457

DTVC  /CODE
0    116616-116740
```

```
1    045460-046743

MATVAR/CODE
0    116741-117013
1    046744-047234

EVAL  /CODE
0    117014-117044
1    047235-047461

C3CD  /CODE
0    117045-117057
1    047462-047562

MTRXPR/CODE
0    117060-117124
1    047563-047727

GAMMA /CODE
0    117125-117161
1    047730-050112

GCONJ /CODE
0    117162-117176
1    050113-050173

ROOTS /******
0    117177-121766

PARAM /******
0    121767-121770

BASIC /******
0    121771-121777

QQUFCF/CODE
0    122000-122072
1    050174-050275

TITLES/******
0    122073-122200
```

```
SKALE /******
0    122201-122202

RLOCUS/******
0    122203-122310

OPTION/******
0    122311-122340

MTRX  /******
0    122341-147420

MATDAT/******
0    147421-147423

INVLT /******
0    147424-155015

FRQRSP/******
0    155016-156054

DCMPLX/******
0    156055-162450

DCDA  /******
0    162451-163005

COMPLX/******
0    163006-163771


END OF ALLOCATION 1103 0036
```