NASA CONTRACTOR REPORT

NASA CR-555

ROOT LOCUS DIAGRAMS BY DIGITAL COMPUTER

by Allan M. Krall and Robert Fornaro

Prepared by
PENNSYLVANIA STATE UNIVERSITY
University Park, Pa.

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • AUGUST 1966

NASA CR-555

ROOT LOCUS DIAGRAMS BY DIGITAL COMPUTER

By Allan M. Krall and Robert Fornaro

Prepared under Grant No. NGR-39-009-041 by
PENNSYLVANIA STATE UNIVERSITY
University Park, Pa.

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – Price $2.00

Introduction. Most continuously operating physical systems being developed today are described by a system of differential equations. The simplest of these are the linear systems, and the simplest of the linear systems are those with constant coefficients. The concept of stability is important in all such systems. In the last case, stability can be determined by examining the characteristic equation which is associated with each such system.

There are several stability techniques available for systems with constant coefficients: criteria due to Bode [1], Evans (Root-Locus) [3], Michailov [7], Neimark [9] and Pontrjagin [11]. Each has its own advantages and disadvantages.

The root locus method, which is to be discussed here, describes how to construct the graph of the roots of the characteristic equation as one parameter varies. This enables the designer to choose the positions of the roots with some freedom, possibly achieving stability.

While most of the techniques are easily adapted for programming on a digital computer, the root locus has proved more difficult. The purpose of this article is to describe just how such a program was developed and to show what information it gives.

The Root Locus. The root locus method has been developed for ordinary systems and for systems with one time delay. We discuss the general system (with delay). Let $g(z) = z^n + az^{n-1} + \ldots\ldots$ and $h(z) = z^m + bz^{m-1} + \ldots.$ be polynomials with (constant) complex coefficients. Let τ and θ be constant real numbers, $\tau \geqq 0$, $0 \leqq \theta < 2\pi$, and let K be real valued. The characteristic equation we wish to consider is $F(z) = g(z) - Ke^{i\theta}e^{-\tau z}h(z) = 0.$

Definition. The root locus of $F(z)$ is the set of all points z such that z is a zero of $h(z)$ or such that z is a zero of $F(z)$ for some real value of K.

The set of all points z on the root locus for values of $K > 0$ is the positive root locus. The set of all points z on the root locus for values of $K < 0$ is the negative root locus.

Theorem 1. The point $z = x + iy$ is on the root locus of $F(z)$ if and only if $\emptyset(x,y) = \cos(\theta - \tau y)\ \mathrm{Im}\ (h(z)\overline{g(z)}\ ) + \sin(\theta - \tau y)\ \mathrm{Re}\ (h(z)\overline{g(z)}\ ) = 0.$

Proof. Suppose z is on the root-locus. If $g(z) \neq 0$, then for some $K \neq 0$, $h(z)Ke^{i\theta}e^{-\tau z}/g(z) = 1$. Thus

$$\frac{h(z)}{g(z)} = K^{-1} e^{\tau x}[\cos(\theta - \tau y) - i \sin (\theta - \tau y)].$$

$$h(z)\overline{g(z)} = K^{-1} e^{\tau x} |g(z)|^2 [\cos (\theta - \tau y) - i \sin (\theta - \tau y)].$$

Since K, τ, x are real,

$$\mathrm{Re}\ (h(z)\overline{g(z)}) = K^{-1}e^{\tau x}|g(z)|^2 \cos (\theta - \tau y),$$

$$\operatorname{Im}\,(h(z)\overline{g(z)}) \;=\; -K^{-1}e^{\tau x}|g(z)|^2 \sin\,(\theta - \tau y).$$

Multiplying the first by $\sin\,(\theta - \tau y)$, the second by $\cos\,(\theta - \tau y)$ and adding completes the first part.

Conversely, if the equation is satisfied, then $\operatorname{Im}\,(e^{i\theta}e^{-\tau z}h(z)\overline{g(z)}) = 0$. So $e^{i\theta}e^{-\tau z}h(z)\overline{g(z)} = R(z)$, where $R(z)$ is real. If $R(z) = 0$, then either $h(z) = 0$, or $g(z) = 0$, and z is on the root-locus. If $R(z) \neq 0$, let $K = |g(z)|^2/R(z)$. If $K = 0$, then $g(z) = 0$, and z is on the root-locus. If $K \neq 0$, then $Ke^{i\theta}e^{-\tau z}h(z)/g(z) = 1$, and $F(z) = 0$.

Note that K can be found by

$$K \;=\; e^{\tau x}|g(z)|^2 \cos\,(\theta - \tau y)/\operatorname{Re}(h(z)\overline{g(z)}),$$

or by

$$K \;=\; -e^{\tau x}|g(z)|^2 \sin\,(\theta - \tau y)/\operatorname{Im}(h(z)\overline{g(z)}).$$

<u>Lemma 1.</u> <u>Let</u> $h(z)$ <u>and</u> $g(z)$ <u>have real coefficients.</u> <u>Then</u>

$$h(z) \;=\; \Sigma_{j=0}^{m}\, h^{(j)}(x)(iy)^j\,/\,j!\,,$$

$$\overline{g(z)} \;=\; \Sigma_{j=0}^{n}\, g^{(j)}(x)(-iy)^j/\,j!\;.$$

Proof. These are just Maclaurin expansions.

<u>Lemma 2.</u> <u>Let</u> $h(z)$ <u>and</u> $g(z)$ <u>have real coefficients.</u> <u>Then</u>

$$h(z)\overline{g(z)} \;=\; \Sigma_{j=0}^{n+m}\frac{(iy)^j}{j!}\,\Sigma_{k=0}^{j}\binom{j}{k}(-1)^{j-k}h^{(k)}(x)g^{(j-k)}(x)\;.$$

Proof. This follows from Lemma 1.

Lemma 3. If $h(z)$ and $g(z)$ have real coefficients, then

$$\mathrm{Re}(h(z)\overline{g(z)}) = \Sigma_{k=0}^{\infty} \frac{(-1)^k y^{2k}}{(2k)!} \Sigma_{i=0}^{2k} \binom{2k}{i} (-1)^{2k-i} h^{(i)}(x) g^{(2k-i)}(x),$$

$$\mathrm{Im}(h(z)\overline{g(z)}) = \Sigma_{k=0}^{\infty} \frac{(-1)^k y^{2k+1}}{(2k+1)!} \Sigma_{i=0}^{2k+1} \binom{2k+1}{i} (-1)^{2k+1-i} h^{(i)}(x) g^{(2k+1-i)}(x).$$

Proof. This follows immediately from Lemma 2.

In any physical situation the coefficients of $h(z)$ and $g(z)$ must be real, and θ must be either 0 or π. If K is permitted to take on all real values, we lose no generality by fixing θ at one of these values. Since negative feedback is most frequently encountered, we let $\theta = \pi$.

Theorem 2. Let $h(z)$ and $g(z)$ have real coefficients, and let $\theta = \pi$. Then $z = x + iy$ is on the root locus of $F(z)$ if and only if

$$\phi(x,y) = \cos \tau y \, \Sigma_{k=0}^{\infty} \frac{(-1)^k y^{2k+1}}{(2k+1)!} \Sigma_{i=0}^{2k+1} \binom{2k+1}{i} (-1)^{2k+1-i} h^{(i)}(x) g^{(2k+1-i)}(x)$$

$$- \sin \tau y \, \Sigma_{k=0}^{\infty} \frac{(-1)^k y^{2k}}{(2k)!} \Sigma_{i=0}^{2k} \binom{2k}{i} (-1)^{2k-i} h^{(i)}(x) g^{(2k-i)}(x) = 0.$$

The graph of this equation in the xy-plane is the root locus of $F(z)$.

<u>The Root Locus by Digital Computer.</u> As might be anticipated, we will use $\phi(x,y) = 0$ to construct the root locus of $F(z)$ rather than $F(z)$ itself. There are several reasons for this. First, in the ordinary case $(\tau = 0)$, we have no control over where the roots of $F(z)$ might lie, so factoring by machine is impractical. In the case with time delay $(\tau \neq 0)$, $F(z) = 0$ has an infinite number of roots, and factoring is not possible.

In addition, the roots of $F(z) = 0$ do not vary at a uniform rate as K varies. It is impossible to say in general how small the increments in K should be in order to construct a reasonable graph.

On the other hand, $\phi(x,y) = 0$ does not involve K. On each vertical line where x is fixed, the y coordinates of the root locus are the real roots of $\phi(x,y) = 0$. The procedure found to be most practical for determining those real roots y (where x is fixed) offers a direct means of controling the error encountered in approximating these roots, and it works just as well for systems with a time delay. It is as follows:

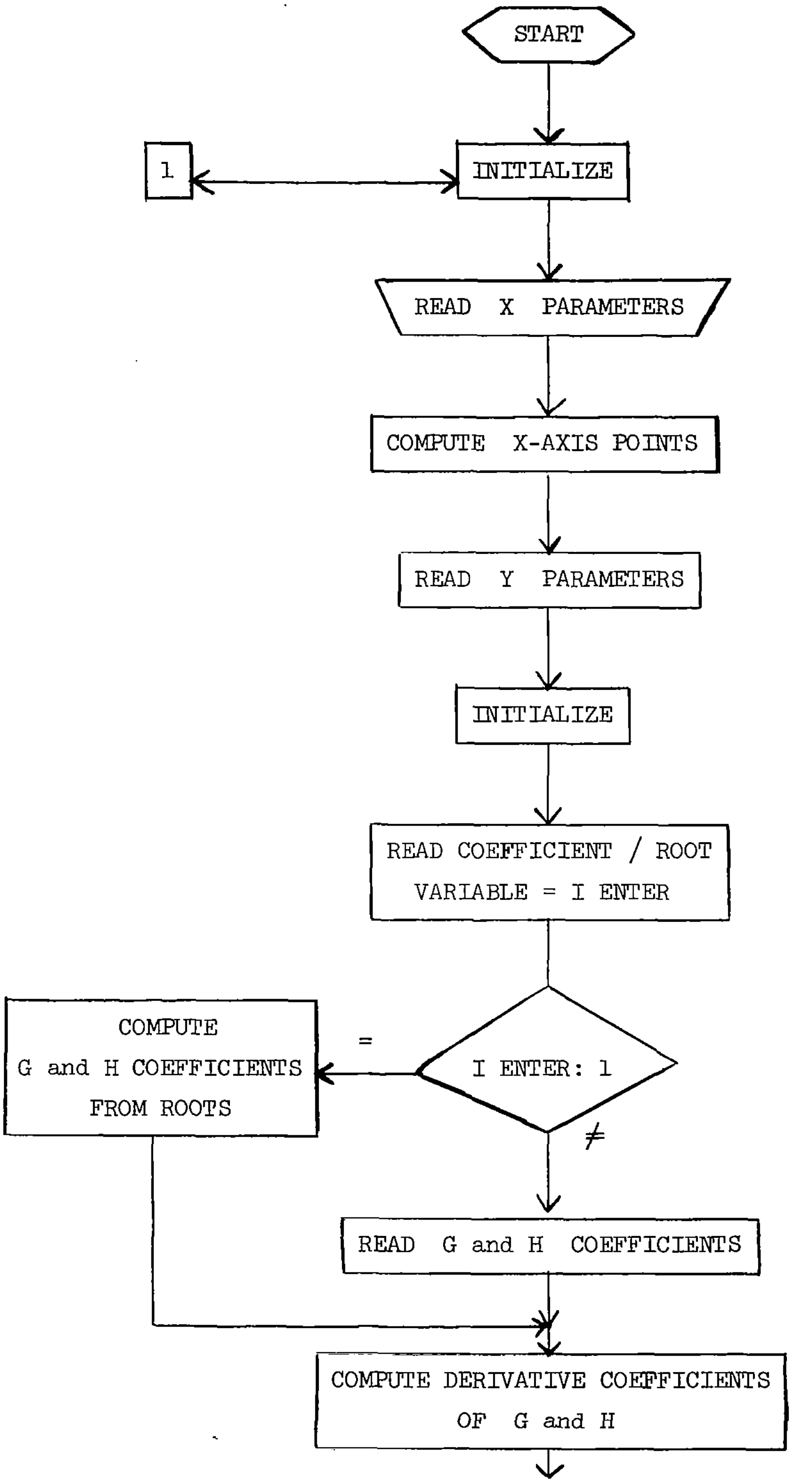
We choose any rectangular region in which the root locus is desired. Let such a region be denoted by $x_\ell \leq x \leq x_r$, $y_b \leq y \leq y_t$. We divide $[x_\ell, x_r]$ into increments $x_\ell, x_\ell + \Delta x$, $x_\ell + 2\Delta x$, $\ldots$, x_r. For each point $x_\ell + m\Delta x$ in turn we divide $[y_b, y_t]$ into increments in a similar manner, $y_t, y_t - \Delta y$, $y_t - 2\Delta y, \ldots y_b$. We then compute $\phi(x,y)$ at these points, first fixing x, then letting y vary from y_t to y_b. In so doing, for a fixed x, we look for intervals in $[y_b, y_t]$ over which $\phi(x,y)$ changes sign. The sign change indicates that a point of the root locus lies in the interval. Each y interval where $\phi(x,y)$ changes sign is divided in half, and these subintervals are considered separately. The subinterval where $\phi(x,y)$ changes sign is itself divided in half, etc. In this manner the y coordinate of the point on the root locus may be found quickly and accurately.

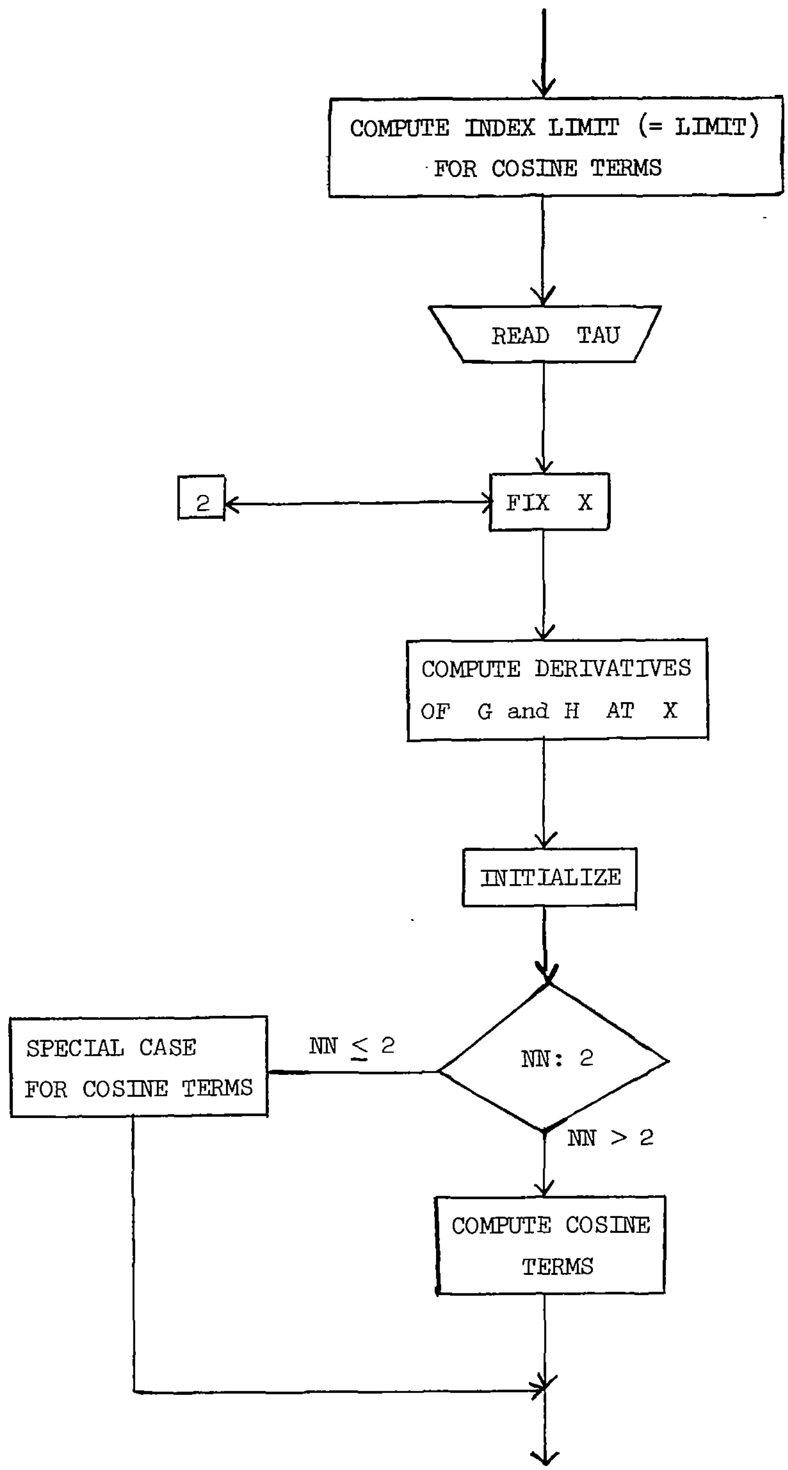
After the vertical strip from y_t to y_b has been exhausted and all the points on the root locus have been found for that fixed value of x, x is increased by Δx, and the process is repeated. As x varies from x_ℓ to x_r every point on the root locus in the rectangle is found. These points are stored and then plotted by the computer.

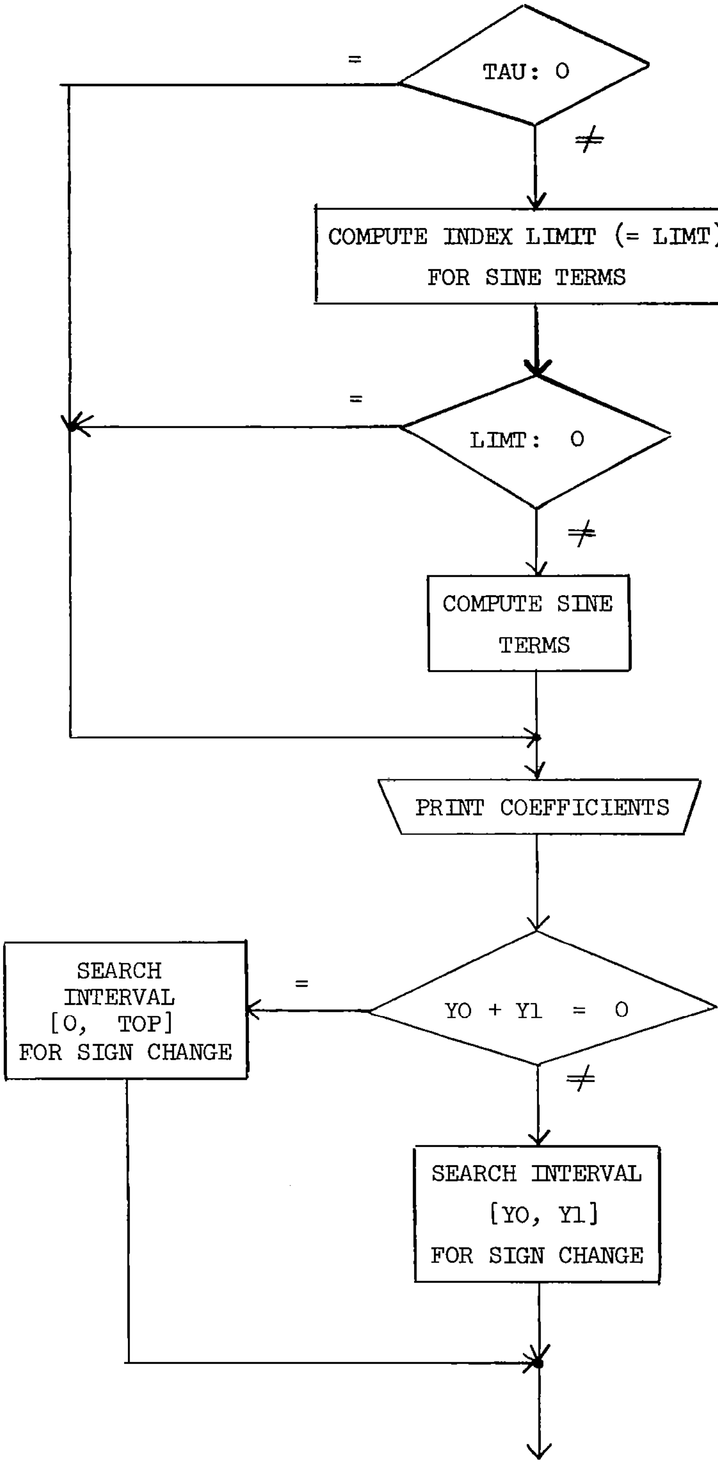
The value of the parameter K is computed for each point on the root locus using the formulas following theorem 1. The program plots + if $K > 0$, and - if $K < 0$. In addition, the triple (x,y,K) is printed out separately for each point on the root locus.

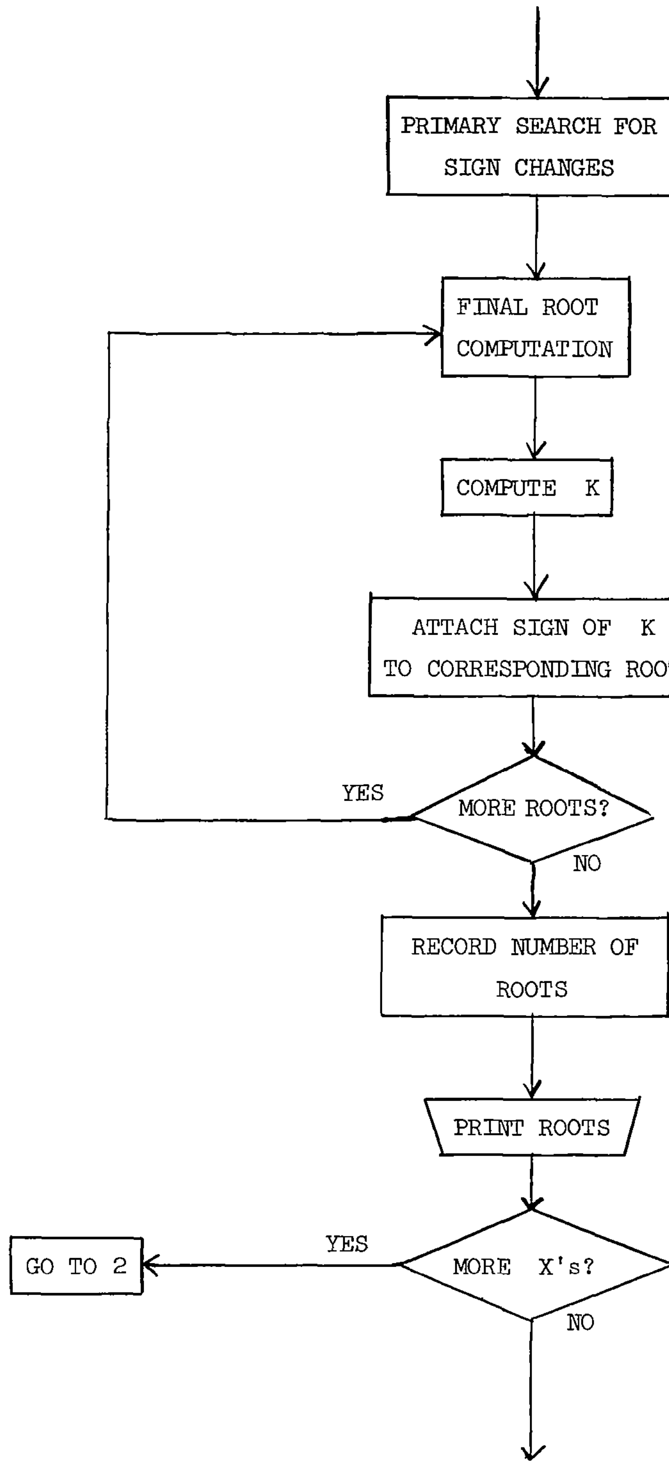
The program was designed for the IBM 7074 at the Computation Center of The Pennsylvania State University, University Park, Pennsylvania, under control of the Dual Autocoder Fortran Translator (DAFT) compiler system. We hope that, with a minmum number of changes, it can be adapted for use elsewhere.

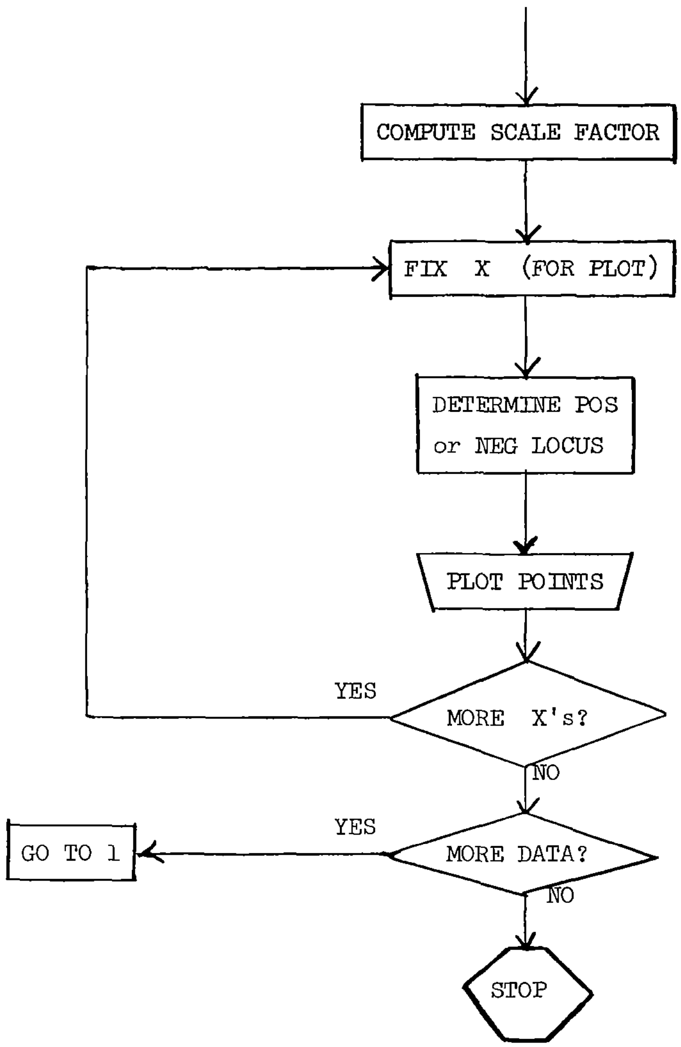
FLOW CHART


START
1
INITIALIZE
READ X PARAMETERS
COMPUTE X-AXIS POINTS
READ Y PARAMETERS
INITIALIZE
READ COEFFICIENT / ROOT
VARIABLE = I ENTER
COMPUTE
G and H COEFFICIENTS
FROM ROOTS
=
I ENTER: 1
≠
READ G and H COEFFICIENTS
COMPUTE DERIVATIVE COEFFICIENTS
OF G and H


COMPUTE INDEX LIMIT (= LIMIT) FOR COSINE TERMS
READ TAU
2
FIX X
COMPUTE DERIVATIVES OF G and H AT X
INITIALIZE
SPECIAL CASE FOR COSINE TERMS
NN ≤ 2
NN: 2
NN > 2
COMPUTE COSINE TERMS


TAU: 0
=
≠
COMPUTE INDEX LIMIT (= LIMT)
FOR SINE TERMS
LIMT: 0
=
≠
COMPUTE SINE
TERMS
PRINT COEFFICIENTS
YO + Y1 = 0
=
≠
SEARCH
INTERVAL
[0, TOP]
FOR SIGN CHANGE
SEARCH INTERVAL
[YO, Y1]
FOR SIGN CHANGE


PRIMARY SEARCH FOR SIGN CHANGES
FINAL ROOT COMPUTATION
COMPUTE K
ATTACH SIGN OF K TO CORRESPONDING ROOT
MORE ROOTS?
YES
NO
RECORD NUMBER OF ROOTS
PRINT ROOTS
MORE X's?
YES
NO
GO TO 2


COMPUTE SCALE FACTOR
FIX X (FOR PLOT)
DETERMINE POS or NEG LOCUS
PLOT POINTS
MORE X's?
YES
NO
MORE DATA?
YES
GO TO 1
NO
STOP

Parameters Necessary for Execution : Data Cards

Card No. 1 Variable Names: DELTA, BGN, END.

Use: Determines points on real axis where coefficients of closed form ploynomial will be evaluated. Left hand endpoint BGN, right hand endpoint END. The interval [BGN, END] divided into subintervals of length DELTA. Evaluation takes place at each endpoint of these subintervals, i.e. at

BGN, BGN-DELTA, BGN-2 DELTA, ... , END.

Card format: columns 1-10 numerical value of: DELTA REAL MODE,

"	11-20	"	BGN	" ,
"	21-30	"	END	" .

Card No. 2 Variable Names: DEC, YO, Yl, TOP, ISIG.

Use: DEC: decrement to be used for isolation of roots.

YO,Yl: Allow detailed inspection of any interval along Y-axis. YO < Yl. If their sum is zero (usual case), program sets upper bound for roots (See comment).

TOP: Maximum value desired along vertical axis (Vertical scale factor).

ISIG: Tolerance desired for roots is 10**(-ISIG), where $1 \leq$ ISIG ≤ 10.

Comment: The usual case is to set YO = 0.0, and Yl = 0.0, and TOP = a (a some value). Occasionally detailed inspection of some interval $[a_1, a_2]$ may be desirable $a_1 < a_2 < a$.

If the magnification factor is to be the same, set YO = a_1, Y1 = a_2, TOP = a_1. To magnify, set YO = a_1, Y1 = a_2, TOP = a_2.

Card format:	columns	1-10	numerical value of:	DEC	REAL MODE,
	"	11-20	"	YO	" ,
	"	21-30	"	Y1	" ,
	"	31-40	"	TOP	" ,
	column	45	"	ISIG	INTEGER MODE.

Card No. 3 Variable Name: I ENTER

Use: A numerical value of 1 will cause the program to assume that h and g are given in factored form and enter appropriate routines to calculate their coefficients. Any other value will cause the program to assume that the coefficients of h and g are being provided directly.

Card format:	column	5	1	Integer Mode,
	columns	1-5	any integer	" " .

If card #3 contains the digit 1 in column 5 these data cards must follow:

Card No. 4* Variable names (Used in COMPCO): M,N

Use: M- number of conjugate pairs of complex roots of g (must be $\leq$ 6).

N- number of real roots of g (must be $\leq$ 12).

Card format:	columns	1-5	(right justified)	M	INTEGER MODE,
	"	6-10	" "	N	" .

Card No. 5*: complex roots of g

Card format: Columns	real part	imaginary part	
	1-5	6-10	REAL MODE,
	11-15	16-20	" ,
	21-25	26-30	" .

etc.

Card No. 6*: real roots of g

Card format: Columns	
	REAL MODE,
1-5	" ,
6-10	" ,
etc.	" .

Card No. 7* and 8*: same as 5* and 6* except information must pertain to h.

If card #3 contains any integer in columns 1-5 other than 1 in column 5 these data cards must follow:

Card No. 4** Variable Names: NG, NH

Use: NG- degree of g (must be $\leq$ 12).

NH- degree of h (must be $\leq$ 12).

Card format: columns				
1-5	(right justified)	NG	INTEGER MODE,
6-10	" "	NH	" .

Card No. 5**: Coefficients of g (highest power first)

Card format: columns			
1-10	coefficients of x^n	REAL MODE,
11-20	" x^{n-1}	" ,
etc.		" .

Card No. 6**: Coefficients of h

Card format: (same as 5**)

Card No. 9* (or 7**) Variable name: TAU

Use: Time Lag Parameter

Card format: columns 1-10 REAL MODE.

Card No. 10* (or 8**) Variable name: I STOP

Use: A numerical value of 1 will cause the program to re-initialize itself, i.e., start over. A numerical value of 2 will result in termination of execution. If the program starts over, additional sets of data cards, as described above, must be included in sequence.

Card format: column 5 INTEGER MODE.

```
*****************************************************************************
*C     IDENTIFICATION
*C
*C    CHAR ,PLOT      --  USED IN PLOTTING SECTION
*C    FMT             --  FORMAT CONTROL
*C    S               --  LEFT ENDPOINT OF SIGN CHANGE INTERVAL
*C    Y,C             --  TEMPORARY STORAGE
*C    A               --  COEFFICIENTS OF POWERS OF Y
*C    X               --  POINTS ALONG X-AXIS
*C    YVAR            --  POINTS TO BE PLOTTED
*C    NROOTS          --  NUMBER OF ROOT LOCUS POINTS AT A GIVEN POINT ON X-AXIS
*C    CG              --  COEFFICIENTS OF G AND ASSOCIATED DERIVATIVES
*C    CH              --  COEFFICIENTS OF H AND ASSOCIATED DERIVATIVES
*C    NG              --  ORDER OF G
*C    NH              --  ORDER OF H
*C    G               --  VALUES OF G AND ASSOCIATED DERIVATIVES AT SOME POINT
*C    H               --  VALUES OF H AND ASSOCIATED DERIVATIVES AT SOME POINT
*C    TAU             --  TIME LAG
*C    KVAL            --  VALUES OF K FOR EACH POINT TO BE PLOTTED
*C
*C         CHARACTER MODE = DAFT FEATURE...
*C         ADDRESSES EACH CHARACTER OF A STRING INDIVIDUALLY
*****************************************************************************
           CHARACTER CHAR(12),PLOT(100),FMT(25)
           DIMENSION S(12),Y(12),A(25),X(200),YVAR(12,200),NROOTS(200),C(15)
           COMMON CG(13,13),CH(13,13),NG,NH,G(26),H(26),TAU
           REAL KVAL(12)
*****************************************************************************
*C
*C         INITIALIZATION
*C
*****************************************************************************
     10    DO 12 I=1,200
           NROOTS(I)=0
           X(I)=0.0
           DO 12 J=1,12
           YVAR(J,I)=0.0
     12    CONTINUE
           DO 13 I=1,100
     13    PLOT(I)=1H
           PRINT 14
     14    FORMAT(1H1,10X)
*****************************************************************************
*C         READ X PARAMETERS
*C    DELTA           --  X-AXIS INCREMENT
*C    BGN             --  LEFT POINT X-AXIS
*C    END             --  RIGHT POINT X-AXIS
*****************************************************************************
           READ 100,  DELTA,BGN,END
    100    FORMAT(3F10.0)
```

```
*****************************************************************************
*C
*C          COMPUTE X-AXIS POINTS
*C
*****************************************************************************
            M=ABS(BGN-END)/DELTA+1.0
            X(1)=BGN
            IF(M-200) 7, 7, 8
      8     M=200
      7     DO 9 I=2,M
      9     X(I)=X(I-1)+DELTA
*****************************************************************************
*C          READ Y PARAMETERS
*C    DEC            --  ROOT SEARCH INCREMENT
*C    Y0,Y1          --  PERMIT DETAILED INSPECTION OF Y-AXIS
*C    TOP            --  UPPER BOUND FOR ROOTS (SCALE FACTOR ON VERTICAL)
*C    ISIG           --  ROOT TOLERANCE..E=10**(-ISIG)
*****************************************************************************
            READ 110,DEC,Y0,Y1,TOP,ISIG
      110   FORMAT(4F10.0,I5)
*****************************************************************************
*C          COMPUTE TOLERANCE FOR ROOT ERROR AND
*C          OBTAIN CORRESPONDING FORMAT STATEMENT
*****************************************************************************
            E=(10.0)**(-ISIG)
            CALL FRMAT(FMT,ISIG)
*****************************************************************************
*C
*C          INITIALIZATION
*C
*****************************************************************************
            DO 15 I=1,13
            DO 15 J=1,13
            CH(I,J)=0.0
      15    CG(I,J)=0.0
*****************************************************************************
*C
*C         COEFFICIENT/ROOT OPTION
*C
*****************************************************************************
            READ 97,IENTER
            IF(IENTER-1) 17, 16, 17
*****************************************************************************
*C
*C          COMPUTE G COEFFICIENTS FROM ROOTS (IENTER=1)
*C
*****************************************************************************
      16    PRINT 99
      99    FORMAT(1H ,12HFACTORS OF G)
            CALL COMPCO(C,NG1)
            NG=NG1-1
            DO  18 I=1,NG1
      18    CG(1,I)=C(I)
```

```
*****************************************************************************
*C
*C        COMPUTE H COEFFICIENTS FROM ROOTS (IENTER=1)
*C
*****************************************************************************
          PRINT 98
   98     FORMAT(1H ,12HFACTORS OF H)
          CALL COMPCO(C,NH1)
          NH=NH1-1
          DO  19 I=1,NH1
   19     CH(1,I)=C(I)
          GO TO 20
*****************************************************************************
*C
*C        READ COEFFICIENTS OF G AND H (IENTER=0)
*C
*****************************************************************************
   17     READ 97,NG,NH
   97     FORMAT(2I5)
          NG1=NG+1
          NH1=NH+1
          READ 96,(CG(1,I),I=1,NG1)
          READ 96,(CH(1,I),I=1,NH1)
   96     FORMAT(8F10.0)
*****************************************************************************
*C
*C        COMPUTE DERIVATIVE COEFFICIENTS AND PRINT
*C
*****************************************************************************
   20     N1G=NG1
          DO 21 I=2,NG1
          N1G=N1G-1
          DO 21 J=1,N1G
          CG(I,J)=FLOAT(N1G+1-J)*CG(I-1,J)
   21     CONTINUE
          N1H=NH1
          DO 22 I=2,NH1
          N1H=N1H-1
          DO 22 J=1,N1H
          CH(I,J)=FLOAT(N1H+1-J)*CH(I-1,J)
   22     CONTINUE
          PRINT 95 ,((CG(I,J),J=1,13),I=1,13)
          PRINT 94 ,((CH(I,J),J=1,13),I=1,13)
   95     FORMAT(50H COEFFICIENTS  OF  G  AND  ASSOCIATED  DERIVATIVES   //
         1(1H ,12F10.1,F9.1))
   94     FORMAT(50H COEFFICIENTS  OF  H  AND  ASSOCIATED  DERIVATIVES   //
         1(1H ,12F10.1,F9.1))
*****************************************************************************
*C
*C        SUMMING INDEX ...NN=DEGREE OF Y POLYNOMIAL
*C
*****************************************************************************
          NN=NG+NH
          LIMIT=(NN-1)/2
```

```
      IF(LIMIT-11)26,26,27
 27   PRINT 93,NN
 93   FORMAT(' DEGREE OF POLYNOMIAL (',I2,') EXCEEDS PROGRAM LIMIT')
      STOP
 26   READ 92,TAU
 92   FORMAT(F10.0)
      DO 28 IM=1,M
      CALL COMPDERIV(X(IM))
************************************************************************
*C
*C         INITIALIZATION
*C
************************************************************************
      DO 11 I=1,12
      S(I)=0.0
      Y(I)=0.0
 11   CONTINUE
      DO 29 I=1,25
 29   A(I)=0.0
************************************************************************
*C
*C         Y COEFFICIENT COMPUTATION..COSINE TERMS
*C
************************************************************************
      IF(NN-2)31,31,32
 31   KK=1
      KI=NN
      DO 33 I=0,KK
      KIK=KK-I
      A(KI)=A(KI)+COMB(KK,I)*(-1.0)**KIK*H(I+1)/FACT(KK)*G(KIK+1)
 33   CONTINUE
      GO TO 34
 32   DO 35 K=0,LIMIT
      KK=2*K+1
      C1=(-1.0)**K
      KI=NN+1-KK
      DO 36 I=0,KK
      KIK=KK-I
      A(KI)=A(KI)+COMB(KK,I)*(-1.0)**KIK*H(I+1)/FACT(KK)*G(KIK+1)
 36   CONTINUE
      A(KI)=A(KI)*C1
 35   CONTINUE
************************************************************************
*C
*C         COMPUTATION OF SINE TERMS
*C
************************************************************************
 34   IF(TAU)37,38,37
 37   LIMT =NN/2
      A(NN+1)=H(1)*G(1)
      IF(LIMT )38,38,39
```

```
39     DO 40 K=1,LIMT
       C1=(-1.0)**K
       KK=2*K
       KI=NN+1-KK
       DO 41 I=0,KK
       KIK=KK-I
       A(KI)=A(KI)+COMB(KK,I)*(-1.0)**KIK*H(I+1)/FACT(KK)*G(KIK+1)
41     CONTINUE
       A(KI)=A(KI)*C1
40     CONTINUE
************************************************************************
*C
*C        PRINT Y COEFFICIENTS
*C
************************************************************************
38     N1=NN+1
       PRINT 112,X(IM),(A(I),I=1,N1)
112    FORMAT(1H ///22HOCOEFFICIENTS FOR X=  ,F6.2//(15X,E15.8))
       N=NN
       IF(YO+Y1)42,55,42
55     Y1=TOP
       YO=0.0
************************************************************************
*C
*C        PRIMARY SEARCH FOR SIGN CHANGES
*C
************************************************************************
42     CALL SEARCH(YO,Y1,DEC,S,J,A,N)
       IF(J) 43,43,44
43     PRINT 102
102    FORMAT('OSEARCH NEGATIVE')
       GO TO 28
************************************************************************
*C
*C        FINAL ROOT COMPUTATION
*C        K  COMPUTATION
*C
************************************************************************
44     K=0
       DO 45 I=1,J
       XL=S(I)
       CALL ROOT (XL,DEC,  A,N,E,AROOT)
       K=K+1
       Y(K)=AROOT
       KVAL(K)=COMPK(AROOT,X(IM))
       YVAR(K,IM)=SIGN(AROOT,KVAL(K))
************************************************************************
*C
*C        SIGN(A,B)...RETURNS VALUE OF A WITH SIGN OF B
*C
************************************************************************
45     CONTINUE
```

```
          NROOTS(IM)=K
          PRINT 113
    113   FORMAT('OROOTS ARE')
          PRINT FMT,(Y(I),KVAL(I),I=1,K)
     28   CONTINUE
*******************************************************************************
*C        SCALING AND GRAPHING
*C
*C        SCALE AND GRAPH...
*C        SYSTEMS LIBRARY ROUTINES WHICH WILL SCALE DATA AND
*C        SET UP PLOT ARRAY AS 100 CHARACTER IMAGE
*C        OF LINE TO BE PRINTED
*******************************************************************************
          PRINT 14
          FACTOR=(TOP-YO)/10.0
          CALL SCALE(YO,FACTOR,YO,FACTOR,YO,FACTOR,YO,FACTOR,YO,FACTOR
         1          ,YO,FACTOR,YO,FACTOR,YO,FACTOR,YO,FACTOR,YO,FACTOR
         2          ,YO,FACTOR,YO,FACTOR)
          DO 46   KK=1,M
          DO 47   I=1,12
          IF(YVAR(I,KK))81,80,81
     80   YVAR(I,KK)=YO
     81   CHAR(I)=1H
     47   CONTINUE
          K=NROOTS(KK)
          DO 48     I=1,K
          IF(YVAR(I,KK))50,51,51
     51   CHAR(I)=1H+
          GO TO 49
     50   CHAR(I)=1H-
     49   YVAR(I,KK)=ABS(YVAR(I,KK))
     48   CONTINUE
          CALL  GRAPH(PLOT,YVAR(1,KK),CHAR(1),YVAR(2,KK),CHAR(2),YVAR(3,KK),
         1CHAR(3),YVAR(4,KK),CHAR(4),YVAR(5,KK),CHAR(5),YVAR(6,KK),CHAR(6),
         2YVAR(7,KK),CHAR(7),YVAR(8,KK),CHAR(8),YVAR(9,KK),CHAR(9),YVAR(10,
         3KK),CHAR(10),YVAR(11,KK),CHAR(11),YVAR(12,KK),CHAR(12))
          PRINT 90,X(KK),PLOT
     90   FORMAT(1HS,F10.2,5X,100C)
     46   CONTINUE
          PRINT 88
     88   FORMAT(16X,'$',98X,'$')
          PRINT 89,TOP,FACTOR
     89   FORMAT(16X,3H0.0,94X,F4.1/17HOSCALE FACTOR IS ,F4.1,16H  UNITS PER
         1 INCH)
          READ 97,ISTOP
          GO TO (10,52),ISTOP
     52   STOP
```

```
            SUBROUTINE COMPCO(C,NN)
*******************************************************************************
*C          SUBROUTINE READS IN REAL AND/OR COMPLEX ROOTS AND COMPUTES
*C          COEFFICIENTS OF THE RESULTING POLYNOMIAL.
*C
*C          C = ARRAY OF COEFFICIENTS
*C          NN = DEGREE OF POLYNOMIAL+1
*******************************************************************************
            CHARACTER FMD(5)
            DIMENSION CRTS(6,2),RRTS(12),C(15),A(6,13),D(15)
            DATA FMD/'(Z -('/
            DO 9,I=1,15
            D(I)=0.0
      9     C(I)=0.0
            DO 1 J=1,6
            DO 1 I=1,2
      1     CRTS(J,I)=0.0
            DO 2 J=1,12
      2     RRTS(J)=0.0
            DO 3 J=1,6
            DO 3 I=1,13
      3     A(J,I)=0.0
            LOWLIM=1
            II=2
            READ 100,M,N
      100   FORMAT(2I5)
            IF(M)25,26,25
      25    READ 101,((CRTS(I,J),J=1,2),I=1,M)
            PRINT 200,(FMD,(CRTS(I,J),J=1,2),I=1,M)
      200   FORMAT(1H0,6(5C,F5.1,3H+/-,F5.1,3HI))))
            GO TO 28
      26    II=1
      28    IF(N)27,32,27
      27    READ 101,(RRTS(I),I=1,N)
            PRINT 201 ,(FMD,RRTS(I),I=1,N)
      201   FORMAT(1H0,10(5C     ,F5.1,2H))))
      32    GO TO(30,31),II
      101   FORMAT(16F5.1)
      31    DO 10,I=1,M
            A(I,1)=1.0
            A(I,2)=-2.0*CRTS(I,1)
      10    A(I,3)=CRTS(I,1)**2+CRTS(I,2)**2
            DO 11,I=1,3
      11    C(I)=A(1,I)
            IF(M-1)12,16,12
      12    J=2
            LIM=3
      17    DO 35,I=1,LIM
      35    D(I)=C(I)
            LIM=LIM+2
```

```
      DO 15,I=1,LIM
      C(I)=0.0
      DO 15,K=1,I
15    C(I)=C(I)+A(J,K)*D(I+1-K)
      J=J+1
      IF(M-J)16,17,17
30    C(1)=1.0
      C(2)=-RRTS(1)
      LOWLIM=2
      NN=2
      GO TO 18
16    NN=2*M+1
      IF(N)18,19,18
18    IF(LOWLIM-N)50,50,19
50    DO 20 J=LOWLIM,N
      T=-RRTS(J)
      NN=NN+1
      DO 20,K=1,NN
      T1=-RRTS(J)*C(K+1)
      C(K+1)=T+C(K+1)
      T=T1
20    CONTINUE
19    RETURN
```

```
      FUNCTION P(X,A,N)
**************************************************************************
*C       ROUTINE TO EVALUATE POLYNOMIAL AT SOME POINT
*C
*C       X = POINT OF EVALUATION
*C       N = DEGREE OF POLYNOMIAL
*C       A = COEFFICIENTS OF POLYNOMIAL WITH A(1) = COEFFICIENT OF X**0
**************************************************************************
      DIMENSION A(25)
      IF(N)1,2,3
2     P=A(1)
      RETURN
1     P=0.0
      RETURN
3     Y=A(1)
      NT=N+1
      DO 10 I=2,NT
10    Y=Y*X+A(I)
      P=Y
      RETURN
```

```
      SUBROUTINE SEARCH(LO,HI,DEC,S,J,A,N)
*******************************************************************************
*C     DIVIDES THE INTERVAL LO , HI  INTO INTERVALS OF  LENGTH DEC
*C     RETURNS LOWER ENDPOINT OF INTERVAL IN WHICH POLYNOMIAL CHANGES SIGN
*C     S = ARRAY OF ENDPOINTS
*C     J = NO. OF SIGN CHANGE INTERVALS
*C     A = COEFFICIENTS OF POLYNOMIAL
*C     N = DEGREE OF POLYNOMIAL
*******************************************************************************
      DIMENSION S(25),A(25)
      REAL LO
      TEMPHI=HI
      J=0
      Y1=T(TEMPHI,A,N)
   10 Y2=T(TEMPHI-DEC,A,N)
      Y=Y1*Y2
      IF(Y)11,13,12
   13 J=J+1
      S(J)=TEMPHI-DEC
      Y1=T(TEMPHI-DEC-DEC/10.0,A,N)
      GO TO 14
   11 J=J+1
      S(J)=TEMPHI-DEC
   12 Y1=Y2
   14 TEMPHI=TEMPHI-DEC
      IF(TEMPHI-LO)15,15,10
   15 RETURN


      SUBROUTINE COMPDERIV(X)
*******************************************************************************
*C     ROUTINE TO EVALUATE DERIVATIVES OF THE POLYNOMIALS
*C     G AND H AT A POINT (X)
*******************************************************************************
      COMMON CG(13,13),CH(13,13),NG,NH,G(26),H(26),TAU
      DIMENSION COEF1(13),COEF2(13)
      DO 10 J=1,26
      G(J)=0.0
      H(J)=0.0
   10 CONTINUE
      DO 12 J=0,12
      JJ=J+1
      DO 13 I=1,13
      COEF1(I)=CG(JJ,I)
   13 COEF2(I)=CH(JJ,I)
      N=NG-J
      K=NH-J
      G(J+1)=P(X,COEF1,NG-J)
      H(J+1)=P(X,COEF2,NH-J)
   12 CONTINUE
      RETURN
```

```
      SUBROUTINE ROOT (XL,DLTA,A,N,E,AROOT)
*******************************************************************************
*C    SUBROUTINE EMPLOYS THE HALF INTERVAL METHOD TO LOCATE A ROOT
*C    OF A POLYNOMIAL GIVEN THAT THERE IS A SIGN CHANGE IN THE INTERVAL
*C    XL,XL+DLTA.
*C    A = COEFFICIENTS OF POLYNOMIAL
*C    N = DEGREE OF POLYNOMIAL
*C    E = ERROR CONTROL
*C    AROOT = APPROXIMATION TO ROOT SUCH THAT..
*C    ABS(AROOT-TRUE VALUE OF ROOT)  LESS THAN  E/2
*******************************************************************************
      DIMENSION A(25)
      H=DLTA
   21 XR=XL+H/2.0
   22 YL=T(XL,A,N)
      YR=T(XR,A,N)
      Y=YL*YR
      IF(Y)9,10,11
    9 IF(ABS(XR-XL)-E)15,20,20
   20 H=H/2.0
      GO TO 21
   11 XL=XR
      XR=XR+H/2.0
      GO TO 22
   10 IF(YL)23,24,23
   23 AROOT=XR
      RETURN
   24 AROOT=XL
      RETURN
   15 AROOT=XL+ABS(XR-XL)/2.0
      RETURN

      FUNCTION T(Y,A,N)
*******************************************************************************
*C    ROUTINE TO COMPUTE TIME LAG FUNCTION,  I.E.,
*C    MULTIPLY EACH A(I) BY  SIN(TAU*Y)  OR  COSINE(TAU*Y)
*C    N = DEGREE OF POLYNOMIAL (COEFFICIENTS A(I)  )
*******************************************************************************
      COMMON CG(13,13),CH(13,13),NG,NH,G(26),H(26),TAU
      DIMENSION A(25),TRIG(2)
      IF(TAU)1,2,1
    2 T=P(Y,A,N)
      RETURN
    1 K=(N-2*(N/2))+1
      IF(K-1)6,7,6
    6 TRIG(1)=SIN(TAU*Y)
      TRIG(2)=-COS(TAU*Y)
      GO TO 20
    7 TRIG(1)=-COS(TAU*Y)
      TRIG(2)=SIN(TAU*Y)
   20 X=A(1)*TRIG(2)
      NT=N+1
      DO 30 I=2,NT
      II=(I-2*(I/2))+1
      X=X*Y+A(I)*TRIG(II)
   30 CONTINUE
      T=X
      RETURN
```

```
      FUNCTION COMPK(Y,X)
***********************************************************************
*C
*C          ROUTINE TO COMPUTE  K
*C
***********************************************************************
      COMMON CG(13,13),CH(13,13),NG,NH,G(26),H(26),TAU
      DIMENSION YP(14)
      REAL IMG,IMH
      MAX=MAXO(NG,NH)+2
      YP(1)=1
      YP(2)=Y
      DO 10 I=3,MAX
      YP(I)=YP(I-1)*Y
   10 CONTINUE
      REALG=0.0
      IMG=0.0
      N2=NG/2
      S=-1.0
      DO 11 J=0,N2
      IO=2*J+1
      IE=2*J
      S=S*(-1.0)
      REALG=REALG+S*G(IE+1)/FACT(IE)*YP(IE+1)
      IMG=IMG+S*G(IO+1)/FACT(IO)*YP(IO+1)
   11 CONTINUE
      M2=NH/2
      S=-1.0
      REALH=0.0
      IMH=0.0
      DO 12 J=0,M2
      IO=2*J+1
      IE=2*J
      S=S*(-1.0)
      REALH=REALH+S*H(IE+1)/FACT(IE)*YP(IE+1)
      IMH=IMH+S*H(IO+1)/FACT(IO)*YP(IO+1)
   12 CONTINUE
      DENOM=REALH*REALG+IMH*IMG
      COMPK=-EXP(TAU*X)*COS(TAU*Y)*(REALG**2+IMG**2)/DENOM
      CALL CKBADARITH($1,$1)
***********************************************************************
*C
*C          CKBADARITH ..SYSTEMS ROUTINE TO CHECK ARITHMETIC INDICATORS
*C
***********************************************************************
    1 RETURN
```

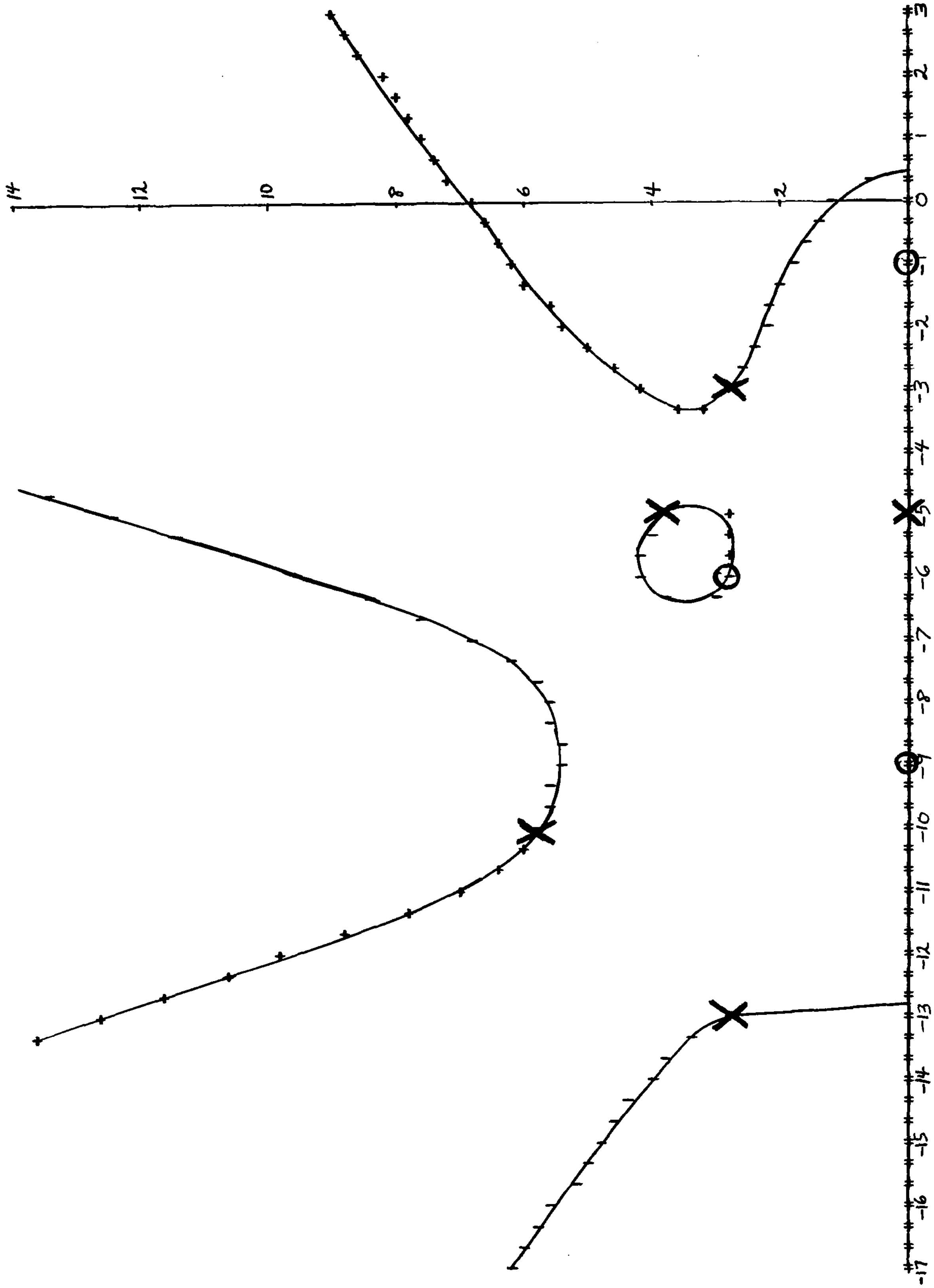
```
          SUBROUTINE FRMAT(FMT,ISIG)
*******************************************************************************
*C
*C       FORMAT CONTROL SUBROUTINE
*C
*******************************************************************************
          CHARACTER FMT(25),FMD(9),FMAT(25)
          DATA FMAT/'(9X,F14.6,9X,3HK =,E17.8)'/,FMD/'123456789'/
          DO 10 I=1,25
          FMT(I)=FMAT(I)
   10     CONTINUE
          FMT( 9)=FMD(ISIG)
          RETURN
```

Examples

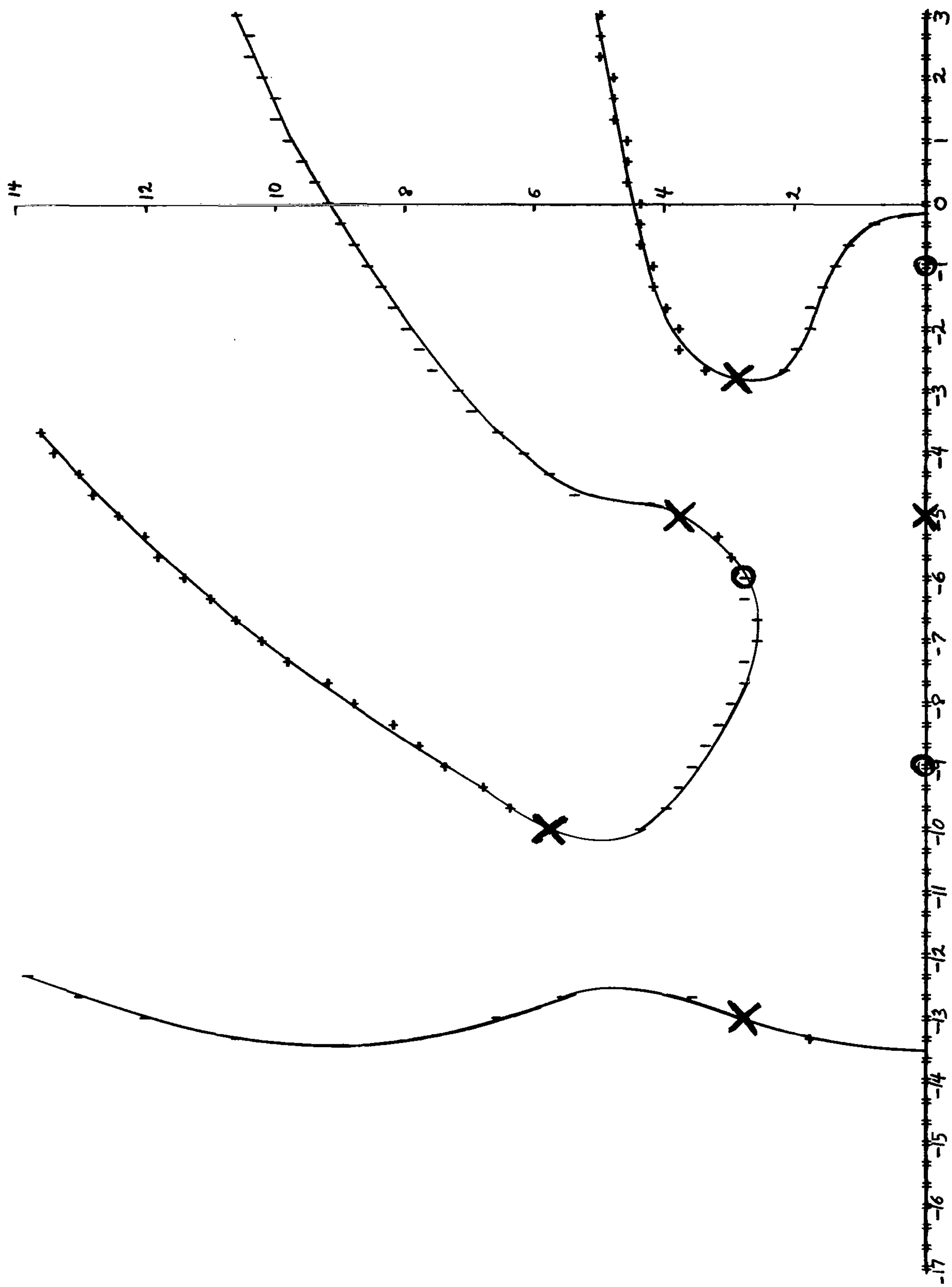
The following diagrams are the root-loci for $g(z) + Ke^{-\tau z}h(z) = 0$ where

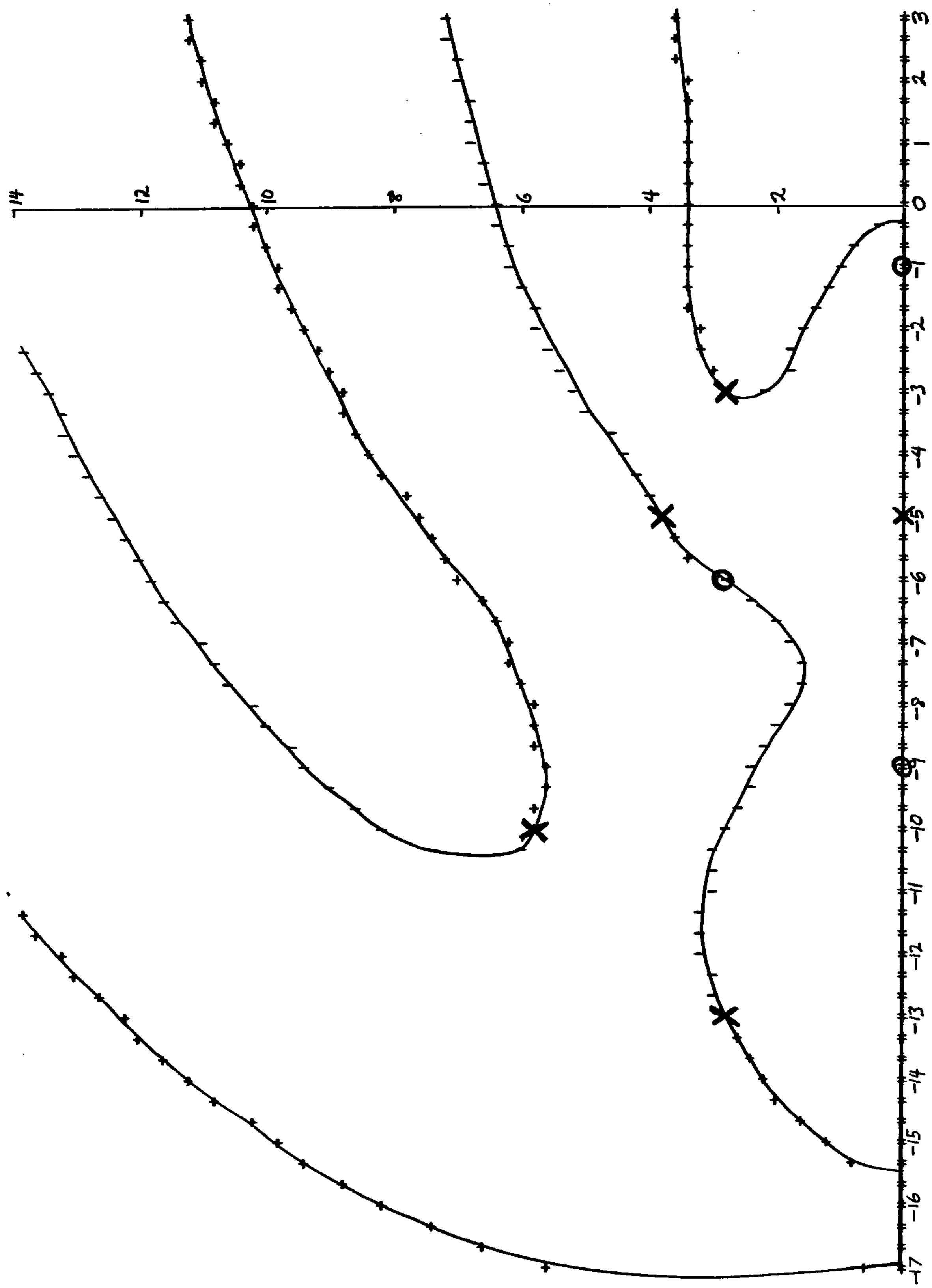
$$g(z) = (z+5)(z+3+3i)(z+3-3i)(z+5+4i)(z+5-4i)$$
$$\cdot\ (z+10+6i)(z+10-6i)(z+13+3i)(z+13-3i),$$

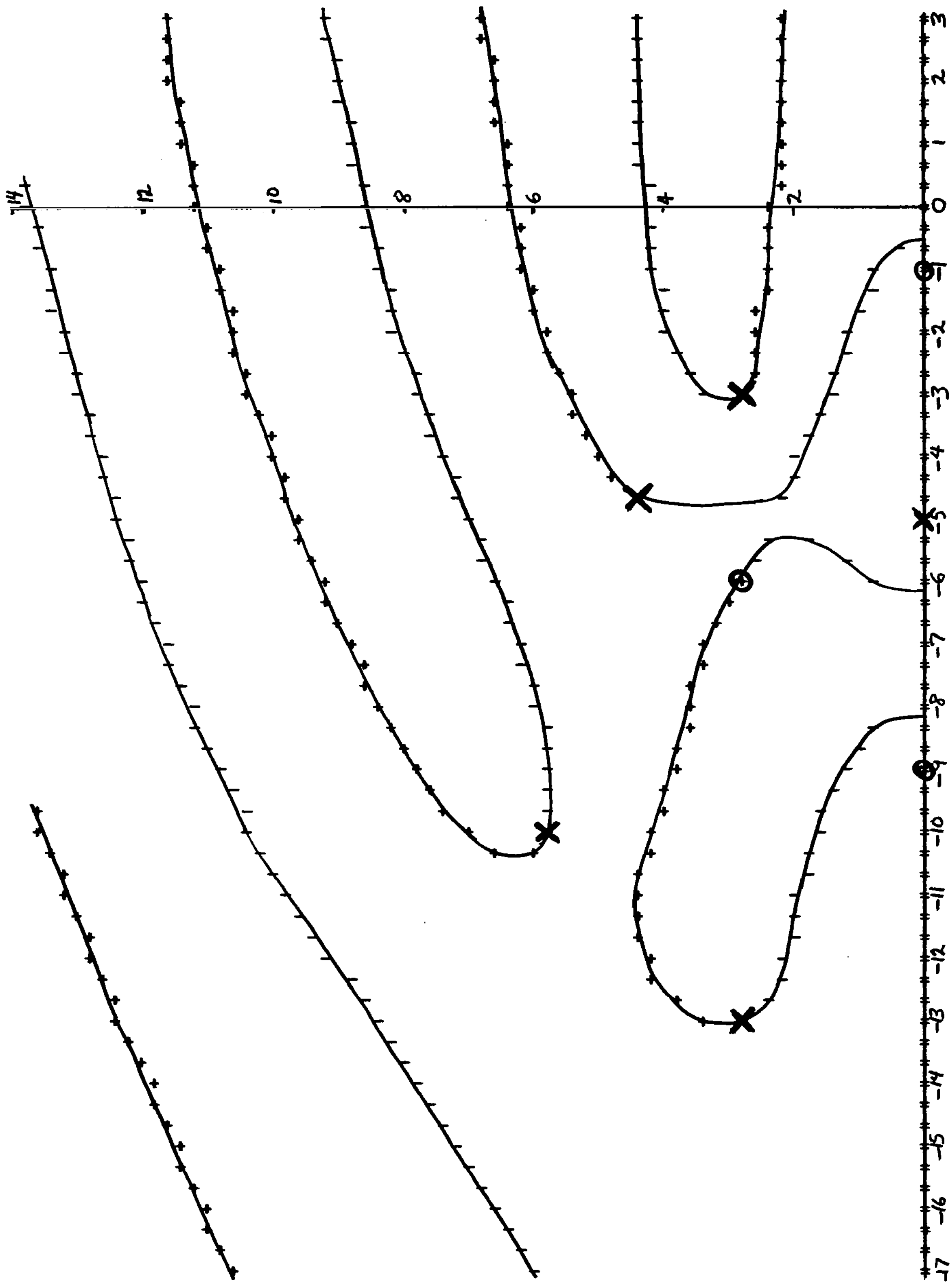
$h(z) = (z+1)(z+a)(z+6+3i)(z+6-3i)$, τ is successively 0, 1/4, 1/2, 1, and 2.

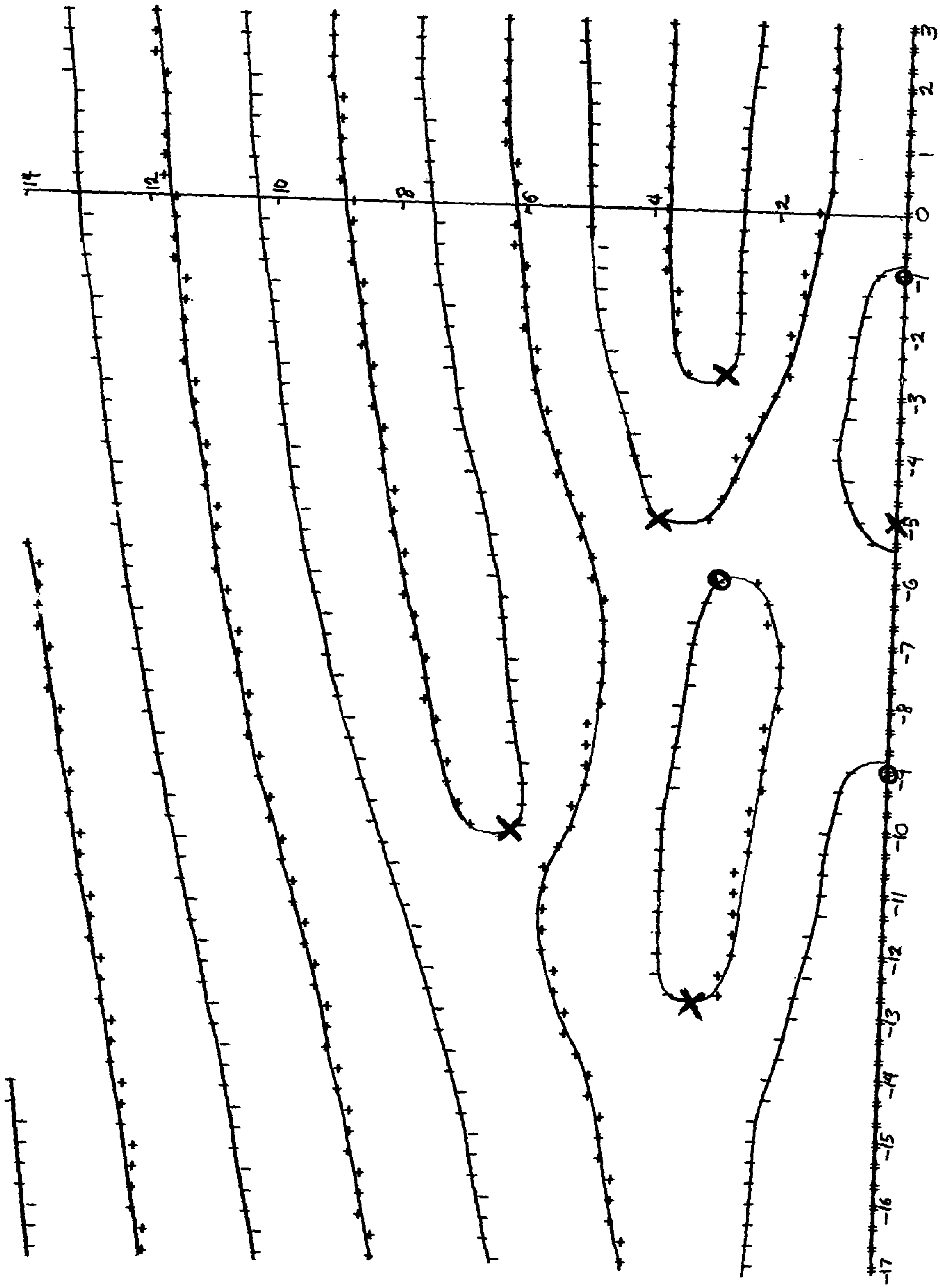


14
12
10
8
6
4
2
3
2
1
0
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
-11
-12
-13
-14
-15
-16
-17


14
12
10
8
6
4
2
0
3
2
1
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
-11
-12
-13
-14
-15
-16
-17


14
12
10
8
6
4
2
0
3
2
1
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
-11
-12
-13
-14
-15
-16
-17


3
2
1
0
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
-11
-12
-13
-14
-15
-16
-17
2
4
6
8
10
12
14


-14
-12
-10
-8
-6
-4
-2
0
3
2
1
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
-11
-12
-13
-14
-15
-16
-17

REFERENCES

1. H. W. Bode, "Network Analysis and Feedback Amplifier Design," D. Van Nostrand Co., New York, 1948.

2. W. R. Evans, "Graphical Analysis of Control Systems," Trans. A.I.E.E., 67, (1948) pp. 547-551.

3. __________, "Control System Synthesis by the Root-Locus Method," Trans. A.I.E.E., 69, (1950), pp. 66-69.

4. A. M. Krall, "An Extension and Proof of the Root-Locus Method," Jour. S.I.A.M., 9, (1961) pp. 644-653.

5. __________, "A Closed Expression for the Root Locus Method," Jour. S.I.A.M., 11, (1963), pp. 700-704.

6. __________, "Stability Criteria for Feedback Systems with a Time Lag," Jour. S.I.A.M., Ser. A, Control, 2.2, (1965), pp. 160-170.

7. A. V. Michailov, "Harmonic Analysis in the Theory of Automatic Control," Auto. i Tele., Moscow, 1938.

8. J. I. Neimark, "On the Distribution of Roots of Polynomials," Dokl. Akad. Nauk., 58, (1947).

9. __________, "On the Structure of the D-Partitions of Polynomials," Dokl. Akad. Nauk, 59, (1948).

10. H. Nyquist, "Regeneration Theory," Bell System Tech. J., 11, (1932).

11. L. S. Pontrjagin, "On the Zeros of Some Elementary Transcendental Functions," Amer. Math. Soc. Transl. Ser. 2, Vol. 1, (1955) pp. 95-110.

NASA-Langley, 1966

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Technical information generated in connection with a NASA contract or grant and released under NASA auspices.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

TECHNICAL REPRINTS: Information derived from NASA activities and initially published in the form of journal articles.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities but not necessarily reporting the results of individual NASA-programmed scientific efforts. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546