ANL-6716
Mathematics and
Computers
(TID-4500, 20th Ed.)
AEC Research and
Development Report

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois

TRANSFER FUNCTION SYNTHESIS AS A RATIO OF TWO COMPLEX POLYNOMIALS

by

C. K. Sanathanan
Reactor Engineering Division

and

Judith Koerner
Applied Mathematics Division

May, 1963

Operated by The University of Chicago
under
Contract W-31-109-eng-38
with the
U.S. Atomic Energy Commission

TABLE OF CONTENTS

LIST OF FIGURES

TABLE

TRANSFER FUNCTION SYNTHESIS AS A RATIO OF TWO COMPLEX POLYNOMIALS

by

C. K. Sanathanan and Judith Koerner

ABSTRACT

Experimental data for frequency response obtained from a linear dynamic system is processed to obtain the transfer function as a ratio of two frequency-dependent polynomials. The difference between the absolute magnitudes of the actual function and the polynomial ratio is the error considered. The polynomial coefficients are evaluated as the result of minimizing the sum of the squares of the above errors at the experimental points. The magnitude and phase angle of the transfer function are evaluated at various frequencies by means of the computed polynomial ratio and are compared with the observed data.

The numerical solution of this problem was obtained by using an IBM 704 FORTRAN program.

The method presented here gives an analytic description of the complex transfer function superior to that given by minimization of the "weighted" sum of the squares of the errors in magnitude.

This method is applicable to both minimum and non-minimum phase systems.

I. INTRODUCTION

It is often desirable to express the transfer function G(s) of a linear dynamic system as a ratio of two frequency-dependent polynomials, namely,

$$G(j\omega)^{*} \simeq \frac{p_0 + p_1(j\omega) + p_2(j\omega)^2 + \ldots}{1 + q_1(j\omega) + q_2(j\omega)^2 + \ldots}$$

$$= \frac{P(j\omega)}{Q(j\omega)} \quad . \qquad (1)$$

*Setting $q_0 = 1$, does not restrict the problem in any way.

Several methods(1-3) have been devised in the past to fit the experimental data with a function such as the above. In the following, one such method is presented briefly along with its deficiencies. A procedure is suggested to eliminate the deficiencies.

The error at frequency ω_k is given by

$$\epsilon_k = G(j\omega_k) - \frac{P(j\omega_k)}{Q(j\omega_k)} \quad . \tag{2}$$

The problem becomes quite difficult to solve when the coefficients $p_0, p_1, p_2, \ldots, q_1, q_2, \ldots$ are evaluated as a result of simply minimizing the sum of $|\epsilon_k|^2$ at all the experimental points. If Eq. (2) is multiplied by $Q(j\omega_k)$, the weighting function, the weighted error at point k is

$$\epsilon'_k = \epsilon_k Q(j\omega_k) = G(j\omega_k)Q(j\omega_k) - P(j\omega_k) \quad , \tag{3}$$

and the sum of $|\epsilon'_k|^2$ for all the experimental frequencies is

$$E(p_0, p_1, p_2, \ldots, q_1, q_2, q_3, \ldots) = \sum_{k=1}^{n} |\epsilon'_k|^2 = \sum_{k=1}^{n} |\epsilon_k|^2 |Q_k|^2 \quad . \tag{4}$$

The sum E is partially differentiated with respect to each polynomial coefficient and equated to zero. The resulting set of linear simultaneous algebraic equations are arranged in the matrix equation form

$$[A][X] = [B] \tag{5}$$

and solved to obtain the polynomial coefficients characterized by the "weighted" minimum mean-square-error criterion.

The above has the following deficiency:

The weighting function $|Q(j\omega_k)|^2$ may vary considerably as ω_k is increased through several decades, and at higher frequencies may attain values considerably higher than those at lower frequencies. Because of the heavy weighting of the errors at the higher frequencies, there is a general tendency for the contributions of the lower frequencies to E to become ineffective. Therefore, this method may be expected to give a poor fit at lower frequencies, which it actually does.

It is suggested that the above deficiency may be overcome by eliminating the weighting by an iterative procedure.

Equation (3) is modified such that

$$\epsilon''_k = \frac{(\epsilon'_k)_L}{Q(j\omega k)_{L-1}} = \frac{\epsilon_k Q(j\omega_k)_L}{Q(j\omega_k)_{L-1}} = \frac{G(j\omega_k)Q(j\omega_k)_L}{Q(j\omega_k)_{L-1}} - \frac{P(j\omega_k)_L}{Q(j\omega_k)_{L-1}} \quad , \tag{6}$$

where the subscript L corresponds to the iteration number. As $Q(j\omega_k)$ is not known to begin with, it is set equal to 1. The subsequent iterations converge rapidly and ϵ_k'' tends to be equal to ϵ_k, and the weighting ceases to exist.

II. ANALYSIS*

From Eq. (6),

$$\left|\epsilon_k''\right|^2 = \left|G(j\omega_k)\, Q(j\omega_k)_L - P(j\omega_k)_L\right|^2 \Big/ \left|Q(j\omega_k)_{L-1}\right|^2 \quad . \tag{7}$$

Substituting $W_{kL} = 1\Big/\left|Q(j\omega_k)_{L-1}\right|^2$ in Eq. (6), summing for all k's, and calling the result E', there is obtained

$$E' = \sum_{k=1}^{n} \left|\epsilon_k''\right|^2 = \sum_{k=1}^{n} \left|\epsilon_k'\right|^2 W_{kL} \quad , \tag{8}$$

where ϵ_k' is a function of $p_0, p_1, p_2, \ldots, q_1, q_2, q_3, \ldots$ The sum E' is now partially differentiated with respect to each of the polynomial coefficients and equated to zero to evaluate the coefficients. This yields the following matrix equation:

$$\begin{bmatrix}
\lambda_0 & 0 & -\lambda_2 & 0 & \lambda_4 & \cdots & T_1 & S_2 & -T_3 & S_4 & \cdots \\
0 & \lambda_2 & 0 & -\lambda_4 & 0 & \cdots & -S_2 & T_3 & S_4 & -T_5 & \cdots \\
\lambda_2 & 0 & -\lambda_4 & 0 & \lambda_6 & \cdots & T_3 & S_4 & -T_5 & -S_6 & \cdots \\
0 & \lambda_4 & 0 & -\lambda_6 & 0 & \cdots & -S_4 & T_5 & S_6 & -T_7 & \cdots \\
\cdot & \cdot & \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot & \cdot & \\
\cdot & \cdot & \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot & \cdot & \\
\cdot & \cdot & \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot & \cdot & \\
T_1 & -S_2 & -T_3 & S_4 & & \cdots & U_2 & 0 & -U_4 & 0 & \cdots \\
S_2 & T_3 & -S_4 & -T_5 & & \cdots & 0 & U_4 & 0 & -U_6 & \cdots \\
T_3 & -S_4 & -T_5 & S_6 & & \cdots & U_4 & 0 & -U_6 & 0 & \cdots \\
S_4 & T_5 & -S_6 & -T_7 & & \cdots & 0 & U_6 & 0 & -U_8 & \cdots \\
\cdot & \cdot & \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot & \cdot & \\
\cdot & \cdot & \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot & \cdot & \\
\cdot & \cdot & \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot & \cdot &
\end{bmatrix}
\begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ \cdot \\ \cdot \\ \cdot \\ q_1 \\ q_2 \\ q_3 \\ q_4 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}
=
\begin{bmatrix} S_0 \\ T_1 \\ S_2 \\ T_3 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ U_2 \\ 0 \\ U_4 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \tag{9}$$

*Since the analysis given here is quite brief, the reader may find it helpful to refer to Levy[2].

where

$$\lambda_i = \sum_{k=1}^{n} (\omega_k)^i W_{kL} \quad ; \tag{10}$$

$$S_i = \sum_{k=1}^{n} (\omega_k)^i R_k W_{kL} \quad ; \tag{11}$$

$$T_i = \sum_{k=1}^{n} (\omega_k)^i I_k W_{kL} \quad ; \tag{12}$$

$$U_i = \sum_{k=1}^{n} (\omega_k)^i (R_k^2 + I_k^2) W_{kL} \quad . \tag{13}$$

Here, R_k and I_k are the real and imaginary parts of the transfer function at ω_k obtained experimentally.

The coefficients q_1 q_2, q_3, . . . evaluated at iteration L - 1 are used to evaluate W_L for the next iteration.

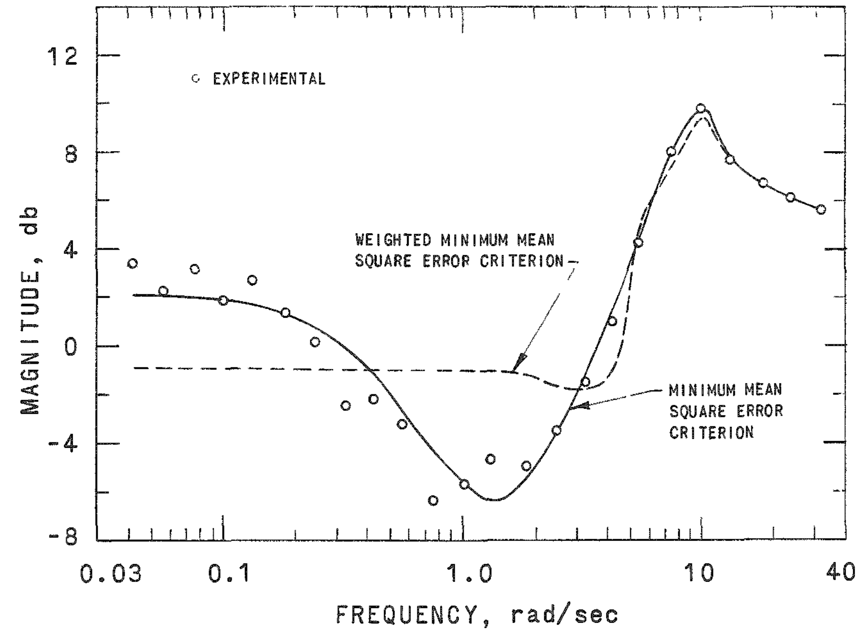
III. EXAMPLE

The experimentally measured data for the transfer function of EBWR (Experimental Boiling Water Reactor) operated at a thermal power of 40 Mw and a pressure of 600 psi are fitted with the transfer functions obtained by the least mean-square-error criterion as well as the "weighted" least mean-square-error criterion.

As the reactor transfer function is believed to have an excess pole over the number of zeros, the numerator polynomial is made to be of one degree less than the denominator polynomial. In this example, the numerator polynomial is chosen to be of degree 6.

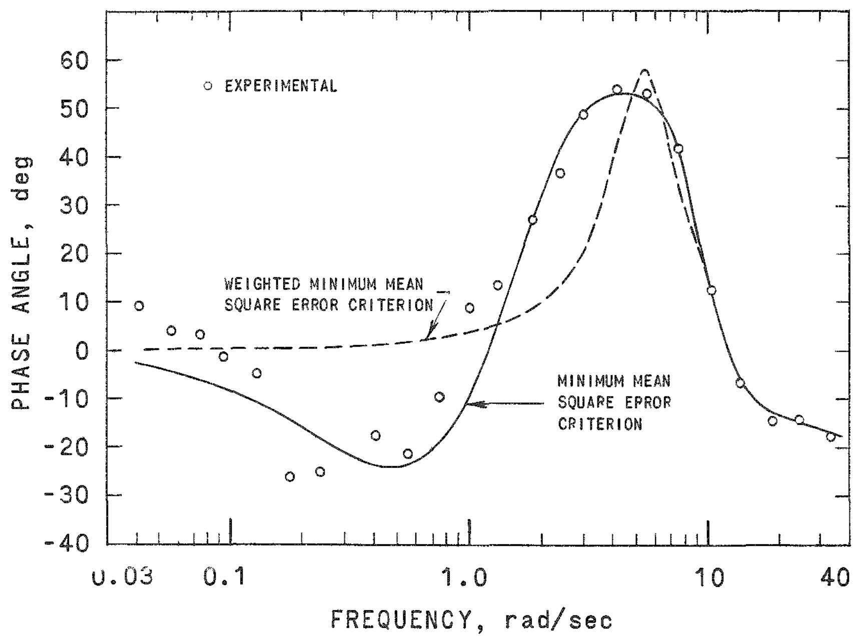
The experimental data consist of the magnitude and the phase angle of the transfer function at 24 frequencies ranging between 0.03 and 40 radians per second. The polynomial coefficients for the "weighted" minimum mean-square error are obtained at the end of the first iteration of Eq. (9), and those for the least mean-square error are obtained at the end of the tenth iteration. In general, the number of iterations depends largely upon the nature of the transfer function and the desired accuracy in the values of the coefficients. The magnitude and the phase angle of the transfer function are also computed at the experimental frequencies from the polynomial coefficients.

The results are shown in Figs. 1 and 2 and in Table I. It is to illustrate the insufficiency of the "weighted" minimum mean-square-error criterion clearly that the authors have chosen the synthesis of a fairly large transfer function such as that of a nuclear reactor.



112-2074

Fig. 1. Magnitude of the Transfer Function vs Frequency



112-2073

Fig. 2. Phase Angle of the Transfer Function vs Frequency

Table I

POLYNOMIAL COEFFICIENTS

	Evaluated by the Weighted Error Criterion (Iteration 1)	Evaluated by the Minimum Mean-square-error Criterion (Iteration 10)
p_0	$9.2894 \cdot 10^{-1}$	1.2768
p_1	$1.7123 \cdot 10^{-1}$	1.2803
p_2	$1.1254 \cdot 10^{-1}$	$7.8233 \cdot 10^{-1}$
p_3	$1.0363 \cdot 10^{-2}$	$7.9181 \cdot 10^{-2}$
p_4	$1.6044 \cdot 10^{-3}$	$8.0884 \cdot 10^{-3}$
p_5	$8.8747 \cdot 10^{-6}$	$2.8947 \cdot 10^{-4}$
p_6	$4.7145 \cdot 10^{-6}$	$2.0136 \cdot 10^{-5}$
q_0	1.0000	1.0000
q_1	$2.1108 \cdot 10^{-1}$	2.5314
q_2	$8.7539 \cdot 10^{-2}$	$4.2697 \cdot 10^{-1}$
q_3	$5.6295 \cdot 10^{-3}$	$5.4644 \cdot 10^{-2}$
q_4	$1.0042 \cdot 10^{-3}$	$4.5364 \cdot 10^{-3}$
q_5	$1.1678 \cdot 10^{-5}$	$1.9840 \cdot 10^{-4}$
q_6	$2.5212 \cdot 10^{-6}$	$1.1446 \cdot 10^{-5}$
q_7	$2.2947 \cdot 10^{-8}$	$1.0519 \cdot 10^{-7}$

Note: Running time on the IBM 704 for the above problem was approximately one minute.

IV. DISCUSSION

In the example cited above, $|Q(j\omega)|^2$ varied from 1.0 to 6.25×10^7 through the frequency range.

The present method is not applicable to functions that have poles at the origin. However, the experimental data may be modified to have them fitted with a function that does not have poles at the origin, and, then, the required number of poles at the origin may be introduced to this function. This procedure is clearly illustrated by Levy.(2)

Suitable scaling of Eq. (9) was necessary for successful computation; namely, Eq. (9) was solved by writing

$$10^{-8}[A][X] = 10^{-8}[B] \quad .$$

The reciprocal of the geometric mean between the first and the last matrix elements is a reasonable scale factor. The expression for $G(j\omega)$ obtained after the first iteration corresponds to the minimization of the sum of the squares of the weighted errors, since W_{kl} is initially set equal to unity for all k's.

Table I shows that the polynomial coefficients are altered considerably in the subsequent iterations.

V. THE COMPUTER PROGRAM

An IBM 704 FORTRAN program, TRAFICORPORATION, was developed for the numerical solution of this problem. This program is given in its complete form in the following discussion.

A. Description of the Programming of the Computing Procedure

Symbols Used	
ALPA	Absolute value of the ratio between the imaginary and the real part of the numerator.
ALTER1	Absolute value of the calculated magnitude of the transfer function in decibels.
ALTER2	Absolute value of the calculated phase of the transfer function in degrees.
AMDA	λ
BETA	Absolute value of the ratio between the imaginary and the real parts of the denominator.
CDBM	Calculated magnitude of the transfer function in decibels.
CEM	Calculated magnitude of the transfer function.
CONV1	Convergence criterion for the magnitude. $\vert\vert ERMAG1\vert - \vert ERMAG2\vert\vert \leq CONV1$
CONV2	Convergence criterion for the phase. $\vert\vert ERFAZ1\vert - \vert ERFAZ2\vert\vert \leq CONV2$
CPH	Calculated phase of the transfer function in degrees.

Symbols Used	
CPHR	Calculated phase of the transfer function in radians.
CURNER	If the error in magnitude exceeds the value (CURNER · ALTER1) at a given point, the calculated value of the magnitude is substituted for the experimental magnitude.
DBM	Experimental magnitude in decibels.
EM	Experimental magnitude.
ERFAZ1	Maximum error in the phase for iteration L - 1.
ERFAZ2	Maximum error in the phase for iteration L.
ERMAG	Same as ERMAG2.
ERMAG1	Maximum error in magnitude for iteration L - 1.
ERMAG2	Maximum error in magnitude for iteration L.
ERORM	Absolute value of the difference between the experimental and calculated magnitudes in decibels.
ERORP	Absolute value of the difference between the experimental and calculated phases in degrees.
ERPASE	Same as ERFAZ2.
IDNO	Identification number of the problem.
L	The current iteration number
MAXPT1	Point at which the maximum error in magnitude has occurred in iteration L - 1.
MAXPT2	Point at which the maximum error in magnitude has occurred for iteration L.
MAXPT3	Point at which the maximum error in phase has occurred in iteration L - 1.
MAXPT4	Point at which the maximum error in phase has occurred in iteration L.
N	The degree of the denominator.
NOLAMD	Number of lambdas in the matrix.

Symbols Used	
NOMEGA	Number of experimental points.
OMEGA	ω
PABS	Magnitude of the numerator.
PABS1	Real part of the numerator.
PABS2	Imaginary part of the numerator.
PABSQ	Square of the magnitude of the numerator.
PH	Experimental phase in radians.
PHE	Experimental phase in degrees.
QABS	Magnitude of the denominator.
QABS1	Real part of the denominator.
QABS2	Imaginary part of the denominator.
QABSQ	Square of the magnitude of the denominator.
R	Real part of the experimental transfer function.
SCALE	Scale factor, used to decrease the size of the matrix elements.
THETA1	$\tan^{-1}$ (ALPA).
THETA2	$\tan^{-1}$ (BETA).
UR	Imaginary part of the experimental transfer function.
W	Reciprocal of the square of the magnitude of the denominator.
WALT	If the error in phase exceeds the value (WALT · ALTER2) at a given point, the calculated value of the phase is substituted for the experimental phase.
WX	Initial value of W.

B. The Computing Procedure

Initially, the required λ's, S's, T's and U's are computed and substituted in the matrices [A] and [B]. The matrix-inversion subroutine, ANF 402,* is used to solve the matrix equation [A] [X] = [B] to obtain the polynomial coefficients $p_0, p_1, \ldots, p_{N-1}$, and $q_1, q_2, \ldots, q_N$. Here, q_0 is set equal to 1. The magnitude, the phase angle, and the errors of the transfer function are computed at each experimental point as follows:

1. Magnitude of the transfer function $= \left[\frac{(PABS1)^2 + (PABS2)^2}{(QABS1)^2 + (QABS2)^2}\right]^{\frac{1}{2}}$

2. Phase of the transfer function $= \tan^{-1}\left(\frac{PABS2}{PABS1}\right) - \tan^{-1}\left(\frac{QABS2}{QABS1}\right)$

3. The errors are computed as the absolute difference between the experimental and calculated values of the magnitude and phase of the transfer function. The maximum errors in magnitude and phase are located.

4. $W = \frac{1}{(QABS1)^2 + (QABS2)^2}$ is used for the next iteration. If, at this time, the convergence criteria are met, the iterative procedure is ended and the output is obtained; if not, the matrix elements are recomputed and the above procedure is repeated.

Suitable scaling may be necessary for the successful solution of the matrix equation. The reciprocal of the geometric mean between the first and the last matrix elements is a reasonable value for SCALE.

C. Input Information

Card Set Number

1 FORMAT (3I6, 2E12.5)

IDNO, NOMEGA, N, CURNER, WALT

Note: IDNO $\leq$ 32, 768
NOMEGA $\leq$ 250
N $\leq$ 12

2 FORMAT (3E12.5)

OMEGA (J), DBM (J), PHE (J) J = 1, . . . , NOMEGA

Note: The OMEGA's need not be in ascending or descending order.

3 FORMAT (4E12.5)

WX, SCALE, CONV1, CONV2

Note: WX = 1.0 gives the polynomial coefficients with the weighted mean square error criterion at the end of the first iteration.

*ANF 402, Matrix Inversion with Accompanying Solution of Linear Equations (FORTRAN II), Burton S. Garbow, February 23, 1959.

D. Output Information

The iteration at which the problem converged, the maximum error in magnitude and phase, the point at which each occurred, the polynomial coefficients, the calculated magnitude, the calculated phase, the errors in magnitude and phase, and the reciprocal of the square of the magnitude of the denominator for each frequency are written on-line on tape 6 for each problem.

E. Operating Instructions

A standard 72-72 reader board, a SHARE 2 printer board, and an underflow switch are necessary for running this program.

SENSE SWITCHES:

1,2,3,4	Not used
5 UP:	Normal
DOWN:	ERMAG, the point at which ERMAG occurred; ERPASE, the point at which ERPASE occurred; the numerator and the denominator coefficients; and OMEGA, CDBM, CPH, ERORM, ERORP, and W at each point are printed on-line for the current iteration.
6 UP:	Normal
DOWN:	The matrix elements as they appear in the matrix and the λ's, S's, T's, and U's are printed on-line for the current iteration.

TAPES:

6	Blank for output

RUNNING PROCEDURE:

1. Mount and ready tape 6.

2. Depress the underflow switch and set the sense switches as desired.

3. Ready the program deck and the input cards in the card reader.

4. CLEAR and LOAD CARDS.

5. At the completion of a series of problems, write an EOF on tape 6 and remove for printing off-line on program control.

VI. THE FORTRAN PROGRAM LISTING

It is believed that the transfer function of a nuclear reactor has an excess pole over the number of zeros. Hence, the program RE 277A was written such that the numerator polynomial is of one degree less than the denominator polynomial.

If, in a problem, the form of the function is unknown, it is suggested that the observed data may be fitted by a ratio of two complex polynomials of equal degree. The program RE 277B was written to do this.

Both versions of the program use the same input information and have the same operating instructions.

The cutput for the sample problem cited in the article is also given. The results of the first iteration (the results obtained by the "weighted" mean square error criterion) were obtained by depressing sense switch 5.

A. RE 277A

```
      RE277A TRAFICORPORATION
C     TRANSFER FUNCTION EXPRESSED AS A RATIO OF TWO COMPLEX POLYNOMIALS
      WHERE THE NUMERATOR IS OF ONE DEGREE LESS THAN THE DENOMINATOR
C
      DIMENSION OMEGA(250),PH(250),DBM(250),R(250),UR(250),W(250),AMDA(5
     10),V(50),S(25),T(25),U(50),A(25,25),B(25),EM(250),P(15),Q(15),QABS
     21(250),QABS2(250),QABSQ(250),QABS(250),PABS1(250),PABS2(250),PABSQ
     3(250),PABS(250),CEM(250),CDBM(250),ERORM(250),CPH(250),THETA1(250)
     4,THETA2(250),ERORP(250),ALPA(250),BETA(250),PHE(250),CPHR(250)
C
C     READ INPUT
C
 1005 FORMAT(6E12.5)
 1006 FORMAT(3I6,4E12.5)
 1000 READ 1006,IDNO,NOMEGA,N,CURNER,WALT
      DO 1010 J = 1,NOMEGA
 1010 READ 1005,OMEGA(J),DBM(J),PHE(J)
 1500 READ 1005,WX,SCALE,CONV1,CONV2
      ERMAG1=0.
      ERFAZ1=0.
      MAXPT1=0
      MAXPT3=0
      ITER=1
      DO 1510 J = 1,NOMEGA
 1510 W(J) = WX
C
C     CONVERSION OF EXPERIMENTAL G(J) INTO THE REAL AND IMAGINARY PARTS
C
      DO 500 J = 1,NOMEGA
      PH(J) = PHE(J)*0.0174533
      EM(J) = EXPF(DBM(J)/8.6858896)
      R(J) = EM(J)*COSF(PH(J))
  500 UR(J) = EM(J)*SINF(PH(J))
      N2 = N*2
      QZERO = 1.
      NOLAMD = N*2-1
C
C     COMPUTATION OF LAMBDA,U,T AND S
C
    7 DO 15 J = 1,N2,2
    5 TEMP = 0.
      DO 10 K = 1,NOMEGA
      J1 = J-1
   10 TEMP = OMEGA(K)**J1*W(K)+TEMP
      AMDA(J) = TEMP
   15 AMDA(J+1) = 0.
   20 DO 35 J = 1,N2,2
   25 TEMP = 0.
      DO 30 K = 1,NOMEGA
      J1 = J+1
   30 TEMP = OMEGA(K)**(J1)*(R(K)**2+UR(K)**2)*W(K)+TEMP
      U(J) = TEMP
   35 U(J+1) = 0.
   40 DO 55 J = 1,N
   45 TEMP = 0.
      DO 50 K = 1,NOMEGA
      J2 = J*2-1
   50 TEMP = OMEGA(K)**(J2)*UR(K)*W(K)+TEMP
   55 T(J) = TEMP
      L=N-1
   60 DO 75 J= 1,L
   65 TEMP = 0.
```

```
      DO 70 K = 1,NOMEGA
      J2 = J*2
   70 TEMP = OMEGA(K)**(J2)*R(K)*W(K)+TEMP
   75 S(J) = TEMP
   80 TEMP = 0.
      DO 81 K = 1,NOMEGA
   81 TEMP = R(K)*W(K)+TEMP
      SZERO = TEMP
C
C     SUBSTITUTION OF PROPER MAGNITUDES IN THE MATRIX
C
      DO 2000 J = 1,N
      DO 2000 I = 1,N
      K = I+J-1
 2000 A(I,J) = AMDA(K)
      MST1 = N+1
      DO 2020 J = MST1,N2
      DO 2020 I = MST1,N2
      K = I + J-1-N2
 2020 A(I,J) = U(K)
      DO 6000 I = 1,N
      IV = 2*I
      V(IV-1) = T(I)
 6000 V(IV) = S(I)
      DO 6005 I = MST1,N2
      DO 6005 J = 1,N
      K = I + J-N-1
 6005 A(I,J) = V(K)
      DO 6010 I = 1,N
      DO 6010 J = MST1,N2
      K = I+J-N-1
 6010 A(I,J) = V(K)
C
C     ASSIGNING CORRECT SIGNS TO THE MATRIX ELEMENTS
C
      LL=N+2
      L=N+1
      J=2
 8000 DO 8010 I=L,N2,2
 8010 A(I,J)=-A(I,J)
      J=J+1
      IF(N-J)8100,8020,8020
 8020 DO 8030 I=1,N2
 8030 A(I,J)=-A(I,J)
      J=J+1
      IF(N-J)8100,8035,8035
 8035 DO 8040 I=2,N,2
 8040 A(I,J)=-A(I,J)
      DO 8050 I=LL,N2,2
 8050 A(I,J)=-A(I,J)
      J=J+2
      IF(N-J)8100,8000,8000
 8100 J=N+1
 8105 DO 8110 I=2,N,2
 8110 A(I,J)=-A(I,J)
      J=J+2
      IF(N2-J)8500,8120,8120
 8120 DO 8130 I=1,N,2
 8130 A(I,J)=-A(I,J)
      DO 8140 I=L,N2,2
 8140 A(I,J)=-A(I,J)
      J=J+1
```

```
      IF(N2-J)8500,8150,8150
 8150 DO 8160 I=1,N2
 8160 A(I,J)=-A(I,J)
      J=J+1
      IF(N2-J)8500,8105,8105
 8500 B(1) = SZERO
      DO 8501 I=2,N
 8501 B(I) = V(I-1)
      B(N+1)=0.
      NTWO = N+2
      DO 8505 I=NTWO,N2
      K=I-N-1
 8505 B(I) = U(K)
C
C     PRINT MATRIX ELEMENTS ROW BY ROW IF SENSE SWITCH 6 IS DEPRESSED
C
      IF(SENSE SWITCH 5)8510,8520
 8510 PRINT 3755,IDNO,ITER
      IF(SENSE SWITCH 6)8513,8520
 8513 PRINT 8511
 8511 FORMAT(42H    THE MATRIX ELEMENTS PRINTED ROW BY ROW/)
      DO 8512 I=1,N2
 8512 PRINT3792,(A(I,J),J=1,N2)
C
C     SCALE MATRIX ELEMENTS
C
 8520 DO 8700 I=1,N2
      DO 8700 J=1,N2
 8700 A(I,J)=A(I,J)*SCALE
      DO 8701 I=1,N2
 8701 B(I)=B(I)*SCALE
C
C     SOLUTION OF MATRIX EQUATION
C
      CALL MATINV(A,N2,B,1,DETRM)
      IF ACCUMULATOR OVERFLOW 8540,8530
 8530 IF QUOTIENT OVERFLOW 8550,8531
 8540 PRINT 8545,IDNO,ITER
 8545 FORMAT(9H1PROBLEM I6,42H HAD AN ACCUMULATOR OVERFLOW IN ITERATION
     1I3)
      GO TO 1000
 8550 PRINT 8555,IDNO,ITER
 8555 FORMAT(9H1PROBLEM I6,38H HAD A QUOTIENT OVERFLOW IN ITERATION I3)
      GO TO 1000
 8531 PZERO = B(1)
      N1=N-1
      DO 2500 I=1,N1
 2500 P(I) = B(I+1)
      N1=N+1
      DO 2505 I=N1,N2
      IMN=I-N
 2505 Q(IMN) = B(I)
C
C     CALCULATION OF THE MAGNITUDE,PHASE AND ERROR
C
      IF (XMODF(N,2))2510,2510,2509
 2509 NSTOP = (N-1)/2
      GO TO 2516
 2510 NSTOP = N/2
 2516 DO 2521 J = 1,NOMEGA
 2515 TEMP = 1.0
      DO 2520 I = 1,NSTOP
```

```
      K = 2*I
2520  TEMP = (-1.0)**I*OMEGA(J)**K*Q(K)+TEMP
2521  QABS1(J) = TEMP
2525  IF (XMODF(N,2))2530,2530,2529
2529  NSTOP = (N+1)/2
      GO TO 2535
2530  NSTOP = N/2
2535  DO 2550 J = 1,NOMEGA
2540  TEMP = 0.
      DO 2545 I = 1,NSTOP
      K = 2*I-1
2545  TEMP = -(-1.)**I*OMEGA(J)**K*Q(K)+TEMP
2550  QABS2(J) = TEMP
      DO 2560 J = 1,NOMEGA
      QABSQ(J) = QABS1(J)**2+QABS2(J)**2
      W(J) = 1.0/QABSQ(J)
2560  QABS(J) = SQRTF(QABSQ(J))
2600  IF (XMODF(N,2))2605,2605,2610
2605  NSTOP = (N-2)/2
      GO TO 2620
2610  NSTOP = (N-1)/2
2620  DO 2626 J = 1,NOMEGA
2621  TEMP = PZERO
      DO 2625 I = 1,NSTOP
      K = 2*I
2625  TEMP = (-1.)**I*OMEGA(J)**K*P(K)+TEMP
2626  PABS1(J) = TEMP
2630  IF (XMODF(N,2))2640,2640,2641
2640  NSTOP = N/2
      GO TO 2645
2641  NSTOP = (N-1)/2
2645  DO 2655 J = 1,NOMEGA
2646  TEMP = 0.
      DO 2650 I = 1,NSTOP
      K = 2*I-1
2650  TEMP = -(-1.)**I*OMEGA(J)**K*P(K)+TEMP
2655  PABS2(J) = TEMP
2660  DO 2670 J = 1,NOMEGA
      PABSQ(J) = PABS1(J)**2+PABS2(J)**2
      PABS(J) = SQRTF(PABSQ(J))
      CEM(J) = PABS(J)/QABS(J)
      CDBM(J)=8.6858896*LOGF(CEM(J))
      ERORM(J)=ABSF(DBM(J)-CDBM(J))
      ALTER1 = ABSF(CDBM(J))
      IF (ERORM(J)-CURNER*ALTER1)2661,2661,2663
2663  IF(ITER-1)2661,2661,2665
2665  DBM(J) = CDBM(J)
      PRINT 2664,J
2664  FORMAT(///66H     EXPERIMENTAL MAGNITUDE CHANGED TO CALCULATED MAGN
     1ITUDE AT J = I4/)
2661  ALPA(J) = ABSF(PABS2(J)/PABS1(J))
      IF  QUOTIENT   OVERFLOW 7300,7301
7301  THETA1(J) = ATANF(ALPA(J))
7304  BETA(J) = ABSF(QABS2(J)/QABS1(J))
      IF  QUOTIENT   OVERFLOW 7302,7303
7303  THETA2(J) = ATANF(BETA(J))
      GO TO 7305
7300  THETA1(J)=1.570796425
      GO TO 7304
7302  THETA2(J)=1.570796425
7305  IF(QABS2(J))2700,2701,2701
2700  THETA2(J) = -THETA2(J)
```

```
2701 IF (QABS1(J))2710,2711,2711
2710 THETA2(J) = 3.1415927-THETA2(J)
2711 IF (PABS2(J))2720,2721,2721
2720 THETA1(J) = -THETA1(J)
2721 IF (PABS1(J))2730,2731,2731
2730 THETA1(J) = 3.1415927-THETA1(J)
2731 CPHR(J) = THETA1(J)-THETA2(J)
     CPH(J) = CPHR(J)*57.2957795
     ERORP(J)=ABSF(CPH(J)-PHE(J))
     ALTER2 = ABSF(CPH(J))
     IF (ERORP(J)-WALT*ALTER2)2670,2670,2671
2671 IF(ITER-1)2670,2670,2673
2673 PHE(J) = CPH(J)
     PRINT 2672,J
2672 FORMAT(///58H    EXPERIMENTAL PHASE CHANGED TO CALCULATED PHASE AT
    1 J = I4/)
2670 CONTINUE
     ERMAG2= ERORM(1)
     MAXPT2 = 1
     DO 3700 I = 2,NOMEGA
     IF(ERMAG2-ERORM(I))3710,3700,3700
3710 ERMAG2= ERORM(I)
     MAXPT2 = I
3700 CONTINUE
     ERFAZ2 = ERORP(1)
     MAXPT4 = 1
     DO 3720 I = 2,NOMEGA
     IF (ERFAZ2-ERORP(I))3715,3720,3720
3715 ERFAZ2 = ERORP(I)
     MAXPT4 = I
3720 CONTINUE
C
C    PRINT RESULTS OF CURRENT ITERATION IF SENSE SWITCH 5 IS DEPRESSED
C
     NLESS1=N-1
     IF (SENSE SWITCH 5)3750,9000
3755 FORMAT(/////111H11520/RE277   COMPLEX CURVE FITTING ROUTINE
    1                                              PROBLEM NUMBER I4//1
    21H ITERATION I4/)
3750 PRINT 3760,ERMAG2,MAXPT2,ERFAZ2,MAXPT4
3760 FORMAT(/18H          ERMAG = F10.5,17H  OCCURED AT J = I4,
    114H     ERPASE = F10.5,17H  OCCURED AT J = I4)
     PRINT 3765
3765 FORMAT(/50H     NUMERATOR COEFFICIENTS P(0),P(1),P(2),ETC. ARE)
     PRINT 3766,PZERO,(P(I),I=1,NLESS1)
3766 FORMAT(/1PE20.5,1P6E15.5)
     PRINT 3770
3770 FORMAT(/52H     DENOMINATOR COEFFICIENTS Q(0),Q(1),Q(2),ETC. ARE)
     PRINT 3766,QZERO,(Q(I),I = 1,N)
     PRINT 3775
3775 FORMAT(/88H   J       OMEGA             CDBM            CPH
    1ERORM          ERORP          W/)
     PRINT 3780,(J,OMEGA(J),CDBM(J),CPH(J),ERORM(J),ERORP(J),W(J),J = 1
    1,NOMEGA)
3780 FORMAT(/I4,1P6E15.5)
     IF (SENSE SWITCH 6) 3781,9000
3781 PRINT 3782
3782 FORMAT(/28H     LAMDAS ARE THE FOLLOWING)
     PRINT 3766,(AMDA(J),J=1,NOLAMD)
     PRINT 3784
3784 FORMAT(/27H     S(0),S(1),S(2),ETC. ARE)
     PRINT 3766,SZERO,(S(J),J=1,NLESS1)
```

```
      PRINT 3786
 3786 FORMAT(/27H       T(1),T(2),T(3),ETC. ARE)
      PRINT 3766,(T(J),J=1,N)
      PRINT 3788
 3788 FORMAT(/27H       U(1),U(2),U(3),ETC. ARE)
      PRINT 3766,(U(J),J=1,N2)
 3792 FORMAT(/1P8E15.5/)
C
C     TEST FOR CONVERGENCE
C
 9000 IF(MAXPT1-MAXPT2)9100,9050,9100
 9050 TEST=ABSF(ERMAG1-ERMAG2)
      IF(TEST-CONV1)9051,9051,9100
 9051 IF(MAXPT3-MAXPT4)9100,9052,9100
 9052 TEST=ABSF(ERFAZ1-ERFAZ2)
      IF(TEST-CONV2)9010,9010,9100
 9100 MAXPT1=MAXPT2
      MAXPT3=MAXPT4
      ERMAG1=ERMAG2
      ERFAZ1=ERFAZ2
      ITER=ITER+1
      GO TO 7
C
C     WRITE OUTPUT TAPE 6
C
 9010 WRITE OUTPUT TAPE 6,3755,IDNO,ITER
      WRITE OUTPUT TAPE 6,3760,ERMAG2,MAXPT2,ERFAZ2,MAXPT4
      WRITE OUTPUT TAPE 6,3765
      WRITE OUTPUT TAPE 6,3766,PZERO,(P(I),I=1,NLESS1)
      WRITE OUTPUT TAPE 6,3770
      WRITE OUTPUT TAPE 6,3766,QZERO,(Q(I),I = 1,N)
      WRITE OUTPUT TAPE 6,3775
      WRITE OUTPUT TAPE 6,3780,(J,OMEGA(J),CDBM(J),CPH(J),ERORM(J),ERORP
     1(J),W(J),J=1,NOMEGA)
 9999 GO TO 1000
      END(0,1,0,0,0)
```

B. RE 277B

```
      RE277B TRAFICORPORATION
C     TRANSFER FUNCTION EXPRESSED AS A RATIO OF TWO COMPLEX POLYNOMIALS.
      WHERE THE NUMERATOR AND THE DENOMINATOR ARE OF EQUAL DEGREE
C
      DIMENSION OMEGA(250),PH(250),DBM(250),R(250),UR(250),W(250),AMDA(5
     10),V(50),S(25),T(25),U(50),A(25,25),B(25),EM(250),P(15),Q(15),QABS
     21(250),QABS2(250),QABSQ(250),QABS(250),PABS1(250),PABS2(250),PABSQ
     3(250),PABS(250),CEM(250),CDBM(250),ERORM(250),CPH(250),THETA1(250)
     4,THETA2(250),ERORP(250),ALPA(250),BETA(250),PHE(250),CPHR(250)
C
C     READ INPUT
C
 1005 FORMAT(6E12.5)
 1006 FORMAT(3I6,4E12.5)
 1000 READ 1006,IDNO,NOMEGA,N,CURNER,WALT
      DO 1010 J = 1,NOMEGA
 1010 READ 1005,OMEGA(J),DBM(J),PHE(J)
 1500 READ 1005,WX,SCALE,CONV1,CONV2
      ERMAG1=0.
      ERFAZ1=0.
      MAXPT1=0
      MAXPT3=0
      ITER=1
      DO 1510 J = 1,NOMEGA
 1510 W(J) = WX
C
C     CONVERSION OF EXPERIMENTAL G(J) INTO THE REAL AND IMAGINARY PARTS
C
      DO 500 J = 1,NOMEGA
      PH(J) = PHE(J)*0.0174533
      EM(J) = EXPF(DBM(J)/8.6858896)
      R(J) = EM(J)*COSF(PH(J))
  500 UR(J) = EM(J)*SINF(PH(J))
      N2 = N*2
      NOLAMD = N2+1
      NOUS = N2-1
      NPONE = N+1
      QZERO = 1.
C
C     COMPUTATION OF LAMBDA,U,T AND S
C
    7 DO 15 J = 1,NOLAMD,2
    5 TEMP = 0.
      DO 10 K = 1,NOMEGA
      J1 = J-1
   10 TEMP = OMEGA(K)**J1*W(K)+TEMP
      AMDA(J) = TEMP
   15 AMDA(J+1) = 0.
   20 DO 35 J = 1,N2,2
   25 TEMP = 0.
      DO 30 K = 1,NOMEGA
      J1 = J+1
   30 TEMP = OMEGA(K)**(J1)*(R(K)**2+UR(K)**2)*W(K)+TEMP
      U(J) = TEMP
   35 U(J+1) = 0.
   40 DO 55 J = 1,N
   45 TEMP = 0.
      DO 50 K = 1,NOMEGA
      J2 = J*2-1
   50 TEMP = OMEGA(K)**(J2)*UR(K)*W(K)+TEMP
   55 T(J) = TEMP
   60 DO 75 J = 1,N
```

```
   65 TEMP = 0.
      DO 70 K = 1,NOMEGA
      J2 = J*2
   70 TEMP = OMEGA(K)**(J2)*R(K)*W(K)+TEMP
   75 S(J) = TEMP
   80 TEMP = 0.
      DO 81 K = 1,NOMEGA
   81 TEMP = R(K)*W(K)+TEMP
      SZERO = TEMP
C
C     SUBSTITUTION OF PROPER MAGNITUDES IN THE MATRIX
C
      DO 2000 J = 1,NPONE
      DO 2000 I = 1,NPONE
      K = I+J-1
 2000 A(I,J) = AMDA(K)
      MST1 = N+2
      DO 2020 J = MST1,NOLAMD
      DO 2020 I = MST1,NOLAMD
      K = I+J-3-N2
 2020 A(I,J) = U(K)
      DO 6000 I = 1,N
      IV = 2*I
      V(IV-1) = T(I)
 6000 V(IV) = S(I)
      DO 6005 I = MST1,NOLAMD
      DO 6005 J = 1,NPONE
      K = I+J-N-2
 6005 A(I,J) = V(K)
      DO 6010 I = 1,NPONE
      DO 6010 J = MST1,NOLAMD
      K = I+J-N-2
 6010 A(I,J) = V(K)
C
C     ASSIGNING CORRECT SIGNS TO THE MATRIX ELEMENTS
C
      J=2
 8000 DO 8010 I=MST1,NOLAMD,2
 8010 A(I,J)=-A(I,J)
      J=J+1
      IF(NPONE-J)8100,8020,8020
 8020 DO 8030 I=1,NOLAMD
 8030 A(I,J)=-A(I,J)
      J=J+1
      IF(NPONE-J)8100,8035,8035
 8035 DO 8040 I=2,NPONE,2
 8040 A(I,J)=-A(I,J)
      DO 8050 I=NTHREE,NOLAMD,2
 8050 A(I,J)=-A(I,J)
      J=J+2
      IF(NPONE-J)8100,8000,8000
 8100 J=MST1
 8105 DO 8110 I=2,NPONE,2
 8110 A(I,J)=-A(I,J)
      J=J+2
      IF(NOLAMD-J)8500,8120,8120
 8120 DO 8130 I=1,NPONE,2
 8130 A(I,J)=-A(I,J)
      DO 8140 I=MST1,NOLAMD,2
 8140 A(I,J)=-A(I,J)
      J=J+1
      IF(NOLAMD-J)8500,8150,8150
```

```
 8150 DO 8160 I=1,NOLAMD
 8160 A(I,J)=-A(I,J)
      J=J+1
      IF(NOLAMD-J)8500,8105,8105
 8500 B(1) = SZERO
      DO 8501 I = 2,NPONE
 8501 B(I) = V(I-1)
      B(N+2) = 0.
      NTHREE = N+3
      DO 8505 I = NTHREE,NOLAMD
      K = I-N-2
 8505 B(I) = U(K)
C
C     PRINT MATRIX ELEMENTS ROW BY ROW IF SENSE SWITCH 6 IS DEPRESSED
C
      IF(SENSE SWITCH 5)8510,8520
 8510 PRINT 3755,IDNO,ITER
      IF(SENSE SWITCH 6)8513,8520
 8513 PRINT 8511
 8511 FORMAT(42H    THE MATRIX ELEMENTS PRINTED ROW BY ROW/)
      DO 8512 I=1,NOLAMD
 8512 PRINT3792,(A(I,J),J=1,NOLAMD)
C
C     SCALE MATRIX ELEMENTS
C
 8520 DO 8700 I=1,NOLAMD
      DO 8700 J=1,NOLAMD
 8700 A(I,J)=A(I,J)*SCALE
      DO 8701 I=1,NOLAMD
 8701 B(I)=B(I)*SCALE
C
C     SOLUTION OF MATRIX EQUATION
C
      CALL MATINV(A,NOLAMD,B,1,DETRM)
      IF ACCUMULATOR OVERFLOW 8540,8530
 8530 IF QUOTIENT OVERFLOW 8550,8531
 8540 PRINT 8545,IDNO,ITER
 8545 FORMAT(9H1PROBLEM I6,42H HAD AN ACCUMULATOR OVERFLOW IN ITERATION
     1I3)
      GO TO 1000
 8550 PRINT 8555,IDNO,ITER
 8555 FORMAT(9H1PROBLEM I6,38H HAD A QUOTIENT OVERFLOW IN ITERATION I3)
      GO TO 1000
 8531 PZERO = B(1)
      DO 2500 I = 1,N
 2500 P(I) = B(I+1)
      DO 2505 I=MST1,NOLAMD
      IMN=I-N-1
 2505 Q(IMN) = B(I)
C
C     CALCULATION OF THE MAGNITUDE,PHASE AND ERROR
C
      IF (XMODF(N,2))2510,2510,2509
 2509 NSTOP = (N-1)/2
      GO TO 2516
 2510 NSTOP = N/2
 2516 DO 2521 J = 1,NOMEGA
 2515 TEMP = 1.0
      DO 2520 I = 1,NSTOP
      K = 2*I
 2520 TEMP = (-1.0)**I*OMEGA(J)**K*Q(K)+TEMP
 2521 QABS1(J) = TEMP
```

```
2525 IF (XMODF(N,2))2530,2530,2529
2529 NSTOP = (N+1)/2
     GO TO 2535
2530 NSTOP = N/2
2535 DO 2550 J = 1,NOMEGA
2540 TEMP = 0.
     DO 2545 I = 1,NSTOP
     K = 2*I-1
2545 TEMP = -(-1.)**I*OMEGA(J)**K*Q(K)+TEMP
2550 QABS2(J) = TEMP
     DO 2560 J = 1,NOMEGA
     QABSQ(J) = QABS1(J)**2+QABS2(J)**2
     W(J) = 1.0/QABSQ(J)
2560 QABS(J) = SQRTF(QABSQ(J))
2600 IF (XMODF(N,2))2605,2605,2610
2605 NSTOP = N/2
     GO TO 2620
2610 NSTOP = (N-1)/2
2620 DO 2626 J = 1,NOMEGA
2621 TEMP = PZERO
     DO 2625 I = 1,NSTOP
     K = 2*I
2625 TEMP = (-1.)**I*OMEGA(J)**K*P(K)+TEMP
2626 PABS1(J) = TEMP
2630 IF (XMODF(N,2))2640,2640,2641
2640 NSTOP = N/2
     GO TO 2645
2641 NSTOP = (N+1)/2
2645 DO 2655 J = 1,NOMEGA
2646 TEMP = 0.
     DO 2650 I = 1,NSTOP
     K = 2*I-1
2650 TEMP = -(-1.)**I*OMEGA(J)**K*P(K)+TEMP
2655 PABS2(J) = TEMP
2660 DO 2670 J = 1,NOMEGA
     PABSQ(J) = PABS1(J)**2+PABS2(J)**2
     PABS(J) = SQRTF(PABSQ(J))
     CEM(J) = PABS(J)/QABS(J)
     CDBM(J) = 20.*LOGF(CEM(J))*0.4342946R
     ERORM(J) = ABSF(DBM(J)-CDBM(J))
     ALTER1 = ABSF(CDBM(J))
     IF (ERORM(J)-CURNER*ALTER1)2661,2661,2663
2663 IF(ITER-1)2661,2661,2665
2665 DBM(J) = CDBM(J)
     PRINT 2664,J
2664 FORMAT(///66H     EXPERIMENTAL MAGNITUDE CHANGED TO CALCULATED MAGN
    1ITUDE AT J = I4/)
2661 ALPA(J) = ABSF(PABS2(J)/PABS1(J))
     IF QUOTIENT OVERFLOW 7300,7301
7301 THETA1(J) = ATANF(ALPA(J))
7304 BETA(J) = ABSF(QABS2(J)/QABS1(J))
     IF QUOTIENT OVERFLOW 7302,7303
7303 THETA2(J) = ATANF(BETA(J))
     GO TO 7305
7300 THETA1(J)=1.570796425
     GO TO 7304
7302 THETA2(J)=1.570796425
7305 IF (QABS2(J))2700,2701,2701
2700 THETA2(J) = -THETA2(J)
2701 IF (QABS1(J))2710,2711,2711
2710 THETA2(J) = 3.1415927-THETA2(J)
2711 IF (PABS2(J))2720,2721,2721
```

```
 2720 THETA1(J) = -THETA1(J)
 2721 IF (PABS1(J))2730,2731,2731
 2730 THETA1(J) = 3.1415927-THETA1(J)
 2731 CPHR(J) = THETA1(J)-THETA2(J)
      CPH(J) = CPHR(J)*57.2957795
      ERORP(J) = ABSF(CPH(J)-PHE(J))
      ALTER2 = ABSF(CPH(J))
      IF (ERORP(J)-WALT*ALTER2)2670,2670,2671
 2671 IF(ITER-1)2670,2670,2673
 2673 PHE(J) = CPH(J)
      PRINT 2672,J
 2672 FORMAT(///58H    EXPERIMENTAL PHASE CHANGED TO CALCULATED PHASE AT
     1 J = I4/)
 2670 CONTINUE
      ERMAG2=ERORM(1)
      MAXPT2=1
      DO 3700 I = 2,NOMEGA
      IF(ERMAG2-ERORM(I))3710,3700,3700
 3710 ERMAG2=ERORM(I)
      MAXPT2=I
 3700 CONTINUE
      ERFAZ2=ERORP(1)
      MAXPT4=1
      DO 3720 I = 2,NOMEGA
      IF(ERFAZ2-ERORP(I))3715,3720,3720
 3715 ERFAZ2=ERORP(I)
      MAXPT4=I
 3720 CONTINUE
C
C     PRINT RESULTS OF CURRENT ITERATION IF SENSE SWITCH 5 IS DEPRESSED
C
      IF (SENSE SWITCH 5)3750,9000
 3755 FORMAT(/////111H11520/RE277   COMPLEX CURVE FITTING ROUTINE
     1                                                 PROBLEM NUMBER I4//1
     21H ITERATION I4/)
 3750 PRINT 3760,ERMAG2,MAXPT2,ERFAZ2,MAXPT4
 3760 FORMAT(/18H          ERMAG = F10.5,17H  OCCURED AT J = I4,
     114H     ERPASE = F10.5,17H  OCCURED AT J = I4)
      PRINT 3765
 3765 FORMAT(/50H    NUMERATOR COEFFICIENTS P(0),P(1),P(2),ETC. ARE)
      PRINT 3766,PZERO,(P(I),I = 1,N)
 3766 FORMAT(/1PE20.5,1P6E15.5)
      PRINT 3770
 3770 FORMAT(/52H     DENOMINATOR COEFFICIENTS Q(0),Q(1),Q(2),ETC. ARE)
      PRINT 3766,QZERO,(Q(I),I = 1,N)
      PRINT 3775
 3775 FORMAT(/88H   J         OMEGA            CDBM            CPH
     1ERORM         ERORP                  W/)
      PRINT 3780,(J,OMEGA(J),CDBM(J),CPH(J),ERORM(J),ERORP(J),W(J),J = 1
     1,NOMEGA)
 3780 FORMAT(/I4,1P6E15.5)
      IF(SENSE SWITCH 6)3781,9000
 3781 PRINT 3782
 3782 FORMAT(/28H     LAMDAS ARE THE FOLLOWING)
      PRINT 3766,(AMDA(J),J=1,NOLAMD)
      PRINT 3784
 3784 FORMAT(/27H    S(0),S(1),S(2),ETC. ARE)
      PRINT 3766,SZERO,(S(J),J=1,N)
      PRINT 3786
 3786 FORMAT(/27H     T(1),T(2),T(3),ETC. ARE)
      PRINT 3766,(T(J),J=1,N)
      PRINT 3788
```

```
 3788 FORMAT(/27H    U(1),U(2),U(3),ETC. ARE)
      PRINT 3766,(U(J),J=1,NOUS)
 3792 FORMAT(/1P8E15.5/)
C
C     TEST FOR CONVERGENCE
C
 9000 IF(MAXPT1-MAXPT2)9100,9050,9100
 9050 TEST=ABSF(ERMAG1-ERMAG2)
      IF(TEST-CONV1)9051,9051,9100
 9051 IF(MAXPT3-MAXPT4)9100,9052,9100
 9052 TEST=ABSF(ERFAZ1-ERFAZ2)
      IF(TEST-CONV2)9010,9010,9100
 9100 MAXPT1=MAXPT2
      MAXPT3=MAXPT4
      ERMAG1=ERMAG2
      ERFAZ1=ERFAZ2
      ITER=ITER+1
      GO TO 7
C
C     WRITE OUTPUT TAPE 6
C
 9010 WRITE OUTPUT TAPE 6,3755,IDNO,ITER
      WRITE OUTPUT TAPE 6,3760,ERMAG2,MAXPT2,ERFAZ2,MAXPT4
      WRITE OUTPUT TAPE 6,3765
      WRITE OUTPUT TAPE 6,3766,PZERO,(P(I),I =1,N)
      WRITE OUTPUT TAPE 6,3770
      WRITE OUTPUT TAPE 6,3766,QZERO,(Q(I),I = 1,N)
      WRITE OUTPUT TAPE 6,3775
      WRITE OUTPUT TAPE 6,3780,(J,OMEGA(J),CDBM(J),CPH(J),ERORM(J),ERORP
     1(J),W(J),J=1,NOMEGA)
 9999 GO TO 100C
      END(0,1,0,0,0)
```

C. MATINV

```
C     MATRIX INVERSION WITH ACCOMPANYING SOLUTION OF LINEAR EQUATIONS
C
      SUBROUTINE MATINV(A,N,B,M,DETERM)
C
      DIMENSION IPIVOT(25), A(25,25), B(25,1), INDEX(25,2), PIVOT(25)
      COMMON PIVOT, INDEX, IPIVOT
      EQUIVALENCE (IROW,JROW), (ICOLUM,JCOLUM), (AMAX, T, SWAP)
C
C     INITIALIZATION
C
   10 DETERM=1.0
   15 DO 20 J=1,N
   20 IPIVOT(J)=0
   30 DO 550 I=1,N
C
C     SEARCH FOR PIVOT ELEMENT
C
   40 AMAX=0.0
   45 DO 105 J=1,N
   50 IF (IPIVOT(J)-1) 60, 105, 60
   60 DO 100 K=1,N
   70 IF (IPIVOT(K)-1) 80, 100, 740
   80 IF (ABSF(AMAX)-ABSF(A(J,K))) 85, 100, 100
   85 IROW=J
   90 ICOLUM=K
   95 AMAX=A(J,K)
  100 CONTINUE
  105 CONTINUE
  110 IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
C
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
  130 IF (IROW-ICOLUM) 140, 260, 140
  140 DETERM=-DETERM
  150 DO 200 L=1,N
  160 SWAP=A(IROW,L)
  170 A(IROW,L)=A(ICOLUM,L)
  200 A(ICOLUM,L)=SWAP
  205 IF(M) 260, 260, 210
  210 DO 250 L=1, M
  220 SWAP=B(IROW,L)
  230 B(IROW,L)=B(ICOLUM,L)
  250 B(ICOLUM,L)=SWAP
  260 INDEX(I,1)=IROW
  270 INDEX(I,2)=ICOLUM
  310 PIVOT(I)=A(ICOLUM,ICOLUM)
  320 DETERM=DETERM*PIVOT(I)
C
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
  330 A(ICOLUM,ICOLUM)=1.0
  340 DO 350 L=1,N
  350 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT(I)
  355 IF(M) 380, 380, 360
  360 DO 370 L=1,M
  370 B(ICOLUM,L)=B(ICOLUM,L)/PIVOT(I)
C
C     REDUCE NON-PIVOT ROWS
C
  380 DO 550 L1=1,N
  390 IF(L1-ICOLUM) 400, 550, 400
  400 T=A(L1,ICOLUM)
```

```
  420 A(L1,ICOLUM)=0.0                                                F4020063
  430 DO 450 L=1,N                                                    F4020064
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T                                   F4020065
  455 IF(M) 550, 550, 460                                             F4020066
  460 DO 500 L=1,M                                                    F4020067
  500 B(L1,L)=B(L1,L)-B(ICOLUM,L)*T                                   F4020068
  550 CONTINUE                                                        F4020069
C                                                                     F4020070
C     INTERCHANGE COLUMNS                                             F4020071
C                                                                     F4020072
  600 DO 710 I=1,N                                                    F4020073
  610 L=N+1-I                                                         F4020074
  620 IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630                        F4020075
  630 JROW=INDEX(L,1)                                                 F4020076
  640 JCOLUM=INDEX(L,2)                                               F4020077
  650 DO 705 K=1,N                                                    F4020078
  660 SWAP=A(K,JROW)                                                  F4020079
  670 A(K,JROW)=A(K,JCOLUM)                                           F4020080
  700 A(K,JCOLUM)=SWAP                                                F4020081
  705 CONTINUE                                                        F4020082
  710 CONTINUE                                                        F4020083
  740 RETURN                                                          F4020084
  750 END (2,2,2,2,0)                                                 F4020085
```

VII. SAMPLE PROBLEM

SAMPLE PROBLEM INPUT DATA

```
    5    24     7 1.0     +06 1.0     +06
4.2     -02 3.5         8.2
5.6     -02 2.4         3.7
7.5     -02 3.3         2.9
1.0     -01 2.0        -6.2
1.3     -01 2.8        -5.1
1.8     -01 1.4       -2.64     +01
2.4     -01 3.0    -01-2.57     +01
3.2     -01-2.3       -1.74     +01
4.2     -01-2.1       -1.77     +01
5.6     -01-3.1       -2.16     +01
7.5     -01-6.2        -9.7
1.0        -5.6        +9.2
1.3        -4.5        1.34     +01
1.8        -4.8        2.63     +01
2.4        -3.4        3.63     +01
3.2        -1.3        6.83     +01
4.2         1.2        5.28     +01
5.6         4.4        5.22     +01
7.6         8.0        4.18     +01
1.0     +01 9.8        1.26     +01
1.32    +01 7.7        -6.1
1.84    +01 6.8       -1.39     +01
2.36    +01 6.2       -1.41     +01
3.19    +01 5.6       -1.76     +01
1.0        1.0      -09 1.0     -03 1.0     -03
```

ITERATION 1

EPMAG = 6.44429 OCCURED AT J = 16 ERPASE = 360.06562 OCCURED AT J = 20

NUMERATOR COEFFICIENTS P(0),P(1),P(2),ETC. ARE

9.28940E-01 1.71[illegible]E-01 1.12538E-01 1.03626E-02 1.60444E-03 8.37465E-06 4.71453E-06

DENOMINATOR COEFFICIENTS Q(0),Q(1),Q(2),ETC. ARE

1.00000E 00 2.11077E-01 8.75486E-02 5.62947E-03 1.00421E-03 1.16778E-05 2.52117E-06
2.29472E-08

J	OMEGA	CDBM	CPH	ERORM	ERORP	[illegible]
1	4.20000E-02	-6.4[illegible]846E-01	-6.45634E-02	4.14085E 00	5.26436E 00	1.0[illegible]23E 00
2	5.60000E-02	-6.41309E-01	-8.58218E-02	3.04131E 00	3.78582E 00	1.00041E 00
3	7.50000E-02	-6.42151E-01	-1.14946E-01	3.94215E 00	3.01495E 00	1.00[illegible]72E 00
4	10.00000E-02	-6.43631E-01	-1.53277E-01	2.64363E 00	6.04672E 00	1.00131E 00
5	1.30000E-01	-6.45967E-01	-1.99225E-01	3.44537E 00	4.90071E 00	1.00221E 00
6	1.80000E-01	-6.51225E-01	-2.76042E-01	2.05123E 00	2.61240E 01	1.00424E 00
7	2.40000E-01	-6.59603E-01	-3.68263E-01	9.59803E-01	2.53317E 01	1.00755E 00
8	3.20000E-01	-6.75141E-01	-4.91495E-01	1.62486E 00	1.69085E 01	1.01347E 00
9	4.20000E-01	-7.00729E-01	-6.46059E-01	1.39927E 00	1.70539E 01	1.02333E 00
10	5.60000E-01	-7.49018E-01	-8.63538E-01	2.35098E 00	2.07365E 01	1.04190E 00
11	7.50000E-01	-8.39528E-01	-1.16035E 00	5.36047E 00	8.53965E 00	1.07659E 00
12	1.00000E 00	-1.00826E 00	-1.54967E 00	4.59174E 00	1.07497E 01	1.14073E 00
13	1.30000E 00	-1.30197E 00	-1.99270E 00	3.19803E 00	1.53927E 01	1.2[illegible]68E 00
14	1.80000E 00	-2.11175E 00	-2.43025E 00	2.68825E 00	2.87302E 01	1.54102E 00
15	2.40000E 00	-4.04199E 00	-2.39834E-01	6.41991E-01	3.65398E 01	2.15490E 00
16	3.20000E 00	-7.74429E 00	3.85394E 01	6.44429E 00	2.97607E 01	2.47992E 00
17	[illegible].[illegible]0000E 00	[illegible]E-0[illegible]	[illegible]E 01	[illegible]0E-0[illegible]	1.[illegible]386E 01	2.[illegible]E 02

18	5.60000E 00	4.23933E 00	5.17112E 01	1.60665E-01	4.88809E-01	1.31191E 00
19	7.60[illegible]00E 0[illegible]	7.65440E 00	4.2655[illegible]E 01	3.45597E-01	8.54994E-01	5.59958E-01
20	1.00[illegible]00E 01	9.60360E 00	-3.47466E 02	1.96395E-01	3.60066E 02	1.48971E-01
21	1.32[illegible]E 01	7.76976E 00	-6.47094E 00	6.97637E-02	3.70937E-01	1.70708E-02
22	1.84[illegible]0E 01	6.79505E 00	-1.39535E 01	4.95142E-03	5.35231E-02	1.52538E-03
23	2.36000E 01	6.19953E 00	-1.40971E 01	4.65751E-04	2.91491E-03	2.81940E-05
24	3.19000E 01	5.60001E 00	-1.76000E 01	6.97374E-06	2.78950E-05	3.10352E-07

1520/RE277 COMPLEX CURVE FITTING ROUTINE PROBLEM NUMBER 5

ITERATION 10

ERMAG = 2.38248 OCCURED AT J = 8 ERPASE = 18.77125 OCCURED AT J = 12

NUMERATOR COEFFICIENTS P(0),P(1),P(2),ETC. ARE

1.27675E 00 1.28033E 00 7.82329E-01 7.91805E-02 8.08841E-03 2.89473E-04 2.01356E-05

DENOMINATOR COEFFICIENTS Q(0),Q(1),Q(2),ETC. ARE

1.00000E 00 2.53138E 00 4.26968E-01 5.46436E-02 4.53640E-03 1.98396E-04 1.14464E-05

1.05189E-07

J	OMEGA	CDBM	CPH	ERORM	ERORP	W
1	4.20000E-02	2.07810E 00	-3.65899E 00	1.42190E 00	1.18590E 01	9.90298E-01
2	5.60000E-02	2.04415E 00	-4.85873E 00	3.55849E-01	8.55873E 00	9.82882E-01
3	7.50000E-02	1.98319E 00	-6.45939E 00	1.31681E 00	9.35939E 00	9.69708E-01
4	10.00000E-02	1.87797E 00	-8.50277E 00	1.22032E-01	2.30277E 00	9.47390E-01
5	1.30000E-01	1.71676E 00	-1.08361E 01	1.08324E 00	5.73608E 00	9.14213E-01
6	1.80000E-01	1.37398E 00	-1.43692E 01	2.60163E-02	1.20308E 01	8.47559E-01
7	2.40000E-01	8.66108E-01	-1.79244E 01	5.66108E-01	7.77558E 00	7.57791E-01
8	3.20000E-01	8.24834E-02	-2.14155E 01	2.38248E 00	4.01553E 00	6.37824E-01
9	4.20000E-01	-9.73049E-01	-2.38418E 01	1.12695E 00	6.14179E 00	5.05801E-01
10	5.60000E-01	-2.43625E 00	-2.41286E 01	6.63753E-01	2.52863E 00	3.65853E-01
11	7.50000E-01	-4.18381E 00	-2.00665E 01	2.01619E 00	1.03665E 01	2.44073E-01
12	1.00000E 00	-5.81407E 00	-9.57125E 00	2.14068E-01	1.87713E 01	1.54589E-01
13	1.30000E 00	-6.58800E 00	6.10798E 00	2.08800E 00	7.29202E 00	9.85887E-02
14	1.80000E 00	-5.77078E 00	2.78866E 01	9.70779E-01	1.58664E 00	5.52373E-02
15	2.40000E 00	-3.72762E 00	4.21772E 01	3.27623E-01	5.87723E 00	3.31256E-02
16	3.20000E 00	-1.17640E 00	5.03271E 01	1.23599E-01	1.79729E 01	2.03603E-02
17	4.20000E 00	1.42988E 00	5.34745E 01	2.29882E-01	6.74522E-01	1.35774E-02

18	5.[illegible]000 E 00	4.44769E 00	5.22161E 01	4.76949E-02	1.60542E-02	9.94545E-03
19	7.[illegible]000 E 00	8.11098E 00	4.10505E 01	1.10976E-01	7.49498E-01	8.67957E-03
20	1.0000 E 01	9.86425E 00	1.30109E 01	6.42477E-02	4.10889E-01	5.82577E-03
21	1.3200 E 01	7.65339E 00	-5.98689E 00	4.66133E-02	1.13110E-01	2.39006E-03
22	1.8400 E 01	6.80158E 00	-1.39296E 01	1.58048E-03	2.95832E-02	1.41981E-04
23	2.3600 E 01	6.10519E 00	-1.39577E 01	9.48080E-02	1.42254E-01	1.27420E-06
24	3.1900 E 01	5.75582E 00	-1.76388E 01	1.55817E-01	3.87533E-02	1.59849E-08

ACKNOWLEDGMENT

The authors wish to thank Mr. W. C. Lipinski and Dr. H. Thacher for their very helpful discussions and suggestions.

BIBLIOGRAPHY

1. G. L. Rabkin, B. A. Mitrofanov, and Yu O. Shterenberg, On the Determination of the Numerical Values of the Coefficients of the Transfer Function of Linearized Links and Systems from Experimental Frequency Characteristics, Automation and Remote Control, 16, No. 5 (1955).

2. E. C. Levy, Complex Curve Fitting, IRE Transactions on Automatic Control, AC-4, No. 1 (May 1959).

3. A. A. Kardashov and L. V. Karnyushin, Determining System Parameters from Experimental Frequency Characteristics, Automation and Remote Control, 19, No. 4, 327 (1958).

4. V. M. Rushchinskin, "Determining an Approximate Expression for a Transfer Function of a Control Object from Its Experimental Frequency Characteristics," (in Russian) from a collection Production Automation, Acad. Sci., U.S.S.R. Press, Ed. 2 (1958).

5. G. I. Monastyrshin, Processing Experimental Frequency Characteristics, Automation and Remote Control, 21, No. 3, 294-300 (1960).

6. R. C. Nicholson and A. K. Orick, A Least Squares Approach to Rational Functions, IBM-704 SHARE Distribution No. 859 (March 9, 1960).