APPLICATION OF NASAP TO THE DESIGN OF COMMUNICATION CIRCUITS AND EXTENSION OF NASAP ROUTINES TO LARGE SCALE CIRCUITS

by

G. J. Herskowitz
M. Sankaran

Prepared for

NASA Electronics Research Center

Under

NASA Research Grant NGR 31-003-050

Department of Electrical Engineering
Stevens Institute of Technology
Hoboken, New Jersey

June 1969

TABLE OF CONTENTS

I. OBJECTIVES

Computer-aided circuit design procedures, in general, are iterative in nature[1-5]. Typically, the circuit designer begins with a set of specifications. He then selects a network configuration and makes an initial choice about the element values based on a combination of established synthesis procedures and past experience. The circuit would then be coded for an analysis[6-8] (using standard analysis programs such as NASAP, ECAP, HYBRID, etc.) and the desired response evaluated. Should the analysis show that the overall response was not satisfactory, the designer would then change either the circuit topology, the components he was using, or the element values of some of the circuit components. This modified circuit would be resubmitted for analysis to see if the circuit's response was improved. He does this, a number of times, until the circuit's response is within a present tolerance limit. This procedure is illustrated in the block diagram shown in Fig. 1.

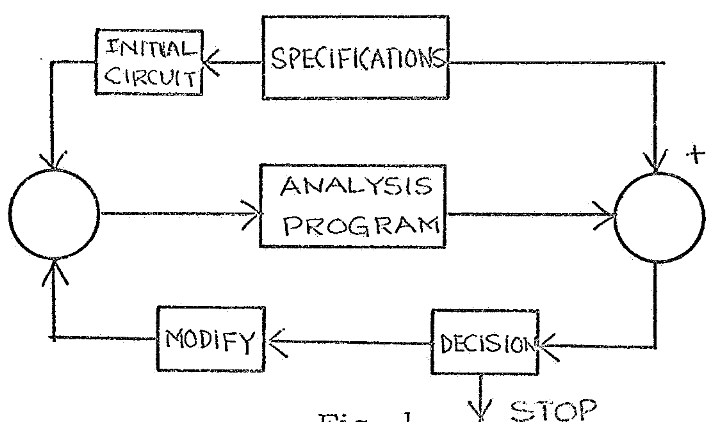

Fig. 1

The major objective of the research conducted at Stevens under NASA Grant No. NGR 31-003-050, was to develop a direct design procedure to arrive at a good initial estimate of the circuit element values, and to incorporate NASAP into a direct design oriented algorithm, similar to the one shown in Fig. 1, and automate and speed up the convergence of the process by using the method of steepest descent.[14] The resulting modified scheme is shown in Fig. 2.

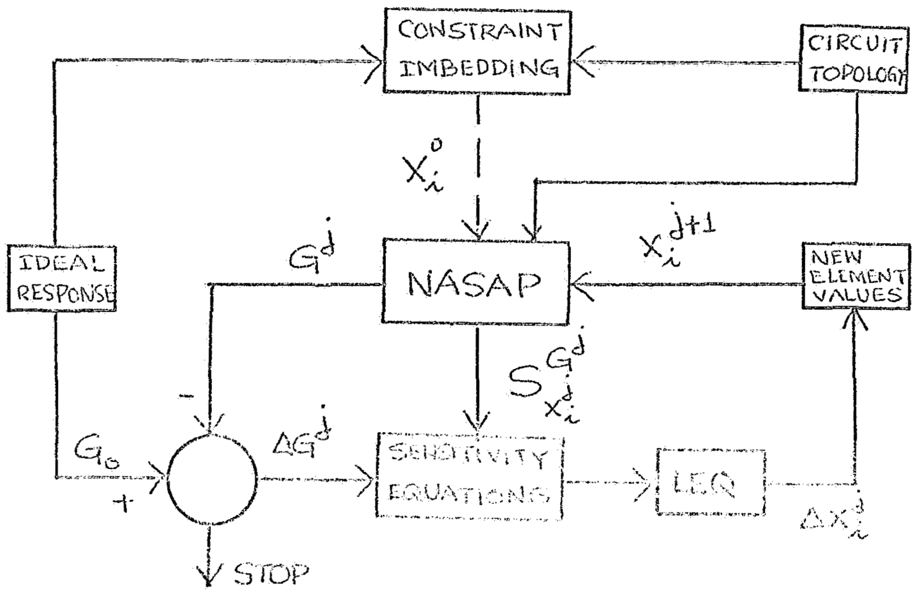

Fig. 2

The required inputs to the program are the circuit topology, desired circuit frequency response and a preselected acceptable error bound between the ideal and achievable frequency response. The outputs of the program are the final values of the circuit elements, satisfying the ideal frequency response within the error bound.

II. APPROACH

The approach taken basically consists of two distinct steps:[9-13]

(i) to determine an initial estimate of the circuit elements applying the method of Constraint Imbedding, using the specification at one selected critical frequency,

(ii) to improve upon this initial estimate in order to match the response characteristic over the entire frequency spectrum, by utilizing the transfer function and sensitivity capabilities of NASAP.

These two procedures are briefly described below.

A. CONSTRAINT IMBEDDING: Given a network N of arbitrary topological structure of known elements (R, L, C and controlled sources), the method of Constraint Imbedding is used to determine the unknown element values required to meet the design specifications at a single frequency. This is accomplished by first converting the design requirements into a set of voltage current constraints on the appropriate nodes and branches of the circuit. These voltage-current constraints are then implanted or imbedded into the network, and their effect on the V-I relationships of the remaining variable elements determined. Application of Ohm's law then yields the nominal values required of the variable elements. The mechanics of implementing this approach is as follows:

The network is divided into two separate parts, one consisting of the fixed portion of the network and the other consisting of the variable portion of the network elements. For the fixed portion of the network, the network equation on a nodal basis is written as:

$$YV = I_s \qquad (1)$$

where

Y is the (n-1) x (n-1) nodal admittance matrix,

V is the (n-1) vector of node voltages,

and

I_s is the (n-1) vector of forcing currents.

For the variable part of the network (consisting of r variable elements), the incidence relationship can be written as:

$$AI = 0 \qquad (2)$$

where

A is the (n-1) x r node incidence matrix,

and

I is the r vector of currents through the variable elements.

Combining equations (1) and (2) yields:

$$[A \mid Y] \left[\frac{I}{V}\right] = [I_s] \qquad (3)$$

Alternately, a consistent network forumulation based on a loop analysis yields:

$$[B \mid Z] \left[\frac{V}{I}\right] = [V_s] \qquad (4)$$

where

B is the (b-n+1) x r circuit matrix for the variable part of the network,

Z is the (b-n+1) x (b-n+1) impedance matrix for the fixed portion of the network,

I is the b vector of branch currents,

and

V is the (b-n+1) vector of source voltages in the various (b-n+1) number of basic meshes.

To either of the equations (3) or (4), equality relations accounting for appropriate forcing constraints can be appended. The problem now is to determine a feasible solution for the augmented or constrained system of network equations. In general these equations can be brought to the form

$$Ax = y \tag{5}$$

where

x is the unknown vector $\begin{bmatrix} I_{var} \\ V_{all} \end{bmatrix}$ or $\begin{bmatrix} V_{var} \\ I_{all} \end{bmatrix}$,

A is the coefficient matrix, in general, rectangular in nature,

and

y is the forcing vector $[I_s]$ or $[V_s]$.

The general solution to this matrix equation can be written as [15-18]

$$x = A^+ y + (I - A^+ A) z \tag{6}$$

where

A^+ is the pseudo inverse

and

z is an arbitrary vector orthogonal to the column space of A^*.

z can be appropriately chosen in order to realize positive network elements.

B. OPTIMIZATION USING NASAP: Once an initial estimate for the variable elements has been found based on a single frequency design using the method of Constraint Imbedding, the problem now is to extend the design using NASAP. This is done as follows:

Let the initially estimated values for the variable elements as obtained through the method of Constraint Imbedding be denoted as r vector $\underline{X}^o$. Using this set of element values, the network is analyzed with the help of NASAP and the actural response characteristic evaluated. It is obvious that at this stage of design there is no assurance that this actual characteristic will match the desired characteristic at frequencies other than the one selected for Constraint Imbedding application. A typical situation that might be expected at this point is illustrated in Fig. 3.

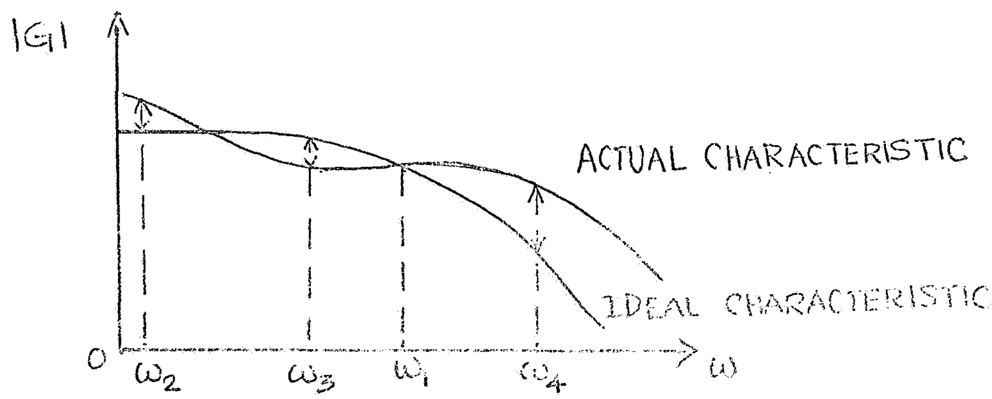

Fig. 3

To force the actual response to coincide with the ideal response, NASAP is employed as follows: a number of frequency points (equal to the number of adjustable parameters) is chosen. At each of these frequency points the sensitivities of $|G(j\omega)|$ with respect to each of the adjustable parameters are evaluated using NASAP. Knowing the deviation of the actual $|G(j\omega)|$ from the desired characteristic at the selected frequency points, the sensitivity equations can be written as

$$\frac{|G^j_{ideal}| - |G^j_o|}{|G^j_o|} = \sum_{i=1}^{r} S^{|G^j_o|}_{x^o_i} \cdot \frac{\Delta x^o_i}{x^o_i} \quad ; \; j = 1, \ldots, r \tag{7}$$

Solving this set of linear simultaneous equations for the changes Δx^o_i, the new improved values (assuming convergence), for x_i's are:

$$\underline{x}^1 = \underline{x}^o + \underline{\Delta x}^o \tag{8}$$

The whole process could now be iterated with this set of new values, in which case equations (7) and (8) are modified as

$$\frac{|G^j_{ideal}| - |G^j_k|}{|G^j_k|} = \sum_{i=1}^{r} S^{|G^j_k|}_{x^k_i} \cdot \frac{\Delta x^k_i}{x^k_i} \quad ; \; j = 1, \ldots, r \tag{9}$$

and

$$\underline{x}^{k+1} = \underline{x}^k + \underline{\Delta x}^k \tag{10}$$

respectively, at the kth iteration.

The entire process can be stopped when the desired accuracy, defined by

$$\sum_{j=1}^{r} \left[\frac{|G^j_{ideal}| - |G^j|}{|G^j|} \right]^2 < \epsilon \tag{11}$$

where ϵ is a small number specified by the designer, is realized. Computer programs based upon implementation of thi s design procedure, given in Appendices I and II, are termed ACCIP and NASAP II.

III. EXAMPLES:

The programs listed in Appendices I and II are now illustrated with a few typical examples.

1. A series resonant circuit

It is desired to determine the values of R, L, and C for the circuit shown in Fig. 4(a), so that the magnitude of the current characteristic will be as shown in Fig. 4(b).

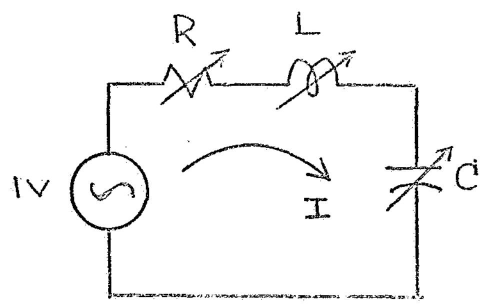

Fig. 4(a)

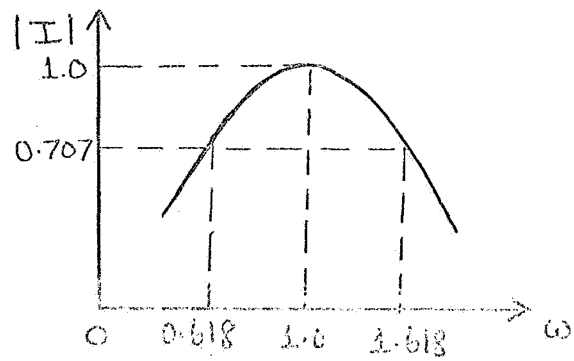

Fig. 4(b)

Solution: (1) First a critical frequency, $\omega = 1.618$ rad/sec is chosen and ACCIP is applied to determine an initial estimate for R, L, and C which gave the following results

$$R = 1.000\,\Omega \;;\; L = 1.385\,h \;;\; C = 0.5\,F.$$

(2) Using this as the starting set of values NASAP II is then employed to determine the element values in order to meet the given specifications at the other two frequencies also. The necessary input cards for NASAP II is given below:

1, 1, 1 ◁ No. of variable resistances, no. of variable inductances, and no. of variable capacitances

2, 3, 4 ◁ Branch no. of the variable elements

0.618, 1.000, 1.618 ◁ The selected frequency values in rad/sec

0.707, 1.000, 0.707 ◁ Ideal response values at the selected frequencies.

NASAP PROBLEM

V1 1 2 1

R1 2 3 1

L1 3 4 1.385

C1 4 1 0.5

OUTPUT

IR1/VV1

EXECUTE

◁ The standard data set as in NASAP application.

The final values:

$R = 1\Omega$; $L = 1h$; $C = 1F$

2. A third-order Butterworth filter

It is desired to select the values of L_1, C_1, and C_2 for the circuit shown in Fig. 5(a), so that the response will be a Butterworth response shown in Fig. 5(b).

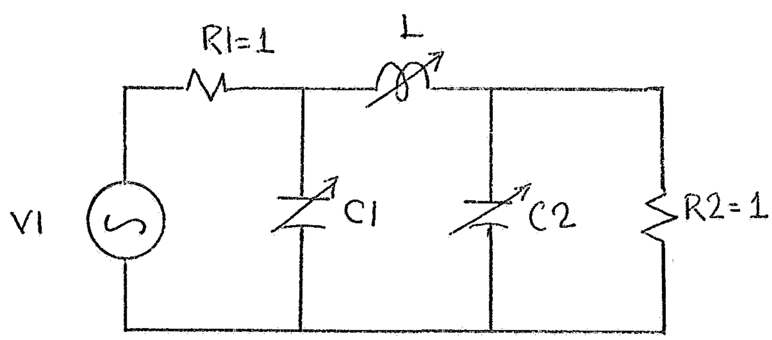

Fig. 5(a)

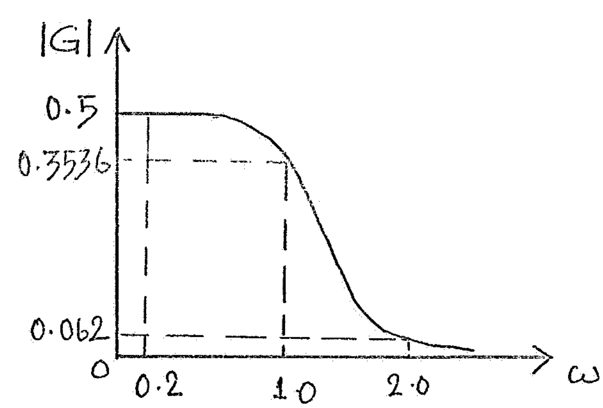

Fig. 5(b)

Solution: (1) First a critical frequency of ω = 1 rad/sec is chosen and ACCIP is applied. The network realized by this program is shown in Fig. 5(c).

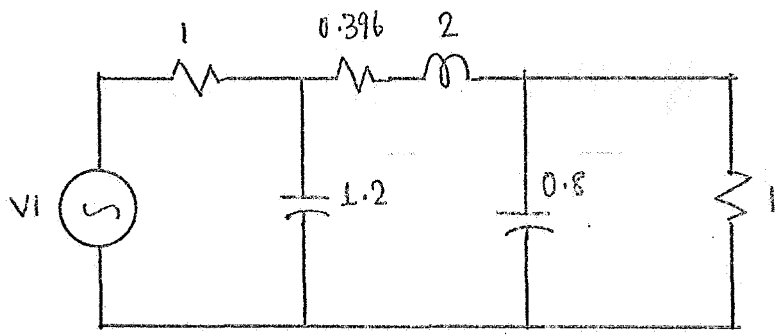

Fig. 5(c)

(2) Using this as the starting set of values NASAP II is then employed to determine the element values in order to meet the given specifications at the other two frequencies also. The necessary input data for NASAP II is given below:

1, 1, 2 ◁ No. of variable resistances, no. of variable inductances, no. of variable capacitances

4, 5, 6, 7 ◁ Branch no. of variable elements

0.2, 1.0, 2.0 ◁ The specified frequency points

0.5, 0.3536, 0.062 ◁ Ideal response at the selected frequencies.

NASAP PROBLEM

V1	1	2	1
R1	2	3	1
R2	5	1	1
R3	4	5	0.396
L1	3	4	2
C1	3	1	1.2
C2	5	1	0.8

◁ The standard data set as in as in NASAP application.

OUTPUT

VR2/VV1

EXECUTE

The final values are:

$$R3 \cong 0\Omega; \quad L1 = 2h; \quad C1 = 1F \;; \; C2 = 1F$$

3. Bridged-T network

It is desired to select the values of R_1, R_2, L_1, and C_1, for the circuit shown in Fig. 6(a), so that the magnitude response is as shown in Fig. 6(b).

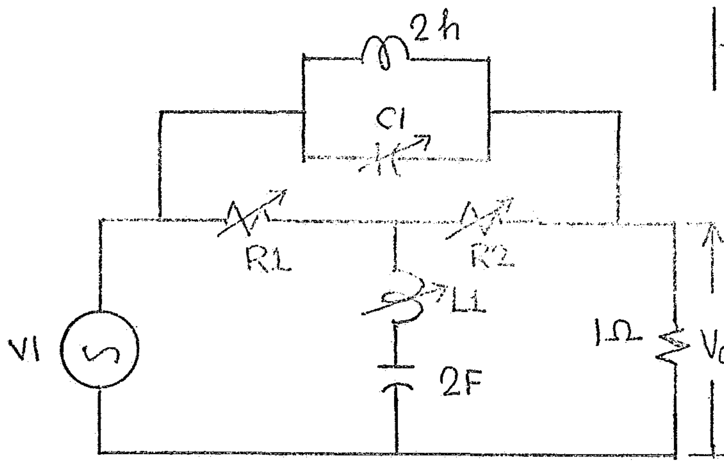

Fig. 6(a)

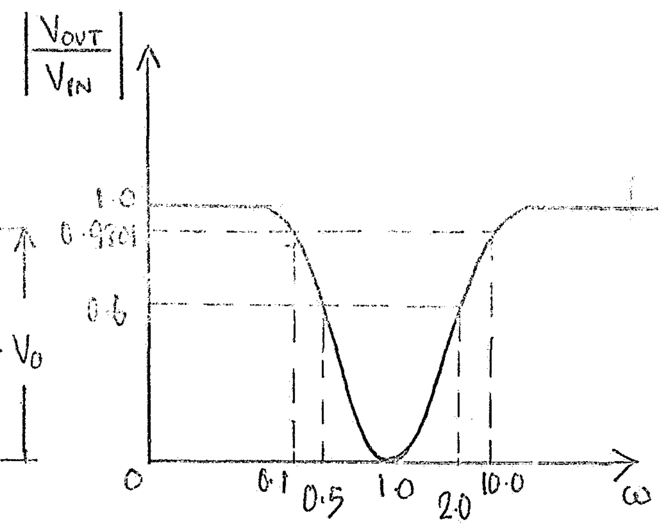

Fig. 6(b)

Solution: (1) Since the phase characteristic even at one frequency was not specified for this problem, ACCIP cannot be applied. Therefore, an initial guess was made for the element values. They are as follows:

$$R_1 = 1.5\Omega; \quad R_2 = 0.5\Omega; \quad L_1 = 1h \;; \; C_1 = 1F$$

(2) Using this as the starting set of values NASAP II is then employed to determine the element values in order to meet the given specifications at all the four frequencies. The final values are:

$$R1 = 1\Omega; \; R2 = 1\Omega \;; \; L1 = 0.5h \;; \; C1 = 0.5F.$$

IV. CONCLUSIONS AND RESULTS

The design procedure described in Section II has been incorporated in a computer program, the use of which is illustrated in Section III with several examples. The program is to be used in two stages, first as a single frequency design using ACCIP (A. C. Constraint Imbedding Program), and then for the optimized design using NASAP II, which incorporates the method of steepest descent as the optimizing scheme. Both these programs are listed in the Appendix. During the course of the development of NASAP II the following advantages have been gained as by products:

Modify capability: NASAP in its present form can be used to find the transfer function only for one set of element values. But now it is possible to determine the effect of element value variations, on the transfer function. Also, the program will evaluate the transfer function, if desired, at specified frequencies.

Multiple sensitivity capability: NASAP in its present form can be used to find the sensitivity of the transfer function with respect to only one element. But now it is possible to determine the sensitivity of the transfer function with respect to any number of parameters. Also, the program will evaluate these quantities, if desired, at specified frequencies.

The computer programs listed in the Appendices I and II were thoroughly tested for possible bugs by the students of a graduate course on 'Computer-Aided Circuit Design' conducted at Stevens during 1968-69, and the resulting experiences are detailed in reference 12.

V. REFERENCES

1. G. J. Herskowitz, Computer-Aided Integrated Circuit Design, McGraw-Hill Book Company, New York, 1968.

2. F. F. Kuo and J. F. Kaiser, System Analysis by Digital Computer, John Wiley and Sons, Inc., New York, 1966.

3. D. A. Calahan, Computer-Aided Network Design, McGraw-Hill Book Company, New York, 1968.

4. G. C. Temes and D. A. Calahan, "Computer-Aided Network Optimization, The State of the Art," Proc. of the IEEE, Vol. 55, No. 11, pp. 1832-63, November, 1967.

5. G. J. Herskowitz, "Active RC Filter Design by Computer," presented at the University of Missouri Computer-Aided Design Workshop, August, 1967.

6. R. Serbagi, "Developement of Network Analysis for Systems Applications Program (NASAP), Final Report, Contract No. 12-564 (NASA)," Digital Programming Services, Inc., Waltham, Mass.

7. L. P. McNamee and H. Potash, "A User's Guide and Programmer's Manual for NASAP," Report No. 68-38, Department of Engineering, University of California, Los Angeles, California, August, 1968.

8. 1620 Electronic Circuit Analysis Program (ECAP) (1620-EE-02X) User's Manual, IBM, Technical Publications Dept., White Plains, N.Y., 1965.

9. E. B. Kozemchak and M. A. Murray-Lasso, "Computer-Aided Circuit Design by Constrained Singular Imbedding," presented at the 1968 IEE International Symposium on Circuit Theory, Miami Beach, Florida, December, 1968.

10. G. J. Herskowitz and M. A. Murray-Lasso, "Application of Constraint Singular Imbedding to Computer-Aided Integrated Circuit Design," to be published in the IEEE PGCT Trans.

11. M. Sankaran, "Linear Circuit Design by Constraint Imbedding," Internal Communications, Dept. of Electrical Engineering, Stevens Institute of Technology, Hoboken, N.J., November, 1968.

12. G. J. Herskowitz and M. Sankaran, "Application of NASAP to the Design of Communication Circuits," Conference Record, 69C15-CT, Twelfth Midwest Symposium on Circuit Theory, Austin, April, 1969.

13. G. J. Herskowitz and M. Sankaran, "Student Experience with an On-Line Linear Circuit Design Program," to appear in the IEEE Trans. on Education, August, 1969.

14. C. B. Tompkins, "Methods of Steepest Descent," in Modern Mathematics for the Engineer (Edwin F. Beckenbach, ed.), Chapter 18, McGraw-Hill Book Company, New York, 1956.

15. R. Penrose, "A Generalized Inverse for Matrices," Proc. Cambridge Phil. Soc., Vol. 51, 1955, pp. 406-413.

16. T. N. E. Greville, "The Pseudo Inverse of a Rectangular or Singular Matrix and Its Application to the Solution of Linear Equations," SIAM Review, Vol. 1, 1959, pp. 38-43.

17. G. Rust and W. R. Burrus, "A Simple Algorithm for Computing the Generalized Inverse of a Matrix," Comm. of the ACM, Vol. 9, No. 5, pp. 381-87, May, 1966.

18. R. P. Voith, W. G. Vogt, and M. H. Mickle, "A Direct Computational Procedure for the Generalized Inverse," Conference Record, 69C15-CT, Twelfth Midwest Symposium on Circuit Theory, Austin, April, 1969.

APPENDIX A

```
      C       THIS IS AC CONSTRAINT IMBEDDING PROGRAM BY PSEUDO INVERSE TECHNIQUE
1             COMMON E(50),BETA(5),CS(20),Y(20,20),CURR(20),VFE(5),VCFE(5),
             1GAIN(5),GM(5),RBP(30),WI(20,20),ZI(20,20),TI(20,20),FI(20,20),
             2SOL(20),NVFE,NNI,KTRIP(30,3),KQUAD(5,2),KQUIN(20,2),KVR(15),
             3KVFE(5,2),NSB(5,2),NNK2,N,NN,KQUAV(5,2),KQUAC(5,2),NVR,KQUAS(5,2)
2             COMPLEX WI,ZI,TI,FI,SCL
3             COMPLEX ADMI(50),E,BETA,CS,Y,CURR,VFE,VCFE,GAIN,GM,CMPLX,CONJG
4             COMPLEX U(20,20),X(20,10),ALPHA(5)
5             EXTERNAL CABS
      C       NO. OF RES,CAP,IND BR.,NODES,CONTROLLED SOURCES,VOLT.& CURR. SOURCES
6        1000 READ 10,NR,NC,NL,NN,NCONSO,NVCVSO,NVCCSO,NCCVSC,NVOSO,NCURRS
7             NBRS=NR+NC+NL
8             N=NN-1
      C       RES,CAP,IND.VALUES
9             READ 20,(RBP(I),I=1,NBRS)
      C       FOR EACH BRANCH INITIAL NODE,FINAL NODE,AND VOLT. SOURCE NUMBER
10            READ 10,((KTRIP(I,J),J=1,3),I=1,NBRS)
      C       VALUES OF THE VOLTAGE SOURCES
11            READ 20,(E(I),I=1,NVOSO)
12            IF(NCURRS.EQ.0) GO TO 98
      C       FOR EACH INDEPENDENT CURR.SOURCE,INITIAL & FINAL NODES
13            READ 10,((KQUIN(I,J),J=1,2),I=1,NCURRS)
      C       VALUES OF THE INDEPENDENT CURRENT SOURCES
14            READ 20,(CS(I),I=1,NCURRS)
15         98 IF (NCONSO) 105,105,106
      C       FOR EACH CURR.CONTROLLED CURR.SOURCE,ITS BRANCH & ITS CONTOLLING BR.
16        106 READ 10,((KQUAD(I,J),J=1,2),I=1,NCONSO)
      C       VALUES OF THE CURRENT GAIN
17            READ 20,(BETA(I),I=1,NCONSO)
18        105 IF(NVCVSO) 107,107,108
      C       FOR EACH VOLT.CONTROLLED VOLT.SOURCE,ITS BRANCH & ITS CONTROLLING BR.
19        108 READ 10,((KQUAV(I,J),J=1,2),I=1,NVCVSO)
      C       VALUES OF THE VOLTAGE GAINS
20            READ 20,(GAIN(I),I=1,NVCVSO)
21        107 IF(NVCCSO) 117,117,118
      C       FOR EACH VOLT.CONTROLLED CURR.SOURCE,ITS BRANCH & ITS CONTROLLING BR.
22        118 READ 10,((KQUAC(I,J),J=1,2),I=1,NVCCSO)
      C       VALUES OF THE TRANS CONDUCTANCES
23            READ 20,(GM(I),I=1,NVCCSO)
24        117 IF(NCCVSO.EQ.0) GO TO 119
      C       FOR EACH CURR.CONTROLLED VOLT.SOURCE,ITS BRANCH & CONTCL.BRANCH
25            READ 10,((KQUAS(I,J),J=1,2),I=1,NCCVSO)
      C       VALUES OF ALPHAS
26            READ 20,(ALPHA(I),I=1,NCCVSO)
27        119 CONTINUE
      C       VALUE OF OMEGA
28            READ 20,OMEGA
      C       NO.OF VARIABLE IMPEDANCES,NO. OF VFE,NO. OF CFE.
29            READ 10,NVR,NVFE,NCFE
      C       BRANCH NUMBERS OF THE VARIABLE IMPEDANCES
30            READ 10,(KVR(I),I=1,NVR)
31            IF(NVFE.EQ.0) GO TO 212
      C       PLUS AND MINUS NODE FOR EACH VFE
32            READ 10, ((KVFE(I,J),J=1,2),I=1,NVFE)
      C       VALUE OF EACH VFE
33            READ 110,(VFE(I),I=1,NVFE)
34        212 IF (NCFE .EQ. 0) GO TO 329
      C       FOR EACH CFE,ITS BRANCH NO. AND INJECTION NODE
35            READ 10,((NSB(I,J),J=1,2),I=1,NCFE)
```

```
C       VALUE OF EACH CFE
        READ 110,(VCFE(I),I=1,NCFE)
    329 DO 317 I=1,20
        CURR(I)=(0.,0.)
        DO 317 J=1,20
    317 Y(I,J)=(0.,0.)
        IF(NR .EQ. 0) GO TO 5
        DO 2 I=1,NR
        XR=1./RBP(I)
      2 ADMI(I)=CMPLX(XR,0.)
      5 IF(NC .EQ. 0)GO TO 4
        DO 3 I=1,NC
        XR=OMEGA*RBP(I+NR)
        K=I+NR
      3 ADMI(K)=CMPLX(0.,XR)
      4 IF(NL .EQ. 0) GO TO 6
        DO 14 I=1,NL
        XR=OMEGA*RBP(I+NR+NC)
        L=I+NR+NC
        ADMI(L)=CMPLX(0.,XR)
     14 ADMI(I+NR+NC)=1./ADMI(I+NR+NC)
      6 CONTINUE
        DO 300 I=1,NVR
    300 ADMI(KVR(I))=(0.0,0.0)
    330 IF(NVFE .EQ. 0) GO TO 213
        DO 303 I=1,NVFE
        L=KVFE(I,2)
        NBRS=NBRS+1
        NN=NN+1
        NCURRS=NCURRS+1
        ADMI(NBRS)=CMPLX(1.,0.)
        KTRIP(NBRS,1)=NN
        KTRIP(NBRS,2)=L
        KTRIP(NBRS,3)=50
        E(50)=(0.,0.)
        CS(NCURRS)=VFE(I)*ADMI(NBRS)
        KQUIN(NCURRS,1)=L
        KQUIN(NCURRS,2)=NN
    303 CONTINUE
    213 IF(NCFE .EQ. 0) GO TO 109
        DO 304 I=1,NCFE
        K=NSB(I,2)
        NN=NN+1
        NCURRS=NCURRS+1
        CS(NCURRS)=VCFE(I)
        KQUIN(NCURRS,1)=K
        KQUIN(NCURRS,2)=NN
        L=NSB(I,1)
        IF(KTRIP(L,1) .EQ. K) KTRIP(L,1)=NN
        IF(KTRIP(L,2) .EQ. K) KTRIP(L,2)=NN
    304 CONTINUE
    109 IF(NCONSO) 102,102,103
    103 DO 91 I=1,NCONSO
        LL=KQUAD(I,2)
        KK=KQUAD(I,1)
        J=KTRIP(KK,1)
        K=KTRIP(KK,2)
        LLL=KTRIP(LL,3)
        CURR(J)=CURR(J)-ADMI(LL)*E(LLL)*BETA(I)
        CURR(K)=CURR(K)+ADMI(LL)*E(LLL)*BETA(I)
```

```
      L=KTRIP(LL,1)
      M=KTRIP(LL,2)
      Y(J,L)=Y(J,L)+BETA(I)*ADMI(LL)
      Y(K,M)=Y(K,M)+BETA(I)*ADMI(LL)
      Y(K,L)=Y(K,L)-BETA(I)*ADMI(LL)
   91 Y(J,M)=Y(J,M)-BETA(I)*ADMI(LL)
  102 IF(NVCVSO) 111,111,112
  112 DO 92 I=1,NVCVSO
      LL=KQUAV(I,2)
      KK=KQUAV(I,1)
      J=KTRIP(KK,1)
      K=KTRIP(KK,2)
      L=KTRIP(LL,1)
      M=KTRIP(LL,2)
      LLL=KTRIP(LL,3)
      CURR(J)=CURR(J)+E(LLL)*GAIN(I)*ADMI(KK)
      CURR(K)=CURR(K)-E(LLL)*GAIN(I)*ADMI(KK)
      Y(K,L)=Y(K,L)+GAIN(I)*ADMI(KK)
      Y(K,M)=Y(K,M)-GAIN(I)*ADMI(KK)
      Y(J,L)=Y(J,L)-GAIN(I)*ADMI(KK)
   92 Y(J,M)=Y(J,M)+GAIN(I)*ADMI(KK)
  111 IF(NVCCSO) 121,121,122
  122 DO 93 I=1,NVCCSO
      LL=KQUAC(I,1)
      KK=KQUAC(I,2)
      J=KTRIP(KK,1)
      K=KTRIP(KK,2)
      LLL=KTRIP(KK,3)
      L=KTRIP(LL,1)
      M=KTRIP(LL,2)
      CURR(L)=CURR(L)+GM(I)*E(LLL)
      CURR(M)=CURR(M)-GM(I)*E(LLL)
      Y(L,J)=Y(L,J)+GM(I)
      Y(L,K)=Y(L,K)-GM(I)
      Y(M,J)=Y(M,J)-GM(I)
   93 Y(M,K)=Y(M,K)+GM(I)
  121 IF(NCCVSO) 131,131,132
  132 DO 94 I=1,NCCVSO
      LL=KQUAS(I,2)
      KK=KQUAS(I,1)
      J=KTRIP(KK,1)
      K=KTRIP(KK,2)
      L=KTRIP(LL,1)
      M=KTRIP(LL,2)
      LLL=KTRIP(LL,3)
      CURR(J)=CURR(J)+E(LLL)*ADMI(LL)*ALPHA(I)*ADMI(KK)
      CURR(K)=CURR(K)-E(LLL)*ADMI(LL)*ALPHA(I)*ADMI(KK)
      Y(J,L)=Y(J,L)-ADMI(LL)*ALPHA(I)*ADMI(KK)
      Y(J,M)=Y(J,M)+ADMI(LL)*ALPHA(I)*ADMI(KK)
      Y(K,L)=Y(K,L)+ADMI(LL)*ALPHA(I)*ADMI(KK)
   94 Y(K,M)=Y(K,M)-ADMI(LL)*ALPHA(I)*ADMI(KK)
  131 CONTINUE
C     CALCULATE CONTRIBUTION OF IND.CURRENT SOURCES TO RSV.
      IF(NCURRS) 101,101,104
  104 DO 17 I=1,NCURRS
      J=KQUIN(I,1)
      K=KQUIN(I,2)
      CURR(J)=CURR(J)-CS(I)
   17 CURR(K)=CURR(K)+CS(I)
  101 CONTINUE
```

```
C     CAL.CONTRI.OF R TO Y MATRIX & CCNT.OF VOLTAGE SOURCES TO RSV
      DO 12 I=1,NBRS
      J=KTRIP(I,1)
      K=KTRIP(I,2)
      L=KTRIP(I,3)
      Y(J,K)=Y(J,K)-ADMI(I)
      Y(K,J)=Y(K,J)-ADMI(I)
      Y(J,J)=Y(J,J)+ADMI(I)
      Y(K,K)=Y(K,K)+ADMI(I)
      CURR(J)=CURR(J)-E(L)*ADMI(I)
   12 CURR(K)=CURR(K)+E(L)*ADMI(I)
      NNI=NN-NVFE-NCFE
      IF(NVFE .EQ. 0)GO TO 214
      DO 310 I=1,NVFE
      K=NNI+I
      L=KVFE(I,1)
      M=KVFE(I,2)
      CURR(M)=CURR(M)+CURR(K)
      DO 500 JL=1,NN
  500 Y(JL,L)=Y(JL,L)+Y(JL,K)
      DO 310 J=1,NN
  310 Y(M,J)=Y(M,J)+Y(K,J)
  214 NNJ=NN-NCFE
      IF(NCFE .EQ. 0) GO TO 341
      DO 311 I=1,NCFE
      K=NNJ+I
      L=NSB(I,2)
      CURR(L)=CURR(L)+CURR(K)
      DO 321 J=1,NN
  321 Y(J,L)=Y(J,L)+Y(J,K)
      DO 311 JL=1,NN
  311 Y(L,JL)=Y(L,JL)+Y(K,JL)
  341 NNK=NNI
      DO 312 JJ=1,NNK
      K=NNK+1-JJ
      DO 312 I=1,NN
  312 Y(I,K+NVR-1)=Y(I,K)
      DO 314 I=1,NN
      DO 314 J=1,NVR
  314 Y(I,J)=(0.,0.)
      DO 313 I=1,NVR
      K=KTRIP(KVR(I),1)
      L=KTRIP(KVR(I),2)
      Y(K,I)=(1.0,0.0)
  313 Y(L,I)=(-1.0,0.0)
      NNK2=NNK+NVR-1
      DO 320 I=2,NN
      CURR(I-1)=CURR(I)
      DO 320 J=1,NNK2
  320 Y(I-1,J)=Y(I,J)
      N=NN-1
      DO 39 I=1,N
   39 PRINT 1,(Y(I,J),J=1,NNK2),CURR(I)
      IF(N.EQ.NNK2) GO TO 42
      DO 41 I=1,N
      DO 41 J=1,NNK2
      WI(J,I)=CONJG(Y(I,J))
   41 CONTINUE
C     WI IS CONJUGATE TRANSPOSE OF Y
      CALL PSEUDO(WI,FI,NNK2,N)
```

```
      DO 44 I=1,N
       DO 44 J=1,NNK2
      TI(J,I)=CONJG(FI(I,J))
   44 CONTINUE
      GO TO 43
   42 CALL CROUT(Y,TI,N)
   43 CALL PRODCT(TI,CURR,SOL,NNK2,N)
      IF(N.EQ.NNK2) GO TO 516
C     THIS SECTION IS TO CONSTRUCT IDENTITY MATRIX OF NULL SPACE
      DO 504 I=1,NNK2
      DO 504 J=1,NNK2
      SUM=(0.,0.)
      DO 505 K=1,N
  505 SUM=SUM+TI(I,K)*Y(K,J)
      IF(I-J) 506,507,506
  507 U(I,J)=CMPLX(1.,0.)-SUM
      GO TO 504
  506 U(I,J)=-SUM
  504 CONTINUE
C     THIS SECTION IS TO PRINT OUT THE NONZERO COLUMNS OF NULL SPACE
      NCOUNT=0
      DO 508 J=1,NNK2
      DO 509 I=1,NNK2
      IF(CABS(U(I,J)).GE.(1.E-2)) GO TO 510
  509 CONTINUE
      GO TO 508
  510 NCOUNT=NCOUNT+1
      DO 511 K=1,NNK2
  511 X(K,NCOUNT)=U(K,J)
  508 CONTINUE
      IF(NCOUNT.EQ.0) GO TO 516
      DO 512 I=1,NNK2
  512 PRINT 513,SOL(I),(X(I,J),J=1,NCOUNT)
  513 FORMAT (/,3(2E12.3,6X))
      PRINT 13
   13 FORMAT (1X,////)
  516 CONTINUE
      DO 2193 I=1,NNK2
      IF(I.GT.NVR) GO TO 2195
      PRINT 2192,KVR(I),SOL(I)
 2192 FORMAT (1X,2HI(,I2,4H) = ,2E15.7)
      GO TO 2193
 2195 IN=I-NVR+1
      PRINT 2194,IN,SOL(I)
 2194 FORMAT (1X,2HV(,I2,4H) = ,2E15.7)
 2193 CONTINUE
      IF(N.NE.NNK2) GO TO 804
      DO 810 I=1,NVR
      II=KTRIP(KVR(I),1)+NVR-1
      JJ=KTRIP(KVR(I),2)+NVR-1
      IF(KTRIP(KVR(I),1).EQ.1) II=NNK2+1
      IF(KTRIP(KVR(I),2).EQ.1) JJ=NNK2+1
      IF(KVR(I).GT.NR) GO TO 811
      ELEVAL=REAL((SOL(II)-SOL(JJ))/SOL(I))
      PRINT 801,KVR(I),ELEVAL
  801 FORMAT (1X,2HR(,I2,4H) = ,1PE15.7,6H  OHMS)
      GO TO 810
  811 IF(KVR(I).GT.(NR+NC)) GO TO 812
      ELEVAL=AIMAG((SOL(I))/((SOL(II)-SOL(JJ))*OMEGA))
      PRINT 802,KVR(I),ELEVAL
```

```
  802 FORMAT (1X,2HC(,I2,4H) = ,1PE15.7,8H  FARADS)
      GO TO 810
  812 ELEVAL=AIMAG((SOL(II)-SOL(JJ))/(SOL(I)*OMEGA))
      PRINT 803,KVR(I),ELEVAL
  803 FORMAT (1X,2HL(,I2,4H) = ,1PE15.7,9H  HENRIES)
  810 CONTINUE
  804 CONTINUE
    1 FORMAT (/,(1X,8E12.3))
   10 FORMAT (40I2)
   20 FORMAT (6E12.3)
  110 FORMAT (5F10.5)
      GO TO 1000
      END
```

```
      SUBROUTINE CROUT(G,F,N)
      DIMENSION G(20,20), A(20,20), F(20,20)
      COMPLEX G,A,F,SUM
      NN=2*N
      DO 10 I=1,N
      DO 11J=1,N
   11 G(I,N+J)=(0.0,0.0)
   10 G(I,N+I)=(1.0,0.0)
C     FIRST COLUMN OF AUXILIARY MATRIX IAS THE FIRST COLUMN OF THE GIVEN MA
      DO 20 I=1,N
   20 A(I,1)=G(I,1)
C FIRST ROW OF AUXILIARY MATRIX IS NORMALISED ROW OF THE GIVEN MATRIX
      DO 30 J=2,NN
   30 A(1,J)=G(1,J)/A(1,1)
      DO 40 I=2,N
      DO 40 J=2,NN
      II=I-1
      JJ=J-1
      IF (I-J) 2,3,3
    3 SUM=(0.0,0.0)
      DO 4 K=1,JJ
    4 SUM=SUM+A(I,K)*A(K,J)
      A(I,J)=G(I,J)-SUM
      GO TO 40
    2 SUM=(0.0,0.0)
      DO 5 K=1,II
    5 SUM=SUM+A(I,K)*A(K,J)
      A(I,J)=  (G(I,J)-SUM)/A(I,I)
   40 CONTINUE
      DO 51 J=1,N
      F(N,J)=A(N,N+J)
      DO 50 I=2,N
      II=I-1
      SUM=(0.0,0.0)
      DO 60 K=1,II
   60 SUM=SUM+A(N-I+1,N-K+1)*F(N-K+1,J)
   50 F(N-I+1,J)=A(N-I+1,N+J)-SUM
   51 CONTINUE
      DO 70 I=1,N
   70 PRINT 6,(F(I,J),J=1,N)
    6 FORMAT(1X,8F10.5)
      RETURN
      END
```

```
      SUBROUTINE PRODCT(A,B,C,M,N)
      DIMENSION A(20,20),B(20),C(20)
      COMPLEX A,B,C
      DO 11 I=1,20
   11 C(I)=(0.0,0.0)
      DO 10 I=1,M
      DO 12 K=1,N
   12 C(I)=C(I)+A(I,K)*B(K)
   10 CONTINUE
      WRITE(1,13)
   13 FORMAT (1X,///)
      RETURN
      END
```

```
      SUBROUTINE PSEUDO(A,T,M,N)
C      M IS THE NUMBER OF ROWS OF THE GIVEN MATRIX
C     N IS THE NUMBER OF COLUMNS OF THE GIVEN MATRIX
C     M IS GREATER THAN N
      DIMENSION A(20,20),B(20,20),D(20,20),T(20,20)
       COMPLEX A,B,D,T,P,W,Q,CONJG
      DO 1 L=1,N
      DO 1 J=1,M
    1 T(L,J)=(0.,0.)
      DO 2 I=1,N
      DO 2 J=1,N
      D(I,J)=(0.,0.)
       DO 3 K=1,M
    3 D(I,J)=D(I,J)+CONJG(A(K,I))*A(K,J)
    2 B(I,J)=D(I,J)
      DO 4 K=1,N
      P=(0.,0.)
      DO 5 J=1,N
    5 P=P+B(K,J)*D(J,K)
      IF(CABS(P).EQ.0.)  GO TO 4
       DO 8 L=1,M
      W=(0.,0.)
      Q=(0.,0.)
       DO 6 J=1,N
      Q=Q+B(K,J)*T(J,L)
      IF((L.GT.N).OR.(L.LE.K))  GO TO 6
       W=W+B(K,J)*D(J,L)
    6  CONTINUE
       DO 7 J=1,N
      T(J,L)=T(J,L)+D(J,K)*(CONJG(A(L,K))-Q)/P
      IF((L.GT.N).OR.(L.LE.K))  GO TO 7
       D(J,L)=D(J,L)-D(J,K)*(W/P)
    7  CONTINUE
    8  CONTINUE
    4  CONTINUE
       RETURN
      END
```

APPENDIX B

```
            C     MAIN SECTION FOR NASAP II
0001              COMMON  /INPUT/DUM1(120)
                 1,IDD(40),DUM2(40),IST(40),DUM3(40),IJTAG(40),IKTAG(40),Z(40),M2
0002              COMMON /LOOPS/ INC(25,42),DUM5(543),M,N,DUM4(3205)
0003              COMMON / SAVE/ IDDP(15),ISTP(15),IJTAGP(15),IKTAGP(15),ZP(15),
                 1INCP(15,15),MPER,NPER,M2PER
0004              COMMON /STORE/ PSCON(15),PSJAY(15),PSN1(15,10),PSD1(15,10),
                 1PSN2(15,10),PSD2(15,10)
0005              COMMON /TRIAL/ OMEGA(10),KTRIAL
0006              COMMON /OUTPUT/ A(10,10), KVR(10),G(10),GI(10),ERROR(10),SOL(10)
0007              COMMON/BINGO/KMOD,NTREE,NLINK,KTREE(10),KLINK(10)
0008              COMPLEX W,UP,DOWN,UP1,DOWN1,UP2,DOWN2
0009            5 KMOD=1
0010              KTRIAL=1
            C     NUMBER OF VARIABLE RESISTANCE,INDUCTANCE AND CAPACITANCE ELEMENTS
0011              READ 7,NR,NL,NC
0012              NFREQ=NR+NL+NC
            C     BRANCH NUMBER OF VARIABLE ELEMENTS
0013              READ 7,(KVR(I),I=1,NFREQ)
            C     THE FREQUENCY POINTS AT WHICH THE MATCH IS DESIRED
0014              READ 8,(OMEGA(I),I=1,NFREQ)
            C     DESIRED MAGNITUDE VALUES AT THE DESIRED FREQUENCIES
0015              READ 8,(GI(I),I=1,NFREQ)
0016            6 GO TO (21,22,23,24,25,26,27,28,29),KTRIAL
0017           21 CALL ZERO
0018              CALL NASINP
            C     IDD,IST,IKTAG,IJTAG,ISTAG,INC,Z,M,N,M2 HAVE BEEN STORED IN NASINP
0019            9 IKTAG(KVR(1))=1
0020            1 CALL GETCON(ISW)
0021              CALL FSORL
0022              CALL HIGORL
            C     NOW IDD,IST,IKTAG,IJTAG,INC,Z,M,N,M2 ARE RESET
0023              N=NPER
0024               M=MPER
0025              M2=M2PER
0026              DO 1001 I=1,M
0027               IDD(I)=IDDP(I)
0028              IST(I)=ISTP(I)
0029                IJTAG(I)=IJTAGP(I)
0030              IKTAG(I)=IKTAGP(I)
0031              Z(I)=ZP(I)
0032         1001 CONTINUE
0033              MNM=N+1
0034              DO 1002 I=1,MNM
0035              DO 1002 J=1,M2
0036              INC(I,J)=INCP(I,J)
0037         1002 CONTINUE
0038              KTRIAL=KTRIAL+1
0039              IF(KTRIAL.GT.NFREQ) GO TO 4
0040               GO TO 6
0041           22 IKTAG(KVR(2))=1
0042              GO TO 1
0043           23  IKTAG(KVR(3))=1
0044              GO TO 1
0045           24  IKTAG(KVR(4))=1
0046              GO TO 1
```

```
0047        25   IKTAG(KVR(5))=1
0048            GO TO 1
0049        26   IKTAG(KVR(6))=1
0050            GO TO 1
0051        27   IKTAG(KVR(7))=1
0052            GO TO 1
0053        28   IKTAG(KVR(8))=1
0054            GO TO 1
0055        29   IKTAG(KVR(9))=1
0056            GO TO 1
0057         4   CONTINUE
0058            SQERR=0.
0059            DO 10 I=1,NFREQ
0060            W=CMPLX(0.,OMEGA(I))
0061            CALL TF(W,PSJAY,UP)
0062            CALL TF(W,PSCCN,DOWN)
0063            G(I)=CABS(UP/DOWN)
0064            ERROR(I)=(GI(I)-G(I))/G(I)
0065            SQERR=(SQERR+ERROR(I)**2)
0066            DO 10 J=1,NFREQ
0067            CALL SF(W,PSN1,UP1,J)
0068            CALL SF(W,PSN2,UP2,J)
0069            CALL SF(W,PSD1,DOWN1,J)
0070            CALL SF(W,PSD2,DOWN2,J)
0071            A(I,J)=REAL((UP1/DOWN1)+(UP2/DOWN2))
0072         10 CONTINUE
0073            AK=1.
0074            IF((SQERR/NFREQ).GT..04) AK=2.
0075            IF((SQERR/NFREQ).GT..09) AK=5.
0076            DO 53 J=1,NFREQ
0077            DO 54 K=1,NTREE
0078            IF(KVR(J).NE.KTREE(K)) GO TO 54
          C     TREE ELEMENT HAS BEEN FIXED
0079            IF(J.GT.(NR+NL)) GO TO 55
0080            DO 56 I=1,NFREQ
0081         56 A(I,J)=A(I,J)/Z(KVR(J))
0082             GO TO 53
0083         55 DO 57 I=1,NFREQ
0084        57  A(I,J)=-A(I,J)*Z(KVR(J))
0085             GO TO 53
0086         54 CONTINUE
0087            DO 58 K=1,NLINK
0088            IF(KVR(J).NE.KLINK(K)) GO TO 58
          C     LINK ELEMENT HAS BEEN FIXED
0089            IF(J.GT.(NR+NL)) GO TO 59
0090            DO 60 I=1,NFREQ
0091        60  A(I,J)=-A(I,J)/Z(KVR(J))
0092            GO TO 53
0093       59   DO 61 I=1,NFREQ
0094         61 A(I,J)=A(I,J)*Z(KVR(J))
0095            GO TO 53
0096         58 CONTINUE
0097         53 CONTINUE
0098            DO 30 I=1,NFREQ
0099            ERROR(I)=ERROR(I)/AK
0100        30  PRINT 31,(A(I,J),J=1,NFREQ),ERROR(I)
```

```
      CALL LEC(A,NFREQ)
      DO 41 I=1,NFREQ
      SOL(I)=0.
      DO 42 J=1,NFREQ
   42 SOL(I)=SOL(I)+A(I,J)*ERROR(J)
   41 CONTINUE
      DO 43 I=1,NFREQ
   44 IF(I.GT.(NR+NL)) GO TO 45
      Z(KVR(I))=Z(KVR(I))+SOL(I)
      PRINT 32,KVR(I),Z(KVR(I))
      GO TO 46
   45 Z(KVR(I))=1./((1./Z(KVR(I)))+SOL(I))
       TEMP=1./Z(KVR(I))
      PRINT 32,KVR(I),TEMP
   46 ZP(KVR(I))=Z(KVR(I))
   43 CONTINUE
      KTRIAL=1
      KMOD=KMOD+1
      IF(KMOD.LE.10) GC TO 9
    7 FORMAT (10I2)
    8 FORMAT (5E12.4)
   31 FORMAT (1X,3E15.7,5X,E15.7)
   32 FORMAT (1X,2HB(,I3,4H) = ,E15.7)
      GO TO 5
      END
```

```
0001            SUBROUTINE ZERO
          C  *  SETS ALL NASAP QUANTITIES TO ZERO
0002            COMMON /LOOPS/ IVV(40,40),ITT(40,40),IWW(40,40)
0003            DIMENSION ZIP(4800)
0004            EQUIVALENCE (ZIP,IVV)
0005            DO 1 I=1,4800
0006          1 ZIP(I)=0.
0007            RETURN
0008            END
```

```
 0001                SUBROUTINE NASINP
 0002                DIMENSION FEQ(3)
 0003                DIMENSION                    ITSK1(15,4),ITSK2(3,6),ITSK3(15,6),
                    1ITSK4(10,5),ITSK5(9,10),INLST(50),INSRT1(80),INSRT2(21,5)
 0004                COMMON  /INPUT/DUM1(120)
                    1,IDD(40),DUM2(40),IST(40),DUM3(40),IJTAG(40),IKTAG(40),Z(40),M2
 0005                COMMON /LOOPS/ INC(25,42),DUM5(543),M,N,DUM4(3205)
 0006                COMMON /RQRMNT/NRQR(10)
 0007                COMMON / SAVE/ IDDP(15),ISTP(15),IJTAGP(15),IKTAGP(15),ZP(15),
                    1INCP(15,15),MPER,NPER,M2PER
 0008                DATA INSRT1/
                    1 16,2*0,17,6*0,18,9*0,19,21,0,20,19*0,3,2,15,12,0,8,7,2*0,6,5,9,0,
                    2 10,11,0,4,3*0,13,14,1,4*0,-1,-2,-3,-4,-5,-6,-7,-8,-9,-10
                    3/
 0009                DATA    INSRT2/
                    1 11,2,1,9,6,5,4,0,7,8,2*0,10,0,3,12,3*0,13,16*0,
                    2 1,3*0,2,0,8,4*0,4,3,2,5,0,6,1,0,7,0,9,10,11,12,
                    3 13,0,5,0,1,4,3,0,2,8*0,6,3*0,7,0,5,0,1,4,3,0,2,4*0,1,3,0,
                    4 8,6,4*0,7
                    5 /
 0010                DATA    ITSK1/
                    1 22,6,21,14,6,21,6,2*14,6,21,6,19,22,23,4,11*22,19,22,23,8,
                    2 11*22,2*7,23,12*22,2*15,23
                    4/
 0011                DATA    ITSK2/
                    1  4,2*19,1,2*2,22,2*18,4,2*19,1,2*3,22,2*15
                    2 /
 0012                DATA    ITSK3/
                    1 10,8*22,19,17,19,5,22,23,10,8*22,26,18,3*22,23,9,
                    2 8*22,26,4*22,23,10,2*19,12,4*11,12,26,18,2*22,
                    3 19,23,9,2*19,12,4*11,12,26,3*22,19,23,22,2*19,6*22,
                    4 15,3*22,15,23
                    4 /
 0013                DATA    ITSK4/
                    1 2*22,13,2*22,13,2*19,2*23,22,6,22,2*6,22,19,
                    2 22,2*23,4,5*22,19,22,2*23,8,5*22,7,22,2*23,
                    3 6*22,19,22,2*23
                    3 /
 0014                DATA    ITSK5/
                    122,24,13,25,22,13,19,22,20,22,5*6,19,2*22,4,5*22,
                    2 19,2*22,8,5*22,19,7,7*22,19,16,3*22,13,2*22,
                    3 13,19,3*22,5*6,19,2*22,4,5*22,19,2*22,8,5*22,
                    4 7,7,2*22,5*27,19,28,22
                    4 /
 0015            101 FORMAT(69A1)
 0016            102 FORMAT(24H FUNCTIONS NOT AVAILABLE)
 0017            103 FORMAT(29H INPUT CODING ERROR IN COLUMN,I3)
 0018            104 FORMAT(27H MACHINE LANGUAGE BREAKDOWN)
 0019            105 FORMAT(44H  USE HEADING CARD LABELED     NASAP PROBLEM)
 0020            106 FORMAT(15I6/)
 0021            107 FORMAT(1X,50A1)
 0022            108 FORMAT(E9.2)
 0023            109 FORMAT(17HTOO MANY ELEMENTS)
 0024            110 FORMAT(1X,45I2)
 0025            111 FORMAT(1X,/////)
 0026            115 FORMAT(1H1)
```

```
0027      116 FORMAT(1X,/)
0028          WRITE (6,115)
0029        1 READ (5,101) INLST
        C     TEST FOR EOF ON INPUT UNIT
0030      986 CONTINUE
0031          IF(ICHAR(INLST(1))-55) 2,4,2
0032        2 WRITE (6,105)
0033        3 WRITE (6,115)
0034      301 READ (5,101) INLST
        C     TEST FOR EOF ON INPUT UNIT
0035          IF(ICHAR(INLST(1))-55) 301,4,301
0036        4 DO 5 I=1,41
0037          Z(I)=0
0038          DO 5 J=1,25
0039        5 INC(J,I)=0
0040          IRQR=0
0041          IFRQ=-1
0042          ITHE=0
0043          FEQ(1)=0.
0044          FEQ(2)=0.
0045          THI=0
0046          M=2
0047          WRITE (6,107) INLST
0048          WRITE (6,116)
0049        6 ISGN=+1
0050          LD=0
0051          L1=1
0052        7 L2=1
0053          IZ=0
0054          II=1
0055          M=M+1
0056          READ (5,101) INLST
        C     TEST FOR EOF ON INPUT UNIT
0057          WRITE (6,107) INLST
0058      260 DO 39 I=II,50
0059          ILST=ICHAR(INLST(I))
0060          IF(INSRT1(ILST+1))10,98,8
0061        8 L3=INSRT1(ILST+1)
0062          ILST=L3
0063          IF(INSRT2(L3,L1))99,98,9
0064        9 L3=INSRT2(L3,L1)+1
0065          GO TO 11
0066       10 L3=1
0067          ILST=-INSRT1(ILST+1)-1
0068       11 GO TO (12,13,14,15,16),L1
0069       12 L3=ITSK1(L3,L2)
0070          J=3
0071          GO TO 17
0072       13 L3=ITSK2(L3,L2)
0073          GO TO 17
0074       14 L3=ITSK3(L3,L2)
0075          GO TO 17
0076       15 L3=ITSK4(L3,L2)
0077          J=4
0078          GO TO 17
0079       16 L3=ITSK5(L3,L2)
```

```
0080            IF(L3-24) 293,200,292
0081        292 IF(L3-25) 293,250,293
0082        293 IRQR=IRQR+1
0083            IF(IRQR-10) 294,294,17
0084        294 NRQR(IRQR)=INLST(I)
0085         17 CONTINUE
0086            GO TO (18,19,19,20,22,23,24,25,26,27,29,32,33,34,35,36,37,38,39,
               143,97,98,99,200,250,265,270,290),L3
0087        270 L2=7
0088            ISGN=1
0089            M=4
0090            GO TO 23
0091        290 L2=7
0092            M=4
0093            ISGN=1
0094            GO TO 39
0095        265 IF(IFRQ) 266,203,203
0096        266 IF(ITME) 39,39,202
0097        202 THI=IZ
0098            THI=THI*10.**(-LD)
0099            L1=5
0100            ITME=0
0101            LD=0
0102            ISGN=1
0103            GO TO 7
0104        203 IFRQ=IFRQ+1
0105            FEQ(IFRQ)=IZ
0106            FEQ(IFRQ)=FEQ(IFRQ)*10.**(-LD)
0107            L1=3
0108            L2=1
0109            LD=0
0110            IZ=0
0111            ISGN=1
0112            II=I
0113            IF(IFRQ-3) 260,204,204
0114        204 L1=5
0115            IFRQ=-1
0116            GO TO 7
0117        200 IFRQ=0
0118            II=II+4
0119            GO TO 255
0120        250 II=I+4
0121            ITME=1
0122        255 L1=3
0123            L2=1
0124            GO TO 260
0125         18 MN=10*MN+ILST
0126         19 INC(MN+3,M)=4*(L2-3)**3/((L2-3)**2+1)**2
0127            GO TO (38,37,35),L3
0128         20 IF(ILST) 99,39,21
0129         21 MN=ILST
0130            GO TO 38
0131         22 ISGN=-1
0132            GO TO 38
0133         23 INC(J,M)=100*ISGN*ILST
0134            GO TO 38
```

```
 24 INC(J,M)=INC(J,M)+ISGN*MN
    GO TO (361,38,35,38),J
 25 INC(J,M)=INC(J,M)+ISGN*(10*MN+ILST)
    GO TO 38
 26 LD=LD+1
 27 IZ=10*IZ+ISGN*ILST
    IF(ILST) 99,39,28
 28 L2=L2+(120+L2*(2-L2*6))/36
    GO TO 39
 29 IF(INC(3,M)/100-4)  30,31,30
 30 LD=LD+9+(ILST-9)*(81-13*(ILST-9))/10
 31 LD=LD+3-ILST
    L2=6
    GO TO 39
 32 L1=4
    L2=1
 33 ISGN=(4-ILST)/3
    GO TO 38
 34 L1=5
    MB=M-1
    J=1
    M=2
    GO TO 7
 35 L1=L1+1
    L2=1
    GO TO 39
36  L2=L2-1
361 J=2
 37 L2=L2+1
 38 L2=L2+1
 39 CONTINUE
    IF((10*L1+L2)/11-3) 98,40,41
 40 INC(4,M)=0
 41 J=1
    IF(L1-5) 42,7,99
 42 Z(M)=IZ
    Z(M)=Z(M)*10.0**(-LD)
    GO TO 6
43  CONTINUE
49  DO 50 I=5,25
    DO 45 J=3,MB
    IF(INC(I,J)) 47,45,47
 45 CONTINUE
51  CONTINUE
 52 N=I-3
    M=MB-2
    M2=M+2
    M1=M+1
    N3=N+3
    N2=N+2
    N1=N+1
    GO TO 150
 47 CONTINUE
53  CONTINUE
 50 CONTINUE
    WRITE (6,109)
```

```
      GO TO 3
   97 WRITE (6,102)
      GO TO 3
   98 WRITE (6,103) I
      WRITE (6,107) INLST
      GO TO 3
   99 WRITE (6,104)
      GO TO 3
  150 DO 151 I=1,M2
      IDD(I)=0
      IST(I)=0
      IJTAG(I)=0
      Z(I)=Z(I+2)
  151 IKTAG(I)=0
  152 IF(INC(2,3)) 153,179,153
  153 DO 154 I=3,M2
      IF(IABS(INC(2,3))-IABS(INC(3,I))) 154,155,154
  155 IJTAG(I-2)=1
      INC(4,I)=INC(1,3)
      GO TO 291
  154 CONTINUE
  179 CONTINUE
      WRITE (6,300)
  300 FORMAT(40HELEMENT NAME IN OUTPUT REQUEST UNDEFINED)
      GO TO 1
  291 IF(INC(2,4)) 279,285,279
  279 DO 280 I=3,M2
      IF(IABS(INC(2,4))-IABS(INC(3,I)))  280,282,280
  282 IKTAG(I-2)=1
      GO TO 285
  280 CONTINUE
      GO TO 99
  285 CONTINUE
      DO 165 I=3,M2
      J=IABS(INC(3,I))/100
      IF(J-3)  156,160,156
  156 IF(J-7)  157,161,157
  157 IF(J-5)  158,162,158
  158 IF(J-4)  159,163,159
  159 IF(J-1)  99,164,99
  160 IDD(I-2)=I-2
      IST(I-2)=-1
      Z(I-2)=1./Z(I-2)
      INC(N3,I)=3
      GO TO 165
  161 IF(INC(4,I)) 174,175,174
  174 INC(N3,I)=6
      DO 182 J=3,M2
      IF(IABS(INC(4,I))-INC(3,J))  182,183,182
  183 IDD(I-2)=J-2
      GO TO 165
  182 CONTINUE
  175 INC(N3,I)=7
      IDD(I-2)=0
      GO TO 165
  162 IDD(I-2)=I-2
```

```
      IST(I-2)=1
      INC(N3,I)=5
      GO TO 165
  163 IDD(I-2)=I-2
      INC(N3,I)=4
      GO TO 165
  164 IF(INC(4,I)) 172,173,172
  172 INC(N3,I)=2
      DO 180 J=3,M2
      IF(IABS(INC(4,I))-INC(3,J))  180,181,180
  181 IDD(I-2)=J-2
      GO TO 165
  180 CONTINUE
  173 INC(N3,I)=1
      IDD(I-2)=0
      GO TO 165
  165 CONTINUE
      DO 171 I=3,M2
      IF(INC(4,I)) 166,171,166
  166 DO 170 J=3,M2
      IF(IABS(INC(4,I))-INC(3,J)) 170,167,170
  167 IF(INC(4,I)) 168,99,169
  168 INC(N3,J)=6
      GO TO 171
  169 INC(N3,J)=2
      GO TO 171
  170 CONTINUE
  171 CONTINUE
      DO 177 I=1,M
  177 INC(1,I+1)=I
      DO 178 I=1,M
      DO 178 J=2,N
  178 INC(J,I+1)=INC(J+3,I+2)
 1178 INC(1,1)=0
 2178 INC(1,M2)=0
 3178 INC(N,1)=0
 4178 INC(N,M2)=0
      DO 1002 I=1,N
      DO 1002 J=1,M2
      INCP(I,J)=INC(I,J)
 1002 CONTINUE
 5178 N=N-1
      M2PER=M2
      NPER=N
      MPER=M
      DO 1001 I=1,M
      IDDP(I)=IDD(I)
      ISTP(I)=IST(I)
      IJTAGP(I)=IJTAG(I)
      IKTAGP(I)=IKTAG(I)
      ZP(I)=Z(I)
 1001 CONTINUE
 1000 RETURN
      END
```

```
0001            FUNCTION ICHAR (NBCD)
0002            INTEGER ALPHAX(64)
0003            DIMENSION NONV(64)
0004            DATA NONV/
          C      0  1  2  3  4  5  6  7  8  9
               170,71,72,73,74,75,76,77,78,79,
               25,5,5,5,5,5,10,
          C      A  B  C  D  E  F  G  H  I
               341,42,43,44,45,46,47,48,49,
          C     I . I I I I -
               75,3,5,5,5,5,20,
          C      J  K  L  M  N  O  P  Q  R
               451,52,53,54,55,56,57,58,59,
          C     I I I I I I BLANK /
               55,5,5,5,5,5,0    ,21,
          C      S  T  U  V  W  X  Y  Z
               662,63,64,65,66,67,68,69 /
          C                0      1      2      3      4      5      6
0005            DATA ALPHAX/4H0    ,4H1    ,4H2    ,4H3    ,4H4    ,4H5    ,4H6    ,
          C      7      8      9    10 11 12 13 14 15 16  17      18      19
               14H7   ,4H8   ,4H9   ,0,0,0,0,0,0,0,4HA   ,4HB   ,4HC   ,
          C      20     21     22     23     24     25     26     27     28
               24HD   ,4HE   ,4HF   ,4HG   ,4HH   ,4HI   ,4HI   ,4H.   ,4HI   ,
          C      29     30     31     32     33     34     35     36     37
               34HI   ,4HI   ,4HI   ,4H-   ,4HJ   ,4HK   ,4HL   ,4HM   ,4HN   ,
          C      38     39     40     41     42     43     44     45     46
               44HO   ,4HP   ,4HQ   ,4HR   ,4HI   ,4HI   ,4HI   ,4HI   ,4HI   ,
          C      47     48     49     50     51     52     53     54
               54HI   ,4H    ,4H/   ,4HS   ,4HT   ,4HU   ,4HV   ,4HW   ,
          C      55     56     57
               64HX   ,4HY   ,4HZ    /
0006            DO 10 J=1,58
0007            IF (NBCD .EQ. ALPHAX(J)) GO TO 11
0008         10 CONTINUE
0009         11 ICHAR=NONV(J)
0010            RETURN
0011            END
```

```
      SUBROUTINE GETCON(ISW)
      COMMON  /INPUT/DUM1(120)
     1,IDD(40),DUM2(40),IST(40),DUM3(40),IJTAG(40),IKTAG(40),Z(40),NE
      COMMON  /LOOPS/INC(25,42),ITSK1(15,4),ITSK2(3,6),ITSK3(15,6),
     1ITSK4(10,5),ITSK5(9,10),INLST(50),INSRT1(80),INSRT2(21,5),M,N,
     2DUM4(3205)
      COMMON /CONV/LCONC(10,40),LPATH(10,40),NLCN(40),ISTART
      COMMON/BINGO/KMOD,NTREE,NLINK,KTREE(10),KLINK(10)
      COMMON/TRIAL/OMEGA(10),KTRIAL
  100 FORMAT(9I2,E12.3)
  101 FORMAT(2I2)
  102 FORMAT(1X,40I2)
  103 FORMAT(1X,/////)
  106 FORMAT(1X,20I3)
  107 FORMAT(2X,9I2,E12.3)
  108 FORMAT(47H ALGORISM FAILURE. REVIEW CIRCUIT CODING RULES.)
      ISW=0
      NE=M
      M1=M+1
      M2=M+2
      N1=N+1
      N2=N+2
      DO 71 II=1,M2
      NLCN(II)=0
      DO 71 JI=1,N2
   71 LCONC(JI,II)=0
    8 CONTINUE
C     END SECTION ONE. INCIDENCE MATRIX.
    9 DO 58 I=1,M
      J=IDD(I)
      IF(I-J) 56,58,56
   56 IF(J) 57,58,57
   57 NLCN(J)=NLCN(J)+1
      LL=NLCN(J)
      LCONC(LL,J)=I
   58 CONTINUE
      I=2
      DO 27 L=1,5
      DO 27 J=2,M1
      IF(INC(N1,J)-L) 27,10,27
   10 IF(INC(I,J)) 11,14,13
   11 DO 12 JI=2,M1
   12 INC(I,JI)=-INC(I,JI)
   13 I1=2
      MEMRY=-1
      GO TO 15
   14 I1=I+1
      MEMRY=+1
   15 IF(I1-N) 16,16,24
   16 DO 23 II=I1,N
   17 IF(INC(II,J)*(II-I)) 18,23,18
   18 IF(MEMRY) 19,28,21
   19 DO 20 JI=2,M1
      MI=INC(I,JI)+INC(II,JI)
   20 INC(II,JI)=((3-MI**2)*MI)/2
      GO TO 17
```

```
         21 DO 22 JI=2,M1
            MI=INC(I,JI)+INC(II,JI)
         22 INC(I,JI)=((3-MI**2)*MI)/2
            GO TO 10
         23 CONTINUE
         24 IF(INC(I,J))28,26,25
         25 INC(I,1)=INC(1,J)
            I=I+1
            IF(I-N1)27,29,28
         26 IF(INC(N1,J)-2)59,59,27
         27 CONTINUE
         28 GO TO 59
         29 CONTINUE
        31  CONTINUE
      C     END SECTION TWO. CUTSET MATRIX.
         32 DO 41 JI=2,M1
            DO 41 II=2,N
            IF(INC(II,1)-INC(1,JI))41,40,41
         40 INC(1,JI)=0
            INC(II,M2)=INC(N1,JI)
         41 CONTINUE
            J=2
         42 IF(INC(1,J))43,44,43
         43 J=J+1
            GO TO 42
         44 J1=J+1
            DO 47 JI=J1,M1
            IF(INC(1,JI))45,47,45
         45 DO 46 II=1,N1
         46 INC(II,J)=INC(II,JI)
            INC(1,JI)=0
            J=J+1
         47 CONTINUE
            DO 48 II=1,N1
         48 INC(II,J)=INC(II,M2)
        33  CONTINUE
      C     END SECTION THREE. COMPRESSED CUTSET MATRIX.
            IF(KTRIAL.GT.1) GO TO 35
            IF(KMOD.GT.1) GO TO 35
            NTREE=N1-2
            NLINK=J-2
            DO 85 I=1,NTREE
         85 KTREE(I)=INC(I+1,1)
            DO 86 I=1,NLINK
         86 KLINK(I)=INC(1,I+1)
         35 JM1=J-1
            N1=N+1
            DO 65 II=2,JM1
            JI=INC(1,II)
            IF(JI-IDD(JI))65,64,65
         64 Z(JI)=1./Z(JI)
            IST(JI)=-IST(JI)
         65 CONTINUE
            DO 55 II=2,N
            DO 55 JI=2,JM1
            IPR=INC(1,JI)
```

```
0107              IPC=INC(II,1)
0108              IF(INC(II,JI))68,55,68
0109           68 IF(IPC-IDD(IPC))67,66,67
          C       ENTER CUT SETS
0110           66 NLCN(IPR)=NLCN(IPR)+1
0111              LL=NLCN(IPR)
0112              LCONC(LL,IPR)=-INC(II,JI)*IPC
          C       ENTER TIE SETS
0113           67 IGN=1
0114           76 IF(IPR-IDD(IPR))55,52,55
0115           52 IF(INC(II,J)-2)55,75,54
0116           75 IF(IPC-IDD(IPC))51,54,51
0117           51 IGN=-IGN
0118           54 NLCN(IPC)=NLCN(IPC)+1
0119              LL=NLCN(IPC)
0120              LCONC(LL,IPC)=IGN*INC(II,JI)*IPR
0121           55 CONTINUE
0122          36  CONTINUE
0123           37 DO 81 I=1,M
0124              N=NLCN(I)
0125              LCONC(N+1,I)=0
0126              IF(N-1) 81,81,77
0127           77 NM1=N-1
0128              DO 80 J=1,NM1
0129              KK=N-J
0130              DO 79 K=1,KK
0131              L1=LCONC(K,I)
0132              L2=LCONC(K+1,I)
0133              IF(IABS(L1)-IABS(L2)) 79,29,78
0134           78 LCONC(K,I)=L2
0135              LCONC(K+1,I)=L1
0136           79 CONTINUE
0137           80 CONTINUE
0138           81 CONTINUE
0139          38  CONTINUE
0140           39 RETURN
0141           59 WRITE (6,108)
0142           60 ISW=1
0143              RETURN
0144              END
```

```
 0001              SUBROUTINE FSORL
            C   *  FINDS ALL FIRST ORDER LOOPS BY THE METHOD OF SERBAGI AND WATERS
 0002              COMMON /CONV/LCONC(10,40),LPATH(10,40),NLCN(40),ISTART
 0003              COMMON/ORDER/IVOE(40),IVTE(40),IELORD(40),IKN(40),IEKN(40),
                  1  IUN(40),IPATH(40),JPATH(40),IAA(40),IAAA(40)
 0004              COMMON/INPUT/IVO(40),IVT(40),ICO(40),IDD(40),IEL(40),IST(40),
                  1  IGE(40),IJTAG(40),IKTAG(40),Z(40),NE
 0005              COMMON /LOOPS/VLOOP(927),ISLOOP(927),JLOOP(927),KLOOP(927),
                  1LOOP(927),JLP(40),ISLP(40),VLP(40),KLP(40),NLOOP,SIGNC,K,DUM(2)
            C      PRINT 25
            C   25 FORMAT(1H1,39H FIRST ORDER LOOPS BY CONSECUTIVE NODES)
 0006              NLOOP=0
 0007              NPINS=NE
 0008              NPINM1=NPINS-1
 0009              DO 4 IP=1,NPINM1
 0010              IF(LCONC(1,IP).EQ.0)  GO TO 24
 0011              ISTART=IP
 0012              K1=IP
 0013              K2=1
 0014            5 CALL REPACK(K1,K2,ITRIG,2)
 0015            6 IF(ITRIG)9,8,7
 0016            7 K1=ITRIG
 0017              K2=K2+1
 0018              GO TO 5
 0019            8 K2=K2-1
 0020              IF(K2.EQ.0)  GO TO 20
 0021           81 CALL REPACK(K2,IDUM,ITRIG,1)
 0022              GO TO 6
            C      THIS IS APATH RECORD IT
 0023            9 NLOOP=NLOOP+1
 0024              IF(NLOOP.LE.927)  GO TO 27
 0025              WRITE (6,26)
 0026              STOP
 0027          26  FORMAT(39H FLOWGRAPH FIRST ORDER LOOPS EXCEED 927)
 0028          27  CONTINUE
 0029              DO 11 I=1,K2
 0030           11 IPATH(I+1)=LPATH(1,I)
 0031              IPATH(1)=ISIGN(ISTART,LPATH(1,K2))
 0032              K2P1=K2+1
            C   12 FORMAT(I5,5X,19I5)
            C      WRITE(6,12)NLOOP,(IPATH(I),I=1,K2P1)
 0033              CALL VISJ(K2,NLOOP)
 0034              GO TO 81
            C
            C      HAVE COMPLETED ALL PATH FROM ONE PIN
            C      DELETE IT FROM THE SET
            C
 0035          20  IPP=NLCN(IP)
 0036              DO 21 I=1,IPP
 0037           21 LCONC(I,IP)=0
 0038          24  IPP1=IP+1
 0039              DO 22 J=IPP1,NPINS
 0040              IF(LCONC(1,J).NE.IP)  GO TO 22
            C
            C      THIS ONE IS BEING DELETED
            C
```

```
      IPP=NLCN(J)-1
      DO 23 K=1,IPP
   23 LCONC(K,J)=LCONC(K+1,J)
      LCONC(IPP+1,J)=0
   22 CONTINUE
    4 CONTINUE
      RETURN
      END
```

```
            SUBROUTINE REPACK (L1,L2,ITRIG,IWAY)
C     **********************************************************************
C     *           LCONC HAS LIST OF CONNECTION. COLUMN 1,THOSE FROM 1
C     *           COLUMN 2,THOSE FROM 2
C     *           IT IS ASSUMED THERE ARE AT MOST 20 EXITS FROM ANY ONE PIN
C     *           THE NUMBERS IN ANY COLUMN MUST BE IN ASCENDING ORDER
C     *           LPATH WILL BE USED TO GENERATE THE PATH
C     *           THIS ROUTINE WILL REPACK A COLUMN, BY DROPPING FIRST ELEMEN
C     *           IF EQUAL TO ONE OF THE PRECEEDING ELEMENTS IN FIRST ROW WIL
C     *           DROP IT ALSO AND REPEAT
C     *           ITRIG=0 NO MORE IN THAT COLUMN
C     *           ITRIG=-1 FOUND STARTING NUMBER
C     *           ITRIG .GT. 0, NEW VALUE
C     **********************************************************************
            COMMON /CONV/LCONC(10,40),LPATH(10,40),NLCN(40),ISTART
            IF(IWAY.EQ.2) GO TO 17
            L=L1
         10 DO 11 I=1,9
            LPATH(I,L)=LPATH(I+1,L)
            IF(LPATH(I,L).EQ.0) GO TO 16
         11 CONTINUE
         16 LPATH(I+1,L)=0
         15 JTRIG=IABS(LPATH(1,L))
            IF(JTRIG.EQ.0) GO TO 14
            IF(JTRIG.EQ.ISTART) GO TO 13
C                 SINCE NOT ZERO  OR ISTART HAS IT APPEARED BEFORE
            IF (L .EQ.1) GO TO 14
            LM1=L-1
            DO 12 J=1,LM1
            IF (IABS(LPATH(1,J)) .EQ. JTRIG) GO TO 10
         12 CONTINUE
            GO TO 14
         13 JTRIG=-1
         14 ITRIG=JTRIG
            RETURN
C
C           ENTRY SHIFT
         17 CONTINUE
C                 THIS ROUTINE WILL MOVE THE L1 COLUMN OF LCONC TO THE L2
C                 COLUMN OF LPATH
C                 AFTER THIS IS DONE IT WILL TEST THE FIRST VALUE
            L=L2
            IP=NLCN(L1)+1
            DO 21 I=1,IP
            LPATH(I,L)=LCONC(I,L1)
         21 CONTINUE
            GO TO 15
            END
```

```
0001            SUBROUTINE VISJ(N,NN)
          C  *  CALCULATES   VLOOP,ISLOOP,JLOOP,KLOOP,LOOP
          C  *  IPATH CONTAIN THE PATH ELEMENTS, N IS THE NUMBER OF ELEMENTS,
          C  *  NN IS THE PATH NUMBER
0002            COMMON /LOOPS/VLOOP(927),ISLOOP(927),JLOOP(927),KLOOP(927),
               1LOOP(927),JLP(40),ISLP(40),VLP(40),KLP(40),NLOOP,SIGNC,K,DUM(2)
0003            COMMON/ORDER/IVOE(40),IVTE(40),IELORD(40),IKN(40),IEKN(40),
               1  IUN(40),IPATH(40),JPATH(40),IAA(40),IAAA(40)
0004            COMMON/INPUT/IVO(40),IVT(40),ICO(40),IDD(40),IEL(40),IST(40),
               1  IGE(40),IJTAG(40),IKTAG(40),Z(40),NE
0005            DIMENSION IPOWER(40)
0006            DATA IPOWER/1,2,4,8,16,32,64,128,256,512,1024,2048,
               14096,8192,
               116384,32768,65536,131072,262144,524288,1048576,2097152,4194304,
               28388608,16777216,33554432,67108864,134217728,268435456,536870912,
               31073741824/
0007            K=IPATH(1)
0008            KK=IABS(K)
0009            J=KK
0010            ZZ=Z(J)
0011            II=IST(J)
0012            IK=0
0013            JP=IPOWER(KK)
0014            JLOOP(NN)=0
0015            ASSIGN 6 TO NNN
0016            IF(IJTAG(J))1,2,1
0017          1 ASSIGN 8 TO NNN
0018            JLOOP(NN)=1
0019          2 CONTINUE
0020            IF(IKTAG(J))4,5,4
0021          4 IK=IK+1
0022          5 CONTINUE
0023            DO 10 I=2,N
0024            K=K*IPATH(I)
0025            KK=IABS(IPATH(I))
0026            J=KK
0027            ZZ=ZZ*Z(J)
0028            II=II+IST(J)
0029            JP=JP+IPOWER(KK)
0030            GO TO NNN,(6,8)
0031          6 IF(IJTAG(J)) 8,8,7
0032          7 ASSIGN 8 TO NNN
0033            JLOOP(NN)=1
0034          8 IF(IKTAG(J))10,10,9
0035          9 IK=IK+1
0036         10 CONTINUE
0037            ZK=K
0038            VLOOP(NN)=ZZ*SIGN(1.,ZK)
0039            ISLOOP(NN)=II
0040            LOOP(NN)=JP
0041            KLOOP(NN)=IK
0042            IF(IK)11,12,11
0043         11 ILAG=1
0044         12 CONTINUE
0045            RETURN
0046            END
```

```
      SUBROUTINE  FLGRPH
      COMMON /LOOPS/VLOOP(927),ISLOOP(927),JLOOP(927),KLOOP(927),
     1LOOP(927),JLP(40),ISLP(40),VLP(40),KLP(40),NLOOP,SIGNC,K,DUM(2)
      COMMON /POLY/ SCON(81),SJAY(81),SKAY(81),SJKZ(81),SJZKO(81),
     1SJOKZ(81),SJKO(81)
      JS=41-ISLP(K)
      VV=VLP(K)*SIGNC
      IF(JLP(K)) 20,20,24
   20 SCON(JS)=SCON(JS)+VV
      IF(KLP(K)) 21,21,22
   21 SJKZ(JS)= SJKZ(JS)+VV
      GO TO 29
   22 SJZKO(JS)=SJZKO(JS)+VV
  23  SKAY(JS)=SKAY(JS)+VV
      GO TO 29
   24 SJAY(JS)=SJAY(JS)+VV
      IF(KLP(K)) 25,25,26
   25 SJOKZ(JS)=SJOKZ(JS)+VV
      GO TO 29
   26 SJKO(JS)=SJKO(JS)+VV
      GO TO 23
   29 CONTINUE
      RETURN
      END
```

```
      FUNCTION AND(K,M)
      IP=M
        IQ=K
    6     P=IP
          P=P/2.
       IP=P
        Q=IQ
        Q=Q/2.
        IQ=Q
       IF((P.NE.IP).AND.(Q.NE.IQ)) GO TO 7
        IF((IP.EQ.0).OR.(IQ.EQ.0)) GO TO 8
        GO TO 6
    7 AND=1
        RETURN
    8 AND=0
          RETURN
          END
```

```
0001          SUBROUTINE TF(FF,ANN,VALUE)
         C    THIS SUBROUTINE EVALUATES THE POLYNOMIAL AT FREQUENCY FF
0002          DIMENSION ANN(1)
0003          COMPLEX TEMP1,TEMP2,TEMP3,FF,VALUE
0004          TEMP1=(0.,0.)
0005          TEMP2=(0.,0.)
0006          TEMP3=(0.,0.)
0007          DO 10 I=1,15
0008          K=8-I
0009          IF(K) 2,3,4
0010        4 TEMP1=(TEMP1+CMPLX(ANN(I),0.))*FF
0011          GO TO 10
0012        3 TEMP2=CMPLX(ANN(8),0.)
0013          GO TO 10
0014        2 TEMP3=(TEMP3+CMPLX(ANN(16+K),0.))/FF
0015       10 CONTINUE
0016          VALUE=TEMP1+TEMP2+TEMP3
0017          RETURN
0018          END
```

```
0001                SUBROUTINE SF(FF,ANA,VALUE,J)
           C        THIS SUBROUTINE EVALUATES THE SENSITIVITY FUNCTION
0002                DIMENSION ANA(15,10)
0003                COMPLEX FF,TEMP1,TEMP2,TEMP3,VALUE
0004                TEMP1=(0.,0.)
0005                TEMP2=(0.,0.)
0006                TEMP3=(0.,0.)
0007                  DO 10 I=1,15
0008                K=8-I
0009                IF(K)2,3,4
0010              4  TEMP1=(TEMP1+CMPLX(ANA(I,J),0.))*FF
0011                GO TO 10
0012             3  TEMP2=CMPLX(ANA(8,J),0.)
0013                GO TO 10
0014              2 TEMP3=(TEMP3+CMPLX(ANA(16+K,J),0.))/FF
0015             10 CONTINUE
0016                VALUE=TEMP1+TEMP2+TEMP3
0017                RETURN
0018                END
```

```
0001              SUBROUTINE LEQ(A,N)
0002              DIMENSION A(10,10)
0003              DIMENSION IPIVOT(64),INDEX(64,2)
0004              DIMENSION PIVOT(64)
0005              EQUIVALENCE (IROW,JROW), (ICOLUM,JCOLUM), (AMAX, T, SWAP)
            C
            C     INITIALIZATION
            C
0006          10  DETERM=1.0
0007          15  DO 20 J=1,N
0008          20  IPIVOT(J)=0
0009          30  DO 550 I=1,N
            C
            C     SEARCH FOR PIVOT ELEMENT
            C
0010          40  AMAX=0.0
0011          45  DO 105 J=1,N
0012          50  IF (IPIVOT(J)-1) 60, 105, 60
0013          60  DO 100 K=1;N
0014          70  IF (IPIVOT(K)-1) 80, 100, 740
0015          80  IF(ABS(AMAX)-ABS(A(J,K)))85,100,100
0016          85  IROW=J
0017          90  ICOLUM=K
0018          95  AMAX=A(J,K)
0019         100  CONTINUE
0020         105  CONTINUE
0021         110  IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
            C
            C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
            C
0022         130  IF (IROW-ICOLUM) 140, 260, 140
0023         140  DETERM=-DETERM
0024         150  DO 200 L=1,N
0025         160  SWAP=A(IROW,L)
0026         170  A(IROW,L)=A(ICOLUM,L)
0027         200  A(ICOLUM,L)=SWAP
0028         260  INDEX(I,1)=IROW
0029         270  INDEX(I,2)=ICOLUM
0030         310  PIVOT(I)=A(ICOLUM,ICOLUM)
0031         320  DETERM=DETERM*PIVOT(I)
            C
            C     DIVIDE PIVOT ROW BY PIVOT ELEMENT
            C
0032         330  A(ICOLUM,ICOLUM)=1.0
0033         340  DO 350 L=1,N
0034         350  A(ICOLUM,L)=A(ICOLUM,L)/PIVOT(I)
            C
            C     REDUCE NON-PIVOT ROWS
            C
0035         380  DO 550 L1=1,N
0036         390  IF(L1-ICOLUM) 400, 550, 400
0037         400  T=A(L1,ICOLUM)
0038         420  A(L1,ICOLUM)=0.0
0039         430  DO 450 L=1,N
0040         450  A(L1,L)=A(L1,L)-A(ICOLUM,L)*T
0041         550  CONTINUE
```

```
              C
              C        INTERCHANGE COLUMNS
              C
 0042          600     DO 710 I=1,N
 0043          610     L=N+1-I
 0044          620     IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630
 0045          630     JROW=INDEX(L,1)
 0046          640     JCOLUM=INDEX(L,2)
 0047          650     DO 705 K=1,N
 0048          660     SWAP=A(K,JROW)
 0049          670     A(K,JROW)=A(K,JCOLUM)
 0050          700     A(K,JCOLUM)=SWAP
 0051          705     CONTINUE
 0052          710     CONTINUE
 0053          740     RETURN
 0054                  END
```