INTERNAL REPORTS IN

SIMULATION, OPTIMIZATION AND CONTROL

No. SOC-69

CANOP2 - INTERACTIVE CASCADED NETWORK OPTIMIZATION PACKAGE

J.W. Bandler and J.R. Popović

December 1974

FACULTY OF ENGINEERING
McMASTER UNIVERSITY
HAMILTON, ONTARIO, CANADA

CANOP2 - INTERACTIVE CASCADED NETWORK OPTIMIZATION PACKAGE

PURPOSE: The program analyzes and optimizes, interactively or by batch processing, cascaded, linear, time-invariant networks in the frequency domain consisting of typical two-port elements, including resonant circuits, transmission-line elements, and microwave C- and D-sections.

AUTHORS: J.W. Bandler and J.R. Popović
Group on Simulation, Optimization and Control
Faculty of Engineering
McMaster University, Hamilton, Canada, L8S 4L7

LANGUAGE: FORTRAN IV

CORE REQUIREMENT: About 17 K_{10} or 41 K_8 words .

AVAILABILITY: This report includes a user's manual with examples and program listings.

DESCRIPTION:

The cascaded network optimization package called CANOP2 will analyze and optimize cascaded,linear, time-invariant networks in the frequency domain. It is an updated version of CANOPT [1,2] and, in short, it accomodates more two-port elements, plots responses, and enforces equality on the variable parameters, if desired. In general, the package features some of the latest and most efficient methods of computer-aided design currently available. The program is organized in such a way that future additions or deletions of performance specifications, constraints, optimization methods and circuit elements are readily implemented. Presently, the network to be optimized is assumed to be a cascade of two-port building blocks terminated in a unit normalized, frequency-independent resistance at the source and a user-specified frequency-independent resistance at the load.

A variety of two-port lumped and distributed elements such as resistors, inductors, capacitors, lossless transmission lines, lossless short-circuited and open-circuited transmission-line stubs, series and parallel LC and RLC resonant circuits and microwave allpass C- and D-sections can be handled. Upper and lower bounds on all relevant parameters can be specified by the user. A generalized least pth objective function, or sequence of least pth objective functions, developed by

Bandler and Charalambous [3] incorporating simultaneously input reflection coefficient, insertion loss, relative group delay and parameter constraints (if any) are automatically created. Constraints are treated by the objective function in essentially the same way as the performance specifications [1]. To distinguish conveniently between the various responses or constraint functions a scheme for interval translation and introduction of artificial points has been developed [1]. The Fletcher method of minimizing unconstrained functions of many variables [4] is available to the user. (The Fletcher-Powell method [5] used in CANOPT was found to be reliable but much slower for network optimization than the Fletcher method, therefore only the latter one is used in this package.) The package was designed to incorporate the adjoint network method of sensitivity evaluation to produce accurate first derivatives needed by these efficient gradient minimization methods [6].

If equality (symmetry) of some parameters can be predicted, symmetry may be forced throughout the optimization. Results may be automatically presented numerically and graphically and analysis of different responses and/or different frequency ranges may be performed at the user's discretion and a new optimization may be requested. A summary of the latest features and options available is given in Table I.

The package written in FORTRAN IV was originally developed for batch processing on a CDC 6400 computer and has been largely extended for use on INTERCOM. The user may interact at many points with the program to change parameters, frequency range, types and options and to request plots. The interactive user enters his data in free format. This report is a manual for both the batch and interactive versions.

The package CANOP2 will analyze and optimize only a cascade connection of the two-port elements listed in Tables II and III. Elements 1 to 19 may be connected in any order (sequentially from the source to the load) using as many as required or as many as the computer being used can accomodate.

The first six elements are one-parameter lumped elements. Their parameter values should be normalized by the user to his center frequency and source resistance, appropriately, as outlined in APPENDIX A.

The next four elements are three-parameter tuned circuits. They are characterized by resonant frequency, quality factor, and slope reactance or susceptance,

TABLE I

SUMMARY OF FEATURES, OPTIONS AND PARAMETERS REQUIRED

Features	Type	Options	Parameters
Objective Functions	Least pth	$1<p<\infty$	Value of p for each of a specified number of optimizations Artificial margin Difference in objective functions for termination
Performance Specifications and Parameter Constraints	Upper (+1.) Lower (-1.) Single (0.)	Reflection coefficient (1) Insertion loss (2) Group delay (3) Parameter value (0)	Normalization frequency Number of points and constraints — Number of bands or intervals For each: Specification/constraint Weighting factor Type Option Frequency (sample point) or parameter — Lower and uppe frequencies (band edges) Number of subintervals
Analysis Optimization	 Gradient	Analysis only (0) Fletcher optimization method (1)	Option Specified or default values for: Number of iterations allowed Estimate of lower bound on objective function Test quantities for termination
Circuit Elements	Cascaded Two-port	See Tables II and III	Number of elements Sequence of code numbers Parameter values Indicator for fixed, variable or equal (symmetrical) parameters Load resistance Parameters for C- and D-sections
Graph	Frequency response	Given response Other response Any frequency range Automatic scaling Specified scaling	As many plots as desired Option Frequency (sample point) — Lower and uppe frequencies (band edges)

TABLE II

ELEMENTS AND CODE NUMBERS

Element	Connection	Code	Parameters
inductor	series	1	inductance
	shunt	4	
capacitor	series	3	capacitance
	shunt	2	
resistor	series	5	resistance
	shunt	6	
resonant RLC circuit	series	7	resonant frequency quality factor slope reactance
	shunt	10	
antiresonant RLC circuit	series	9	antiresonant frequency quality factor slope susceptance
	shunt	8	
resonant LC circuit	series	16	resonant frequency slope reactance
	shunt	19	
antiresonant LC circuit	series	18	antiresonant frequency slope susceptance
	shunt	17	
lossless transmission line	series shorted	11	length characteristic impedance
	shunt shorted	14	
	series open	13	
	shunt open	12	
	cascade	15	

TABLE III

ALLPASS SECTIONS

Parameters	
All fixed or all variable (determined by one indicator)	Fixed
location of real zeros of C-sections	number of C-sections
location of real parts of zeros of D-sections location of imaginary parts of zeros of D-sections	number of D-sections
delay level	cutoff frequency

as appropriate. The last four elements are two-parameter tuned circuits. They are characterized also by resonant frequency and slope reactance or susceptance. Normalization as before must again be carried out by the user.

Elements 11 to 15 are two-parameter lossless transmission-line components. All are characterized by normalized length and characteristic impedance (see APPENDIX A).

The allpass sections (Table III) are treated in the same way as, for example, Kudsia [7]. Group delay relative to delay level in nsec is calculated.

The source and load are real constant resistances, the source being assumed to be unity.

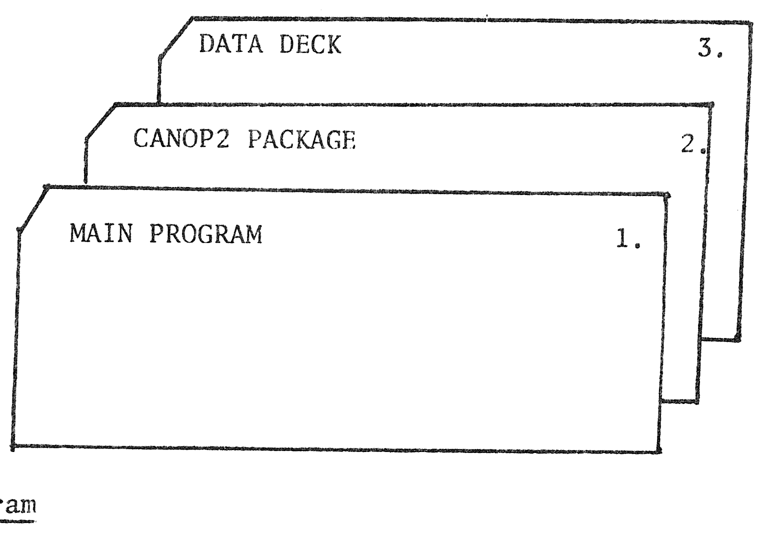
HOW TO USE THE BATCH VERSION OF CANOP2

The package, which consists of 1621 cards, requires the CDC system routine SECOND which keeps track of elapsed time. For a different system the cards A280, A295, D8 and D48 should be replaced by cards appropriate to the system or removed together with cards A305 and D49.

Data cards No. A22 and A23 may also have different forms for different systems, and should be replaced by appropriate cards.

A core requirement of about 41 K_8 or 17 K_{10} is sufficient to optimize, e.g., a seven parameter problem using 25 sample points.

Set up the input deck as follows:



1. Main Program

Write the main program as indicated below:

```
PROGRAM TST (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT)
```

Dimension the following arrays:

DIMENSION A1(NE),A2(NE), A3(J), A4(NINT), A5(NINT), A6(3,NINT), A7(N), A8(N), A9(N), A10(N), A11(N), A12(N), A13(K), A14(K), A15(K), A16(K), A17(N*K), A18(K(K+7)/2) IA1(MM+NE), IA2(NINT), IA3(JJ)

where

NE	is the total number of parameters in all elements.
$J \triangleq 2*ND+NC+1$	where NC and ND are the numbers of C- and D-sections, respectively.
$NINT \triangleq NINTD+NINTS$	where NINTD is the number of frequency bands or intervals and NINTS is number of other frequency points and constraints. Also double those intervals with single or both upper and lower specifications.
N	is the total number of frequency points to be used (those counted twice when single or both upper and lower specifications are used for the same frequency points).
K	is the total number of variable parameters.
MM	is the number of elements in the circuit.

$JJ \triangleq max[NINT,NE]$.

Call the subroutine CANOP2 as follows:

```
CALL CANOP2 (A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13, A14,
        A15, A16, A17, A18, IA1, IA2, IA3)
```

Add

```
CALL EXIT
END
```

A user may also use a standard main program listed in APPENDIX E and stored in permanent file of CDC 6400 at McMaster University by control card

ATTACH, ℓfn_1, CANOP2M, ID=*******.*

2. The CANOP2 Package

A listing is contained in APPENDICES F & G.

It is also stored in permanent file of CDC 6400 at McMaster University and may be attached by two control cards:

ATTACH, ℓfn_2, CANOP2B, ID=*******.

ATTACH, ℓfn_3, CANOP2C, ID=*******.

*ℓfn stands for logical file name of no more than 7 characters. It should be the same in the ATTACH command and in the LOAD command.

3. Data Deck

Parameters to be supplied as data are defined below:

MM	The number of elements in the circuit, not including C and D sections. Set to 0 if you do not want any.
IC(I),I=1,MM	A sequence of code numbers of elements which specify the order in which the elements are sequentially connected from source to load. (See Table II for element numbers).
AA(I),I=1,NE	Values of parameters in the circuit including starting values for variables. (The total number of parameters is NE).
IC(MM+I),I=1,NE	Indicates whether the parameters in the circuit are fixed or variable. Set to 0 if fixed and a positive integer if variable such that the numerical integer value indicates whether the variable is new (increase integer value by one) or repeated (set its integer to the integer value of a twin variable). (For example, the following sequence: 0,1,0,2,0,2,3,0,1 indicates that four parameters are fixed and five variable, among which three are independent, the last is forced to have the same value as second, and sixth as fourth.)
NC	The number of C sections. Set to 0 if you do not want any.
ND	The number of D sections. Set to 0 if you do not want any.
KVR	Denotes whether C and D section parameters are variable or fixed. Set to 1 if variable and 0 if fixed.
AB(I),I=1,J	Values of the parameters of the C and D sections and d level. (See Kudsia [7]). The d level is treated like any other circuit parameter.
R	The load resistance.
NINTD	The number of frequency bands or intervals.
NINTS	The number of other frequency points and constraints.
XX(1,I),I=1,NINT where NINT≜NINTD+NINTS	The lower frequency bounds (band edges) for bands, single frequency points, parameters to be constrained (artificial frequency points).
XX(2,J),J=1,NINTD	The upper frequency bounds (band edges) for bands.
NUMB(I),I=1, NINTD	The number of subintervals (equals sample points minus one).

FUN(I),I=1,NINT	A sequence of numbers to be used as specifications or constraints.
XX(3,I),I=1,NINT	Indicates whether a specification or constraint for any given I is an upper or lower one. Set to 1. for upper, to -1. for lower and to 0. for single.
WT(I),I=1,NINT	The weighting factors (positive). Set to 1. if unsure.
IOBJ(I),I=1,NINT	The approximating function: For reflection coefficient set to 1. For insertion loss (dB) set to 2. For group delay (nsec) set to 3*. For parameter constraints set to 0.
FM	The center frequency (e.g., in MHz, for normalization).
WC	The cut-off frequency for C- and D-sections (e.g., in MHz).
MET	Indicator for analysis or optimization. Set to 0 if only analysis is required. Set to 1 for optimization method to be used (Fletcher).
IDEF	Set to 0 if default values for the optimization are not to be used, otherwise specify differently.
MAX	The maximum number of allowable iterations (e.g., 100).
IPRINT	Intermediate output is printed out after every specified number of iterations. Set to 0 if no intermediate output is desired.
EPS(I),I=1,K	The small quantities for testing convergence in the Fletcher method (e.g., 10^{-4}). K is the number of unequal variable parameters.
EST	A realistic under-estimate (lower bound) of the value of the objective function.
DIF	The value of the difference between objective function values in successive optimizations. Set to 0. if not sure.

---

*Group delay is calculated in nsec if the frequencies are given in MHz.

KSI — A small quantity by which specifications could be shifted artificially [1]. Set to 0. if not sure.

IPA(I),I=1,ITER — Vector containing the values of p (positive integer, greater than one) to be used successively in a total of ITER complete optimizations.

Table IV shows the arrangement of the data deck.

If the parameter IDEF is different from zero, the following default values will be used, and need not to be included in data deck:

MAX = 100

IPRINT = 0

EPS(I) = 10^{-4} for all I's

EST = -0.1

DIF = 0

KSI = 0

ITER = 1

IPA(1) = 2

INPUT-OUTPUT EXAMPLE

A test example will be presented here to illustrate the approach. The same one with more extensive input and output will be presented for the interactive version.

We consider a seven-section equal-ripple band-pass microwave filter of 3 to 1 bandwidth (ratio of upper band edge to lower band edge) consisting of two unit elements and five stubs which has been previously considered by Horton and Wenzel [8] as represented in Fig. 1. The terminations of the filter are unity. The filter is to have a 0.1 dB ripple in the passband, from 1.0875 to 3.2625 GHz, and an attenuation above 50 dB at frequency points 0.6 and 3.75 GHz in the stop-band. All section lengths were kept fixed at normalized values of 1, and the normalized characteristic impedances are used as variables. The starting value of the variable vector (see Fig. 1) was $\underset{\sim}{Z}_0 = [.63 \ .33 \ 1.27 \ .26 \ 1.27 \ .33 \ .63]^T$. 21 uniformly spaced sample points were used in the passband. The weighting is set to be 1 everywhere and the default values are used for the optimization. Symmetry of the variable characteristic impedances is taken into account. The user's written main program and data deck are shown in Fig. 2. The typical output for analysis only is shown in Fig. 3. A plot is obtained automatically. Effort was made to keep both input and output as similar as possible for the

TABLE IV - THE ARRANGEMENT OF THE DATA DECK

Condition	Number of cards	Parameters	Type	Format
-	1	MM	INTEGER	I10
MM > 0	As many as required	IC(I),I=1,MM	INTEGER	8I10
MM > 0	As many as required	AA(I),I=1,NE	REAL	5E16.8
MM > 0	As many as required	IC(MM+I),I=1,NE	INTEGER	8I10
-	1	NC,ND	INTEGER	2I10
NC≠0 & ND≠0	1	KVR	INTEGER	I10
NC≠0 & ND≠0	As many as required	AB(I),I=1,J	REAL	5E16.8
-	1	R	REAL	E16.8
-	1	NINTD,NINTS	INTEGER	2I10
NINTD≠0	As many as required by NINTD	XX(1,I),XX(2.I).FUN(I),WT(I),XX(3,I), IOBJ(I),NUMB(I)	5 REALS AND 2 INTEGERS	4E16.8,F6.0, 2I5
NINTS≠0	As many as required by NINTS	XX(1,I),FUN(I),WT(I),XX(3,I), IOBJ(I)	4 REALS and 1 INTEGER	4E16.8, I6
-	1	FM,WC	REAL	2E16.8
-	1	MET,IDEF	INTEGER	2I10
MET=0	No more cards are required	Analysis is required only	-	-
IDEF≠0	No more cards are required	Default values are used for optimization	-	-
MET=1	1	MAX,IPRINT	INTEGER	2I10
-	As many as required	EPS(I),I=1,K	REAL	5E16.8
-	1	EST,DIF,RSI	REAL	3E16.8
-	1	ITER	INTEGER	I10
-	As many as required	IPA(I),I=1,ITER	INTEGER	8I10

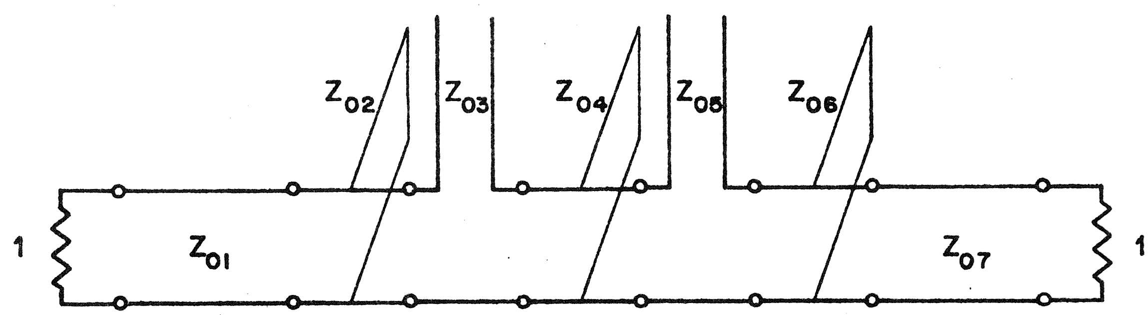


Fig. 1. Seven-section band-pass filter example.

```
      PROGRAM TST (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C              M A I N    P R O G R A M
C
      DIMENSION A1(14),A2(14),A3(1),A4(3),A5(3),A6(3,3),A7(25),A8(25),
     1 A9(25),A10(25),A11(25),A12(25),A13(7),A14(7),A15(7),A16(7),
     2 A17(175),A18(49),IA1(21),IA2(3),IA3(14)
      CALL CANOP2 (A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12,A13,A14,A15,
     + A16,A17,A18,IA1,IA2,IA3)
      CALL EXIT
      END


                               CDTOT    0011
```

```
    7
   15         14          13          14         13          14          15
    1.           0.63                  1.           0.33                   1.
 1.27               1.           0.26                           1.           1.27
 1.              0.33                  1.                         0.63
    0          1          0            2          0             3         0          4
    0          3          0            2          0             1
    0          0
  1.
    1          2
   1087.5       3262.5            0.1              1.               1.        2      20
600.            50.             1.              -1.                2
 3750.          50.             1.              -1.                2
  2175.
    1          1


                               CDTOT    0015
```

Fig. 2. Main program and data deck for the example.

INPUT DATA

NUMBER OF ELEMENTS 7

THE CALCULATED NUMBER OF PARAMETERS 14

CODE NUMBER	PARAMETER NUMBER	PARAMETER VALUE	PARAMETER CONDITION
15	1	.10000000E+01	FIXED
15	2	.63000000E+00	VARIABLE
14	3	.10000000E+01	FIXED
14	4	.33000000E+00	VARIABLE
13	5	.10000000E+01	FIXED
13	6	.12700000E+01	VARIABLE
14	7	.10000000E+01	FIXED
14	8	.26000000E+00	VARIABLE
13	9	.10000000E+01	FIXED
13	10	.12700000E+01	VARIABLE
14	11	.10000000E+01	FIXED
14	12	.33000000E+00	VARIABLE
15	13	.10000000E+01	FIXED
15	14	.63000000E+00	VARIABLE

NUMBER OF C SECTIONS 0

NUMBER OF D SECTIONS 0

LOAD RESISTANCE .10000000E+01

NUMBER OF FREQUENCY INTERVALS 1

NUMBER OF FREQUENCY POINTS 2

LOWER FREQUENCY	UPPER FREQUENCY	NUMBER OF SUBINTERVALS	SPECIFICATION		TYPE	WEIGHTING FACTOR
.10875000E+04	.32625000E+04	20	.10000000E+00	INSERTION LOSS	UPPER	.10000000E+01

FREQUENCY	SPECIFICATION		TYPE	WEIGHTING FACTOR
.60000000E+03	.50000000E+02	INSERTION LOSS	LOWER	.10000000E+01
.37500000E+04	.50000000E+02	INSERTION LOSS	LOWER	.10000000E+01

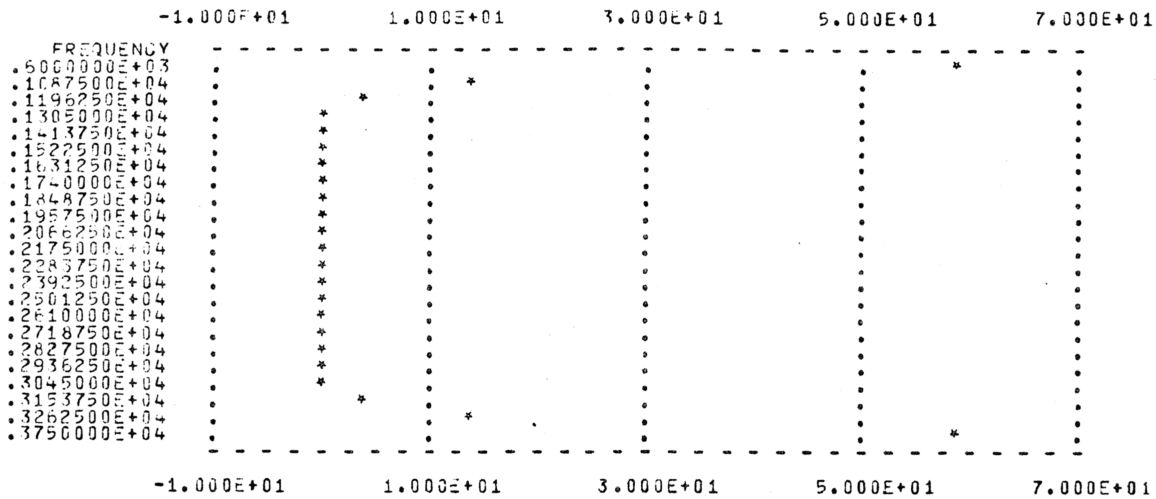
THE CALCULATED TOTAL NUMBER OF INTERVALS 3

CENTER FREQUENCY .21750000E+04

RESPONSE AT THE STARTING POINT

FREQUENCY	INSERTION LOSS
.10875000E+04	.13524955E+02
[illegible]	.37772373E+01
[illegible]	.24539809E+00
[illegible]	.51137893E-03
.15225000E+04	.78165992E-01
.16312500E+04	.25444966E+00
.17400000E+04	.44370802E+00
.18487500E+04	.49699340E+00
.19575000E+04	.35268927E+00
.20662500E+04	.11687159E+00
.21750000E+04	-.12301271E-12
.22837500E+04	.11687159E+00
.23925000E+04	.35268927E+00
.25012500E+04	.49699340E+00
.26100000E+04	.44370802E+00
.27187500E+04	.25444966E+00
.28275000E+04	.78165992E-01
.29362500E+04	.51137893E-03
.30450000E+04	.24539809E+00
.31537500E+04	.37772373E+01
.32625000E+04	.13524955E+02
.60000000E+03	.58882108E+02
.37500000E+04	[illegible]

Fig. 3. Typical output for analysis only.



SOME DETAILS FROM THE ABOVE GRAPH

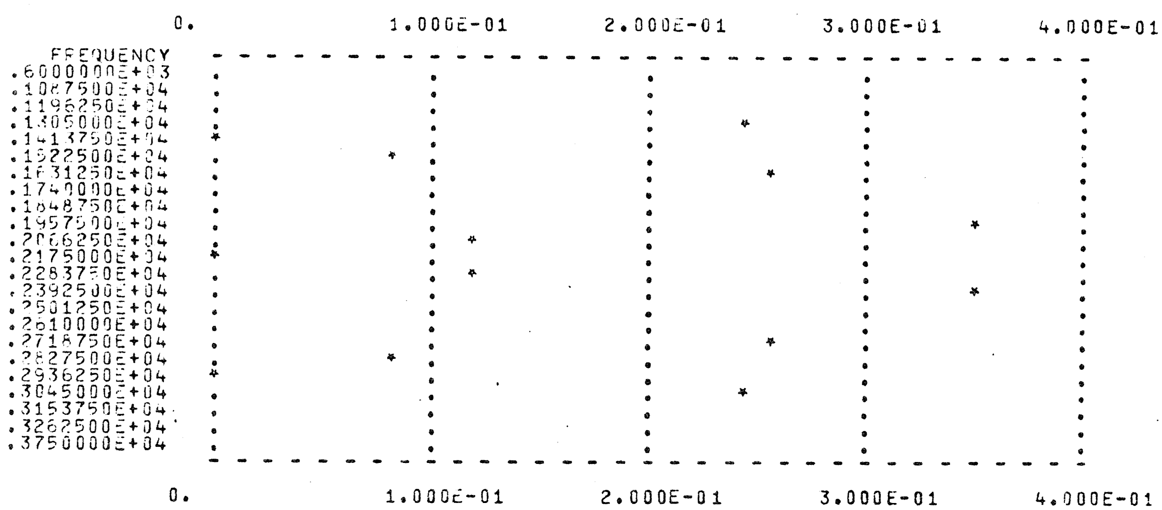


Fig. 3. [Continued]

same desired features as CANOPT. For the complete, although not plotted, output when symmetry was not considered in the optimization refer to [9]. (Note: The convergence criterion for the optimization method in [1,9] was stricter using 10^{-6}.)

HOW TO USE THE INTERACTIVE VERSION OF CANOP2

The interactive version of CANOP2 is developed for the INTERCOM System of the CDC 6400 computer. Numerical data is supplied in the system's free format and also system routine SECOND which keeps track of elapsed time is employed. Batch routine AXXXX from APPENDIX F is replaced with interactive routine AXXXX listed in APPENDIX E and, with the common part from APPENDIX G, completes the interactive version of CANOP2.

1. Accessing

On the time-sharing system INTERCOM of the CDC6400 at McMaster University the user should access the file by typing two commands:

ATTACH, ℓfn_1, CANOP2I, ID=*******.*

ATTACH, ℓfn_2, CANOP2C, ID=*******.

2. Main Program

The user then either employs the standard main program (APPENDIX E) if it provides enough dimensions for all his arrays by typing

ATTACH, ℓfn_3, CANOP2M, ID=*******.

or he gets into the EDITOR mode to create the main program for his problem. This can be done by typing the commands

EDITOR.

CREATE

The main program should be written in the same manner as described for the batch version in the previous section.

After typing the main program the user may leave the EDITOR mode by typing the commands

SAVE, filename

BYE

The program just written has to be compiled. The user should type the

---

*ℓfn stands for logical file name of no more than 7 characters. It should be the same in the ATTACH command and in the LOAD command. Two commands should be added for extra long time: ETL, 300., and more columns per a line: SCREEN, 80.

command

FTN,I = filename.

3. Execution

To execute the program the necessary commands are CONNECT, INPUT, OUTPUT.

(i) XEQ,LOAD = LGO, ℓfn_1, ℓfn_2.

with the user's written main program, or

(ii) XEQ, LOAD = ℓfn_1, ℓfn_2, ℓfn_3.

with the standard main program.

4. Interaction

In the interactive mode the user will be instructed how to enter required data. Input format is arbitrary. The user may separate each value by a comma, a blank, a few blanks or by typing the RETURN key. Unless specified, questions are expected to be answered by YES or NO. The user is allowed to modify his data before typing RETURN key by backspacing (type the following two keys at the same time: CTRL - H) or answering the following question "IS DATA OK" by NO. At a few convenient points the user may also modify his data as instructed.

Depending on how data is supplied, only relevant questions appear. For example, when analysis only is required, there is no need for parameters concerning optimization. (Questions for initial data are listed in APPENDIX D.) Also some features might be changed and the problem completely modified at appropriate points when asked for modifications. It is advisable to print the input data before the computer processes it, make changes if desired, and correct all discovered misprints. After obtaining the results, the program may be rerun with some different input data or terminated.

Warning to the user: Make sure that all dimensions supplied through the main program are consistent with all the changes being made.

INPUT-OUTPUT EXAMPLE

The same example is chosen as before, and symmetry on the variable parameters is enforced. Partial interactive output is presented in Fig. 4. Results are obtained for $p = 2$ and $p = 10^3$ and default values for the optimization. 21 uniformly spaced sample points were used in the passband to demonstrate the work of the package, although there may be a need for a larger number of points as

```
YOU ARE WELCOME TO USE THE CASCADED NETWORK OPTIMIZATION PROGRAM.
ENTER YOUR DATA IN ANY FORMAT, HOWEVER, BE REASONABLE.
PLEASE SEPARATE EACH VALUE BY A COMMA, A BLANK OR TYPING THE RETURN
 KEY.
GOOD LUCK.


DO YOU WANT TO SEE THE TABLE OF ELEMENTS AND CODE NUMBERS.          NO

DO YOU WANT QUESTIONS FULLY WORDED TO BE PRINTED OUT.     YES

SPECIFY THE NUMBER OF ELEMENTS IN THE CIRCUIT NOT INCLUDING C- AND D-SECTIONS.

SET TO 0 IF YOU DO NOT WANT ANY.
 1)     7

SUPPLY A SEQUENCE OF   7 CODE NUMBERS OF ELEMENTS TO BE CONNECTED SEQUENTIALLY

FROM SOURCE TO LOAD.
(SEE TABLE FOR ELEMENTS AND CODE NUMBERS.)
 2)     15 14 13 14 13 14 15

IS DATA OK.        OK

SPECIFY VALUES OF  14 PARAMETERS IN THE CIRCUIT INCLUDING STARTING VALUES
FOR VARIABLES. (FOLLOW THE SUPPLIED SEQUENCE OF THE CODE NUMBERS OF ELEMENTS.)

(SEE TABLE FOR THE SEQUENCE OF PARAMETERS.)
 3)     1 .63 1 .33 1 1.27 1 .26 1 1.27 1 .33 1 .63

IS DATA OK.        OK

INDICATE WHICH OF THE  14 PARAMETERS ARE FIXED OR VARIABLE.
SET TO 0 IF FIXED AND A POSITIVE INTEGER IF VARIABLE SUCH
THAT THE INTEGER INDICATES WHETHER THE VARIABLE IS NEW OR REPEATED.
 4)     0 1 0 2 0 3 0 4 0 3 0 2 0 1

IS DATA OK.        OK

SPECIFY THE NUMBER OF C-SECTIONS.
SET TO 0 IF YOU DO NOT WANT ANY.
 5)     0

SPECIFY THE NUMBER OF D-SECTIONS.
SET TO 0 IF YOU DO NOT WANT ANY.
 6)     0

SPECIFY THE LOAD RESISTANCE.
 9)     1

SPECIFY THE NUMBER OF FREQUENCY BANDS OR INTERVALS.
10)     1

SPECIFY THE NUMBER OF OTHER FREQUENCY POINTS AND CONSTRAINTS.
11)     2

FOR EACH INTERVAL, SUPPLY THE FOLLOWING INFORMATION:
1.  LOWER FREQUENCY BOUND (BAND EDGE),
2.  UPPER FREQUENCY BOUND (BAND EDGE),
3.  NUMBER OF SUBINTERVALS (EQUALS SAMPLE POINTS MINUS ONE),
4.  PERFORMANCE SPECIFICATION,
5.  WEIGHTING FACTOR (POSITIVE). SET TO 1 IF UNSURE,
6.  TYPE OF SPECIFICATION:
          SET TO  1 FOR UPPER,
          SET TO -1 FOR LOWER,
          SET TO  0 FOR SINGLE,
7.  APPROXIMATING FUNCTION:
          SET TO 1 FOR REFLECTION COEFFICIENT,
          SET TO 2 FOR INSERTION LOSS (DB),
          SET TO 3 FOR GROUP DELAY (NSEC).
12) INTERVAL( 1)             1037.5 3262.5 20 .1 1 1 2

IS DATA OK.        OK
```

Fig. 4. Partial interactive output.

```
FOR EACH FREQUENCY POINT AND CONSTRAINT SUPPL[illegible] THE FOLLOWING INFORMATION:
1.   FREQUENCY,
2.   PERFORMANCE SPECIFICATION OR CONSTRAINT,
3.   WEIGHTING FACTOR (POSITIVE). SET TO 1 IF UNSURE,
4.   TYPE OF SPECIFICATION OR CONSTRAINT:
          SET TO  1 FOR UPPER,
          SET TO -1 FOR LOWER,
          SET TO  0 FOR SINGLE,
5.   APPROXIMATING FUNCTION:
          SET TO 1 FOR REFLECTION COEFFICIENT,
          SET TO 2 FOR INSERTION LOSS (DB),
          SET TO 3 FOR GROUP DELAY (NSEC),
          SET TO 0 FOR PARAMETER CONSTRAINT.
13) FREQUENCY POINT( 1)     600 50 1 -1 2

IS DATA OK.        OK

13) FREQUENCY POINT( 2)     3750 50 1 -1 2

IS DATA OK.        OK

[illegible] [illegible]FY THE CENTER FREQUENCY (FOR NORMALIZATION).
[illegible]     2175

SET TO 1 IF YOU WANT OP[illegible]IZATION.
SET TO 0 IF OPTIMIZATION IS NOT TO BE USED.
16)     1

DO YOU WANT TO USE DEFAULT VALUES FOR THE OPTIMIZATION.  YES

ANY MODIFICATION
"       NO

DO YOU WANT TO PRINT OUT YOUR INPUT DATA.
"       YES
```

INPUT DATA

```
NUMBER OF ELEMENTS                                                 7
THE CALCULATED NUMBER OF PARAMETERS                                14
```

NODE NUMBER	PARAMETER NUMBER	PARAMETER VALUE	PARAMETER CONDITION
15	1	.100000E+01	FIXED
15	2	.[illegible]30000E+00	VARIABLE
14	3	.100000E+01	FIXED
14	4	.[illegible]30000E+00	VARIABLE
13	5	.100000E+01	FIXED
13	6	.127000E+01	VARIABLE
14	7	.100000E+01	FIXED
[illegible]	8	.2[illegible]0000E+00	VARIABLE
[illegible]	9	.[illegible]40000E+01	FIXED
[illegible]	10	.[illegible]7000E+01	VARIABLE
[illegible]	11	.100000E+01	FIXED
[illegible]	12	.[illegible]30000E+00	VARIABLE
[illegible]	13	.100000E+01	FIXED
15	14	.[illegible]30000E+00	VARIABLE

```
NUMBER OF C-SECTIONS                                               0
NUMBER OF D-SECTIONS                                               0
LOAD RESISTANCE                                            .100000E+01
NUMBER OF FREQUENCY INTERVALS                                      1
NUMBER OF FREQUENCY POINTS                                         2
```

LOWER FREQUENCY	UPPER FREQUENCY	NO. OF SUBINT.	SPECIFICATION		TYPE	WEIGHTING FACTOR
.108750E+04	.326250E+04	20	.1000E+00	INSERTION LOSS	UPPER	.10E+01

FREQUENCY	SPECIFICATION		TYPE	WEIGHTING FACTOR
.600000E+03	.500000E+02	INSERTION LOSS	LOWER	.100000E+01
.375000E+04	.500000E+02	INSERTION LOSS	LOWER	.100000E+01

```
THE CALCULATED TOTAL NUMBER OF INTERVALS                           3
CENTER FREQUENCY                                           .217500E+04
CUT-OFF FREQUENCY                                          0.
```

Fig. 4. [Continued]

```
DEFAULT VALUES ARE USED FOR THE OPTIMIZATION
FLETCHER METHOD WILL BE USED
TEST QUANTITIES TO BE USED IN FLETCHER METHOD
                                                                 .100000E-03
                                                                 .100000E-03
                                                                 .100000E-03
                                                                 .100000E-03
ESTIMATE OF LOWER BOUND ON FUNCTION TO BE MINIMIZED             -.100000E+00
DIFFERENCE IN THE OBJECTIVE FUNCTION
IN SUCCESSIVE OPTIMIZATIONS                                     0.
ARTIFICIAL MARGIN                                               0.
NUMBER OF COMPLETE OPTIMIZATIONS                                       1
VALUES OF P
                                                                       2
MAXIMUM NUMBER OF ALLOWABLE ITERATIONS                               100



ANY MODIFICATION
"       NO


YOUR DATA IS NOW BEING PROCESSED.  IT MAY TAKE SOME TIME BEFORE
RESULTS ARE AVAILABLE.  PLEASE BE PATIENT.
```

RESPONSE AT THE STARTING POINT

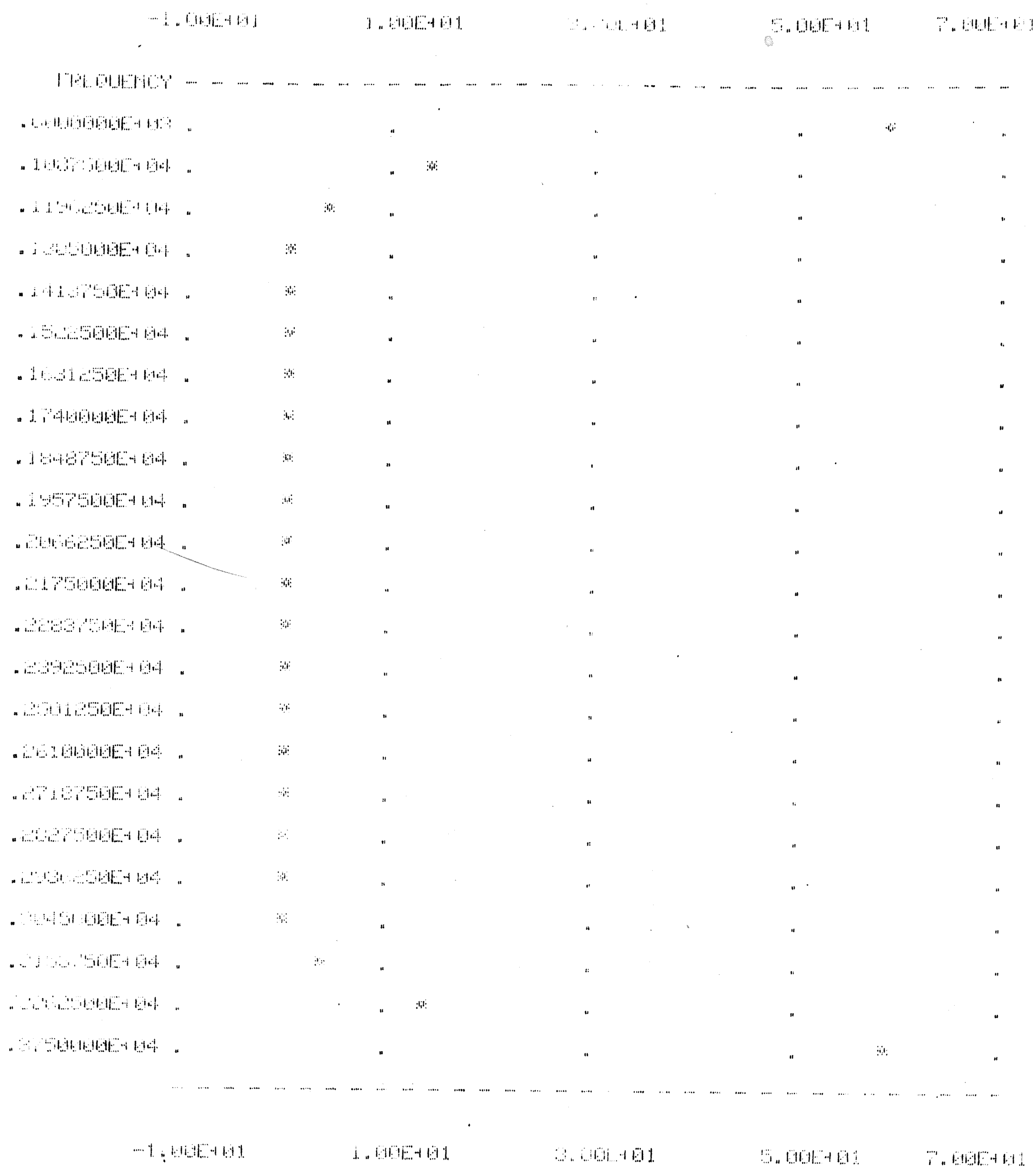
FREQUENCY	INSERTION LOSS
.108750E+04	.135250E+02
.119625E+04	.377724E+01
.130500E+04	.245398E+00
.141375E+04	.511379E-03
.152250E+04	.781660E-01
.163125E+04	.254450E+00
.174000E+04	.443708E+00
.184875E+04	.496993E+00
.195750E+04	.352689E+00
.206625E+04	.116872E+00
.217500E+04	-.123013E-12
.228375E+04	.116872E+00
.239250E+04	.352689E+00
.250125E+04	.496993E+00
.261000E+04	.443708E+00
.271875E+04	.254450E+00
.282750E+04	.781660E-01
.293625E+04	.511379E-03
.304500E+04	.245398E+00
.315375E+04	.377724E+01
.326250E+04	.135250E+02
.600000E+03	.588821E+02
.375000E+04	.588821E+02

```
DO YOU WANT A PLOT.          YES

DO YOU WANT TO PLOT THE ABOVE RFSPONSE.          YES

DO YOU WANT TO SCALE AUTOMATICALLY.   YES
```

Fig. 4. [Continued]



DO YOU WANT MORE PLOTS. NO

Fig. 4. [Continued]

```
OPTIMIZATION BY FLETCHER METHOD
-------------------------------

IEXIT = 1
CRITERION FOR OPTIMUM HAS BEEN SATISFIED

OPTIMUM SOLUTION
----------------

ITER.   FUNCT.      EXECUTION    OBJECTIVE       VARIABLE       GRADIENT
 NO.    EVALU.      TIME(SEC)    FUNCTION        VECTOR          VECTOR

 26       30        .916E+01   -.986505E-02    .607306E+00    -.765771E-04
                                                .301942E+00    -.126016E-01
                                                .719486E+00     .858622E-03
                                                .234645E+00     .193623E-01

VARIABLE PARAMETERS
      IN TOTAL

  .60730638E+00
  .30194192E+00
  .71948640E+00
  .23464546E+00
  .71948640E+00
  .30194192E+00
  .60730638E+00


VALUE OF Q            -2

              FINAL RESPONSE OF THE CIRCUIT
              -----------------------------

          FREQUENCY                  INSERTION LOSS
         .108750E+04                  .616778E-01
         .119625E+04                  .705244E-01
         .130500E+04                  .334635E-02
         .141375E+04                  .675778E-01
         .152250E+04                  .492002E-01
         .163125E+04                  .233392E-02
         .174000E+04                  .183063E-01
         .184875E+04                  .593839E-01
         .195750E+04                  .601589E-01
         .206625E+04                  .227465E-01
         .217500E+04                 -.612843E-13
         .228375E+04                  .227465E-01
         .239250E+04                  .601589E-01
         .250125E+04                  .593839E-01
         .261000E+04                  .183063E-01
         .271875E+04                  .233392E-02
         .282750E+04                  .492002E-01
         .293625E+04                  .675778E-01
         .304500E+04                  .334635E-02
         .315375E+04                  .705244E-01
         .326250E+04                  .616778E-01
         .600000E+03                  .500812E+02
         .375000E+04                  .500812E+02
DO YOU WANT A PLOT.         NO


DO YOU WANT TO TERMINATE THE PROGRAM
"       NO



ANY MODIFICATION
"       YES

WHICH ENTRY
"       04

240     1000
```

Fig. 4. [Continued]

```
ANY MODIFICATION
"       NO

DO YOU WANT THE STARTING VALUES FOR VARIABLES TO BE THE SAME
AS ORIGINALLY DEFINED.      NO

DO YOU WANT THE STARTING VALUES FOR VARIABLES TO BE THE SAME
AS THOSE OBTAINED IN THE LAST PRINTOUT.          YES

DO YOU WANT TO PRINT OUT YOUR INPUT DATA.
"       NO


YOUR DATA IS NOW BEING PROCESSED.  IT MAY TAKE SOME TIME BEFORE
RESULTS ARE AVAILABLE.  PLEASE BE PATIENT.
```

OPTIMIZATION BY FLETCHER METHOD

IEXIT = 1
CRITERION FOR OPTIMUM HAS BEEN SATISFIED

OPTIMUM SOLUTION

ITER. NO.	FUNCT. EVALU.	EXECUTION TIME(SEC)	OBJECTIVE FUNCTION	VARIABLE VECTOR	GRADIENT VECTOR
15	49	.150E+02	-.345781E-01	.606463E+00	-.793726E+00
				.303051E+00	.839850E+00
				.722061E+00	.339665E+01
				.235593E+00	-.875485E+01

VARIABLE PARAMETERS
IN TOTAL

.60646253E+00
.30305056E+00
.72206138E+00
.23559306E+00
.72206138E+00
.30305056E+00
.60646253E+00

VALUE OF Q   -1000

FINAL RESPONSE OF THE CIRCUIT

FREQUENCY	INSERTION LOSS
.108750E+04	.653364E-01
.119625E+04	.652946E-01
.130500E+04	.374049E-02
.141375E+04	.653451E-01
.152250E+04	.448794E-01
.163125E+04	.124570E-02
.174000E+04	.221218E-01
.184875E+04	.652510E-01
.195750E+04	.644821E-01
.206625E+04	.241569E-01
.217500E+04	-.612843E-13
.228375E+04	.241569E-01
.239250E+04	.644821E-01
.250125E+04	.652510E-01
.261000E+04	.221218E-01
.271875E+04	.124570E-02
.282750E+04	.448794E-01
.293625E+04	.653451E-01
.304500E+04	.374049E-02
.315375E+04	.652946E-01
.326250E+04	.653364E-01
.000000E+00	.500356E+02
.375000E+04	.500356E+02

Fig. 4. [Continued]

```
DO YOU WANT A PLOT.          YES

DO YOU WANT TO PLOT THE ABOVE RESPONSE.          YES

DO YOU WANT TO SCALE AUTOMATICALLY.   NO

SUPPLY MINIMUM AND MAXIMUM RESPONSE VALUES TO BE PLOTTED.0
.2

             0.             5.00E-02        1.00E-01        1.50E-01     2.00E-01

   FREQUENCY - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
.6000000E+03 .               .               .               .               .
.1087500E+04 .               .    *          .               .               .
.1196250E+04 .               .    *          .               .               .
.1305000E+04 .*              .               .               .               .
.1413750E+04 .               .    *          .               .               .
.1522500E+04 .             * .               .               .               .
.1631250E+04 *               .               .               .               .
.1740000E+04 .      *        .               .               .               .
.1848750E+04 .               .    *          .               .               .
.1957500E+04 .               .    *          .               .               .
.2066250E+04 .       *       .               .               .               .
.2175000E+04 *               .               .               .               .
.2283750E+04 .       *       .               .               .               .
.2392500E+04 .               .    *          .               .               .
.2501250E+04 .               .    *          .               .               .
.2610000E+04 .      *        .               .               .               .
.2718750E+04 *               .               .               .               .
.2827500E+04 .             * .               .               .               .
.2936250E+04 .               .    *          .               .               .
.3045000E+04 .*              .               .               .               .
.3153750E+04 .               .    *          .               .               .
.3262500E+04 .               .    *          .               .               .
.3750000E+04 .               .               .               .               .
             - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

             0.             5.00E-02        1.00E-01        1.50E-01     2.00E-01
DO YOU WANT MORE PLOTS.     NO
```

Fig. 4. [Continued]

indicated by the 51 points that are used in plotting the results (Fig. 5). Fig. 5 also shows a plot of the group delay of the filter for which analysis was done with the parameters obtained in the insertion loss optimization for p = 1000. Table V compares results with and without forcing the symmetry on the variable parameters. There is a considerable saving in the number of function evaluations and running time for the symmetrical case.

ADDITIONAL ELEMENTS

Presently, the source and load are real constant resistances, the source being assumed to be unity. Frequency - dependent, complex source and load impedances are readily accommodated or can be constructed or modelled, where appropriate, by defining suitable fixed components.

The simplest way of handling two-port sections not in the present list (Tables II and III) is to replace an existing element, frequent use of which is not anticipated, by the desired element preferably with the same number of parameters. In this case, only a few FORTRAN lines dealing with the ABCD matrix of the element and its sensitivities need be changed. Many sensitivity expressions have been published in reference [6]. How to derive those which are not available is demonstrated in APPENDIX B. If the parameters of the new element are not to be changed then sensitivity formulas are not necessary.

Adding elements is slightly more complicated in that more FORTRAN lines need adjustment. The procedure which should be followed is contained in APPENDIX C. Distributed RC lines, nonuniform lines and transistor amplifier stages are examples of two-ports that can be added.

COMMENTS

Low values of p, e.g. 2, intermediately large values of p, e.g., 10, 10^3, and larger, are optional to the user depending on how close to a minimax (Chebyshev, equal-ripple) solution he wants to come. Low values of p will generally allow quicker optimization to non equal - ripple solutions. Large values of p may slow down optimization but better near equal - ripple solutions will be obtained. Recommendation: start with 2, increase to 10 then to 100, etc., as needed. Optimization for larger values of p starts automatically at the optimum of the previous optimization in the batch version and it is optional in the interactive version.

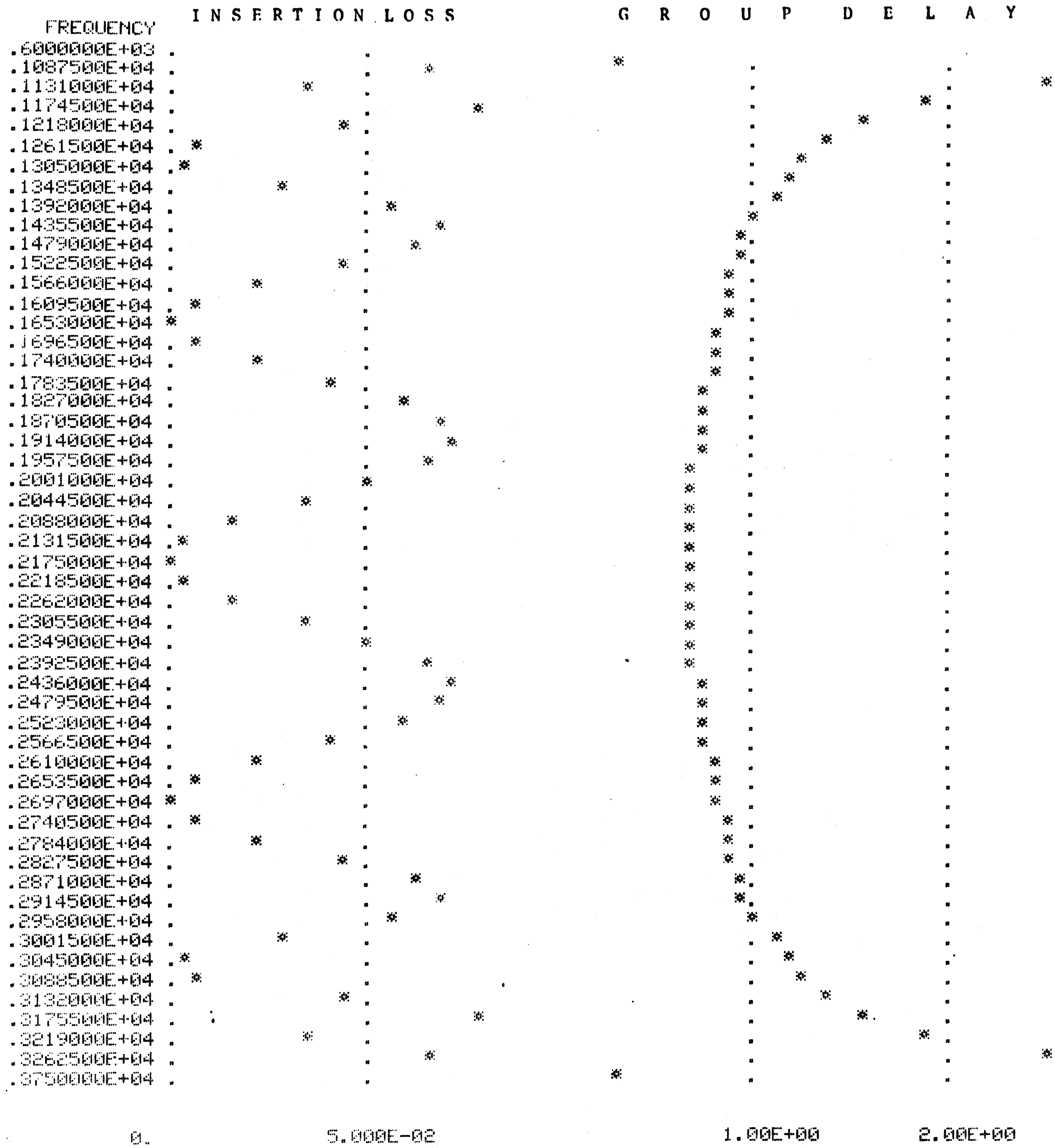


Fig. 5. Interactive plot of insertion loss and group delay of the optimized results for the filter of Fig. 2. Linear scales are used.

TABLE V

RESULTS FOR THE MICROWAVE FILTER

Parameters	p = 2	p = 2 symmetry enforced	p = 1000	p = 1000 symmetry enforced
Z_{01}	.607337	.607306	.606456	.606463
Z_{02}	.301907	.301942	.303864	.303051
Z_{03}	.719283	.719486	.722362	.722061
Z_{04}	.234544	.234645	.235607	.235593
Z_{05}	.719251	Z_{03}	.721749	Z_{03}
Z_{06}	.301928	Z_{02}	.302224	Z_{02}
Z_{07}	.607352	Z_{01}	.606429	Z_{01}
No. of function evaluations	77	30	84	49
Execution time (sec)	23.2	9.16	25.4	15

Optimization for p = 1000 was started at the optimum for p = 2.

The program terminates when the stopping criterion for the optimization method is satisfied, or when the relative change in the objective function in two successive optimizations is less than a small prescribed quantity.

APPENDIX A

NORMALIZATION OF PARAMETER VALUES

To illustrate the normalization process we may consider the following examples. For element 1, a series inductance, we consider a parameter L_n such that $\omega_n L_n$, where ω_n is the normalized frequency, yields the desired reactance in ohms. Thus, if the normalization frequency is 3 GHz, the inductance 2nH, then at the normalized frequency

$$L_n = 2\pi f . L = 12\pi .$$

For elements 11-15, for example, we consider a length ℓ_n such that $\tan \frac{\pi}{2} \omega_n \ell_n$ yields the desired value of the frequency variable for lossless transmission lines.

APPENDIX B

SENSITIVITY EXPRESSIONS

Sensitivity expressions for elements 1 to 19 have either been published [6,10] or are readily obtainable using a procedure similar to the example which follows. For element 7, for example, the quantity $\underset{\sim}{I}^T \Delta \underset{\sim}{Z}^T \hat{\underset{\sim}{I}}$ (See Table I of Bandler and Seviora [6]) is given by

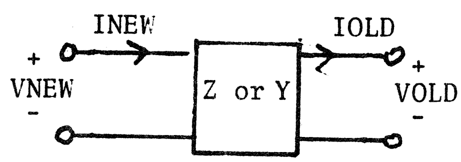
$$\begin{aligned} I\hat{I}\Delta Z = & \left[\frac{-I\hat{I}X' \omega_R}{2Q^2}\right] \Delta Q \\ & + \left[I\hat{I}\left(\frac{X'}{2Q} - j \frac{\omega_R X'}{\omega}\right)\right] \Delta\omega_R \\ & + \left[I\hat{I} \frac{Z}{X'}\right] \Delta X' \\ & + \left[I\hat{I}\, j \frac{X'}{2} \left(1 + \frac{\omega_R^2}{\omega^2}\right)\right] \Delta\omega \end{aligned}$$

where Z is the impedance of the element, I is the original network current and $\hat{I}$ is the adjoint network current through it, ω_R is the resonant frequency, Q is the quality factor and X' the slope reactance at $\omega = \omega_R$. The expressions in square brackets are appropriate sensitivity expressions with respect to Q, ω_R, X' and ω, respectively.

APPENDIX C

INSERTING ADDITIONAL ELEMENTS

The procedure used for existing elements with one, two and three parameters has been followed. New elements are defined as subroutine CODE20, CODE21 and CODE22 with one, two and three parameters, respectively. The listing of subroutine APPROX, the only one affected by these additions, with the changes marked at the places where they could be inserted between the lines made of asterisks. At the beginning of each group of changes it is pointed out which case is effective by C20, C21 and C22 for the one, two and three parameter problem. The following FORTRAN notation is used for any element



with OMEGA for the normalized frequency and VAR1, VAR2 and VAR3 for the element's parameter to be varied. A few FORTRAN lines dealing with the ABCD matrix of the element and its sensitivities need to be defined as in the following example. Write the first lines of the subroutines with one, two or three variable parameters respectively, for example, as

SUBROUTINE CODE20 (IOLD,VOLD,OMEGA,VAR1,INEW,VNEW)

SUBROUTINE CODE21 (IOLD,VOLD,OMEGA,VAR1,VAR2,INEW,VNEW)

SUBROUTINE CODE22 (IOLD,VOLD,OMEGA,VAR1,VAR2,VAR3,INEW,VNEW)

Declare

COMPLEX IOLD,VOLD,INEW,VNEW, Z or Y

Define either impedance Z or admittance Y

If series connection, define input current INEW and input voltage VNEW as follows:

INEW = IOLD

VNEW=VOLD+Z*IOLD or VNEW=VOLD+IOLD/Y

If parallel connection define

VNEW=VOLD

```
INEW=IOLD+VOLD/Z or INEW=IOLD+VOLD*Y
Add
RETURN
END
```

```
      SUBROUTINE APPROX (OMEGN,N1,X,APP,GRAD,IC,A,B,AB)
      DIMENSION X(1), GRAD(1), IC(1), A(1), B(1), AB(1)
      DIMENSION AD(50), ADJJ(50), G(50)
      COMPLEX AD,ADJJ,CONRHO,RHO,I,IHAT,V,VHAT,INEW,VNEW,IOLD,VOLD,G,V1,
     1V2,IHAT1,IHAT2,PSIL,RL,GL,GC,GLLTEL,GLLTZO,GSCTEL,GSCTZO,GOCTEL,GO
     2CTZO,GLCPBP,GLCPQ,GLCPOR,GLCSQ,GLCSOR,GLCSXP
      COMPLEX POLD,PHASE
      COMMON /BLACK/ M,NE,RL,NC,ND,KVR,EM,NC,MET
      COMMON /S16/ PIE,WCC,WCSQ,WCS1,FMC
      LOGICAL B
C
C     M IS THE TOTAL NUMBER OF ELEMENTS IN THE CIRCUIT
C      NE IS THE TOTAL NUMBER OF ELEMENTS IN THE CIRCUIT
C       A ARRAY CONTAINS PARAMETER VALUES
C     B ARRAY CONTAINS LOGICAL VARIABLES
C     IC ARRAY CONTAINS CODE NUMBER GIVING ORDER IN WHICH BLOCKS ARE CON
C     NECTED
C
C     FOLLOWING FUNCTION STATEMENTS DEFINE SENSITIVITIES
C
      THETA(EL)=PIE/2.*OMEGA*EL
      GC(OMEGA,V,VHAT)=-CMPLX(0.,OMEGA)*V*VHAT
      GL(OMEGA,I,IHAT)=CMPLX(0.,OMEGA)*I*IHAT
      GLLTZO(V1,V2,IHAT1,IHAT2,ZO)=(V1*IHAT1-V2*IHAT2)/ZO
      GLLTEL(OMEGA,EL,V1,V2,IHAT1,IHAT2)=PIE/2.*OMEGA/SIN(THETA(EL))*(V1
     1*IHAT2-V2*IHAT1)
      GSCTEL(OMEGA,EL,ZO,I,IHAT)=PIE/2.*(1./COS(THETA(EL)))**2*CMPLX(0.,
     1ZO*OMEGA)*I*IHAT
      GSCTZO(EL,I,IHAT)=CMPLX(0.,SIN(THETA(EL))/COS(THETA(EL)))*I*IHAT
      GOCTEL(ZO,EL,OMEGA,I,IHAT)=CMPLX(0.,ZO*THETA(EL)/EL*(1./SIN(THETA(
     1EL)))**2)*I*IHAT
      GOCTZO(OMEGA,EL,I,IHAT)=CMPLX(0.,-COS(THETA(EL))/SIN(THETA(EL)))*I
     1*IHAT
      GLCPBP(BP,OMEGA,Q,OMEGAR,V,VHAT)=-V*VHAT*CMPLX((OMEGAR/Q),((OMEGA*
     1OMEGA-OMEGAR*OMEGAR)/OMEGA))/2.0
      GLCPQ(BP,Q,OMEGAR,V,VHAT)=V*VHAT*CMPLX(BP*OMEGAR/(2.*Q*Q),0.)
      GLCPOR(BP,OMEGA,OMEGAR,Q,V,VHAT)=-BP*V*VHAT*CMPLX(1./(2.*Q),-OMEGA
     1R/OMEGA)
      GLCSQ(XP,OMEGAR,Q,I,IHAT)=-CMPLX((XP*OMEGAR)/(2.*Q*Q),0.)*I*IHAT
      GLCSOR(OMEGA,XP,OMEGAR,Q,I,IHAT)=CMPLX((XP/(2.*Q)),-OMEGAR*XP/OMEG
     1A)*I*IHAT
      GLCSXP(OMEGA,OMEGAR,Q,I,IHAT)=CMPLX(OMEGAR/Q,((OMEGA*OMEGA-OMEGAR*
     1OMEGAR)/OMEGA))*I*IHAT/2.
      GESXP(OMEGA,OMEGAR,I,IHAT)=CMPLX(0.,(OMEGA*OMEGA-OMEGAR*OMEGAR)/OM
     1EGA)*I*IHAT/2.
      GESOR(OMEGA,XP,OMEGAR,I,IHAT)=CMPLX(0.,-XP*OMEGAR/OMEGA)*I*IHAT
      GEPBP(OMEGA,OMEGAR,V,VHAT)=-CMPLX(0.,(OMEGA*OMEGA-OMEGAR*OMEGAR)/O
     1MEGA)*V*VHAT/2.
      GEPOR(BP,OMEGA,OMEGAR,V,VHAT)=V*VHAT*CMPLX(0.,BP*OMEGAR/OMEGA)
```

```
*****************************************************************************
*                                                                           *
* Define sensitivity expressions as function statements of all relevant parameters *
* as described in the APPENDIX B and [6]. For the example chosen the following sen- *
* sitivity expressions are defined where the name clearly states which code and with *
* respect to which variable parameter:                                      *
* G20VAR1 (OMEGA,VAR1,I or V, IHAT or VHAT)  =  define                      *
* G21VAR1 (OMEGA,VAR1,VAR2,I or V, IHAT or VHAT)  =  define                 *
* G21VAR2 (OMEGA,VAR1,VAR2,I or V, IHAT or VHAT)  =  define                 *
* G22VAR1 (OMEGA, VAR1,VAR2,VAR3,I or V, IHAT or VHAT)  =  define           *
* G22VAR2 (OMEGA,VAR1,VAR2,VAR3,I or V, IHAT or VHAT)  =  define            *
* G22VAR3 (OMEGA,VAR1,VAR2,VAR3,I or V, IHAT or VHAT)  =  define            *
*****************************************************************************
```

```
      IF (N1.GT.50) GO TO 149
      OLDIL=0.
      GD1=0.
      GD2=0.
      N=N1
      IF (KVR.EQ.1) N=N1-NC-2*ND-1
      IF (OMEGN.LE.30.) GO TO 1
      GO TO 147
```

```
1       CONTINUE
        OMEGA=OMEGN
        IF (OMEGN.GT.10..AND.OMEGN.LE.20.) OMEGA=OMEGN-10.
        IF (OMEGN.GT.20.) OMEGA=OMEGN-20.
        IF (M.EQ.0) GO TO 84
        DELO=1.E-7*OMEGA
        KKGD=0
        IF (KKGD.EQ.0) GO TO 3
2       DO 83 KKV=1,N
        KKGD=KKGD+1
        DELX=1.E-4*X(KKV)
        X(KKV)=X(KKV)+DELX
3       CONTINUE
        KGD=0
        IF (OMEGN.LE.20.) GO TO 4
        OMEGA=OMEGA-DELO
4       CONTINUE
        DO 62 KV=1,2
        J=0
        K=0
        VOLD=EL
        IOLD=1.0
        DO 73 L=1,M
        MM=M+1-L
        NN=IC(MM)                                                    ,5,39,20
        GO TO (5,5,5,5,5,5,20,20,20,20,39,39,39,39,39,39,39,39,39, NN
5       KKK=NE-K
        IF (B(KKK)) GO TO 6
        GO TO 7
6       JJ=N-J
        A(KKK)=X(JJ)
        J=J+1                                      IF (NN.EQ.20) GO TO 2000
7       GO TO (8,9,10,11,15,16), NN
8       CALL CODE1 (IOLD,VOLD,A(KKK),OMEGA,INEW,VNEW)
        GO TO 12
9       CALL CODE2 (IOLD,VOLD,A(KKK),OMEGA,INEW,VNEW)
        GO TO 12
10      CALL CODE3 (IOLD,VOLD,A(KKK),OMEGA,INEW,VNEW)
        VOLD=VNEW-VOLD
        GO TO 12
11      CALL CODE4 (IOLD,VOLD,A(KKK),OMEGA,INEW,VNEW)
        IOLD=INEW-IOLD
        C20*********************************************************************
        *        GO TO 12                                                      *
        *  2000  CALL CODE20 (IOLD,VOLD,OMEGA,A(KKK),INEW,VNEW)                *
        * Use either IOLD=INEW - IOLD or VOLD=VNEW-VOLD or neither whichever is applicable *
        * for the sensitivity calculation                                      *
        ************************************************************************
12      IF (B(KKK).AND.OMEGN.GT.10.) GO TO 14
        IF (B(KKK)) GO TO 13
        GO TO 19
13      IF (NN.EQ.1.OR.NN.EQ.4) G(JJ)=GL(OMEGA,IOLD,IOLD)
        IF (NN.EQ.2.OR.NN.EQ.3) G(JJ)=GC(OMEGA,VOLD,VOLD)
14      IF (NN.EQ.1.OR.NN.EQ.4) AD(JJ)=IOLD
        IF (NN.EQ.2.OR.NN.EQ.3) AD(JJ)=VOLD
        C20*********************************************************************
        *                                                                      *
        * IF (NN.EQ.20)  G(JJ) = G20VAR1(OMEGA,VAR1,IOLD or VOLD,IOLD or VOLD) *
        * IF (NN.EQ.20)  AD(JJ) = IOLD or VOLD                                 *
        *                                                                      *
        ************************************************************************
        GO TO 19
1       CALL CODE5 (IOLD,VOLD,A(KKK),INEW,VNEW)
        GO TO 17
1       CALL CODE6 (IOLD,VOLD,A(KKK),INEW,VNEW)
        IOLD=INEW-IOLD
17      IF (B(KKK)) GO TO 18
        GO TO 19
18      G(JJ)=IOLD*IOLD
```

```
      AD(JJ)=IOLD
19    KN=K+1
      GO TO 72
20    KK=K
      JOLD=J
      DO 23 II=1,3
      KKK=NE-KK
      IF (B(KKK)) GO TO 21
      GO TO 22
21    JJ=N-J
      A(KKK)=X(JJ)
      J=J+1
22    KK=KK+1
23    CONTINUE
      JK=NE-K-2
      JV=NE-K-1
      JH=NE-K
      II=NN-6                    IF (NN.EQ.22)  GO TO 2200
      GO TO (24,25,26,27), NI
24    CALL CODE7 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      GO TO 28
25    CALL CODE8 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      GO TO 28
26    CALL CODE9 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      VOLD=VNEW-VOLD

C22***********************************************************************
*                                                                        *
*       GO TO 28                                                         *
*  2200 CALL CODE22 (IOLD,VOLD,OMEGA,A(JK),A(JV),A(JH),INEW,VNEW)        *
*  Use either IOLD=INEW-IOLD or VOLD=VNEW-VOLD or neither,               *
*  whichever is applicable for the sensitivity calculation.              *
**************************************************************************
      GO TO 28
27    CALL CODE10 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      IOLD=INEW-IOLD
28    CONTINUE
      IF (NN.EQ.7.OR.NN.EQ.10) NI=7
      IF (NN.EQ.8.OR.NN.EQ.9) NI=8
      DO 38 II=1,3
      GO TO (29,30,31), II
29    KVH=JH
      GO TO 32
30    KVH=JV
      GO TO 32
31    KVH=JK
32    CONTINUE
      NJ=N-JOLD
      IF (B(KVH).AND.OMEGN.GT.10.) GO TO 37
      IF (B(KVH)) GO TO 33
      GO TO 38
33    IF (KVH.EQ.JH) GO TO 34
      IF (KVH.EQ.JV) GO TO 35
      IF (KVH.EQ.JK) GO TO 36
34    IF (NI.EQ.7) G(NJ)=GLCSXP(OMEGA,A(JK),A(JV),IOLD,IOLD)
      IF (NI.EQ.8) G(NJ)=GLCPBP(A(JH),OMEGA,A(JV),A(JK),VOLD,VOLD)
C22***********************************************************************
*                                                                        *
*IF (NN.EQ.22) G(NJ) = G22VAR3  (OMEGA,A(JK),A(JV),A(JH),IOLD or VOLD,IOLD or VOLD)
**************************************************************************
      GO TO 37
35    IF (NI.EQ.7) G(NJ)=GLCSQ(A(JH),A(JK),A(JV),IOLD,IOLD)
      IF (NI.EQ.8) G(NJ)=GLCPQ(A(JH),A(JV),A(JK),VOLD,VOLD)

C22***********************************************************************
*                                                                        *
*IF (NN.EQ.22) G(NJ) = G22VAR2 (OMEGA,A(JK),A(JV),A(JH),IOLD or VOLD, IOLD or VOLD)
**************************************************************************
      GO TO 37
36    IF (NI.EQ.7) G(NJ)=GLCSOR(OMEGA,A(JH),A(JK),A(JV),IOLD,IOLD)
      IF (NI.EQ.8) G(NJ)=GLCPOR(A(JH),OMEGA,A(JK),A(JV),VOLD,VOLD)
```

```
C22*************************************************************************************
*  IF (NN.EQ.22) G(NJ) = G22VAR1 (OMEGA,A(JK),A(JV),A(JH),IOLD or VOLD,IOLD or VOLD).  *
*                                                                                     *
*                                                                                     *
*                                                                                     *
***************************************************************************************

      IF (NI.EQ.7) AD(NJ)=IOLD
37    IF (NI.EQ.8) AD(NJ)=VOLD

C22*************************************************************************************
*                                                                                     *
*  IF (NN.EQ.22) AD(NJ) = IOLD or VOLD whichever is used in sensitivity expression    *
*                                                                                     *
***************************************************************************************
      JOLD=JOLD+1
38    CONTINUE
      KN=K+3
      GO TO 72
39    KK=K
      JOLD=J
      DO 42 II=1,2
      KKK=NE-KK
      IF (B(KKK)) GO TO 40
      GO TO 41
40    JJ=N-J
      A(KKK)=X(JJ)
      J=J+1
41    KK=KK+1
42    CONTINUE
      JV=NE-K-1
      JH=NE-K
      NI=NN-10 ——— IF (NN.EQ.21)  GO TO 2100
      GO TO (43,44,45,46,54,59,60,61,62), NI
43    CALL CODE11 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
      GO TO 48
44    CALL CODE12 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
      GO TO 47
45    CALL CODE13 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
      GO TO 48
46    CALL CODE14 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
      GO TO 47
47    IOLD=INEW-IOLD
48    NJ=N-JOLD
      IF (NN.EQ.11.OR.NN.EQ.14) NI=11
      IF (NN.EQ.12.OR.NN.EQ.13) NI=12
      IF (B(JH)) GO TO 49
      GO TO 51
49    IF (OMEGN.GT.10.) GO TO 50
      IF (NI.EQ.11) G(NJ)=GSCTZO(A(JV),IOLD,IOLD)
      IF (NI.EQ.12) G(NJ)=GOCTZO(OMEGA,A(JV),IOLD,IOLD)
50    CONTINUE
      JOLD=JOLD+1
      AD(NJ)=IOLD
51    NJ=N-JOLD
      IF (B(JV)) GO TO 52
      GO TO 58
52    IF (OMEGN.GT.10.) GO TO 53
      IF (NI.EQ.11) G(NJ)=GSCTEL(OMEGA,A(JV),A(JH),IOLD,IOLD)
      IF (NI.EQ.12) G(NJ)=GOCTEL(A(JH),A(JV),OMEGA,IOLD,IOLD)
53    CONTINUE
      AD(NJ)=IOLD
      GO TO 58
54    CALL CODE15 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
      NJ=N-JOLD
      IF (B(JH)) GO TO 55
      GO TO 56
55    AD(NJ)=VOLD
      ADJJ(NJ)=VNEW
      G(NJ)=GLLTZO(VNEW,VOLD,INEW,IOLD,A(JH))
      JOLD=JOLD+1
56    NJ=N-JOLD
      IF (B(JV)) GO TO 57
```

```
      GO TO 58
57    AD(NJ)=VOLD
      G(NJ)=GLLTEL(OMEGA,A(JV),VNEW,VOLD,INEW,IOLD)
      AD JJ(NJ)=VNEW
58    KN=K+2
      GO TO 72
59    CALL CODE16 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW)
      GO TO 63
60    CALL CODE17 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW)
      GO TO 63
61    CALL CODE18 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW)
      VOLD=VNEW-VOLD
      GO TO 63
62    CALL CODE19 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW)
      IOLD=INEW-IOLD
C21***************************************************************************
*                                                                            *
*                                                                            *
*     GO TO 63                                                               *
*2100 CALL CODE21 (IOLD,VOLD,OMEGA,A(JV),A(JH),INEW,VNEW)                    *
*Use either IOLD=INEW-IOLD or VOLD=VNEW-VOLD or neither,                     *
*whichever is applicable for the sensitivity calculation                     *
******************************************************************************
63    CONTINUE
      IF (NN.EQ.16.OR.NN.EQ.19) NI=16
      IF (NN.EQ.17.OR.NN.EQ.18) NI=17
      DO 71 II=1,2
      GO TO (64,65), II
64    KVH=JH
      GO TO 66
65    KVH=JV
66    CONTINUE
      NJ=N-JOLD
      IF (B(KVH).AND.OMEGN.GT.10.) GO TO 70
      IF (B(KVH)) GO TO 67
      GO TO 71
67    IF (KVH.EQ.JH) GO TO 68
      IF (KVH.EQ.JV) GO TO 69
68    IF (NI.EQ.16) G(NJ)=GESXP(OMEGA,A(JV),IOLD,IOLD)
      IF (NI.EQ.17) G(NJ)=GEPBP(OMEGA,A(JV),VOLD,VOLD)
C21***************************************************************************
*                                                                            *
*IF (NN.EQ.21) G(NJ) = G21VAR2 (OMEGA,A(JV),A(JH),IOLD or VOLD, IOLD or VOLD) *
******************************************************************************
      GO TO 70
69    IF (NI.EQ.16) G(NJ)=GESOR(OMEGA,A(JH),A(JV),IOLD,IOLD)
      IF (NI.EQ.17) G(NJ)=GEPOR(A(JH),OMEGA,A(JV),VOLD,VOLD)
C21***************************************************************************
*                                                                            *
*IF (NN.EQ.21) G(NJ) = G21VAR1 (OMEGA,A(JV),A(JH),IOLD or VOLD, IOLD or VOLD) *
******************************************************************************
70    IF (NI.EQ.16) AD(NJ)=IOLD
      IF (NI.EQ.17) AD(NJ)=VOLD
C21***************************************************************************
*                                                                            *
*IF (NN.EQ.21) AD(NJ) = IOLD or VOLD whichever is used in sensitivity expression *
*                                                                            *
******************************************************************************
```

```
      JOLD=JOLD+1
71    CONTINUE
      KN=K+2
72    VOLD=VNEW
      IOLD=INEW
      K=KN
73    CONTINUE
      IF (OMEGN.GT.20.) GO TO 79
      IF (OMEGN.GT.10.) GO TO 87
      RHO=1.-2.*INEW/(VNEW+INEW)
      CONRHO=CONJG(RHO)
      APP=CABS(RHO)
      IF (MET) 74,76,74
74    DO 75 L=1,N
      GRAD(L)=REAL((CONRHO/APP)*2.*G(L)/((VNEW+INEW)**2))
75    CONTINUE
76    IF (NC.GT.0.OR.ND.GT.0) GO TO 77
      RETURN
77    IF (KVR.EQ.0) RETURN
      NNN=N+1
      DO 78 L=NNN,N1
      GRAD(L)=0.
78    CONTINUE
      RETURN
79    CONTINUE
      PHASE=(1./(VNEW+INEW))
      IF (KGD.EQ.0) GO TO 80
      GO TO 81
80    OMEGA=OMEGA+2.*DELO
      POLD=PHASE
81    IF (KGD.GT.0) DELAY=-AIMAG((2.0/(POLD+PHASE))*((PHASE-POLD)/(2*DEL
     1O)))*(1000./(2.*PIE*FM))
      KGD=KGD+1
82    CONTINUE
      OMEGA=OMEGA-DELO
      IF (KKGD.EQ.0) OLDEL=DELAY
      IF (KKGD.EQ.0) GO TO 2
      GRAD(KKV)=(DELAY-OLDEL)/DELX
      X(KKV)=X(KKV)-DELX
83    CONTINUE
      APP=OLDEL
      IF (NC.GT.0.OR.ND.GT.0) GO TO 84
      RETURN
84    IF (KVR.EQ.0) GO TO 86
      NNN=N1-N
      DO 85 L=1,NNN
      LL=N+L
      AB(L)=X(LL)
85    CONTINUE
86    WCC=WC/FM
      IF (NC.GT.0) CALL CODEC (NC,N,AB,WCC,OMEGA,GD1,GRAD,FM)
      IF (ND.GT.0) CALL CODED (ND,N,NC,AB,WCC,OMEGA,GD2,GRAD,FM)
      APP=OLDEL+GD1+GD2-AB(NNN)
      IF (KVR.EQ.1) GRAD(N1)=-1.0
      RETURN
87    APP=-20.*ALOG10(CABS((1.+RL)/(VNEW+INEW)))
      K=1
      J=1
      VOLD=1.0
      IOLD=1.0
      DO 141 L=1,M
      NN=IC(L)                 ,931, 117,99
      GO TO (88,89,90,91,92,93,99,99,99,99,117,117,117,117,117,117,117,1
     117,117), NN
88    CALL CODE1 (IOLD,VOLD,A(K),OMEGA,INEW,VNEW)
      GO TO 94
89    CALL CODE2 (IOLD,VOLD,A(K),OMEGA,INEW,VNEW)
      GO TO 94
90    CALL CODE3 (IOLD,VOLD,A(K),OMEGA,INEW,VNEW)
      VOLD=VNEW-VOLD
      GO TO 94
91    CALL CODE4 (IOLD,VOLD,A(K),OMEGA,INEW,VNEW)
      IOLD=INEW-IOLD
      GO TO 94
92    CALL CODE5 (IOLD,VOLD,A(K),INEW,VNEW)
      GO TO 94
93    CALL CODE6 (IOLD,VOLD,A(K),INEW,VNEW)
      IOLD=INEW-IOLD
```

```
C20*****************************************************************************
*                                                                              *
*     GO TO 94                                                                 *
*931 CALL CODE20 (IOLD,VOLD,OMEGA,A(K),INEW,VNEW)                               *
*     Use either IOLD=INEW-IOLD or VOLD=VNEW-VOLD or neither, whichever is appli-*
*     cable                                                                    *
********************************************************************************
 94      IF  (B(K))  GO TO 95
         GO  TO 98
 95      IF  (NN.EQ.1.OR.NN.EQ.4)  G(J)=GL(OMEGA,AD(J),IOLD)
         IF  (NN.EQ.2.OR.NN.EQ.3)  G(J)=GC(OMEGA,AD(J),VOLD)
         IF  (NN.EQ.5.OR.NN.EQ.6)  G(J)=IOLD*AD(J)

C20*****************************************************************************
*                                                                              *
*    IF (NN.EQ.20) G(J) = G20VAR1 (OMEGA,VAR1,AD(J),IOLD or VOLD)              *
*    IF (NN.EQ.20) GO TO 97                                                    *
********************************************************************************
         GO TO (96,97,96,97,96,97), NN
 96      G(J)=-G(J)
 97      J=J+1
 98      K=K+1
         GO TO 140
 99      JK=K
         JV=K+1
         JH=K+2
         NII=NN-6

C22*****************************************************************************
*                                                                              *
*   IF (NN,EQ.22) GO TO 1040                                                   *
*                                                                              *
********************************************************************************
         GO TO (100,101,102,103), NII
 100     CALL CODE7  (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
         GO TO 104
 101     CALL CODE8  (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
         GO TO 104
 102     CALL CODE9  (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
         VOLD=VNEW-VOLD
         GO TO 104
 103     CALL CODE10 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
         IOLD=INEW-IOLD
 104     IF  (NN.EQ.7.OR.NN.EQ.10)  NI=7
         IF  (NN.EQ.8.OR.NN.EQ.9)  NI=8

C22*****************************************************************************
*                                                                              *
*      GO to 1050                                                              *
*1040 CALL CODE22 (IOLD,VOLD,OMEGA,A(JK),A(JV),A(JH),INEW,VNEW)                 *
*    Use either IOLD=INEW-IOLD or VOLD=VNEW-VOLD or neither, whichever applicable*
*1050  CONTINUE                                                                *
********************************************************************************
         DO 116 II=1,3
         GO TO  (105,106,107), II
 105     KVH=JK
         GO TO 108
 106     KVH=JV
         GO TO 108
 107     KVH=JH
 108     IF  (B(KVH))  GO TO 109
         GO TO 116
 109     IF  (KVH.EQ.JK)  GO TO 110
         IF  (KVH.EQ.JV)  GO TO 111
         IF  (KVH.EQ.JH)  GO TO 112
 110     IF  (NI.EQ.7)  G(J)=GLCSOR(OMEGA,A(JH),A(JK),A(JV),AD(J),IOLD)
         IF  (NI.EQ.8)  G(J)=GLCPOR(A(JH),OMEGA,A(JK),A(JV),AD(J),VOLD)
```

```
C22*********************************************************************
*                                                                     *
*  IF (NN.EQ.22) G(J) = G22VAR3  (OMEGA,A(JK),A(JV),A(JH),AD(J),IOLD or VOLD)  *
*                                                                     *
***********************************************************************
      GO TO 113
111   IF (NI.EQ.7) G(J)=GLCSQ(A(JH),A(JK),A(JV),AD(J),IOLD)
      IF (NI.EQ.8) G(J)=GLCPQ(A(JH),A(JV),A(JK),AD(J),VOLD)
C22*********************************************************************
*                                                                     *
*  IF (NN.EQ.22) G(J) = G22VAR 2  (OMEGA,A(JK),A(JV),A(JH),AD(J),IOLD or VOLD)  *
*                                                                     *
***********************************************************************
      GO TO 113
112   IF (NI.EQ.7) G(J)=GLCSXP(OMEGA,A(JK),A(JV),AD(J),IOLD)
      IF (NI.EQ.8) G(J)=GLCPBP(A(JH),OMEGA,A(JV),A(JK),AD(J),VOLD)
C22*********************************************************************
*                                                                     *
*  IF (NN.EQ.22) G(J) =  G22VAR1 (OMEGA, A(JK),A(JV),A(JH),AD(J),IOLD or VOLD)  *
*  IF (NN.EQ.22) GO TO 115                                            *
*                                                                     *
***********************************************************************
113   GO TO (114,115,114,115), NII
114   G(J)=-G(J)
115   J=J+1
116   CONTINUE
      K=K+3
      GO TO 140
117   JV=K
      JH=K+1
C21*********************************************************************
*                                                                     *
*   IF (NN.EQ.21)  GO TO 1270                                         *
*                                                                     *
*                                                                     *
***********************************************************************

      NII=NN-10
      GO TO (118,119,120,121,123,124,125,126,127), NII
118   CALL CODE11 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
      GO TO 122
119   CALL CODE12 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
      IOLD=INEW-IOLD
      GO TO 122
120   CALL CODE13 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
      GO TO 122
121   CALL CODE14 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
      IOLD=INEW-IOLD
122   IF (NN.EQ.11.OR.NN.EQ.14) NI=11
      IF (NN.EQ.12.OR.NN.EQ.13) NI=12
      GO TO 129
123   CALL CODE15 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
      GO TO 129
124   CALL CODE16 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW)
      GO TO 128
125   CALL CODE17 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW)
      GO TO 128
126   CALL CODE18 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW)
      VOLD=VNEW-VOLD
      GO TO 128
127   CALL CODE19 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW)
      IOLD=INEW-IOLD
```

```
      C21*********************************************************************
      *                                                                     *
      *     GO TO 128                                                       *
      *1270 CALL CODE21 (IOLD,VOLD,OMEGA,A(JV),A(JH),INEW,VNEW)             *
      *  Use either IOLD=INEW-IOLD or VOLD=VNEW-VOLD or neither, whichever is applicable *
      ***********************************************************************
128   IF (NN.EQ.16.OR.NN.EQ.19) NI=16
      IF (NN.EQ.17.OR.NN.EQ.18) NI=17
123   DO 139 II=1,2
      GO TO (130,131), II
130   KVH=JV
      GO TO 132
131   KVH=JH
132   IF (B(KVH)) GO TO 133
      GO TO 139
133   IF (KVH.EQ.JV) GO TO 134
      IF (KVH.EQ.JH) GO TO 135
134   IF (NI.EQ.11) G(J)=GSCTEL(OMEGA,A(JV),A(JH),AD(J),IOLD)
      IF (NI.EQ.12) G(J)=GOCTEL(A(JH),A(JV),OMEGA,AD(J),IOLD)
      IF (NN.EQ.15) G(J)=-GLLTEL(OMEGA,A(JV),ADJJ(J),AD(J),IOLD,INEW)
      IF (NI.EQ.16) G(J)=GESOR(OMEGA,A(JH),A(JV),AD(J),IOLD)
      IF (NI.EQ.17) G(J)=GEPOR(A(JH),OMEGA,A(JV),AD(J),VOLD)

      C21*********************************************************************
      *                                                                     *
      *                                                                     *
      *  IF (NN,EQ.21) G(J) = G21VAR1 (OMEGA,A(JV),A(JH),AD(J),IOLD or VOLD) *
      *  IF (NN,EQ.21) GO TO 138                                            *
      *                                                                     *
      ***********************************************************************
      GO TO 136
135   IF (NI.EQ.11) G(J)=GSCTZO(A(JV),AD(J),IOLD)
      IF (NI.EQ.12) G(J)=GOCTZO(OMEGA,A(JV),AD(J),IOLD)
      IF (NN.EQ.15) G(J)=-GLLTZO(ADJJ(J),AD(J),IOLD,INEW,A(JH))
      IF (NI.EQ.16) G(J)=GESXP(OMEGA,A(JV),AD(J),IOLD)
      IF (NI.EQ.17) G(J)=GEPBP(OMEGA,A(JV),AD(J),VOLD)

      C21*********************************************************************
      *                                                                     *
      *  IF (NN.EQ.21) G(J) = G21VAR2  (OMEGA,A(JV), A(JH),AD(J),IOLD or VOLD) *
      *  IF (NN.EQ.21)  GO TO 137                                           *
      *                                                                     *
      ***********************************************************************
136   GO TO (137,138,137,138,138,137,138,137,138), NII
137   G(J)=-G(J)
138   J=J+1
139   CONTINUE
      K=K+2
140   IOLD=INEW
      VOLD=VNEW
141   CONTINUE
      PSIL=VNEW+INEW*RL
      CONSTN=20./ALOG(10.)
      IF (MET) 142,144,142
142   DO 143 L=1,N
      GRAD(L)=-REAL(G(L)/PSIL)*CONSTN
143   CONTINUE
144   IF (NC.GT.0.OR.ND.GT.0) GO TO 145
      RETURN
145   IF (KVR.EQ.0) RETURN
      NNN=N+1
      DO 146 L=NNN,N1
      GRAD(L)=0.
146   CONTINUE
      RETURN
147   INT=IFIX(OMEGN-30.)
      APP=X(INT)
      DO 148 L=1,N1
      GRAD(L)=0.
148   CONTINUE
      GRAD(INT)=1.0
      RETURN
149   WRITE (6,150)
      CALL EXIT
C
C
150   FORMAT (/////5X,69HNUMBER OF VARIABLE PARAMETERS N EXCEEDS 50. PLE
     1ASE CHANGE THE LENGTHS,/5X,55HOF ARRAYS AD(N), ADJJ(N) AND G(N) IN
     2 SUBROUTINE APPROX.)
      END
```

APPENDIX D

QUESTIONS REQUIRING NUMERICAL DATA IN THE INTERACTIVE CANOP 2

Q. No.

1. Specify the number of elements in the circuit not including C- and D-sections. Set to 0 if you do not want any.
2. Supply a sequence of MM code numbers of elements to be connected sequentially from source to load. (See table for elements and code numbers.)
3. Specify values of NE parameters in the circuit including starting values for variables. (Follow the supplied sequence of the code numbers of elements.) (See table for the sequence of parameters.)
4. Indicate which of the NE parameters are fixed or variable. Set to 0 if fixed and a positive integer if variable such that the integer indicates whether the variable is new or repeated.
5. Specify the number of C-sections. Set to 0 if you do not want any.
6. Specify the number of D-sections. Set to 0 if you do not want any.
7. Indicate whether C- and D-section parameters are all fixed or variable. Set to 0 if fixed and 1 if variable.
8. Specify NC values of the parameters of the C- and 2*ND values of the D-sections and a d level.
9. Specify the load resistance.
10. Specify the number of frequency bands or intervals.
11. Specify the number of other frequency points and constraints.
12. For each interval supply the following information:
    1. lower frequency bound (band edge),
    2. upper frequency bound (band edge),
    3. number of subintervals (equals sample points minus one),
    4. performance specification,
    5. weighting factor (positive). Set to 1 if unsure,
    6. type of specification:

       set to 1 for upper,

       set to -1 for lower,

       set to 0 for single,

7. approximating function

   set to 1 for reflection coefficient,

   set to 2 for insertion loss (dB),

   set to 3 for group delay (nsec).

13. For each frequency point and constraint supply the following information:
    1. frequency,
    2. performance specification or constraint,
    3. weighting factor (positive). Set to 1 if unsure,
    4. type of specification or constraint:

       set to 1 for upper,

       set to -1 for lower,

       set to 0 for single,
    5. approximating function:

       set to 1 for reflection coefficient,

       set to 2 for insertion loss (dB)

       set to 3 for group delay (nsec),

       set to 0 for parameter constraint.
14. Specify the center frequency (for normalization).
15. Specify the cut-off frequency for C- and D-section.
16. Set to 1 if you want optimization.

    Set to 0 if optimization is not to be used.
17. Specify K small quantities for testing convergence in the Fletcher method.
    (e.g., 1.E-4).
18. Specify the maximum number of iterations.
    (e.g., 100).
19. Specify the number of iterations after which you want an intermediate output to be printed out.

    Set to 0 if no intermediate output is desired.
20. Supply a realistic under-estimate (lower bound) of the value of the objective function.
21. Specify the difference in objective function in successive optimizations.
    Set to 0 if not sure.
22. Specify a small quantity by which specification would be shifted artificially.
    Set to 0 if not sure.

23. Specify the number of complete optimizations.
24. Supply values of p (positive integer, greater than one) to be used successively for each complete optimization.

If the default feature for the optimization is required, questions 17 to 24 do not appear and the following default parameters would be considered:

Q. No.	Default parameters
17	1.E-4 for all required values
18	100
19	0
20	-0.1
21	0.
22	0.
23	1
24	2

These parameters may be changed, after default parameters have been chosen, by entering the question number and the new parameter value.

APPENDIX E

STANDARD MAIN PROGRAM AND INTERACTIVE VERSION

```
      PROGRAM TST (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      DIMENSION A1(14),A2(14),A3(1),A4(3),A5(3),A6(3,3),A7(25),A8(25),
     1 A9(25),A10(25),A11(25),A12(25),A13(7),A14(7),A15(7),A16(7),
     2 A17(175),A18(49),IA1(21),IA2(3),IA3(14)
      CALL CANOP2 (A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12 ,A13,A14,
     1A15,A16,A17,A18,IA1,IA2,IA3)
      CALL EXIT
      END
```

CDTOT 0008

```
      SUBROUTINE CANOP2 (AA,B,AB,FUN,WT,XX,X,X1,FX,ERROR,EHELP,AP,A,G,GR
     1AD,EPS,GGRAD,H,IC,IOBJ,NUMB)
C
C          SUBROUTINE WHICH COORDINATES
C          THE OTHER SUBROUTINES
C
      EXTERNAL OBJECT
      DIMENSION DD(3)
      DIMENSION NUMC(50), IPA(25)
      DIMENSION XXX(150)
      DIMENSION TEXT(11)
      DIMENSION IA(100)
      DIMENSION ASTRT(100)
      DIMENSION A(1), G(1), EPS(1), AA(1), B(1), AB(1), IC(1), IOBJ(1),
     1H(1), GRAD(1), XX(3,1), X(1), X1(1), ERROR(1), EHELP(1), AP(1), GG
     2RAD(1), FUN(1), WT(1), NUMB(1), FX(1)
      COMMON /BLK/ KO,T1,KOUNT,NUMF
      COMMON /BLACK/ MM,NE,RL,NC,ND,KVR,FM,WC,MET
      COMMON /EXT/ APP,PSI,EMAX,N,NINT,IP
      COMMON /S16/ PIE,WCC,WCSQ,WCS1,FMC
      COMMON /TEST/ FREQ(150),RESP(150,1),ICALC,XLOG10,IDENT
      COMMON /TOTAL/ IRE,NTOT,ATOT(25),GRADT(25),ICC(25)
      DIMENSION IOBXF(25), NUMBXF(25), XF(2,25)
      LOGICAL B
      LOGICAL CONV,UNITH
      COMPLEX RL
      DATA FI,V/10HFIXED     ,10HVARIABLE  /
      DATA TEXT/10HPARAMETER ,10HREFLECTION,10HINSERTION ,10HGROUP DELA,
     17HVALUE  ,7H COEFF.,7HLOSS   ,7HY      ,6HLOWER ,6HSINGLE,6HUPPER
     2/
      DATA DD/2HNO,2HYE,2HOK/
      ERR(Z)=ERROX(Z,IINT,FUN,WT,A,N1,GRAD,APP,PSI,XX,1,IC,AA,B,AB)
      UNITH=.TRUE.
      CONV=.FALSE.
      T1=0.
      PSI=0.
      ITER=1
      IPA(1)=2
      NOPT=0
      KTY1=0
      KTY2=0
      IVAR1=0
      IVAR2=0
      NEW=0
      NE=0
      NRE=0
      IRE=0
      JW=0
      IPRINT=0
      NPRINT=1
      NINS=0
      NINX=0
      WC=0.
      PRINT 254
      PRINT 255
    1 READ (5,298) ANS
      IF (ANS.NE.DD(1).AND.ANS.NE.DD(2)) GO TO 2
      IF (ANS.EQ.DD(1)) GO TO 3
      WRITE (6,256)
      PRINT 257
      WRITE (6,258)
      GO TO 3
    2 PRINT 299
      GO TO 1
    3 CONTINUE
      PRINT 259
    4 READ (5,298) ANS
      IF (ANS.NE.DD(1).AND.ANS.NE.DD(2)) GO TO 5
      IF (ANS.EQ.DD(1)) GO TO 6
      CONV=.TRUE.
      GO TO 6
    5 PRINT 299
      GO TO 4
    6 CONTINUE
      IF (CONV) 7,8
    7 PRINT 261
      PRINT 262
    8 IJK=1
      PRINT 260, IJK
      READ (5,*) MM
      IF (JW.EQ.0) GO TO 9
      IF (MM.EQ.MMO.OR.MM.EQ.0) GO TO 119
      IF (MM.NE.0) PRINT 263, MM
    9 IF (MM.EQ.0) GO TO 33
      IF (CONV) 10,11
   10 PRINT 263, MM
      PRINT 264
   11 IJK=2
   12 PRINT 260, IJK
      READ (5,*) (IC(L),L=1,MM)
      PRINT 265
      READ (5,298) ANS
      IF (ANS.EQ.DD(1)) GO TO 12
      NE=0
      DO 18 I=1,MM
      IF (IC(I).LE.6) 13,14
   13 NUMC(NE+1)=IC(I)
      NE=NE+1
   14 IF (IC(I).GE.7.AND.IC(I).LE.10) 15,16
   15 NUMC(NE+1)=IC(I)
```

```
      NUMC(NE+2)=IC(I)
      NUMC(NE+3)=IC(I)
      NE=NE+3
 16   IF (IC(I).GE.11.AND.IC(I).LE.19) 17,18
 17   NUMC(NE+1)=IC(I)
      NUMC(NE+2)=IC(I)
      NE=NE+2
 18   CONTINUE
      IF (JW.EQ.0) GO TO 19
      IF (NE.EQ.NEO) GO TO 119
 19   IF (CONV) 20,21
 20   PRINT 266, NE
 21   IJK=3
 22   PRINT 260, IJK
      READ (5,*) (AA(I),I=1,NE)
      NPRINT=1
      IVAR1=1
      PRINT 265
      READ (5,298) ANS
      IF (ANS.EQ.DD(1)) GO TO 22
      IF (JW.EQ.0) GO TO 23
      IF (NE.EQ.NEO) GO TO 27
 23   IF (CONV) 24,25
 24   PRINT 267, NE
 25   IJK=4
 26   PRINT 260, IJK
      READ (5,*) (IC(MM+I),I=1,NE)
      PRINT 265
      READ (5,298) ANS
      IF (ANS.EQ.DD(1)) GO TO 26
 27   IRE=0
      NRE=1
      KTY1=0
      DO 32 I=1,NE
      B(I)=.TRUE.
      IF (IC(MM+I).EQ.0) B(I)=.FALSE.
      IF (B(I)) 28,32
 28   KTY1=KTY1+1
      ICC(KTY1)=IC(MM+I)
      ATOT(KTY1)=AA(I)
      KCU=KTY1-1
      IF (KCU) 29,29,30
 29   A(1)=ATOT(1)
      ASTRT(1)=A(1)
      GO TO 32
 30   NR=0
      DO 31 KRE=1,KCU
      IF (ICC(KTY1).LE.ICC(KRE)) GO TO 32
      NR=NR+1
 31   CONTINUE
      IF (NR.NE.KCU) GO TO 32
      NRE=NRE+1
      A(NRE)=ATOT(KTY1)
      ASTRT(NRE)=A(NRE)
 32   CONTINUE
      NTOT=KTY1
      N1=NRE
      IF (JW.EQ.1) GO TO 119
 33   IF (CONV) 34,35
 34   PRINT 268
      PRINT 262
 35   IJK=5
      PRINT 260, IJK
      READ (5,*) NC
      IF (JW.EQ.0) GO TO 38
      IF (NC.EQ.NCO) GO TO 119
      NEW=1
      IF (NC.EQ.0) GO TO 119
      PRINT 269
 36   READ (5,298) ANS
      IF (ANS.NE.DD(1).AND.ANS.NE.DD(2)) GO TO 37
      IF (ANS.EQ.DD(2)) GO TO 45
      GO TO 40
 37   PRINT 299
      GO TO 36
 38   IF (CONV) 39,40
 39   PRINT 270
      PRINT 262
 40   IJK=6
      PRINT 260, IJK
      READ (5,*) ND
      IF (JW.EQ.0) GO TO 41
      IF (ND.EQ.NDO) GO TO 119
      NEW=1
      IF (NC.EQ.NCO.AND.ND.EQ.0) GO TO 119
 41   IF (NC.EQ.0.AND.ND.EQ.0) KVR=0
      IF (NC.EQ.0.AND.ND.EQ.0) GO TO 48
      IF (NEW.EQ.1) GO TO 45
      IF (CONV) 42,43
 42   PRINT 271
 43   IJK=7
      PRINT 260, IJK
      READ (5,*) KVR
      IF (JW.EQ.1) GO TO 119
      IF (CONV) 44,45
 44   JJ=2*ND
      WRITE (6,272) NC,JJ
 45   JJ=2*ND
      J=JJ+NC+1
      JCD=J
```

```
      IJK=8
46    PRINT 260, IJK
      READ (5,*) (AB(I),I=1,J)
      PRINT 265
      READ (5,298) ANS
      IF (ANS.EQ.DD(1)) GO TO 46
      IF (KVR.EQ.0) KTY2=0
      IF (KVR.NE.0) KTY2=J
      N1=NRE+KTY2
      IF (KVR.EQ.0) GO TO 47
      IVAR2=1
47    CONTINUE
48    IF (JW.EQ.0) GO TO 49
      GO TO 119
49    IF (CONV) 50,51
50    PRINT 273
51    IJK=9
      PRINT 260, IJK
      READ (5,*) R
      RL=CMPLX(R,0.0)
      IF (JW.EQ.1) GO TO 119
      IF (CONV) 52,53
52    PRINT 274
53    IJK=10
      PRINT 260, IJK
      READ (5,*) NINTD
      IF (JW.EQ.0) GO TO 54
      IF (NINTD.EQ.NINTDO) GO TO 119
      IF (NINTD.EQ.0) GO TO 119
      GO TO 63
54    IF (CONV) 55,56
55    PRINT 275
56    IJK=11
      PRINT 260, IJK
      READ (5,*) NINTS
      IF (JW.EQ.0) GO TO 62
      IF (NINTS.EQ.NINTSO) GO TO 119
      IF (NINTS.EQ.0) GO TO 119
      PRINT 276
57    READ (5,298) ANS
      IF (ANS.NE.DD(1).AND.ANS.NE.DD(2)) GO TO 58
      IF (ANS.EQ.DD(1)) GO TO 59
      GO TO 112
58    PRINT 299
      GO TO 57
59    PRINT 277
60    READ (5,298) ANS
      IF (ANS.NE.DD(1).AND.ANS.NE.DD(2)) GO TO 61
      IF (ANS.EQ.DD(2)) GO TO 53
      GO TO 112
61    PRINT 299
      GO TO 60
62    IF (NINTD.EQ.0) GO TO 68
63    IF (CONV) 64,65
64    PRINT 278
65    IJK=12
      NINT=0
      DO 67 I=1,NINTD
      NINT=NINT+1
66    PRINT 279, I
      READ (5,*) XX(1,NINT),XX(2,NINT),NUMB(NINT),FUN(NINT),WT(NINT),XX(
     13,NINT),IOBJ(NINT)
      PRINT 265
      READ (5,298) ANS
      IF (ANS.EQ.DD(1)) GO TO 66
      XXX(NINT)=XX(3,NINT)
      IA(NINT)=I
      IF (XX(3,NINT).NE.0.) GO TO 67
      XX(3,NINT)=1.
      XX(3,NINT+1)=-1.
      XXX(NINT+1)=-1.
      XX(1,NINT+1)=XX(1,NINT)
      XX(2,NINT+1)=XX(2,NINT)
      FUN(NINT+1)=FUN(NINT)
      WT(NINT+1)=WT(NINT)
      NUMB(NINT+1)=NUMB(NINT)
      IOBJ(NINT+1)=IOBJ(NINT)
      NINT=NINT+1
      IA(NINT)=-I
67    CONTINUE
      NINS=NINT
      IF (JW.EQ.1) GO TO 119
68    IF (NINTS.EQ.0) GO TO 73
      IF (CONV) 69,70
69    PRINT 280
70    IJK=13
      IF (JW.EQ.1) NINT=NINS
      IF (NINTD.EQ.0) NINT=0
      DO 72 I=1,NINTS
      NINT=NINT+1
      NUMB(NINT)=0
71    PRINT 281, I
      READ (5,*) XX(1,NINT),FUN(NINT),WT(NINT),XX(3,NINT),IOBJ(NINT)
      XX(2,NINT)=XX(1,NINT)
      PRINT 265
      READ (5,298) ANS
      IF (ANS.EQ.DD(1)) GO TO 71
      XXX(NINT)=XX(3,NINT)
      IA(NINT)=I
      IF (XX(3,NINT).NE.0.) GO TO 72
```

```
      XX(3,NINT)=1.
      XX(3,NINT+1)=-1.
      XXX(NINT+1)=-1.
      XX(1,NINT+1)=XX(1,NINT)
      XX(2,NINT+1)=XX(2,NINT)
      FUN(NINT+1)=FUN(NINT)
      WT(NINT+1)=WT(NINT)
      NUMB(NINT+1)=0
      IOBJ(NINT+1)=IOBJ(NINT)
      NINT=NINT+1
      IA(NINT)=-I
72    CONTINUE
      NINX=NINT
      IF (JW.EQ.1) GO TO 119
73    IF (CONV) 74,75
74    PRINT 282
75    IJK=14
      PRINT 260, IJK
      READ (5,*) FM
      IF (JW.EQ.1) GO TO 119
      IF (NC.EQ.0.AND.ND.EQ.0) GO TO 78
      IF (CONV) 76,77
76    PRINT 283
77    IJK=15
      PRINT 260, IJK
      READ (5,*) WC
78    IF (JW.EQ.1) GO TO 119
      IF (CONV) 79,80
79    PRINT 284
80    IJK=16
81    PRINT 260, IJK
      READ (5,*) MET
      IF (JW.EQ.0) GO TO 82
      IF (MET.EQ.METO) GO TO 119
82    IF (MET.EQ.0) GO TO 118
      IF (MET.EQ.2) GO TO 85
      PRINT 251
83    READ (5,298) ANS
      IF (ANS.NE.DD(1).AND.ANS.NE.DD(2)) GO TO 87
      IF (ANS.EQ.DD(1)) GO TO 88
      IDEF=1
      DO 84 I=1,N1
      EPS(I)=1.E-4
84    CONTINUE
      GO TO 86
85    PRINT 252
      GO TO 81
86    N1K=N1
      MAX=100
      EST=-0.1
      DIF=0.
      PSI=0.
      ITER=1
      IPA(1)=2
      GO TO 118
87    PRINT 299
      GO TO 83
88    IDEF=0
      IF (MET.NE.1) GO TO 118
      IF (CONV) 89,90
89    PRINT 285, N1
90    IJK=17
91    PRINT 260, IJK
      READ (5,*) (EPS(I),I=1,N1)
      PRINT 265
      READ (5,298) ANS
      IF (ANS.EQ.DD(1)) GO TO 91
      N1K=N1
      IF (JW.EQ.1) GO TO 119
      IF (MET.NE.1) GO TO 118
      IF (CONV) 92,93
92    PRINT 287
93    IJK=18
      PRINT 260, IJK
      READ (5,*) MAX
      IF (JW.EQ.1) GO TO 119
      IF (CONV) 94,95
94    PRINT 286
95    IJK=19
      PRINT 260, IJK
      READ (5,*) IPRINT
      IF (JW.EQ.1) GO TO 119
      IF (CONV) 96,97
96    PRINT 288
97    IJK=20
      PRINT 260, IJK
      READ (5,*) EST
      IF (JW.EQ.1) GO TO 119
      IF (CONV) 98,99
98    PRINT 289
99    IJK=21
      PRINT 260, IJK
      READ (5,*) DIF
      IF (JW.EQ.1) GO TO 119
      IF (CONV) 100,101
100   PRINT 290
101   IJK=22
      PRINT 260, IJK
      READ (5,*) PSI
      IF (JW.EQ.1) GO TO 119
```

```
      IF (CONV) 102,103
102   PRINT 291
103   IJK=23
      PRINT 260, IJK
      READ (5,*) ITER
      IF (JW.EQ.1) GO TO 119
      IF (CONV) 104,105
104   PRINT 292
105   IJK=24
      PRINT 260, IJK
      READ (5,*) (IPA(I),I=1,ITER)
      IF (JW.EQ.1) GO TO 119
      GO TO 118
106   IF (NINTD.EQ.1) GO TO 65
      PRINT 293, NINTD
107   READ (5,298) ANS
      IF (ANS.NE.DD(1).AND.ANS.NE.DD(2)) GO TO 108
      IF (ANS.EQ.DD(2)) GO TO 65
      GO TO 109
108   PRINT 299
      GO TO 107
109   PRINT 294
      READ (5,*) I
      DO 111 IINT=1,NINS
      IF (IA(IINT).NE.I) GO TO 111
110   PRINT 279, I
      READ (5,*) XX(1,IINT),XX(2,IINT),NUMB(IINT),FUN(IINT),WT(IINT),XX(
     13,IINT),IOBJ(IINT)
      XXX(IINT)=XX(3,IINT)
      PRINT 265
      READ (5,298) ANS
      IF (ANS.EQ.DD(1)) GO TO 110
      IF (XX(3,IINT).NE.0.) GO TO 111
      XX(3,IINT)=1.
      XX(3,IINT+1)=-1.
      XXX(IINT+1)=-1.
      XX(1,IINT+1)=XX(1,IINT)
      XX(2,IINT+1)=XX(2,IINT)
      FUN(IINT+1)=FUN(IINT)
      WT(IINT+1)=WT(IINT)
      NUMB(IINT+1)=NUMB(IINT)
      IOBJ(IINT+1)=IOBJ(IINT)
111   CONTINUE
      GO TO 119
112   IF (NINTS.EQ.1) GO TO 70
      PRINT 295, NINTS
113   READ (5,298) ANS
      IF (ANS.NE.DD(1).AND.ANS.NE.DD(2)) GO TO 114
      IF (ANS.EQ.DD(2)) GO TO 70
      GO TO 115
114   PRINT 299
      GO TO 113
115   PRINT 296
      READ (5,*) I
      NIN=NINS+1
      DO 117 IINT=NIN,NINX
      IF (IA(IINT).NE.I) GO TO 117
116   PRINT 281, I
      READ (5,*) XX(1,IINT),FUN(IINT),WT(IINT),XX(3,IINT),IOBJ(IINT)
      XXX(IINT)=XX(3,IINT)
      PRINT 265
      READ (5,298) ANS
      IF (ANS.EQ.DD(1)) GO TO 116
      XX(2,IINT)=XX(1,IINT)
      IF (XX(3,IINT).NE.0.) GO TO 117
      XX(3,IINT)=1.
      XX(3,IINT+1)=-1.
      XXX(IINT+1)=-1.
      XX(1,IINT+1)=XX(1,IINT)
      XX(2,IINT+1)=XX(2,IINT)
      FUN(IINT+1)=FUN(IINT)
      WT(IINT+1)=WT(IINT)
      NUMB(IINT+1)=0
      IOBJ(IINT+1)=IOBJ(IINT)
117   CONTINUE
      GO TO 119
118   CONTINUE
      INO=1
      JW=1
119   MMO=MM
      NEO=NE
      NCO=NC
      NDO=ND
      NINTDO=NINTD
      NINTSO=NINTS
      NEW=0
      METO=MET
      PRINT 297
120   READ (5,298) ANS
      IF (ANS.NE.DD(1).AND.ANS.NE.DD(2)) GO TO 121
      IF (ANS.EQ.DD(1)) GO TO 123
      GO TO 122
121   PRINT 299
      GO TO 120
122   PRINT 300
      READ (5,*) IJK
      CONV=.FALSE.
      GO TO (8,11,21,25,35,40,43,46,51,53,56,106,112,75,77,80,90,93,95,9
     17,99,101,103,105), IJK
123   CONV=.FALSE.
```

```
      KK=0
      IF (KVR.EQ.0) GO TO 126
      DO 124 I=1,JCD
      KK=KK+1
      AA(NE+I)=AB(I)
124   CONTINUE
      KK=NE+KK
      NE1=NE+1
      KTY2=0
      DO 125 I=NE1,KK
      KTY2=KTY2+1
      NCE=NRE+KTY2
      ASTRT(NCE)=AA(I)
      ICC(KTY1+KTY2)=NCE
      ATOT(KTY1+KTY2)=AA(I)
      A(NCE)=AA(I)
125   CONTINUE
126   CONTINUE
      NCE=NRE+KTY2
      N1=NCE
      NTOT=KTY1+KTY2
      IF (N1.NE.NTOT) IRE=1
      IF (MET.EQ.1.AND.N1K.EQ.N1) GO TO 127
      IF (MET.NE.1) GO TO 157
      PRINT 285, N1
      READ (5,*) (EPS(I),I=1,N1)
127   IF (NOPT.EQ.0) GO TO 154
      IF (NOPT.EQ.NOPTO) GO TO 154
      IF (IVAR1.EQ.1.OR.IVAR2.EQ.1) GO TO 154
      IF (IDATA.EQ.1) GO TO 154
      PRINT 301
128   READ (5,298) ANS
      IF (ANS.NE.DD(1).AND.ANS.NE.DD(2)) GO TO 129
      IF (ANS.EQ.DD(1)) GO TO 130
      NPRINT=1
      GO TO 134
129   PRINT 299
      GO TO 128
130   PRINT 302
131   READ (5,298) ANS
      IF (ANS.NE.DD(1).AND.ANS.NE.DD(2)) GO TO 133
      IF (ANS.NE.DD(1)) GO TO 132
      GO TO 139
132   NPRINT=0
      GO TO 155
133   PRINT 299
      GO TO 131
134   DO 135 I=1,N1
      IF (IRE.EQ.0) ATOT(I)=ASTRT(I)
      A(I)=ASTRT(I)
135   CONTINUE
      IF (IRE) 138,138,136
136   DO 137 I=1,NTOT
      INDEX=ICC(I)
      ATOT(I)=A(INDEX)
137   CONTINUE
138   CONTINUE
      GO TO 154
139   IF (KTY1.EQ.0) GO TO 149
      PRINT 303, KTY1
      NPRINT=1
140   READ (5,*) (ATOT(I),I=1,KTY1)
      PRINT 265
      READ (5,298) ANS
      IF (ANS.NE.DD(1)) GO TO 141
      PRINT 304
      GO TO 140
141   K=0
      NCE=0
      DO 148 I=1,NE
      IF (B(I)) 142,148
142   K=K+1
      AA(I)=ATOT(K)
      IF (IRE) 147,147,143
143   KCU=K-1
      IF (KCU) 144,144,145
144   A(1)=ATOT(1)
      GO TO 148
145   NR=0
      DO 146 KRE=1,KCU
      IF (ICC(K).LE.ICC(KRE)) GO TO 148
      NR=NR+1
146   CONTINUE
      IF (NR.NE.KCU) GO TO 148
      NCE=NCE+1
      A(NCE)=ATOT(K)
      GO TO 148
147   A(K)=ATOT(K)
148   CONTINUE
149   IF (KTY2.EQ.0) GO TO 154
      PRINT 272, NC,JJ
150   READ (5,*) (AB(I),I=1,JCD)
      PRINT 265
      IF (ANS.NE.DD(1)) GO TO 151
      PRINT 304
      GO TO 150
151   CONTINUE
      KK=0
      DO 152 I=1,JCD
      KK=KK+1
```

```
      AA(NE+I)=AB(I)
152   CONTINUE
      KK=NE+KK
      IF (KVR.EQ.0) GO TO 154
      NE1=NE+1
      KTY2=0
      DO 153 I=NE1,KK
      KTY2=KTY2+1
      ATOT(KTY1+KTY2)=AA(I)
      A(KTY1+KTY2)=AA(I)
153   CONTINUE
154   NPRINT=1
155   CONTINUE
      IF (IND.EQ.1) GO TO 156
      IDATA=0
      GO TO 161
156   IND=0
157   PRINT 305
158   READ (5,298) ANS
      IF (ANS.NE.DD(1).AND.ANS.NE.DD(2)) GO TO 159
      GO TO 160
159   PRINT 299
      GO TO 158
160   IDATA=0
      IF (ANS.EQ.DD(2)) IDATA=1
161   CONTINUE
      IF (IDATA.EQ.0) GO TO 180
      NINT=0
      PRINT 322
      PRINT 343, MM
      PRINT 344, NE
      IF (MM.EQ.0) GO TO 165
      PRINT 364
      KNE=0
      DO 164 I=1,NE
      IF (B(I)) 162,163
162   KNE=KNE+1
      AA(I)=ATOT(KNE)
      PRINT 365, NUMC(I),I,AA(I),V
      GO TO 164
163   PRINT 365, NUMC(I),I,AA(I),FI
164   CONTINUE
165   PRINT 346, NC
      PRINT 347, ND
      IF (NC.EQ.0.AND.ND.EQ.0) GO TO 169
      PRINT 348
      IF (KVR.EQ.0) GO TO 167
      DO 166 I=1,J
      PRINT 345, AB(I),V
166   CONTINUE
      GO TO 169
167   DO 168 I=1,J
      PRINT 345, AB(I),FI
168   CONTINUE
169   PRINT 349, R
      PRINT 350, NINTD
      PRINT 351, NINTS
      IF (NINTD.EQ.0) GO TO 171
      DO 170 I=1,NINTD
      NINT=NINT+1
      PRINT 352
      IND1=IOBJ(NINT)+1
      IND2=IND1+4
      IND3=XXX(NINT)+10
      PRINT 354, XX(1,NINT),XX(2,NINT),NUMB(NINT),FUN(NINT),TEXT(IND1),T
     1EXT(IND2),TEXT(IND3),WT(NINT)
      IF (XXX(NINT).NE.0.) GO TO 170
      NINT=NINT+1
170   CONTINUE
171   IF (NINTS.EQ.0) GO TO 173
      PRINT 353
      DO 172 I=1,NINTS
      NINT=NINT+1
      IND1=IOBJ(NINT)+1
      IND2=IND1+4
      IND3=XXX(NINT)+10
      PRINT 355, XX(1,NINT),FUN(NINT),TEXT(IND1),TEXT(IND2),TEXT(IND3),W
     1T(NINT)
      IF (XXX(NINT).NE.0.) GO TO 172
      NINT=NINT+1
172   CONTINUE
173   NINT=MAX0(NINX,NINS)
      PRINT 363, NINT
      PRINT 356, FM
      IF (NC.EQ.0.AND.ND.EQ.0) GO TO 174
      PRINT 357, WC
174   IF (MET.NE.1) MET=0
      IF (MET) 175,175,176
175   PRINT 307
      GO TO 177
176   IF (IDEF.EQ.1) PRINT [illegible]3
      PRINT 323
      PRINT 325
      PRINT 326, (EPS(I),I=1,N1)
      PRINT 327, EST
      PRINT 358, DIF
      PRINT 359, PSI
      PRINT 360, ITER
      PRINT 361
      PRINT 362, (IPA(I),I=1,ITER)
```

```
      PRINT 324, MAX
177   PRINT 297
178   READ (5,298) ANS
      IF (ANS.NE.DD(1).AND.ANS.NE.DD(2)) GO TO 179
      IF (ANS.EQ.DD(1)) GO TO 127
      IND=1
      GO TO 122
179   PRINT 299
      GO TO 178
180   PRINT 306
      IF (NINTO.EQ.0) NINT=NINX
      IF (NINTS.EQ.0) NINT=NINS
      NINT=MAX0(NINX,NINS)
      DO 182 I=1,NINT
      IF (IOBJ(I).EQ.0) GO TO 181
      XX(1,I)=XX(1,I)/FM+(IOBJ(I)-1)*10.
      XX(2,I)=XX(2,I)/FM+(IOBJ(I)-1)*10.
      GO TO 182
181   XX(1,I)=XX(1,I)+30.
      XX(2,I)=XX(1,I)
182   CONTINUE
      K=0
      PIE=4.*ATAN(1.)
      WCC=WC/FM
      WCSQ=WCC*WCC
      WCS1=1.-WCSQ
      FMC=FM*0.001
      XLOG10=ALOG(10.0)
      PMIN=0.
      PMAX=0.
      ICALC=0
      IPLOT=1
      IF (NPRINT.EQ.0) GO TO 183
      WRITE (6,341)
183   DO 190 J=1,NINT
      IINT=J
      IF (NPRINT.EQ.0) GO TO 186
      IF (J-1) 186,185,184
184   IF (IOBJ(J)-IOBJ(J-1)) 185,186,185
185   IF (XX(1,J).LE.10.) WRITE (6,338)
      IF (XX(1,J).LE.20.0.AND.XX(1,J).GT.10.) WRITE (6,339)
      IF (XX(1,J).LE.30.0.AND.XX(1,J).GT.20.) WRITE (6,340)
186   L=NUMB(J)+1
      IF (NUMB(J).EQ.0) Z=XX(1,J)
      DO 190 I=1,L
      IF (NUMB(J).GT.0) Z=XX(1,J)+(XX(2,J)-XX(1,J))*(I-1)/NUMB(J)
      K=K+1
      X(K)=Z
      ERROR(K)=ERR(Z)
      EHELP(K)=ERROR(K)*XX(3,J)
      ER=ERROR(K)
      IF (PSI.NE.0.) ER=ER+PSI*XX(3,IINT)
      ERT=(ER/WT(IINT))+FUN(IINT)
      IF (IOBJ(J).EQ.1) FX(K)=Z*FM
      IF (IOBJ(J).EQ.2) FX(K)=(Z-10.)*FM
      IF (IOBJ(J).EQ.3) FX(K)=(Z-20.)*FM
      IF (IOBJ(J).EQ.0) GO TO 189
      IF (K.EQ.1) GO TO 187
      IF (FX(K).EQ.FX(K-1)) GO TO 189
187   ICALC=ICALC+1
      IF (NPRINT.EQ.0) GO TO 188
      WRITE (6,337) FX(K),ERT
188   FREQ(ICALC)=FX(K)
      RESP(ICALC)=ERT
189   CONTINUE
190   AP(K)=APP
      EMAX=EHELP(1)
      DO 191 M=2,K
      EMAX=AMAX1(EMAX,EHELP(M))
191   CONTINUE
      NOPTO=NOPT
      IVAR1=0
      IVAR2=0
C
C          PLOT
C
      DO 245 KK=1,ITER
      IP=IPA(KK)
      IF (NPRINT.EQ.0) GO TO 225
C
      PRINT 308
192   READ (5,298) ANS
      IF (ANS.NE.DD(1).AND.ANS.NE.DD(2)) GO TO 193
      IF (ANS.EQ.DD(1).AND.IPLOT.EQ.1) GO TO 225
      IF (ANS.EQ.DD(1).AND.IPLOT.EQ.2) GO TO 245
      GO TO 194
193   PRINT 299
      GO TO 192
194   PRINT 309
195   READ (5,298) ANS
      IF (ANS.NE.DD(1).AND.ANS.NE.DD(2)) GO TO 196
      IF (ANS.EQ.DD(1)) GO TO 197
      GO TO 202
196   PRINT 299
      GO TO 195
197   MINT=1
      ICALC=0
      PRINT 310
      READ (5,*) IOBXF(MINT)
      PRINT 312
```

```
198   PRINT 311, MINT
      READ (5,*) XF(1,MINT)
      XF(2,MINT)=XF(1,MINT)
      IF (XF(1,MINT).LT.0.) GO TO 199
      READ (5,*) NUMBXF(MINT)
      IF (NUMBXF(MINT).NE.0) READ (5,*) XF(2,MINT)
      MINT=MINT+1
      IOBXF(MINT)=IOBXF(MINT-1)
      GO TO 198
199   MINT=MINT-1
      DO 200 I=1,MINT
      XF(1,I)=XF(1,I)/FM+(IOBXF(I)-1)*10.
      XF(2,I)=XF(2,I)/FM+(IOBXF(I)-1)*10.
200   CONTINUE
      DO 201 J=1,MINT
      IF (XF(1,J).LE.10.) WRITE (6,338)
      IF (XF(1,J).LE.20.AND.XF(1,J).GT.10.) WRITE (6,339)
      IF (XF(1,J).LE.30.AND.XF(1,J).GT.20.) WRITE (6,340)
      L=NUMBXF(J)+1
      IF (NUMBXF(J).EQ.0) Z=XF(1,J)
      DO 201 I=1,L
      IF (NUMBXF(J).GT.0) Z=XF(1,J)+(XF(2,J)-XF(1,J))*(I-1)/NUMBXF(J)
      ICALC=ICALC+1
      CALL APPROX (Z,N1,A,APP,GRAD,IC,AA,B,AB)
      IF (IOBXF(J).EQ.1) FREQ(ICALC)=Z*FM
      IF (IOBXF(J).EQ.2) FREQ(ICALC)=(Z-10.)*FM
      IF (IOBXF(J).EQ.3) FREQ(ICALC)=(Z-20.)*FM
      IF (IOBXF(J).EQ.0) GO TO 201
      RESP(ICALC)=APP
      WRITE (6,337) FREQ(ICALC),APP
201   CONTINUE
202   PRINT 313
      IAUTO=1
      PMIN=0.0
      PMAX=0.0
203   READ (5,298) ANS
      IF (ANS.NE.DD(1).AND.ANS.NE.DD(2)) GO TO 204
      IF (ANS.NE.DD(1)) GO TO 205
      PRINT 314
      READ (5,*) PMIN,PMAX
      IAUTO=0
      GO TO 205
204   PRINT 299
      GO TO 203
205   IDENT=0
      DO 213 I=2,ICALC
      IF (FREQ(I)-FREQ(I-1)) 206,213,213
206   IF (FREQ(I)-FREQ(1)) 207,207,209
207   POMOC=FREQ(I)
      POM=RESP(I)
      DO 208 J=2,I
      K=I-J+2
      RESP(K)=RESP(K-1)
      FREQ(K)=FREQ(K-1)
208   CONTINUE
      FREQ(1)=POMOC
      RESP(1)=POM
      GO TO 213
209   IF (FREQ(I).LT.FREQ(2).AND.I.GT.3) GO TO 210
      GO TO 212
210   POMOC=FREQ(I)
      POM=RESP(I)
      DO 211 J=3,I
      K=I-J+3
      RESP(K)=RESP(K-1)
      FREQ(K)=FREQ(K-1)
211   CONTINUE
      FREQ(2)=POMOC
      RESP(2)=POM
      GO TO 213
212   POMOC=FREQ(I)
      POM=RESP(I)
      FREQ(I)=FREQ(I-1)
      RESP(I)=RESP(I-1)
      FREQ(I-1)=POMOC
      RESP(I-1)=POM
      IDENT=1
213   CONTINUE
      IF (IDENT) 205,214,205
214   CONTINUE
      IDENT=0
215   CALL PLOT (1,IAUTO,PMIN,PMAX)
      PRINT 315
216   READ (5,298) ANS
      IF (ANS.NE.DD(1).AND.ANS.NE.DD(2)) GO TO 217
      IF (ANS.NE.DD(1)) GO TO 218
      IF (IPLOT.EQ.1) GO TO 2[illegible]5
      IF (IPLOT.EQ.2) GO TO 2[illegible]5
217   PRINT 299
      GO TO 216
218   PRINT 316
219   READ (5,298) ANS
      IF (ANS.NE.DD(1).AND.ANS.NE.DD(2)) GO TO 221
      IF (ANS.EQ.DD(1)) GO TO 222
220   PRINT 317
      READ (5,*) PMIN,PMAX
      IAUTO=0
      GO TO 215
221   PRINT 299
      GO TO 219
```

```
222    PRINT 318
223    READ (5,298) ANS
       IF (ANS.NE.DD(1).AND.ANS.NE.DD(2)) GO TO 224
       IF (ANS.EQ.DD(1)) GO TO 220
       GO TO 197
224    PRINT 299
       GO TO 223
225    NPRINT=0
       IF (MET.NE.1) GO TO 226
       GO TO 227
226    WRITE (6,328)
       GO TO 245
C
C           OPTIMIZATION
C
227    CALL SECOND (T1)
       WRITE (6,329)
       WRITE (6,330)
       IF (IPRINT.EQ.0) GO TO 228
C
C           PRINTS THE INTERMEDIATE RESULTS
C           FOR THE FLETCHER METHOD
C
       WRITE (6,331)
228    CONTINUE
       CALL OPTIM1 (N1,A,F,G,H,UNITH,EST,EPS,MAX,IPRINT,IEXIT,GRAD,NUMB,X
      1X,X,X1,ERROR,EHELP,AP,GGRAD,FUN,WT,IC,AA,B,AB,2)
       GO TO (229,230,231,230,232), IEXIT
229    PRINT 366, IEXIT
       GO TO 233
230    PRINT 367, IEXIT
       GO TO 233
231    PRINT 368, IEXIT
       GO TO 233
232    PRINT 369, IEXIT
233    CALL SECOND (T2)
C
C           PRINTS THE RESULTS
C           FOR THE OPTIMIZATION PROCESS
C
       IF (KO.EQ.0) GO TO 234
       WRITE (6,332)
       GO TO 235
234    WRITE (6,334)
235    CONTINUE
       T=T2-T1
       WRITE (6,333)
       WRITE (6,335) KOUNT,NUMF,T,F,((A(I),G(I)),I=1,N1)
       IF (IRE.EQ.0) GO TO 236
       PRINT 319
       WRITE (6,320) (ATOT(I),I=1,NTOT)
236    WRITE (6,336) IP
       KQ=0
       WRITE (6,342)
       ICALC=0
       IPLOT=2
       DO 242 J=1,NINT
       IINT=J
       IF (J-1) 239,238,237
237    IF (IOBJ(J)-IOBJ(J-1)) 238,239,238
238    IF (XX(1,J).LE.10.) WRITE (6,338)
       IF (XX(1,J).LE.20.0.AND.XX(1,J).GT.10.) WRITE (6,339)
       IF (XX(1,J).LE.30.0.AND.XX(1,J).GT.20.) WRITE (6,340)
239    KQ=KQ+1
       KL=KQ+NUMB(J)
       DO 241 I=KQ,KL
       L=I-KQ+1
       ER=ERR(X(I))
       IF (PSI.NE.0.) ER=ER+PSI*XX(3,IINT)
       ERT=(ER/WT(IINT))+FUN(IINT)
       IF (IOBJ(J).EQ.0) GO TO 241
       IF (I.EQ.1) GO TO 240
       IF (FX(I).EQ.FX(I-1)) GO TO 241
240    ICALC=ICALC+1
       WRITE (6,337) FX(I),ERT
       FREQ(ICALC)=FX(I)
       RESP(ICALC)=ERT
241    CONTINUE
       KQ=KL
242    CONTINUE
       PRINT 308
243    READ (5,298) ANS
       IF (ANS.NE.DD(1).AND.ANS.NE.DD(2)) GO TO 244
       IF (ANS.NE.DD(1)) GO TO 194
       GO TO 245
244    PRINT 299
       GO TO 243
245    CONTINUE
       PRINT 321
246    READ (5,298) ANS
       IF (ANS.NE.DD(1).AND.ANS.NE.DD(2)) GO TO 249
       IF (ANS.EQ.DD(2)) GO TO 250
       DO 248 I=1,NINT
       IF (IOBJ(I).EQ.0) GO TO 247
       XX(1,I)=(XX(1,I)-(IOBJ(I)-1)*10.0)*FM
       XX(2,I)=(XX(2,I)-(IOBJ(I)-1)*10.0)*FM
       GO TO 248
247    XX(1,I)=XX(1,I)-30.0
       XX(2,I)=XX(1,I)
248    CONTINUE
```

```
      NOPT=NOPT+1
      GO TO 118
249   PRINT 299
      GO TO 246
250   RETURN
C
C
C
251   FORMAT (* DO YOU WANT TO USE DEFAULT VALUES FOR THE OPTIMIZATION.*
     1)
252   FORMAT (* FLETCHER-POWELL METHOD IS NOT AVAILABLE.*/* SUPPLY 0 FOR
     1 ANALYSIS ONLY OR 1 FOR FLETCHER OPTIMIZATION METHOD.*)
253   FORMAT (/1X,*DEFAULT VALUES ARE USED FOR THE OPTIMIZATION:*)
254   FORMAT (1H1,*YOU ARE WELCOME TO USE THE CASCADED NETWORK OPTIMIZAT
     1ION PROGRAM.*/,* ENTER YOUR DATA IN ANY FORMAT, HOWEVER, BE REASON
     2ABLE.*/* PLEASE SEPARATE EACH VALUE BY A COMMA, A BLANK OR TYPING
     3THE RETURN*/*  KEY.*/* GOOD LUCK.*//)
255   FORMAT (* DO YOU WANT TO SEE THE TABLE OF ELEMENTS AND CODE NUMBER
     1S.*)
256   FORMAT (///1X,20X,*TABLE OF ELEMENTS AND CODE NUMBERS*//,1H0,71(1H
     1-)/,1X,*ELEMENT*,15X,*CONNECTION*,7X,*CODE*,6X,*PARAMETERS*/,1X,71
     2(1H-)/,25X,*SERIES*,10X,*1*,/1X,*INDUCTOR*,41X,*INDUCTANCE*/,1X,24
     3X,*SHUNT*,11X,*4*/,1X,71(1H-)/,25X,*SERIES*,10X,*3*,/1X,*CAPACITOR
     4*,40X,*CAPACITANCE*/,1X,24X,*SHUNT*,11X,*2*,/1X,71(1H-)/,25X,*SERI
     5ES*,10X,*5*,/1X,*RESISTOR*,41X,*RESISTANCE*/,1X,24X,*SHUNT*,11X,*6
     6*,/1X,71(1H-)/,1X,*RESONANT*,16X,*SERIES*,10X,*7*,8X,*RESONANT FRE
     7QUENCY*/,1X,*RLC CIRCUIT*,38X,*QUALITY FACTOR*/,1X,24X,*SHUNT*,10X
     8,*10*,8X,*SLOPE REACTANCE*/,1X,71(1H-)/,1X,*ANTIRESONANT*,12X,*SER
     9IES*,10X,*9*,8X,*ANTIRESONANT FREQUENCY*/,1X,*RLC CIRCUIT*,38X,*QU
     $ALITY FACTOR*/,1X,24X,*SHUNT*,11X,*8*,8X,*SLOPE SUSCEPTANCE*/,1X,7
     $1(1H-))
257   FORMAT (1H0,*RESONANT*,16X,*SERIES*,9X,*16*,8X,*RESONANT FREQUENCY
     1*,/,1X,*LC CIRCUIT*,14X,*SHUNT*,10X,*19*,8X,*SLOPE REACTANCE*/1X,7
     21(1H-)/,1X,*ANTIRESONANT*,12X,*SERIES*,9X,*18*,8X,*ANTIRESONANT FR
     3EQUENCY*/,1X,*LC CIRCUIT*,14X,*SHUNT*,10X,*19*,8X,*SLOPE SUSCEPTAN
     4CE*/1X,71(1H-))
258   FORMAT (1H0,24X,*SERIES*,9X,*11*/,1X,24X,*SHORTED*,//,1X,24X,*SHUN
     1T*,10X,*14*/,1X,24X,*SHORTED*/,1X,*LOSSLESS*/,1X,*TRANSMISSION*,2X
     2,33(1H-),2X,*LENGTH*/,1X,*LINE*,20X,*SERIES*,9X,*13*/,1X,24X,*OPEN
     3ED*,/1X,49X,*CHARACTERISTIC*/,1X,24X,*SHUNT*,10X,*12*,8X,*IMPEDANC
     4E*/,1X,24X,*OPENED*/,1X,14X,33(1H-)/,1X,24X,*CASCADE*,8X,*15*/,1X,
     571(1H-)//)
259   FORMAT (* DO YOU WANT QUESTIONS FULLY WORDED TO BE PRINTED OUT.*)
260   FORMAT (I3,*) *)
261   FORMAT (* SPECIFY THE NUMBER OF ELEMENTS IN THE CIRCUIT NOT INCLUD
     1ING C- AND D-SECTIONS.*)
262   FORMAT (* SET TO 0 IF YOU DO NOT WANT ANY.*)
263   FORMAT (* SUPPLY A SEQUENCE OF*,I4,1X,*CODE NUMBERS OF ELEMENTS TO
     1 BE CONNECTED SEQUENTIALLY*/* FROM SOURCE TO LOAD.*)
264   FORMAT (* (SEE TABLE FOR ELEMENTS AND CODE NUMBERS.)*)
265   FORMAT (* IS DATA OK. *
266   FORMAT (* SPECIFY VALUES OF*,I4,1X,*PARAMETERS IN THE CIRCUIT INCL
     1UDING STARTING VALUES*/* FOR VARIABLES. (FOLLOW THE SUPPLIED SEQUE
     2NCE OF THE CODE NUMBERS OF ELEMENTS.)*/* (SEE TABLE FOR THE SEQUEN
     3CE OF PARAMETERS.)*)
267   FORMAT (* INDICATE WHICH OF THE*,I4,* PARAMETERS ARE FIXED OR VARI
     1ABLE.*/* SET TO 0 IF FIXED AND A POSITIVE INTEGER IF VARIABLE SUCH
     2*/* THAT THE INTEGER INDICATES WHETHER THE VARIABLE IS NEW OR REPE
     3ATED.*)
268   FORMAT (* SPECIFY THE NUMBER OF C-SECTIONS.*)
269   FORMAT (* IS THE NUMBER OF D-SECTION THE SAME AS DEFINED PREVIOUSL
     1Y.*)
270   FORMAT (* SPECIFY THE NUMBER OF D-SECTIONS.*)
271   FORMAT (* INDICATE WHETHER C- AND D-SECTION PARAMETERS ARE ALL FIX
     1ED OR VARIABLE.*/* SET TO 0 IF FIXED AND 1 IF VARIABLE.*)
272   FORMAT (* SPECIFY*,I4,1X,*VALUES OF THE PARAMETERS OF THE C- AND*,
     1/1X,I4,1X,*VALUES OF THE D-SECTIONS AND A D-LEVEL.*)
273   FORMAT (* SPECIFY THE LOAD RESISTANCE.*)
274   FORMAT (* SPECIFY THE NUMBER OF FREQUENCY BANDS OR INTERVALS.*)
275   FORMAT (* SPECIFY THE NUMBER OF OTHER FREQUENCY POINTS AND CONSTRA
     1INTS.*)
276   FORMAT (* DID YOU CHANGE THE NUMBER OF INTERVALS.*)
277   FORMAT (* DO YOU INTEND TO CHANGE THE NUMBER OF INTERVALS.*)
278   FORMAT (* FOR EACH INTERVAL, SUPPLY THE FOLLOWING INFORMATION:*/*
     11.  LOWER FREQUENCY BOUND (BAND EDGE),*/* 2.  UPPER FREQUENCY BOUN
     2D (BAND EDGE),*/* 3.  NUMBER OF SUBINTERVALS (EQUALS SAMPLE POINTS
     3 MINUS ONE),*/* 4.  PERFORMANCE SPECIFICATION,*/* 5.  WEIGHTING FA
     4CTOR (POSITIVE). SET TO 1 IF UNSURE,*/* 6.  TYPE OF [illegible]
     5*/10X,*SET TO  1 FOR UPPER,*/10X,,*SET TO -1 FOR L[illegible]
     6TO  0 FOR SINGLE,*/* 7.  APPROXIMATING FUNCTION:*/[illegible]
     7R REFLECTION COEFFICIENT,*/10X,*SET TO 2 FOR INSERTION LOSS (DB),*
     8/10X,*SET TO 3 FOR GROUP DELAY (NSEC).*)
279   FORMAT (* 12) INTERVAL(*,I2,*)*,4X)
280   FORMAT (* FOR EACH FREQUENCY POINT AND CONSTRAINT SUPPLY THE FOLLO
     1WING INFORMATION:*/* 1.  FREQUENCY,*/* 2.  PERFORMANCE SPECIFICATI
     2ON OR CONSTRAINT,*/* 3.  WEIGHTING FACTOR (POSITIVE). SET TO 1 IF
     3UNSURE,*/* 4.  TYPE OF SPECIFICATION OR CONSTRAINT:*/10X,*SET TO
     41 FOR UPPER,*/10X,*SET TO -1 FOR LOWER,*/10X,*SET TO  0 FOR SINGLE
     5,*/* 5.  APPROXIMATING FUNCTION:*/10X,*SET TO 1 FOR REFLECTION COE
     6FFICIENT,*/10X,*SET TO 2 FOR INSERTION LOSS (DB),*/10X,*SET TO 3 F
     7OR GROUP DELAY (NSEC),*/10X,*SET TO 0 FOR PARAMETER CONSTRAINT.*)
281   FORMAT (* 13) FREQUENCY POINT(*,I2,*)*,4X)
282   FORMAT (* SPECIFY THE CENTER FREQUENCY (FOR NORMALIZATION).*)
283   FORMAT (* SPECIFY THE CUT-OFF FREQUENCY FOR C- AND D-SECTIONS.*)
284   FORMAT (* SET TO 1 IF YOU WANT OPTIMIZATION.*/* SET TO 0 IF OPTIMI
     1ZATION IS NOT TO BE USED.*)
285   FORMAT (* SPECIFY*,I4,1X,*SMALL QUANTITIES FOR TESTING CONVERGENCE
     1 IN THE FLETCHER METHOD.*/* (E.G., 1.E-4)*)
286   FORMAT (* SPECIFY THE NUMBER OF ITERATIONS AFTER WHICH YOU WANT AN
     1 INTERMEDIATE OUTPUT */* TO BE PRINTED OUT.*/* SET TO 0 IF NO INTE
```

```
      2RMEDIATE OUTPUT IS DESIRED.*)
287    FORMAT (* SPECIFY THE MAXIMUM NUMBER OF ITERATIONS.*/* (E.G., 100)
      1*)
288    FORMAT (* SUPPLY A REALISTIC UNDER-ESTIMATE (LOWER BOUND) OF THE V
      1ALUE*/* OF THE OBJECTIVE FUNCTION.*)
289    FORMAT (* SPECIFY THE DIFFERENCE IN OBJECTIVE FUNCTION IN SUCCESSI
      1VE OPTIMIZATIONS.*/* SET TO 0 IF NOT SURE.*)
290    FORMAT (* SPECIFY A SMALL QUANTITY BY WHICH SPECIFICATIONS WOULD B
      1E SHIFTED ARTIFICIALLY.*/* SET TO 0 IF NOT SURE.*)
291    FORMAT (* SPECIFY THE NUMBER OF COMPLETE OPTIMIZATIONS.*)
292    FORMAT (* SUPPLY VALUES OF P (POSITIVE INTEGER, GREATER THAN ONE)*
      1/* TO BE USED SUCCESSIVELY FOR EACH COMPLETE OPTIMIZATION.*)
293    FORMAT (* DO YOU WANT TO READ DATA FOR ALL*,I3,* INTERVALS.*)
294    FORMAT (* WHICH INTERVAL.*)
295    FORMAT (* DO YOU WANT TO READ DATA FOR ALL*,I3,* FREQUENCY POINTS
      1AND CONSTRAINTS.*)
296    FORMAT (* WHICH FREQUENCY POINT.*)
297    FORMAT (////* ANY MODIFICATION*/* ≠ *)
298    FORMAT (A2)
299    FORMAT (*       -- PLEASE RETYPE --*,4X)
300    FORMAT (* WHICH ENTRY*/* ≠ *)
301    FORMAT (* DO YOU WANT THE STARTING VALUES FOR VARIABLES TO BE THE
      1SAME*/* AS ORIGINALLY DEFINED.*)
302    FORMAT (* DO YOU WANT THE STARTING VALUES FOR VARIABLES TO BE THE
      1SAME*/* AS THOSE OBTAINED IN THE LAST PRINTOUT.*)
303    FORMAT (* SPECIFY VALUES OF*,I4,* VARIABLE PARAMETERS IN THE CIRCU
      1IT.*/)
304    FORMAT (* SUPPLY DATA AGAIN.*/)
305    FORMAT (* DO YOU WANT TO PRINT OUT YOUR INPUT DATA.*/* ≠ *)
306    FORMAT (//,* YOUR DATA IS NOW BEING PROCESSED.  IT MAY TAKE SOME T
      1IME BEFORE*/* RESULTS ARE AVAILABLE.  PLEASE BE PATIENT.*)
307    FORMAT (* NO OPTIMIZATION METHOD TO BE USED, ONLY ANALYSIS IS REQU
      1IRED.*)
308    FORMAT (* DO YOU WANT A PLOT.*)
309    FORMAT (* DO YOU WANT TO PLOT THE ABOVE RFSPONSE.*)
310    FORMAT (* ENTER 1 FOR REFLECTION COEFFICIENT, OR 2 FOR INSERTION
      1LOSS*/* OR 3 FOR THE GROUP DELAY.*)
311    FORMAT (* INTERVAL(*,I2,*)*,4X)
312    FORMAT (* SUPPLY THE FOLLOWING INFORMATION:*/*  1. LOWER FREQUENCY
      1,*/*  2. NUMBER OF SUBINTERVALS (EQUALS SAMPLE POINTS MINUS ONE),
      2*/*     SET TO 0 FOR SINGLE FREQUENCY (POSITIVE) POINT,*/*  3. UPPE
      3R FREQUENCY (IF NO. OF SUBINTERVALS DIFFERENT THAN 0)*//*    NOTE:
      4 NEGATIVE LOWER FREQUENCY (NOT  COUNTED)*/* WILL INDICATE THAT NO
      5MORE INTERVALS WOULD BE SUPPLIED.*)
313    FORMAT (* DO YOU WANT TO SCALE AUTOMATICALLY.*)
314    FORMAT (* SUPPLY MINIMUM AND MAXIMUM RESPONSE VALUES TO BE PLOTTED
      1.*)
315    FORMAT (* DO YOU WANT MORE PLOTS.*)
316    FORMAT (* DO YOU WANT SOME MORE DETAILS FROM THE ABOVE PLOT.*)
317    FORMAT (* ENTER MINIMUM AND MAXIMUM RESPONSE RANGE TO BE PLOTTED.*
      1)
318    FORMAT (* DO YOU WANT DIFFERENT FREQUENCY RANGE AND/OR DIFFERENT R
      1ESPONSE.*)
319    FORMAT (///* VARIABLE PARAMETERS*/7X,*IN TOTAL*/)
320    FORMAT (E16.8)
321    FORMAT (1H0/,* DO YOU WANT TO TERMINATE THE PROGRAM*/* ≠ *)
322    FORMAT (1H1,10HINPUT DATA,/,1X,10(1H-)/)
323    FORMAT (* FLETCHER METHOD WILL BE USED*)
324    FORMAT (* MAXIMUM NUMBER OF ALLOWABLE ITERATIONS*,18X,I5)
325    FORMAT (* TEST QUANTITIES TO BE USED IN FLETCHER METHOD*)
326    FORMAT (1H ,51X,E14.6)
327    FORMAT (* ESTIMATE OF LOWER BOUND ON FUNCTION TO BE MINIMIZED*,E14
      1.6)
328    FORMAT (1H0,49HNONE OF THE OPTIMIZATION METHODS HAVE BEEN CALLED,/
      1,1X,29HPLEASE CHECK QUESTION NO. 16.,/,1X,9HREMAINDER,/,1X,41H
      21      FLETCHER METHOD WOULD BE CALLED.)
329    FORMAT (1H1)
330    FORMAT (1H0,31HOPTIMIZATION BY FLETCHER METHOD,/,1H0,31(1H-))
331    FORMAT (1H ,*ITER.*,2X,*FUNCT.*,3X,*TIME ELAPSED*,4X,*OBJECTIVE*,6
      1X,*VARIABLE*,7X,*GRADIENT*/1H ,1X,*NO.*,3X,*EVALU.*,5X,*(SECONDS)*
      2,5X,*FUNCTION*,8X,*VECTOR*,9X,*VECTOR*,/)
332    FORMAT (1H0/,1X,16HOPTIMUM SOLUTION,/,1X,16(1H-)/)
333    FORMAT (1H ,*ITER.*,2X,*FUNCT.*,5X,*EXECUTION*,5X,*OBJECTIVE*,6X,*
      1VARIABLE*,7X,*GRADIENT*/1H ,1X,*NO.*,3X,*EVALU.*,5X,*TIME(SEC)*,5X
      2,*FUNCTION*,8X,*VECTOR*,9X,*VECTOR*,/)
334    FORMAT (1H1,25HRESULTS AT LAST ITERATION/,1X,25(1H-))
335    FORMAT (1H ,I3,5X,I3,6X,E10.3,1X,E14.6,1X,80(E14.6,1X,E14.6/,44X))
336    FORMAT (//,* VALUE OF Q*,I12)
337    FORMAT (6X,E14.6,13X,E14.6)
338    FORMAT (/,10X,*FREQUENCY*,14X,*REFLECTION COEFF.*)
339    FORMAT (/,10X,*FREQUENCY*,15X,*INSERTION LOSS*)
340    FORMAT (/,10X,*FREQUENCY*,17X,*GROUP DELAY*)
341    FORMAT (1H1,13X,*RESPONSE AT THE STARTING POINT*,/14X,30(*-*))
342    FORMAT (1H1,13X,*FINAL RESPONSE OF THE CIRCUIT*,/14X,29(*-*))
343    FORMAT (* NUMBER OF ELEMENTS*,40X,I3)
344    FORMAT (* THE CALCULATED NUMBER OF PARAMETERS*,21X,I5)
345    FORMAT (42X,E14.6,3X,A10)
346    FORMAT (* NUMBER OF C-SECTIONS*,36X,I5)
347    FORMAT (* NUMBER OF D-SECTIONS*,36X,I5)
348    FORMAT (1X,*PARAMETERS OF THE C AND/OR D SECTIONS*)
349    FORMAT (* LOAD RESISTANCE*,36X,E14.6)
350    FORMAT (* NUMBER OF FREQUENCY INTERVALS*,27X,I5)
351    FORMAT (* NUMBER OF FREQUENCY POINTS*,30X,I5)
352    FORMAT (6X,*LOWER*,9X,*UPPER*,4X,*NO. OF*,5X,*SPECIFICATION*,12X,*
      1TYPE*,3X,*WEIGHTING*,/1X,3X,*FREQUENCY*,5X,*FREQUENCY*,2X,*SUBINT.
      2*,38X,*FACTOR*)
353    FORMAT (4X,9HFREQUENCY,12X,13HSPECIFICATION,10X,4HTYPE,5X,16HWEIGH
      1TING FACTOR)
354    FORMAT (1X,E13.6,1X,E13.6,1X,I5,1X,E11.4,1X,A10,A7,1X,A6,1X,E9.2)
355    FORMAT (1H ,2E14.6,2X,A10,A7,A6,3X,E14.6)
```

```
356    FORMAT (* CENTER FREQUENCY*,35X,E14.6)
357    FORMAT (* CUT-OFF FREQUENCY*,34X,E14.6)
358    FORMAT (1X,*DIFFERENCE IN THE OBJECTIVE FUNCTION*,/1X,*IN SUCCESSI
      1VE OPTIMIZATIONS*,24X,E14.6)
359    FORMAT (* ARTIFICIAL MARGIN*,34X,E14.6)
360    FORMAT (* NUMBER OF COMPLETE OPTIMIZATIONS*,24X,I5)
361    FORMAT (1X,*VALUES OF P*)
362    FORMAT (57X,I5)
363    FORMAT (* THE CALCULATED TOTAL NUMBER OF INTERVALS*,16X,I5)
364    FORMAT (3X,4HCODE,9X,9HPARAMETER,10X,9HPARAMETER,10X,9HPARAMETER,/
      12X,6HNUMBER,9X,6HNUMBER,14X,5HVALUE,12X,9HCONDITION)
365    FORMAT (4X,I2,12X,I3,10X,E14.6,10X,A10)
366    FORMAT (1H0,*IEXIT =*,I2,/* CRITERION FOR OPTIMUM HAS BEEN SATISFI
      1ED*)
367    FORMAT (1H0,*IEXIT =*,I2,/* EITHER OF THE FOLLOWING THINGS HAS HAP
      1PENED*/* 1. EPS CHOSEN IS TOO SMALL*/* 2. MATRIX H GOES SINGULAR*)
368    FORMAT (1H0,*IEXIT =*,I2,/* MAXIMUM NUMBER OF ALLOWABLE ITERATIONS
      1 HAS BEEN EXCEEDED*)
369    FORMAT (1H0,*IEXIT =*,I2,/* FUNCTION VALUE LESS THAN MINIMUM ESTIM
      1ATED HAS BEEN DETECTED*)
       END
```

APPENDIX F

BATCH VERSION

```
      SUBROUTINE CANOP2 (AA,B,AB,FUN,WT,XX,X,X1,FX,ERROR,EHELP,AP,A,G,GR
     1AD,EPS,GGRAD,H,IC,IOBJ,NUMB)
C
C          SUBROUTINE WHICH COORDINATES
C          THE OTHER SUBROUTINES
C
      EXTERNAL OBJECT
      DIMENSION TEXT(11)
      DIMENSION A(1), G(1), EPS(1), AA(1), B(1), AB(1), IC(1), IOBJ(1),
     1H(1), GRAD(1), XX(3,1), X(1), X1(1), ERROR(1), EHELP(1), AP(1), GG
     2RAD(1), FUN(1), WT(1), NUMB(1), FX(1)
      DIMENSION IPA(25)
      COMMON /BLK/ KO,T1,KOUNT,NUMF
      COMMON /BLACK/ MM,NE,RL,NC,ND,KVR,FM,WC,MET
      COMMON /EXT/ APP,PSI,EMAX,N,NINT,IP
      COMMON /S16/ PIE,WCC,WCSQ,WCS1,FMC
      COMMON /TEST/ FREQ(150),RESP(150,1),ICALC,XLOG10,IDENT
      COMMON /TOTAL/ IRE,NTOT,ATOT(25),GRADT(25),ICC(25)
      LOGICAL B
      LOGICAL CONV,UNITH
      COMPLEX RL
      DATA F,V/10HFIXED     ,10HVARIABLE  /
      DATA TEXT/10HPARAMETER ,10HREFLECTION,10HINSERTION ,10HGROUP DELA,
     17HVALUE  ,7H COEFF.,7HLOSS   ,7HY      ,6HLOWER ,6HSINGLE,6HUPPER
     2/
      ERR(Z)=ERROX(Z,IINT,FUN,WT,A,N1,GRAD,APP,PSI,XX,1,IC,AA,B,AB)
      UNITH=.TRUE.
      T1=0.
      PSI=0.
      ITER=1
      IPA(1)=2
      K=0
      KK=0
      NE=0
      NINT=0
      WRITE (6,69)
      READ (5,86) MM
      WRITE (6,97) MM
      IF (MM.EQ.0) GO TO 14
      READ (5,86) (IC(L),L=1,MM)
      DO 6 I=1,MM
      IF (IC(I).LE.6) 1,2
1     NUMB(NE+1)=IC(I)
      NE=NE+1
2     IF (IC(I).GE.7.AND.IC(I).LE.10) 3,4
3     NUMB(NE+1)=IC(I)
      NUMB(NE+2)=IC(I)
      NUMB(NE+3)=IC(I)
      NE=NE+3
4     IF (IC(I).GE.11.AND.IC(I).LE.19) 5,6
5     NUMB(NE+1)=IC(I)
      NUMB(NE+2)=IC(I)
      NE=NE+2
6     CONTINUE
      WRITE (6,98) NE
      READ (5,87) (AA(I),I=1,NE)
      READ (5,86) (IC(MM+I),I=1,NE)
      IRE=0
      NRE=1
      WRITE (6,118)
      DO 13 I=1,NE
      B(I)=.TRUE.
      IF (IC(MM+I).EQ.0) B(I)=.FALSE.
      IF (B(I)) 7,12
7     K=K+1
      ICC(K)=IC(MM+I)
      ATOT(K)=AA(I)
      KCU=K-1
      IF (KCU) 8,8,9
8     A(1)=ATOT(1)
      GO TO 11
9     NR=0
      DO 10 KRE=1,KCU
      IF (ICC(K).LE.ICC(KRE)) GO TO 11
      NR=NR+1
10    CONTINUE
      IF (NR.NE.KCU) GO TO 11
      NRE=NRE+1
      A(NRE)=ATOT(K)
11    WRITE (6,119) NUMB(I),I,AA(I),V
      GO TO 13
12    WRITE (6,119) NUMB(I),I,AA(I),F
13    CONTINUE
14    READ (5,86) NC,ND
      WRITE (6,100) NC
      WRITE (6,101) ND
      IF (NC.EQ.0.AND.ND.EQ.0) GO TO 18
      READ (5,86) KVR
      J=2*ND+NC+1
      READ (5,87) (AB(I),I=1,J)
      WRITE (6,102)
      IF (KVR.EQ.0) GO TO 17
      DO 15 I=1,J
      KK=KK+1
      WRITE (6,99) AB(I),V
      AA(NE+I)=AB(I)
15    CONTINUE
      KK=NE+KK
      NE1=NE+1
      DO 16 I=NE1,KK
```

```
      K=K+1
      NRE=NRE+1
      ICC(K)=NRE
      ATOT(K)=AA(I)
      A(NRE)=AA(I)
16    CONTINUE
      GO TO 13
17    WRITE (6,99) AA(I),F
18    CONTINUE
      N1=NRE
      NTOT=K
      IF (NRE.NE.NTOT) IRE=1
      READ (5,87) R
      WRITE (6,103) R
      RL=CMPLX(R,0.)
      READ (5,86) NINTD,NINTS
      WRITE (6,104) NINTD
      WRITE (6,105) NINTS
      IF (NINTD.EQ.0) GO TO 20
      WRITE (6,106)
      DO 19 I=1,NINTD
      NINT=NINT+1
      READ (5,95) XX(1,NINT),XX(2,NINT),FUN(NINT),WT(NINT),XX(3,NINT),IO
     1BJ(NINT),NUMB(NINT)
      IND1=IOBJ(NINT)+1
      IND2=IND1+4
      IND3=XX(3,NINT)+10
      WRITE (6,108) XX(1,NINT),XX(2,NINT),NUMB(NINT),FUN(NINT),TEXT(IND1
     1),TEXT(IND2),TEXT(IND3),WT(NINT)
      IF (XX(3,NINT).NE.0) GO TO 19
      XX(3,NINT)=1.
      XX(3,NINT+1)=-1.
      XX(1,NINT+1)=XX(1,NINT)
      XX(2,NINT+1)=XX(2,NINT)
      FUN(NINT+1)=FUN(NINT)
      WT(NINT+1)=WT(NINT)
      NUMB(NINT+1)=NUMB(NINT)
      IOBJ(NINT+1)=IOBJ(NINT)
      NINT=NINT+1
19    CONTINUE
20    IF (NINTS.EQ.0) GO TO 22
      WRITE (6,107)
      DO 21 I=1,NINTS
      NINT=NINT+1
      NUMB(NINT)=0
      READ (5,96) XX(1,NINT),FUN(NINT),WT(NINT),XX(3,NINT),IOBJ(NINT)
      IND1=IOBJ(NINT)+1
      IND2=IND1+4
      IND3=XX(3,NINT)+10
      WRITE (6,109) XX(1,NINT),FUN(NINT),TEXT(IND1),TEXT(IND2),TEXT(IND3
     1),WT(NINT)
      XX(2,NINT)=XX(1,NINT)
      IF (XX(3,NINT).NE.0) GO TO 21
      XX(3,NINT)=1.
      XX(3,NINT+1)=-1.
      XX(1,NINT+1)=XX(1,NINT)
      XX(2,NINT+1)=XX(2,NINT)
      FUN(NINT+1)=FUN(NINT)
      WT(NINT+1)=WT(NINT)
      NUMB(NINT+1)=0
      IOBJ(NINT+1)=IOBJ(NINT)
      NINT=NINT+1
21    CONTINUE
22    WRITE (6,117) NINT
      READ (5,87) FM,WC
      WRITE (6,110) FM
      IF (NC.NE.0.[illegible].NE.0) WRITE (6,111) WC
      READ (5,86) [illegible]
      IF (MET.EQ.0) GO TO 26
      IF (MET.EQ.2) GO TO 37
      IF (MET.EQ.1) GO TO 23
      GO TO 25
23    WRITE (6,70)
      IF (IDEF.NE.0) PRINT 65
      MAX=100
      IPRINT=0
      IF (IDEF.EQ.0) READ (5,86) MAX,IPRINT
      DO 24 I=1,N1
      EPS(I)=1.E-4
24    CONTINUE
      IF (IDEF.EQ.0) READ (5,87) (EPS(I),I=1,N1)
      WRITE (6,73)
      WRITE (6,74) (EPS(I),I=1,N1)
25    CONTINUE
      IF (MET.NE.1.AND.MET.NE.0) GO TO 37
      IF (MET.NE.1) GO TO 26
      EST=-0.1
      DIF=0.
      PSI=0.
      IF (IDEF.EQ.0) READ (5,87) EST,DIF,PSI
      WRITE (6,75) EST
      WRITE (6,112) DIF
      WRITE (6,113) PSI
      ITER=1
      IF (IDEF.EQ.0) READ (5,86) ITER
      WRITE (6,114) ITER
      IPA(1)=2
      IF (IDEF.EQ.0) READ (5,86) (IPA(I),I=1,ITER)
      WRITE (6,115)
      WRITE (6,116) (IPA(I),I=1,ITER)
```

```
      WRITE (6,71) MAX
      WRITE (6,72) IPRINT
 26   CONTINUE
      DO 28 I=1,NINT
      IF (IOBJ(I).EQ.0) GO TO 27
      XX(1,I)=XX(1,I)/FM+(IOBJ(I)-1)*10.
      XX(2,I)=XX(2,I)/FM+(IOBJ(I)-1)*10.
      GO TO 28
 27   XX(1,I)=XX(1,I)+30.
      XX(2,I)=XX(1,I)
 28   CONTINUE
      K=0
      PIE=4.*ATAN(1.)
      WCC=WC/FM
      WCSQ=WCC*WCC
      WCS1=1.-WCSQ
      FMC=FM*0.001
      XLOG10=ALOG(10.0)
      PMAX=0.
      PMIN=0.
      ICALC=0
      DO 29 I=1,NINT
      PMIN=AMIN1(PMIN,FUN(I))
      IF (XX(3,I).NE.1.) GO TO 29
      PMAX=AMAX1(PMAX,FUN(I))
 29   CONTINUE
      WRITE (6,93)
      DO 35 J=1,NINT
      IINT=J
      IF (J-1) 32,31,30
 30   IF (IOBJ(J)-IOBJ(J-1)) 31,32,31
 31   IF (XX(1,J).LE.10.) WRITE (6,90)
      IF (XX(1,J).LE.20.0.AND.XX(1,J).GT.10.) WRITE (6,91)
      IF (XX(1,J).LE.30.0.AND.XX(1,J).GT.20.) WRITE (6,92)
 32   L=NUMB(J)+1
      IF (NUMB(J).EQ.0) Z=XX(1,J)
      DO 35 I=1,L
      IF (NUMB(J).GT.0) Z=XX(1,J)+(XX(2,J)-XX(1,J))*(I-1)/NUMB(J)
      K=K+1
      X(K)=Z
      ERROR(K)=ERR(Z)
      EHELP(K)=ERROR(K)*XX(3,J)
      ER=ERROR(K)
      IF (PSI.NE.0.) ER=ER+PSI*XX(3,IINT)
      ERT=(ER/WT(IINT))+FUN(IINT)
      IF (IOBJ(J).EQ.1) FX(K)=Z*FM
      IF (IOBJ(J).EQ.2) FX(K)=(Z-10.)*FM
      IF (IOBJ(J).EQ.3) FX(K)=(Z-20.)*FM
      IF (IOBJ(J).EQ.0) GO TO 34
      IF (K.EQ.1) GO TO 33
      IF (FX(K).EQ.FX(K-1)) GO TO 34
 33   WRITE (6,89) FX(K),ERT
      ICALC=ICALC+1
      FREQ(ICALC)=FX(K)
      RESP(ICALC)=ERT
 34   CONTINUE
 35   AP(K)=APP
      EMAX=EHELP(1)
      DO 36 M=2,K
      EMAX=AMAX1(EMAX,EHELP(M))
 36   CONTINUE
      IF (ITER.GT.25) GO TO 64
C
C          OPTIMIZATION
C
      DO 63 KK=1,ITER
      IP=IPA(KK)
C
C          PRINTS THE INPUT DATA
C          FOR THE OPTIMIZATION PROCESS
      IF (MET.NE.1) GO TO 37
      GO TO 39
 37   IF (MET.NE.0) GO TO 38
      WRITE (6,66)
      GO TO 53
 38   WRITE (6,76)
      CALL EXIT
C
C
 39   CALL SECOND (T1)
      WRITE (6,77)
      WRITE (6,78)
      IF (IPRINT.EQ.0) GO TO 41
C
C          PRINTS THE INTERMEDIATE RESULTS
C          FOR THE FLETCHER METHOD
C
      IF (T1.EQ.0.) GO TO 40
      WRITE (6,79)
      GO TO 41
 40   WRITE (6,80)
 41   CONTINUE
      CALL OPTIM1 (N1,A,F,G,H,UNITH,EST,EPS,MAX,IPRINT,IEXIT,GRAD,NUMB,X
     1X,X,X1,ERROR,EHELP,AP,GGRAD,FUN,WT,IC,AA,B,AB,1)
      CALL SECOND (T2)
C
C          PRINTS THE RESULTS
C          FOR THE OPTIMIZATION PROCESS
C
      IF (KO.EQ.0) GO TO 42
```

```
      WRITE (6,81)
      GO TO 43
42    WRITE (6,83)
43    CONTINUE
      T=T2-T1
      IF (T1.EQ.0.) GO TO 44
      WRITE (6,82)
      WRITE (6,84) KOUNT,NUMF,T,F,((A(I),G(I)),I=1,N1)
      GO TO 45
44    WRITE (6,80)
      WRITE (6,85) KOUNT,NUMF,F,((A(I),G(I)),I=1,N1)
45    IF (IRE.EQ.0) GO TO 46
      WRITE (6,67)
      WRITE (6,68) (ATOT(I),I=1,NTOT)
46    WRITE (6,88) IP
      KQ=0
      WRITE (6,94)
      ICALC=0
      DO 52 J=1,NINT
      IINT=J
      IF (J-1) 49,48,47
47    IF (IOBJ(J)-IOBJ(J-1)) 48,49,48
48    IF (XX(1,J).LE.10.) WRITE (6,90)
      IF (XX(1,J).LE.20.0.AND.XX(1,J).GT.10.) WRITE (6,91)
      IF (XX(1,J).LE.30.0.AND.XX(1,J).GT.20.) WRITE (6,92)
49    KQ=KQ+1
      KL=KQ+NUMB(J)
      DO 51 I=KQ,KL
      L=I-KQ+1
      ER=ERR(X(I))
      IF (PSI.NE.0.) ER=ER+PSI*XX(3,IINT)
      ERT=(ER/WT(IINT))+FUN(IINT)
      IF (IOBJ(J).EQ.0) GO TO 51
      IF (I.EQ.1) GO TO 50
      IF (FX(I).EQ.FX(I-1)) GO TO 51
50    WRITE (6,89) FX(I),ERT
      ICALC=ICALC+1
      FREQ(ICALC)=FX(I)
      RESP(ICALC)=ERT
51    CONTINUE
      KQ=KL
52    CONTINUE
C
C          PLOT OF THE FINAL RESPONSE
C
53    IDENT=0
      DO 61 I=2,ICALC
      IF (FREQ(I)-FREQ(I-1)) 54,61,61
54    IF (FREQ(I)-FREQ(1)) 55,55,57
55    POMOC=FREQ(I)
      POM=RESP(I)
      DO 56 J=2,I
      K=I-J+2
      RESP(K)=RESP(K-1)
      FREQ(K)=FREQ(K-1)
56    CONTINUE
      FREQ(1)=POMOC
      RESP(1)=POM
      GO TO 61
57    IF (FREQ(I).LT.FREQ(2).AND.I.GT.3) GO TO 58
      GO TO 60
58    POMOC=FREQ(I)
      POM=RESP(I)
      DO 59 J=3,I
      K=I-J+3
      RESP(K)=RESP(K-1)
      FREQ(K)=FREQ(K-1)
59    CONTINUE
      FREQ(2)=POMOC
      RESP(2)=POM
      GO TO 61
60    POMOC=FREQ(I)
      POM=RESP(I)
      FREQ(I)=FREQ(I-1)
      RESP(I)=RESP(I-1)
      FREQ(I-1)=POMOC
      RESP(I-1)=POM
      IDENT=1
61    CONTINUE
      IF (IDENT) 53,62,53
62    CONTINUE
      IDENT=0
      CALL PLOT (1,1,0.0,0.0)
      IF (PMIN.LT.-1.E-3) GO TO 63
      IF (IOBJ(1).EQ.3) GO TO 63
      IF (IOBJ(1).EQ.3) GO TO 63
      IDENT=1
      CALL PLOT (1,0,0.0,4.*PMAX)
63    CONTINUE
      RETURN
64    WRITE (6,120)
      CALL EXIT
C
C
C
65    FORMAT (/1X,45HDEFAULT VALUES ARE USED FOR THE OPTIMIZATION:)
66    FORMAT (//1X,25HANALYSIS IS REQUIRED ONLY//)
67    FORMAT (///1X,19HVARIABLE PARAMETERS,/7X,8HIN TOTAL/)
68    FORMAT (E16.8)
69    FORMAT (1H1,10HINPUT DATA,/,1X,10(1H-),//)
```

```
70     FORMAT (1X,28HFLETCHER METHOD WILL BE USED//)
71     FORMAT (///1H0,38HMAXIMUM NUMBER OF ALLOWABLE ITERATIONS,18X,I5//)
72     FORMAT (1H0,39HINTERMEDIATE OUTPUT TO BE PRINTED EVERY,I8,5X,10HIT
      1ERATIONS)
73     FORMAT (1H0,/,1X,45HTEST QUANTITIES TO BE USED IN FLETCHER METHOD)
74     FORMAT (1H0,51X,E16.8)
75     FORMAT (//1X,51HESTIMATE OF LOWER BOUND ON FUNCTION TO BE MINIMIZE
      1D,E16.8//)
76     FORMAT (1H0,29HPLEASE CHECK THE VALUE OF MET,/,1X,9HREMAINDER,/,1X
      1,34HMET=0    ANALYSIS IS REQUIRED ONLY,/,1X,40HMET=1    FLETCHER M
      2ETHOD WOULD BE CALLED)
77     FORMAT (1H1)
78     FORMAT (1H0,31HOPTIMIZATION BY FLETCHER METHOD,/,1H0,31(1H-))
79     FORMAT (1H0,9HITERATION,2X,8HFUNCTION,6X,12HTIME ELAPSED,8X,9HOBJE
      1CTIVE,16X,16HVARIABLE VECTOR ,13X,16HGRADIENT VECTOR ,/1X,6HNUMBER
      2,5X,11HEVALUATIONS,3X,9H(SECONDS),11X,8HFUNCTION,/)
80     FORMAT (1H0,9HITERATION,2X,8HFUNCTION,8X,9HOBJECTIVE,16X,16HVARIAB
      1LE VECTOR ,13X,16HGRADIENT VECTOR ,/1X,6HNUMBER,5X,11HEVALUATIONS,
      25X,8HFUNCTION,/)
81     FORMAT (////1X,16HOPTIMUM SOLUTION,/,1X,16(1H-)/)
82     FORMAT (1H0,9HITERATION,2X,8HFUNCTION,6X,14HEXECUTION TIME,6X,9HOB
      1JECTIVE,16X,16HVARIABLE VECTOR ,13X,16HGRADIENT VECTOR ,/1X,6HNUMB
      2ER,5X,11HEVALUATIONS,3X,9H(SECONDS),11X,8HFUNCTION,/)
83     FORMAT (////1X,25HRESULTS AT LAST ITERATION/,1X,25(1H-))
84     FORMAT (1H0,I5,7X,I5,5X,E16.8,3X,E16.8,12X,95(E16.8,13X,E16.8,/,70
      1X))
85     FORMAT (1H0,I5,7X,I5,8X,E16.8,7X,95(E16.8,13X,E16.8,/,49X))
86     FORMAT (8I10)
87     FORMAT (5E16.8)
88     FORMAT (5(/),1X,10HVALUE OF Q,I12)
89     FORMAT (15X,E16.8,10X,E16.8)
90     FORMAT (/20X,9HFREQUENCY,14X,17HREFLECTION COEFF.)
91     FORMAT (/20X,9HFREQUENCY,14X,14HINSERTION LOSS)
92     FORMAT (/20X,9HFREQUENCY,14X,11HGROUP DELAY)
93     FORMAT (1H1,23X,30HRESPONSE AT THE STARTING POINT,/23X,31(1H-))
94     FORMAT (1H1,23X,29HFINAL RESPONSE OF THE CIRCUIT,/23X,30(1H-))
95     FORMAT (4E16.6,F6.0,2I5)
96     FORMAT (4E16.8,I6)
97     FORMAT (1X,18HNUMBER OF ELEMENTS,40X,I3//)
98     FORMAT (//1X,35HTHE CALCULATED NUMBER OF PARAMETERS,20X,I5///)
99     FORMAT (52X,E16.8,3X,A10)
100    FORMAT (///1X,20HNUMBER OF C SECTIONS,36X,I5//)
101    FORMAT (1X,20HNUMBER OF D SECTIONS,36X,I5//)
102    FORMAT (1X,37HPARAMETERS OF THE C AND/OR D SECTIONS)
103    FORMAT (/1X,15HLOAD RESISTANCE,36X,E16.8//)
104    FORMAT (1X,29HNUMBER OF FREQUENCY INTERVALS,27X,I5//)
105    FORMAT (1X,26HNUMBER OF FREQUENCY POINTS,30X,I5//)
106    FORMAT (/6X,5HLOWER,13X,5HUPPER,15X,9HNUMBER OF,20X,13HSPECIFICATI
      1ON,14X,4HTYPE,10X,9HWEIGHTING,/4X,9HFREQUENCY,9X,9HFREQUENCY,11X,1
      22HSUBINTERVALS,61X,6HFACTOR/)
107    FORMAT (///4X,9HFREQUENCY,24X,13HSPECIFICATION,16X,4HTYPE,7X,16HWE
      1IGHTING FACTOR/)
108    FORMAT (2(E16.8,4X),I10,10X,E16.8,3X,A10,A7,4X,A6,4X,E16.8)
109    FORMAT (E16.8,10X,E16.8,3X,A10,A7,4X,A6,4X,E16.8)
110    FORMAT (/1X,16HCENTER FREQUENCY,35X,E16.8//)
111    FORMAT (1X,17HCUT-OFF FREQUENCY,34X,E16.8//)
112    FORMAT (1X,36HDIFFERENCE IN THE OBJECTIVE FUNCTION,/1X,27HIN SUCCE
      1SSIVE OPTIMIZATIONS,24X,E16.8//)
113    FORMAT (1X,17HARTIFICIAL MARGIN,34X,E16.8//)
114    FORMAT (1X,32HNUMBER OF COMPLETE OPTIMIZATIONS,24X,I5//)
115    FORMAT (1X,11HVALUES OF P)
116    FORMAT (57X,I5)
117    FORMAT (////1X,40HTHE CALCULATED TOTAL NUMBER OF INTERVALS,16X,I5/
      1/)
118    FORMAT (6X,4HCODE,12X,9HPARAMETER,13X,9HPARAMETER,13X,9HPARAMETER,
      1/5X,6HNUMBER,12X,6HNUMBER,17X,5HVALUE,15X,9HCONDITION/)
119    FORMAT (7X,I2,15X,I3,13X,E16.8,11X,A10)
120    FORMAT (/////5X,55HNUMBER OF COMPLETE OPTIMIZATIONS ITER EXCEEDS V
      1ALUE 25.,//5X,65HPLEASE CHANGE THE LENGTH OF ARRAY IPA(ITER) IN SU
      2BROUTINE CANOPT.)
       END
```

APPENDIX G

ROUTINES COMMON TO BOTH THE INTERACTIVE AND BATCH VERSIONS

```
      FUNCTION ERROX (Z,IINT,FUNCS,W,A,N1,GRAD,APP,PSI,XX,IPOINT,IC,AA,B
     1,AB)
C
C          FUNCTION SUBROUTINE WHICH CALCULATES
C          UPPER AND LOWER WEIGHTED ERROR FUNCTION
C
      COMMON /BLACK/ MM,NE,RL,NC,ND,KVR,FM,WC,MET
      COMMON /TOTAL/ IRE,NTOT,ATOT(25),GRADT(25),ICC(25)
      DIMENSION A(1), GRAD(1), XX(3,1), FUNCS(1), W(1), AA(1), B(1), AB(
     11), IC(1)
      IF (IPOINT) 1,7,1
    1 IF (IRE) 2,2,3
    2 CALL APPROX (Z,N1,A,APP,GRAD,IC,AA,B,AB)
      GO TO 7
    3 DO 4 K=1,NTOT
      INDEX=ICC(K)
      ATOT(K)=A(INDEX)
    4 CONTINUE
      CALL APPROX (Z,NTOT,ATOT,APP,GRADT,IC,AA,B,AB)
      IF (MET.EQ.0) GO TO 7
      DO 6 KK=1,N1
      SUMA=0.
      DO 5 K=1,NTOT
      IF (ICC(K).NE.KK) GO TO 5
      SUMA=GRADT(K)+SUMA
    5 CONTINUE
      GRAD(KK)=SUMA
    6 CONTINUE
    7 IF (PSI) 8,9,8
    8 ERROX=(APP-FUNCS(IINT))*W(IINT)-PSI*XX(3,IINT)
      RETURN
    9 ERROX=(APP-FUNCS(IINT))*W(IINT)
      RETURN
      END
```

```
      SUBROUTINE OBJECT (N1,A,OBJ,G,GRAD,NUMB,XX,X,X1,ERROR,EHELP,AP,GGR
     1AD,FUN,W,IC,AA,B,AB)
C
C          SUBROUTINE WHICH COMPUTES THE OBJECTIVE FUNCTION
C          AND ITS GRADIENTS W.R.T. THE VARIABLE PARAMETERS
C          IN THE LEAST P-TH SENSE
C
      DIMENSION A(1), GRAD(1), NUMB(1), XX(3,1), X(1), X1(1), ERROR(1),
     1EHELP(1), AP(1), GGRAD(1), G(1), FUN(1), W(1), IC(1), AA(1), B(1),
     2 AB(1)
      COMMON /BLACK/ MM,NE,RL,NC,ND,KVR,FM,WC,MET
      COMMON /EXT/ APP,PSI,EMAX,N,NINT,IP
      ERR(Z)=ERROX(Z,IINT,FUN,W,A,N1,GRAD,APP,PSI,XX,IPOINT,IC,AA,B,AB)
      MET=1
      OBJP=0.
      GRADP=0.
      DO 1 K=1,N1
      G(K)=0.
    1 CONTINUE
      INDIC=1
      IPOINT=1
      K=0
      KL=0
      KG=0
      DO 8 J=1,NINT
      IINT=J
      IF (J.EQ.1) GO TO 2
      KL=KL+L
    2 L=NUMB(J)+1
      DO 7 I=1,L
      K=K+1
      IF (J.EQ.1) GO TO 5
      DO 4 KK=1,KL
      IF (ABS(X(K)-X(KK)).GT.10E-6) GO TO 4
      AP(K)=AP(KK)
      APP=AP(K)
      KGK=(K-1)*N1+1
      KGKK=(KK-1)*N1+1
      DO 3 KN1=1,N1
      GGRAD(KGK)=GGRAD(KGKK)
      GRAD(KN1)=GGRAD(KGKK)
      KGK=KGK+1
      KGKK=KGKK+1
    3 CONTINUE
      IPOINT=0
      GO TO 5
    4 CONTINUE
    5 ERROR(K)=ERR(X(K))
      EHELP(K)=ERROR(K)*XX(3,J)
      DO 6 KN1=1,N1
```

```
      KG=KG+1
      GGRAD(KG)=GRAD(KN1)
    6 CONTINUE
      IF (IPOINT.NE.0) AP(K)=APP
      IPOINT=1
    7 CONTINUE
    8 CONTINUE
      EMAX=EHELP(1)
      DO 9 M=2,K
      EMAX=AMAX1(EMAX,EHELP(M))
    9 CONTINUE
      IF (EMAX) 10,11,11
   10 IP=-IABS(IP)
      GO TO 12
   11 IP=IABS(IP)
   12 K=0
      N=0
      DO 18 J=1,NINT
      IINT=J
      L=NUMB(J)+1
      DO 17 I=1,L
      K=K+1
      IF (IP) 14,13,13
   13 IF (EHELP(K)) 17,14,14
   14 N=N+1
      X1(N)=X(K)
      ERROR(N)=ERROR(K)
      EHELP(N)=AP(K)
      KGK=(K-1)*N1+1
      KGKK=(N-1)*N1+1
      KGN=KGKK
      DO 15 KN1=1,N1
      GGRAD(KGN)=GGRAD(KGK)
      KGK=KGK+1
      KGN=KGN+1
   15 CONTINUE
      DEL=ERROR(N)/EMAX
      DELP=DEL**(IP-1)
      OBJI=DELP*DEL
      GRADI=DELP
      OBJP=OBJP+OBJI
      KGN=KGKK
      DO 16 KN1=1,N1
      GRAD(KN1)=GRADI*W(IINT)*GGRAD(KGN)
      G(KN1)=G(KN1)+GRAD(KN1)
      KGN=KGN+1
   16 CONTINUE
   17 CONTINUE
   18 CONTINUE
      PR=1./IP
      GRP=OBJP**(PR-1.)
      OBJ=GRP*OBJP*EMAX
      DO 19 K=1,N1
      G(K)=GRP*G(K)
   19 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE OPTIM1 (N,X,F,G,H,UNITH,FEST,EPS,MAXFN,IPRINT,IEXIT,GRA
     1D,NUMB,XX,XP,X1,ERROR,EHELP,AP,GGRAD,FUN,WT,IC,AA,B,AB,INTER)
      DIMENSION X(1), G(1), H(1), EPS(1), GRAD(1), NUMB(1), XX(3,1), XP(
     11), X1(1), ERROR(1), EHELP(1), AP(1), GGRAD(1), FUN(1), WT(1), AA(
     21), B(1), AB(1), IC(1)
      LOGICAL CONV,UNITH
      COMMON /BLK/ KO,T1,ITN,NFNS
      CALL SECOND (T3)
      KO=0
      CALL OBJECT (N,X,F,G,GRAD,NUMB,XX,XP,X1,ERROR,EHELP,AP,GGRAD,FUN,W
     1T,IC,AA,B,AB)
      IF (F.LT.FEST) GO TO 26
      NFNS=1
      ITN=0
      STEP=1.
      IDX=N
      IDG=N+N
      IH=IDG+N
      IF (.NOT.UNITH) GO TO 2
      IJ=IH+1
      DO 1 I=1,N
      DO 1 J=I,N
      H(IJ)=0.
      IF (I.EQ.J) H(IJ)=1.0
    1 IJ=IJ+1
    2 CONV=.TRUE.
      GDX=0.
      DO 6 I=1,N
      Z=0.
      IJ=IH+I
      IF (I.EQ.1) GO TO 4
      II=I-1
      DO 3 J=1,II
      Z=Z-H(IJ)*G(J)
      IJ=IJ+N-J
    3 CONTINUE
    4 DO 5 J=I,N
```

```
      Z=Z-H(IJ)*G(J)
      IJ=IJ+1
5     CONTINUE
      IF (ABS(Z).GT.EPS(I)) CONV=.FALSE.
      H(IDX+I)=Z
      GDX=GDX+G(I)*Z
6     CONTINUE
C
      IF (IPRINT.EQ.0) GO TO 10
      IF (MOD(ITN,IPRINT).NE.0) GO TO 10
      CALL SECOND (T4)
      TIME=T4-T3
      IF (T1.EQ.0.) GO TO 9
      GO TO (7,8), INTER
7     WRITE (6,34) ITN,NFNS,TIME,F,((X(I),G(I)),I=1,N)
      GO TO 10
8     WRITE (6,33) ITN,NFNS,TIME,F,((X(I),G(I)),I=1,N)
      GO TO 10
9     WRITE (6,35) ITN,NFNS,F,((X(I),G(I)),I=1,N)
10    IEXIT=1
      IF (CONV) GO TO 27
      IEXIT=2
      IF (GDX.GE.0.) GO TO 27
      Z=1.
      IF (ITN.LT.N.AND.UNITH) Z=STEP
      W=2.*(FEST-F)/GDX
      IF (W.LT.Z) Z=W
      STEP=Z
11    GDX=GDX*Z
      DO 12 I=1,N
      H(IDX+I)=H(IDX+I)*Z
      X(I)=X(I)+H(IDX+I)
12    CONTINUE
      CALL OBJECT (N,X,FP,H,GRAD,NUMB,XX,XP,X1,ERROR,EHELP,AP,GGRAD,FUN,
     1WT,IC,AA,B,AB)
      IF (FP.LT.FEST) GO TO 26
      NFNS=NFNS+1
      IEXIT=3
      IF (ITN.EQ.MAXFN) GO TO 27
      GPDX=0.
      DO 13 I=1,N
      H(IDG+I)=H(I)-G(I)
      GPDX=GPDX+H(I)*H(IDX+I)
13    CONTINUE
      DGDX=GPDX-GDX
      IF (F.GT.FP-.0001*GDX) GO TO 15
      IEXIT=4
      IF (GPDX.LT.0..AND.ITN.GT.N) GO TO 27
      Z=3.*(F-FP)+GPDX+GDX
      W=SQRT(1.-GDX/Z*GPDX/Z)*ABS(Z)
      Z=1.-(GPDX+W-Z)/(DGDX+2.*W)
      IF (Z.LT.0.1) Z=0.1
      DO 14 I=1,N
      X(I)=X(I)-H(IDX+I)
14    CONTINUE
      GO TO 17
15    F=FP
      DO 16 I=1,N
      G(I)=H(I)
16    CONTINUE
      IF (DGDX.GT.0.) GO TO 18
      GDX=GPDX
      Z=4.
17    STEP=Z*STEP
      GO TO 11
18    IF (GPDX.LT.0.5*GDX) STEP=2.*STEP
      DGHDG=0.
      DO 22 I=1,N
      Z=0.
      IJ=IH+I
      IF (I.EQ.1) GO TO 20
      II=I-1
      DO 19 J=1,II
      Z=Z+H(IJ)*H(IDG+J)
      IJ=IJ+N-J
19    CONTINUE
20    DO 21 J=I,N
      Z=Z+H(IJ)*H(IDG+J)
      IJ=IJ+1
21    CONTINUE
      DGHDG=DGHDG+Z*H(IDG+I)
      H(I)=Z
22    CONTINUE
      IF (DGHDG.LT.0.0) DGHDG=DGDX*0.01
      IF (DGDX.LT.DGHDG) GO TO 24
      W=1.0+DGHDG/DGDX
      DO 23 I=1,N
      H(IDX+I)=W*H(IDX+I)-H(I)
23    CONTINUE
      DGDX=DGDX+DGHDG
      DGHDG=DGDX
24    IJ=IH
      DO 25 I=1,N
      W=H(IDX+I)/DGDX
      Z=H(I)/DGHDG
      DO 25 J=I,N
      IJ=IJ+1
25    H(IJ)=H(IJ)+W*H(IDX+J)-Z*H(J)
      ITN=ITN+1
      GO TO 2
26    IEXIT=5
27    IF (IEXIT.EQ.1) KO=1
      IF (IPRINT.EQ.0) RETURN
```

```
      GO TO (28,29,30,29,31), IEXIT
28    WRITE (6,36) IEXIT
      GO TO 32
29    WRITE (6,37) IEXIT
      GO TO 32
30    WRITE (6,38) IEXIT
      GO TO 32
31    WRITE (6,39) IEXIT
32    CONTINUE
      RETURN
C
C
C
33     FORMAT (1H ,I3,5X,I3,6X,E10.3,1X,E14.6,1X,80(E14.6,1X,E14.6/,44X))
34     FORMAT (1H0,I5,7X,I5,5X,E16.8,3X,E16.8,12X,95(E16.8,13X,E16.8,/,70
     1X))
35     FORMAT (1H0,I5,7X,I5,8X,E16.8,7X,95(E16.8,13X,E16.8,/,49X))
36     FORMAT (/,1H0,6HIEXIT=,I2,40HCRITERION FOR OPTIMUM HAS BEEN SATISF
     1IED)
37     FORMAT (/,1H0,6HIEXIT=,I2,43HEITHER OF THE FOLLOWING THINGS HAS HA
     1PPENED,/,9X,26H1. EPS CHOSEN IS TOO SMALL,/,9X,25H2. MATRIX H GOES
     2 SINGULAR)
38     FORMAT (/,1H0,6HIEXIT=,I2,56HMAXIMUM NUMBER OF ALLOWABLE ITERATION
     1S HAS BEEN EXCEEDED)
39     FORMAT (/,1H0,6HIEXIT=,I2,61HFUNCTION VALUE LESS THAN MINIMUM ESTI
     1MATED HAS BEEN DETECTED )
       END


       SUBROUTINE APPROX (OMEGN,N1,X,APP,GRAD,IC,A,B,AB)
       DIMENSION X(1), GRAD(1), IC(1), A(1), B(1), AB(1)
       DIMENSION AD(50), ADJJ(50), G(50)
       COMPLEX AD,ADJJ,CONRHO,RHO,I,IHAT,V,VHAT,INEW,VNEW,IOLD,VOLD,G,V1,
     1V2,IHAT1,IHAT2,PSIL,RL,GL,GC,GLLTEL,GLLTZO,GSCTEL,GSCTZO,GOCTEL,GO
     2CTZO,GLCPBP,GLCPQ,GLCPOR,GLCSQ,GLCSOR,GLCSXP
       COMPLEX POLD,PHASE
       COMMON /BLACK/ M,NE,RL,NC,ND,KVR,FM,WC,MET
       COMMON /S16/ PIE,WCC,WCS0,WCS1,FMC
       LOGICAL B
C
C      M IS THE TOTAL NUMBER OF ELEMENTS IN THE CIRCUIT
C       NE IS THE TOTAL NUMBER OF ELEMENTS IN THE CIRCUIT
C       A ARRAY CONTAINS PARAMETER VALUES
C      B ARRAY CONTAINS LOGICAL VARIABLES
C      IC ARRAY CONTAINS CODE NUMBER GIVING ORDER IN WHICH BLOCKS ARE CON
C      NECTED
C
C      FOLLOWING FUNCTION STATEMENTS DEFINE SENSITIVITIES
C
       THETA(EL)=PIE/2.*OMEGA*EL
       GC(OMEGA,V,VHAT)=-CMPLX(0.,OMEGA)*V*VHAT
       GL(OMEGA,I,IHAT)=CMPLX(0.,OMEGA)*I*IHAT
       GLLTZO(V1,V2,IHAT1,IHAT2,ZO)=(V1*IHAT1-V2*IHAT2)/ZO
       GLLTEL(OMEGA,EL,V1,V2,IHAT1,IHAT2)=PIE/2.*OMEGA/SIN(THETA(EL))*(V1
     1*IHAT2-V2*IHAT1)
       GSCTEL(OMEGA,EL,ZO,I,IHAT)=PIE/2.*(1./COS(THETA(EL)))**2*CMPLX(0.,
     1ZO*OMEGA)*I*IHAT
       GSCTZO(EL,I,IHAT)=CMPLX(0.,SIN(THETA(EL))/COS(THETA(EL)))*I*IHAT
       GOCTEL(ZO,EL,OMEGA,I,IHAT)=CMPLX(0.,ZO*THETA(EL)/EL*(1./SIN(THETA(
     1EL)))**2)*I*IHAT
       GOCTZO(OMEGA,EL,I,IHAT)=CMPLX(0.,-COS(THETA(EL))/SIN(THETA(EL)))*I
     1*IHAT
       GLCPBP(BP,OMEGA,Q,OMEGAR,V,VHAT)=-V*VHAT*CMPLX((OMEGAR/Q),((OMEGA*
     1OMEGA-OMEGAR*OMEGAR)/OMEGA))/2.0
       GLCPQ(BP,Q,OMEGAR,V,VHAT)=V*VHAT*CMPLX(BP*OMEGAR/(2.*Q*Q),0.)
       GLCPOR(BP,OMEGA,OMEGAR,Q,V,VHAT)=-BP*V*VHAT*CMPLX(1./(2.*Q),-OMEGA
     1R/OMEGA)
       GLCSQ(XP,OMEGAR,Q,I,IHAT)=-CMPLX((XP*OMEGAR)/(2.*Q*Q),0.)*I*IHAT
       GLCSOR(OMEGA,XP,OMEGAR,Q,I,IHAT)=CMPLX((XP/(2.*Q)),-OMEGAR*XP/OMEG
     1A)*I*IHAT
       GLCSXP(OMEGA,OMEGAR,Q,I,IHAT)=CMPLX(OMEGAR/Q,((OMEGA*OMEGA-OMEGAR*
     1OMEGAR)/OMEGA))*I*IHAT/2.
       GESXP(OMEGA,OMEGAR,I,IHAT)=CMPLX(0.,(OMEGA*OMEGA-OMEGAR*OMEGAR)/OM
     1EGA)*I*IHAT/2.
       GESOR(OMEGA,XP,OMEGAR,I,IHAT)=CMPLX(0.,-XP*OMEGAR/OMEGA)*I*IHAT
       GEPBP(OMEGA,OMEGAR,V,VHAT)=-CMPLX(0.,(OMEGA*OMEGA-OMEGAR*OMEGAR)/O
     1MEGA)*V*VHAT/2.
       GEPOR(BP,OMEGA,OMEGAR,V,VHAT)=V*VHAT*CMPLX(0.,BP*OMEGAR/OMEGA)
       IF (N1.GT.50) GO TO 149
```

```
      OLDEL=0.
      GD1=0.
      GD2=0.
      N=N1
      IF (KVR.EQ.1) N=N1-NC-2*ND-1
      IF (OMEGN.LE.30.) GO TO 1
      GO TO 147
1     CONTINUE
      OMEGA=OMEGN
      IF (OMEGN.GT.10..AND.OMEGN.LE.20.) OMEGA=OMEGN-10.
      IF (OMEGN.GT.20.) OMEGA=OMEGN-20.
      IF (M.EQ.0) GO TO 84
      DELO=1.E-7*OMEGA
      KKGD=0
      IF (KKGD.EQ.0) GO TO 3
2     DO 83 KKV=1,N
      KKGD=KKGD+1
      DELX=1.E-4*X(KKV)
      X(KKV)=X(KKV)+DELX
3     CONTINUE
      KGD=0
      IF (OMEGN.LE.20.) GO TO 4
      OMEGA=OMEGA-DELO
4     CONTINUE
      DO 82 KV=1,2
      J=0
      K=0
      VOLD=RL
      IOLD=1.0
      DO 73 L=1,M
      MM=M+1-L
      NN=IC(MM)
      GO TO (5,5,5,5,5,5,20,20,20,20,39,39,39,39,39,39,39,39,39), NN
5     KKK=NE-K
      IF (B(KKK)) GO TO 6
      GO TO 7
6     JJ=N-J
      A(KKK)=X(JJ)
      J=J+1
7     GO TO (8,9,10,11,15,16), NN
8     CALL CODE1 (IOLD,VOLD,A(KKK),OMEGA,INEW,VNEW)
      GO TO 12
9     CALL CODE2 (IOLD,VOLD,A(KKK),OMEGA,INEW,VNEW)
      GO TO 12
10    CALL CODE3 (IOLD,VOLD,A(KKK),OMEGA,INEW,VNEW)
      VOLD=VNEW-VOLD
      GO TO 12
11    CALL CODE4 (IOLD,VOLD,A(KKK),OMEGA,INEW,VNEW)
      IOLD=INEW-IOLD
12    IF (B(KKK).AND.OMEGN.GT.10.) GO TO 14
      IF (B(KKK)) GO TO 13
      GO TO 19
13    IF (NN.EQ.1.OR.NN.EQ.4) G(JJ)=GL(OMEGA,IOLD,IOLD)
      IF (NN.EQ.2.OR.NN.EQ.3) G(JJ)=GC(OMEGA,VOLD,VOLD)
14    IF (NN.EQ.1.OR.NN.EQ.4) AD(JJ)=IOLD
      IF (NN.EQ.2.OR.NN.EQ.3) AD(JJ)=VOLD
      GO TO 19
15    CALL CODE5 (IOLD,VOLD,A(KKK),INEW,VNEW)
      GO TO 17
16    CALL CODE6 (IOLD,VOLD,A(KKK),INEW,VNEW)
      IOLD=INEW-IOLD
17    IF (B(KKK)) GO TO 18
      GO TO 19
18    G(JJ)=IOLD*IOLD
      AD(JJ)=IOLD
19    KN=K+1
      GO TO 72
20    KK=K
      IOLD=J
      DO 23 II=1,3
      KKK=NE-KK
      IF (B(KKK)) GO TO 21
      GO TO 22
21    JJ=N-J
      A(KKK)=X(JJ)
      J=J+1
22    KK=KK+1
23    CONTINUE
      JK=NE-K-2
      JV=NE-K-1
      JH=NE-K
      NI=NN-6
      GO TO (24,25,26,27), NI
24    CALL CODE7 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      GO TO 28
25    CALL CODE8 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      GO TO 28
26    CALL CODE9 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      VOLD=VNEW-VOLD
      GO TO 28
27    CALL CODE10 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      IOLD=INEW-IOLD
28    CONTINUE
      IF (NN.EQ.7.OR.NN.EQ.10) NI=7
      IF (NN.EQ.8.OR.NN.EQ.9) NI=8
      DO 38 II=1,3
      GO TO (29,30,31), II
29    KVH=JH
      GO TO 32
30    KVH=JV
```

```
      GO TO 32
31    KVH=JK
32    CONTINUE
      NJ=N-JOLD
      IF (B(KVH).AND.OMEGN.GT.10.) GO TO 37
      IF (B(KVH)) GO TO 33
      GO TO 38
33    IF (KVH.EQ.JH) GO TO 34
      IF (KVH.EQ.JV) GO TO 35
      IF (KVH.EQ.JK) GO TO 36
34    IF (NI.EQ.7) G(NJ)=GLCSXP(OMEGA,A(JK),A(JV),IOLD,IOLD)
      IF (NI.EQ.8) G(NJ)=GLCP3P(A(JH),OMEGA,A(JV),A(JK),VOLD,VOLD)
      GO TO 37
35    IF (NI.EQ.7) G(NJ)=GLCSQ(A(JH),A(JK),A(JV),IOLD,IOLD)
      IF (NI.EQ.8) G(NJ)=GLCPQ(A(JH),A(JV),A(JK),VOLD,VOLD)
      GO TO 37
36    IF (NI.EQ.7) G(NJ)=GLCSOR(OMEGA,A(JH),A(JK),A(JV),IOLD,IOLD)
      IF (NI.EQ.8) G(NJ)=GLCPOR(A(JH),OMEGA,A(JK),A(JV),VOLD,VOLD)
37    IF (NI.EQ.7) AD(NJ)=IOLD
      IF (NI.EQ.8) AD(NJ)=VOLD
      JOLD=JOLD+1
38    CONTINUE
      KN=K+3
      GO TO 72
39    KK=K
      JOLD=J
      DO 42 II=1,2
      KKK=NE-KK
      IF (B(KKK)) GO TO 40
      GO TO 41
40    JJ=N-J
      A(KKK)=X(JJ)
      J=J+1
41    KK=KK+1
42    CONTINUE
      JV=NE-K-1
      JH=NE-K
      NI=NN-10
      GO TO (43,44,45,46,54,59,60,61,62), NI
43    CALL CODE11 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
      GO TO 48
44    CALL CODE12 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
      GO TO 47
45    CALL CODE13 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
      GO TO 48
46    CALL CODE14 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
      GO TO 47
47    IOLD=INEW-IOLD
48    NJ=N-JOLD
      IF (NN.EQ.11.OR.NN.EQ.14) NI=11
      IF (NN.EQ.12.OR.NN.EQ.13) NI=12
      IF (B(JH)) GO TO 49
      GO TO 51
49    IF (OMEGN.GT.10.) GO TO 50
      IF (NI.EQ.11) G(NJ)=GSCTZO(A(JV),IOLD,IOLD)
      IF (NI.EQ.12) G(NJ)=GOCTZO(OMEGA,A(JV),IOLD,IOLD)
50    CONTINUE
      JOLD=JOLD+1
      AD(NJ)=IOLD
51    NJ=N-JOLD
      IF (B(JV)) GO TO 52
      GO TO 58
52    IF (OMEGN.GT.10.) GO TO 53
      IF (NI.EQ.11) G(NJ)=GSCTEL(OMEGA,A(JV),A(JH),IOLD,IOLD)
      IF (NI.EQ.12) G(NJ)=GOCTEL(A(JH),A(JV),OMEGA,IOLD,IOLD)
53    CONTINUE
      AD(NJ)=IOLD
      GO TO 58
54    CALL CODE15 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
      NJ=N-JOLD
      IF (B(JH)) GO TO 55
      GO TO 56
55    AD(NJ)=VOLD
      ADJJ(NJ)=VNEW
      G(NJ)=GLLTZO(VNEW,VOLD,INEW,IOLD,A(JH))
      JOLD=JOLD+1
56    NJ=N-JOLD
      IF (B(JV)) GO TO 57
      GO TO 58
57    AD(NJ)=VOLD
      G(NJ)=GLLTEL(OMEGA,A(JV),VNEW,VOLD,INEW,IOLD)
      ADJJ(NJ)=VNEW
58    KN=K+2
      GO TO 72
59    CALL CODE16 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW)
      GO TO 63
60    CALL CODE17 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW)
      GO TO 63
61    CALL CODE18 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW)
      VOLD=VNEW-VOLD
      GO TO 63
62    CALL CODE19 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW)
      IOLD=INEW-IOLD
63    CONTINUE
      IF (NN.EQ.16.OR.NN.EQ.19) NI=16
      IF (NN.EQ.17.OR.NN.EQ.18) NI=17
      DO 71 II=1,2
      GO TO (64,65), II
64    KVH=JH
      GO TO 66
```

```
65     KVH=JV
66     CONTINUE
       NJ=N-JOLD
       IF (B(KVH).AND.OMEGN.GT.10.) GO TO 70
       IF (B(KVH)) GO TO 67
       GO TO 71
67     IF (KVH.EQ.JH) GO TO 68
       IF (KVH.EQ.JV) GO TO 69
68     IF (NI.EQ.16) G(NJ)=GESXP(OMEGA,A(JV),IOLD,IOLD)
       IF (NI.EQ.17) G(NJ)=GEPBP(OMEGA,A(JV),VOLD,VOLD)
       GO TO 70
69     IF (NI.EQ.16) G(NJ)=GESOR(OMEGA,A(JH),A(JV),IOLD,IOLD)
       IF (NI.EQ.17) G(NJ)=GEPOR(A(JH),OMEGA,A(JV),VOLD,VOLD)
70     IF (NI.EQ.16) AD(NJ)=IOLD
       IF (NI.EQ.17) AD(NJ)=VOLD
       JOLD=JOLD+1
71     CONTINUE
       KN=K+2
72     VOLD=VNEW
       IOLD=INEW
       K=KN
73     CONTINUE
       IF (OMEGN.GT.[illegible]) GO TO 79
       IF (OMEGN.GT.10.) GO TO 87
       RHO=1.-2.*INEW/(VNEW+INEW)
       CONRHO=CONJG(RHO)
       APP=CABS(RHO)
       IF (MET) 74,76,74
74     DO 75 L=1,N
       GRAD(L)=REAL((CONRHO/APP)*2.*G(L)/((VNEW+INEW)**2))
75     CONTINUE
76     IF (NC.GT.0.OR.ND.GT.0) GO TO 77
       RETURN
77     IF (KVR.EQ.0) RETURN
       NNN=N+1
       DO 78 L=NNN,N1
       GRAD(L)=0.
78     CONTINUE
       RETURN
79     CONTINUE
       PHASE=(1./(VNEW+INEW))
       IF (KGD.EQ.0) GO TO 80
       GO TO 81
80     OMEGA=OMEGA+2.*DELO
       POLD=PHASE
81     IF (KGD.GT.0) DELAY=-AIMAG((2.0/(POLD+PHASE))*((PHASE-POLD)/(2*DEL
      1O)))*(1000./(2.*PIE*FM))
       KGD=KGD+1
82     CONTINUE
       OMEGA=OMEGA-DELO
       IF (KKGD.EQ.0) OLDEL=DELAY
       IF (KKGD.EQ.0) GO TO 2
       GRAD(KKV)=(DELAY-OLDEL)/DELX
       X(KKV)=X(KKV)-DELX
83     CONTINUE
       APP=OLDEL
       IF (NC.GT.0.OR.ND.GT.0) GO TO 84
       RETURN
84     IF (KVR.EQ.0) GO TO 86
       NNN=N1-N
       DO 85 L=1,NNN
       LL=N+L
       AB(L)=X(LL)
85     CONTINUE
86     WCC=WC/FM
       IF (NC.GT.0) CALL CODEC (NC,N,AB,WCC,OMEGA,GD1,GRAD,FM)
       IF (ND.GT.0) CALL CODED (ND,N,NC,AB,WCC,OMEGA,GD2,GRAD,FM)
       APP=OLDEL+GD1+GD2-AB(NNN)
       IF (KVR.EQ.1) GRAD(N1)=-1.0
       RETURN
87     APP=-20.*ALOG10(CABS((1.+RL)/(VNEW+INEW)))
       K=1
       J=1
       VOLD=1.0
       IOLD=1.0
       DO 141 L=1,M
       NN=IC(L)
       GO TO (88,89,90,91,92,93,99,99,99,99,117,117,117,117,117,117,117,1
      117,117), NN
88     CALL CODE1 (IOLD,VOLD,A(K),OMEGA,INEW,VNEW)
       GO TO 94
89     CALL CODE2 (IOLD,VOLD,A(K),OMEGA,INEW,VNEW)
       GO TO 94
90     CALL CODE3 (IOLD,VOLD,A(K),OMEGA,INEW,VNEW)
       VOLD=VNEW-VOLD
       GO TO 94
91     CALL CODE4 (IOLD,VOLD,A(K),OMEGA,INEW,VNEW)
       IOLD=INEW-IOLD
       GO TO 94
92     CALL CODE5 (IOLD,VOLD,A(K),INEW,VNEW)
       GO TO 94
93     CALL CODE6 (IOLD,VOLD,A(K),INEW,VNEW)
       IOLD=INEW-IOLD
94     IF (B(K)) GO TO 95
       GO TO 98
95     IF (NN.EQ.1.OR.NN.EQ.4) G(J)=GL(OMEGA,AD(J),IOLD)
       IF (NN.EQ.2.OR.NN.EQ.3) G(J)=GC(OMEGA,AD(J),VOLD)
       IF (NN.EQ.5.OR.NN.EQ.6) G(J)=IOLD*AB(J)
       GO TO (96,97,96,97,96,97), NN
96     G(J)=-G(J)
```

```
97      J=J+1
98      K=K+1
        GO TO 140
99      JK=K
        JV=K+1
        JH=K+2
        NII=NN-6
        GO TO (100,101,102,103), NII
100     CALL CODE7 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
        GO TO 104
101     CALL CODE8 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
        GO TO 104
102     CALL CODE9 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
        VOLD=VNEW-VOLD
        GO TO 104
103     CALL CODE10 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
        IOLD=INEW-IOLD
104     IF (NN.EQ.7.OR.NN.EQ.10) NI=7
        IF (NN.EQ.8.OR.NN.EQ.9) NI=8
        DO 116 II=1,3
        GO TO (105,106,107), II
105     KVH=JK
        GO TO 108
106     KVH=JV
        GO TO 108
107     KVH=JH
108     IF (B(KVH)) GO TO 109
        GO TO 116
109     IF (KVH.EQ.JK) GO TO 110
        IF (KVH.EQ.JV) GO TO 111
        IF (KVH.EQ.JH) GO TO 112
110     IF (NI.EQ.7) G(J)=GLCSOR(OMEGA,A(JH),A(JK),A(JV),AD(J),IOLD)
        IF (NI.EQ.8) G(J)=GLCPOR(A(JH),OMEGA,A(JK),A(JV),AD(J),VOLD)
        GO TO 113
111     IF (NI.EQ.7) G(J)=GLCSQ(A(JH),A(JK),A(JV),AD(J),IOLD)
        IF (NI.EQ.8) G(J)=GLCPQ(A(JH),A(JV),A(JK),AD(J),VOLD)
        GO TO 113
112     IF (NI.EQ.7) G(J)=GLCSXP(OMEGA,A(JK),A(JV),AD(J),IOLD)
        IF (NI.EQ.8) G(J)=GLCPBP(A(JH),OMEGA,A(JV),A(JK),AD(J),VOLD)
113     GO TO (114,115,114,115), NII
114     G(J)=-G(J)
115     J=J+1
116     CONTINUE
        K=K+3
        GO TO 140
117     JV=K
        JH=K+1
        NII=NN-10
        GO TO (118,119,120,121,123,124,125,126,127), NII
118     CALL CODE11 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
        GO TO 122
119     CALL CODE12 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
        IOLD=INEW-IOLD
        GO TO 122
120     CALL CODE13 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
        GO TO 122
121     CALL CODE14 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
        IOLD=INEW-IOLD
122     IF (NN.EQ.11.OR.NN.EQ.14) NI=11
        IF (NN.EQ.12.OR.NN.EQ.13) NI=12
        GO TO 129
123     CALL CODE15 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
        GO TO 129
124     CALL CODE16 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW)
        GO TO 128
125     CALL CODE17 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW)
        GO TO 128
126     CALL CODE18 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW)
        VOLD=VNEW-VOLD
        GO TO 128
127     CALL CODE19 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW)
        IOLD=INEW-IOLD
128     IF (NN.EQ.16.OR.NN.EQ.19) NI=16
        IF (NN.EQ.17.OR.NN.EQ.18) NI=17
129     DO 139 II=1,2
        GO TO (130,131), II
130     KVH=JV
        GO TO 132
131     KVH=JH
132     IF (B(KVH)) GO TO 133
        GO TO 139
133     IF (KVH.EQ.JV) GO TO 134
        IF (KVH.EQ.JH) GO TO 135
134     IF (NI.EQ.11) G(J)=GSCTEL(OMEGA,A(JV),A(JH),AD(J),IOLD)
        IF (NI.EQ.12) G(J)=GOCTEL(A(JH),A(JV),OMEGA,AD(J),IOLD)
        IF (NN.EQ.15) G(J)=-GLLTEL(OMEGA,A(JV),ADJJ(J),AD(J),IOLD,INEW)
        IF (NI.EQ.16) G(J)=GESOR(OMEGA,A(JH),A(JV),AD(J),IOLD)
        IF (NI.EQ.17) G(J)=GEPOR(A(JH),OMEGA,A(JV),AD(J),VOLD)
        GO TO 136
135     IF (NI.EQ.11) G(J)=GSCTZO(A(JV),AD(J),IOLD)
        IF (NI.EQ.12) G(J)=GOCTZO(OMEGA,A(JV),AD(J),IOLD)
        IF (NN.EQ.15) G(J)=-GLLTZO(ADJJ(J),AD(J),IOLD,INEW,A(JH))
        IF (NI.EQ.16) G(J)=GESXP(OMEGA,A(JV),AD(J),IOLD)
        IF (NI.EQ.17) G(J)=GEPBP(OMEGA,A(JV),AD(J),VOLD)
136     GO TO (137,138,137,138,138,137,138,137,138), NII
137     G(J)=-G(J)
138     J=J+1
139     CONTINUE
        K=K+2
140     IOLD=INEW
```

```
      VOLD=VNEW
141   CONTINUE
      PSIL=VNEW+INEW*RL
      CONSTN=20./ALOG(10.)
      IF (MET) 142,144,142
142   DO 143 L=1,N
      GRAD(L)=-REAL(G(L)/PSIL)*CONSTN
143   CONTINUE
144   IF (NC.GT.0.OR.ND.GT.0) GO TO 145
      RETURN
145   IF (KVR.EQ.0) RETURN
      NNN=N+1
      DO 146 L=NNN,N1
      GRAD(L)=0.
146   CONTINUE
      RETURN
147   INT=IFIX(OMEGN-30.)
      APP=X(INT)
      DO 148 L=1,N1
      GRAD(L)=0.
148   CONTINUE
      GRAD(INT)=1.0
      RETURN
149   WRITE (6,150)
      CALL EXIT
C
C
150   FORMAT (/////5X,69HNUMBER OF VARIABLE PARAMETERS N EXCEEDS 50. PLE
     1ASE CHANGE THE LENGTHS,/5X,55HOF ARRAYS AD(N), ADJJ(N) AND G(N) IN
     2 SUBROUTINE APPROX.)
      END


      SUBROUTINE CODE1 (IOLD,VOLD,AL,OMEGA,INEW,VNEW)
C
C          SERIES INDUCTOR
C
      COMPLEX IOLD,VOLD,INEW,VNEW,Z
      Z=CMPLX(0.,OMEGA*AL)
      VNEW=VOLD+Z*IOLD
      INEW=IOLD
      RETURN
      END


      SUBROUTINE CODE2 (IOLD,VOLD,C,OMEGA,INEW,VNEW)
C
C          SHUNT CAPACITOR
C
      COMPLEX IOLD,VOLD,INEW,VNEW,Y
      Y=CMPLX(0.,OMEGA*C)
      VNEW=VOLD
      INEW=IOLD+VOLD*Y
      RETURN
      END


      SUBROUTINE CODE3 (IOLD,VOLD,C,OMEGA,INEW,VNEW)
C
C          SERIES CAPACITOR
C
      COMPLEX IOLD,VOLD,INEW,VNEW,Z
      Z=CMPLX(0.,-(1./(OMEGA*C)))
      INEW=IOLD
      VNEW=VOLD+Z*IOLD
      RETURN
      END


      SUBROUTINE CODE4 (IOLD,VOLD,AL,OMEGA,INEW,VNEW)
C
C          SHUNT INDUCTOR
C
      COMPLEX IOLD,VOLD,INEW,VNEW,Y
      Y=CMPLX(0.,(-1./(OMEGA*AL)))
      VNEW=VOLD
      INEW=IOLD+VOLD*Y
      RETURN
      END


      SUBROUTINE CODE5 (IOLD,VOLD,R,INEW,VNEW)
C
C          SERIES RESISTOR
C
      COMPLEX IOLD,VOLD,INEW,VNEW
      VNEW=VOLD+R*IOLD
      INEW=IOLD
      RETURN
      END
```

```
      SUBROUTINE CODE6 (IOLD,VOLD,R,INEW,VNEW)
C
C         SHUNT RESISTOR
C
      COMPLEX IOLD,VOLD,INEW,VNEW
      VNEW=VOLD
      INEW=IOLD+1./R*VOLD
      RETURN
      END
```

```
      SUBROUTINE CODE7 (IOLD,VOLD,Q,OMEGAR,OMEGA,XP,INEW,VNEW)
C
C         RLC SERIES RESONANT CIRCUIT
C
      COMPLEX IOLD,VOLD,INEW,VNEW,Z
      Z=(XP/2.)*CMPLX(OMEGAR/Q,((OMEGA*OMEGA-OMEGAR*OMEGAR)/OMEGA))
      INEW=IOLD
      VNEW=VOLD+Z*IOLD
      RETURN
      END
```

```
      SUBROUTINE CODE8 (IOLD,VOLD,Q,OMEGAR,OMEGA,BP,INEW,VNEW)
C
C         RLC SHUNT ANTIRESONANT CIRCUIT
C
      COMPLEX IOLD,VOLD,INEW,VNEW,Y
      VNEW=VOLD
      Y=(BP/2.)*CMPLX((OMEGAR/Q),((OMEGA*OMEGA-OMEGAR*OMEGAR)/OMEGA))
      INEW=IOLD+VOLD*Y
      RETURN
      END
```

```
      SUBROUTINE CODE9 (IOLD,VOLD,Q,OMEGAR,OMEGA,BP,INEW,VNEW)
C
C         RLC SERIES ANTIRESONANT CIRCUIT
C
      COMPLEX IOLD,VOLD,INEW,VNEW,Y
      INEW=IOLD
      Y=(BP/2.)*CMPLX((OMEGAR/Q),((OMEGA*OMEGA-OMEGAR*OMEGAR)/OMEGA))
      VNEW=VOLD+IOLD/Y
      RETURN
      END
```

```
      SUBROUTINE CODE10 (IOLD,VOLD,Q,OMEGAR,OMEGA,XP,INEW,VNEW)
C
C         RLC SHUNT RESONANT CIRCUIT
C
      COMPLEX IOLD,VOLD,INEW,VNEW,Z
      Z=(XP/2.)*CMPLX(OMEGAR/Q,(OMEGA*OMEGA-OMEGAR*OMEGAR)/OMEGA)
      VNEW=VOLD
      INEW=IOLD+VOLD/Z
      RETURN
      END
```

```
      SUBROUTINE CODE11 (IOLD,VOLD,EL,OMEGA,ZO,INEW,VNEW,PIE)
C
C         SERIES SHORTED LOSSLESS TRANSMISSION LINE
C
      COMPLEX IOLD,VOLD,INEW,VNEW
      THETA=PIE/2.*OMEGA*EL
      INEW=IOLD
      VNEW=VOLD+CMPLX(0.,ZO*SIN(THETA)/COS(THETA))*IOLD
      RETURN
      END
```

```
      SUBROUTINE CODE12 (IOLD,VOLD,EL,OMEGA,ZO,INEW,VNEW,PIE)
C
C         SHUNT OPENED LOSSLESS TRANSMISSION LINE
C
      COMPLEX IOLD,VOLD,INEW,VNEW
      THETA=PIE/2.*OMEGA*EL
      VNEW=VOLD
      INEW=IOLD+CMPLX(0.,SIN(THETA)/(ZO*COS(THETA)))*VOLD
      RETURN
      END
```

```
      SUBROUTINE CODE13 (IOLD,VOLD,EL,OMEGA,ZO,INEW,VNEW,PIE)
C
C         SERIES OPENED LOSSLESS TRANSMISSION LINE
C
      COMPLEX IOLD,VOLD,INEW,VNEW
      THETA=PIE/2.*OMEGA*EL
      INEW=IOLD
      VNEW=VOLD-CMPLX(0.,ZO*COS(THETA)/SIN(THETA))*INEW
      RETURN
      END
```

```
      SUBROUTINE CODE14 (IOLD,VOLD,EL,OMEGA,ZO,INEW,VNEW,PIE)
C
C         SHUNT SHORTED LOSSLESS TRANSMISSION LINE
C
      COMPLEX IOLD,VOLD,INEW,VNEW
      THETA=PIE/2.*OMEGA*EL
      VNEW=VOLD
      INEW=IOLD-CMPLX(0.,COS(THETA)/(ZO*SIN(THETA)))*VOLD
      RETURN
      END
```

```
      SUBROUTINE CODE15 (IOLD,VOLD,EL,OMEGA,ZO,INEW,VNEW,PIE)
C
C         CASCADED LOSSLESS TRANSMISSION LINE
C
      COMPLEX VOLD,IOLD,INEW,VNEW,JSINE
      THETA=PIE/2.*OMEGA*EL
      CT=COS(THETA)
      JSINE=CMPLX(0.,SIN(THETA))
      VNEW=CT*VOLD+ZO*JSINE*IOLD
      INEW=JSINE*VOLD/ZO+CT*IOLD
      RETURN
      END
```

```
      SUBROUTINE CODEC (K,N,A,WC,OMEGA,GD,GRAD,FM)
C
C         ALLPASS C-SECTION
C
      DIMENSION A(1), GRAD(1)
      COMMON /S16/ PIE,WCC,WCSQ,WCS1,FMC
      OMSQ=OMEGA*OMEGA
      OMSQ=OMSQ-WCSQ
      FC=PIE/2.*SQRT(OMSQ/WCS1)
      TGFC=SIN(FC)/COS(FC)
      TGFC=TGFC*TGFC
      S=OMEGA/SQRT(OMSQ*WCS1)
      S=S/FMC
      QUA2=(1.+TGFC)/2.
      GD=0.
      DO 1 J=1,K
      QUA1=(A(J)*A(J)+TGFC)
      GD=GD+A(J)/QUA1
      GRAD(J)=-S*(A(J)*A(J)-TGFC)/(QUA1*QUA1)*QUA2
    1 CONTINUE
      GD=S*QUA2*GD
      RETURN
      END
```

```
      SUBROUTINE CODED (M,N,K,A,WC,OMEGA,GD,GRAD,FM)
C
C         ALLPASS D-SECTION
C
      DIMENSION A(1), GRAD(1)
      COMMON /S16/ PIE,WCC,WCSQ,WCS1,FMC
      OMSQ=OMEGA*OMEGA
      OMSQ=OMSQ-WCSQ
      FC=PIE/2.*SQRT(OMSQ/WCS1)
      TGFC=SIN(FC)/COS(FC)
      TGFC=TGFC*TGFC
      TGSQ=TGFC*TGFC
      S=OMEGA/SQRT(OMSQ*WCS1)
      S=S/FMC
      S=(1.+TGFC)*S
      GD=0.
      DO 1 I=1,M
      J=I+K
      JJ=J+M
      OMSQ=A(J)*A(J)
      AJSQ=A(JJ)*A(JJ)
      ASUM=OMSQ+AJSQ
      ASUB=OMSQ-AJSQ
      DEN=TGSQ+2.*TGFC*ASUB+ASUM*ASUM
      GD=GD+A(J)*(TGFC+ASUM)/DEN
      DEN1=TGFC+ASUM+2.*OMSQ
      DEN2=A(J)*(TGFC+ASUM)
      DEN3=4.*DEN2
      DEN4=4.*A(JJ)*(ASUM-TGFC)
      GRAD(J)=DEN*DEN1-DEN2*DEN3
      GRAD(JJ)=2.*DEN*A(J)*A(JJ)-DEN2*DEN4
      DEN=DEN*DEN
      GRAD(J)=S*GRAD(J)/DEN
      GRAD(JJ)=S*GRAD(JJ)/DEN
    1 CONTINUE
      GD=S*GD
      RETURN
      END
```

```
      SUBROUTINE CODE16 (IOLD,VOLD,OMEGAR,OMEGA,XP,INEW,VNEW)
C
C     LC SERIES RESONANT CIRCUIT
C
      COMPLEX IOLD,VOLD,INEW,VNEW,Z
      Z=XP/2.*CMPLX(0.,(OMEGA*OMEGA-OMEGAR*OMEGAR)/OMEGA)
      INEW=IOLD
      VNEW=VOLD+Z*IOLD
      RETURN
      END

      SUBROUTINE CODE17 (IOLD,VOLD,OMEGAR,OMEGA,BP,INEW,VNEW)
C
C     LC SHUNT ANTIRESONANT CIRCUIT
C
      COMPLEX IOLD,VOLD,INEW,VNEW,Y
      Y=BP/2.*CMPLX(0.,(OMEGA*OMEGA-OMEGAR*OMEGAR)/OMEGA)
      VNEW=VOLD
      INEW=IOLD+VOLD*Y
      RETURN
      END

      SUBROUTINE CODE18 (IOLD,VOLD,OMEGAR,OMEGA,BP,INEW,VNEW)
C
C     LC SERIES ANTIRESONANT CIRCUIT
C
      COMPLEX IOLD,VOLD,INEW,VNEW,Y
      Y=BP/2.*CMPLX(0.,(OMEGA*OMEGA-OMEGAR*OMEGAR)/OMEGA)
      INEW=IOLD
      VNEW=VOLD+IOLD/Y
      RETURN
      END

      SUBROUTINE CODE19 (IOLD,VOLD,OMEGAR,OMEGA,XP,INEW,VNEW)
C
C     LC SHUNT RESONANT CIRCUIT
C
      COMPLEX IOLD,VOLD,INEW,VNEW,Z
      Z=XP/2.*CMPLX(0.,(OMEGA*OMEGA-OMEGAR*OMEGAR)/OMEGA)
      VNEW=VOLD
      INEW=IOLD+VOLD/Z
      RETURN
      END

      SUBROUTINE PLOT (INUM,IAUTO,PMIN,PMAX)
      COMMON /TEST/ FREQ(150),ROUT(150,1),ICALC,XLOG10,IDENT
      DIMENSION COOR(5), IPOINT(13)
      DATA IPOINT/5755555555B,2*5555555555B,5557555555B,2*5555555555B,55
     1555755555B,2*5555555555B,5555555755B,2*5555555555B,5555555557B/
      DATA ILET1,ILET2,ISHIFT/06B,10B,100B/
C
C
      IF (IAUTO.EQ.1) GO TO 2
      XMIN=PMIN
      DEL=PMAX-PMIN
      IF (DEL.GT.0.0) GO TO 1
      XMIN=PMAX
      DEL=-DEL
1     IF (DEL.LT.1.0E-20) DEL=1.0E-20
      DEL=DEL/4.0
      GO TO 8
C
C     DETERMINE MAX AND MIN VALUES
C
C
2     XMAX=ROUT(1,INUM)
      XMIN=XMAX
      DO 3 I=2,ICALC
      IF (ROUT(I,INUM).LT.XMIN) XMIN=ROUT(I,INUM)
      IF (ROUT(I,INUM).GT.XMAX) XMAX=ROUT(I,INUM)
3     CONTINUE
C
C     SCALING
C
      AVER=(XMAX+XMIN)/2.0
      DEL=XMAX-XMIN
      DELMIN=ABS(AVER)*1.0E-3
      IF (DELMIN.LT.1.0E-20) DELMIN=1.0E-20
      IF (DEL.LT.DELMIN) DEL=DELMIN
      IEXP=-IFIX(ALOG10(DEL))
      FACTOR=EXP(IEXP*XLOG10)
      DEL=DEL*FACTOR
      IF (DEL.LT.0.75) FACTOR=FACTOR*10.0
      IF (DEL.GT.8.0) FACTOR=FACTOR/10.0
      IAVER=IFIX(FACTOR*AVER+0.5)
      AVER=FLOAT(IAVER)/FACTOR
      DEL1=XMAX-AVER
      DEL=AVER-XMIN
      IF (DEL1.GT.DEL) DEL=DEL1
      DEL=DEL*FACTOR
      IF (DEL.GT.1.0) GO TO 4
      DEL=1.0
      GO TO 7
4     IF (DEL.GT.2.0) GO TO 5
```

```
      DEL=2.0
      GO TO 7
5     IF (DEL.GT.4.0) GO TO 6
      DEL=4.0
      GO TO 7
6     DEL=8.0
7     DEL=DEL/FACTOR
      XMIN=AVER-DEL
      DEL=DEL/2.0
C
C     DETERMINE COORDINATES
C
8     COOR(1)=XMIN
      SMALL=DEL*1.0E-4
      DO 9 I=1,4
      COOR(I+1)=COOR(I)+DEL
      IF (ABS(COOR(I+1)).LT.SMALL) COOR(I+1)=0.0
9     CONTINUE
      DEL=DEL/16.0
C
C     PRINT COORDINATES
C
      WRITE (6,13)
      IF (IDENT.GT.0) WRITE (6,14)
      WRITE (6,15) (COOR(I),I=1,5)
C
C     BEGIN PLOTTING
C
      DO 12 I=1,ICALC
      VOUT=ROUT(I,INUM)
      JPOINT=IFIX((VOUT-XMIN)/DEL+0.5)
      IF (JPOINT.LT.0) JPOINT=0
      IF (JPOINT.GT.103) JPOINT=103
      JADDR=JPOINT/5+1
      JSPOT=JPOINT-(JADDR-1)*5
      LNEW=ILET1
      IF ((JPOINT/16)*16.EQ.JPOINT) LNEW=ILET2
      ITEMP=IPOINT(JADDR)
      JTEMP=1
      JSTOP=4-JSPOT
      IF (JSTOP.LT.1) GO TO 11
      DO 10 J=1,JSTOP
      JTEMP=JTEMP*ISHIFT
10    CONTINUE
11    IPOINT(JADDR)=ITEMP-LNEW*JTEMP
      WRITE (6,16) FREQ(I),(IPOINT(J),J=1,13)
      IPOINT(JADDR)=ITEMP
12    CONTINUE
      WRITE (6,17) (COOR(I),I=1,5)
      WRITE (6,18)
      RETURN
C
13    FORMAT (1H0)
14    FORMAT (24X,60HS O M E   D E T A I L S   F R O M   T H E   A B O V E
     1G R A P H//)
15    FORMAT (/17X,5(6X,1PE10.3)//15X,9HFREQUENCY,2X,33(1X,1H-))
16    FORMAT (10X,E14.7,3X,13A5)
17    FORMAT (26X,33(1X,1H-)//,17X,5(6X,1PE10.3))
18    FORMAT (1HR)
      END
```

ACKNOWLEDGEMENT

The authors thank W.Y. Chu, who initiated some early interactive features.

REFERENCES

[1] J.W. Bandler, J.R. Popović and V.K. Jha, "Cascaded network optimization program", IEEE Trans. Microwave Theory Tech., vol. MTT-22, Mar. 1974, 300-308.

[2] J.W. Bandler and J.R. Popović, "CANOPT-cascaded network optimization package", Faculty of Engineering, McMaster University, Hamilton, Canada, Report CRL-12, May 1974.

[3] J.W. Bandler and C. Charalambous, "Practical least pth optimization of networks", IEEE Trans. Microwave Theory Tech., vol. MTT-20, Dec. 1972, pp. 834-840.

[4] R Fletcher, "A new approach to variable metric algorithms", Computer J., vol. 13, Aug. 1970, pp. 317-322.

[5] R. Fletcher and M.J.D. Powell, "A rapidly convergent descent method for minimization", Computer J., vol. 6, June 1963, pp. 163-168.

[6] J.W. Bandler and R.E. Seviora, "Current trends in network optimization", IEEE Trans. Microwave Theory Tech., vol. MTT-18, Dec. 1970, pp. 1159-1170.

[7] C.M. Kudsia, "Synthesis of optimum reflection type microwave equalizers RCA Review, vol. 31, Sept. 1970, pp. 571-595.

[8] M.C. Horton and R.J. Wenzel "General theory and design of optimum quarter-wave TEM filters", IEEE Trans. Microwave Theory Tech., vol. MTT-13, May 1965, pp. 316-327.

[9] J.W. Bandler, W.Y. Chu and J.R. Popović, "Efficient, interactive semi-automated optimization of models and designs", Computers, Electronics and Control Symp. (Calgary, Alberta, May 1974), paper VI-4.

[10] J.W. Bandler, "Computer-aided circuit optimization", in Modern Filter Theory and Design, G.C. Temes and S.K. Mitra, Eds. New York: Wiley-Interscience, 1973.

P.PROGA_L.7

program P.mmm
line L.number

XDBG.

SET Breakpoint SB <location>, <first>, <limit>, set.

LIST

CLEAR

EXEC, GO

PRINT