CRL INTERNAL REPORT SERIES

No. CRL-6

Network Optimization Program

J.W. Bandler and J.R. Popović

COMMUNICATIONS RESEARCH LABORATORY

FACULTY OF ENGINEERING

McMASTER UNIVERSITY

HAMILTON, ONTARIO, CANADA

CRL INTERNAL REPORT SERIES

No. CRL-6

Network Optimization Program

J.W. Bandler and J.R. Popović

CRL NETWORK OPTIMIZATION PROGRAM

VERSION CRL1

by

JOHN W. BANDLER and J.R. POPOVIĆ
Communications Research Laboratory
McMaster University
Hamilton, Ontario, Canada

May 1973

CRL NETWORK OPTIMIZATION PROGRAM: VERSION CRL1

J.W. BANDLER and J.R. POPOVIĆ

Abstract The program described analyzes and optimizes certain electrical networks arranged as a cascade of two-ports such as microwave filters and allpass networks.

INTRODUCTION

The package features some of the latest and most efficient methods of computer-aided design currently available. At the user's command, either the well-known and highly respected Fletcher-Powell [1] method of minimizing unconstrained functions of many variables may be used, or the more recent, and generally more efficient, method by Fletcher [2].

State-of-the-art techniques in least pth approximation generalized for such tasks as filter design as proposed by Bandler and Charalambous [3] and programmed by Popović [4] are incorporated. Thus, a variety of upper and lower response specifications as well as simple upper and lower desired bounds for variable parameters are catered for. Low values of p, e.g., 2, intermediately large values of p, e.g., 10 to 1,000, as well as extremely large values of p, e.g., 1,000,000 are optional to the user depending on how close to a minimax (Chebyshev, equal-ripple) solution he wants to come.

The package has been designed to incorporate the adjoint network method of sensitivity evaluation to produce accurate first derivatives needed by these efficient gradient minimization methods. Many formulas published by Bandler and Seviora [5] have been built into the package.

As it stands at present, the package is capable of analyzing and optimizing certain linear, time-invariant, lumped and distributed networks in the frequency domain subject to the following specifications.

The network 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 (taken as unity when allpass networks are present).

Resistors, inductors, capacitors, lossless short-circuited and open-circuited transmission-line stubs, and series and parallel RLC resonant circuits can be called upon by the user and connected as series or shunt elements, in any order. Lossless transmission-lines as well as microwave allpass C- and D-sections can also be added.

Any circuit parameter may be fixed or varied as specified by the user. If variable parameters are to be constrained, then each may have an associated lower and upper desired bound supplied by the user. For upper and lower parameter constraints, fictitious frequency points of value 1, 2, 3, ... etc. are associated with each variable parameter in correct sequence. The constraints are treated exactly like single point specifications. (For a single point specification, let the number of subintervals be zero and set the upper bound of the interval equal to the lower bound.)

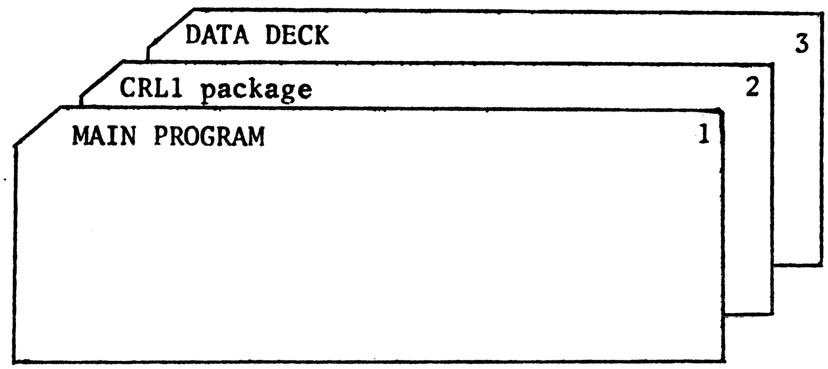
Continuous specifications require program modification.

Gradients are automatically checked before optimization. Responses before and after optimization are printed out. Much other useful information which can be used to check on the progress of the optimization process and to diagnose errors is printed out at the user's discretion.

The user should also consult an earlier report on a similar package [6].

HOW TO USE THE PROGRAM

Set up the input deck as follows:



Control cards when permanent file is used:

1. job card, CM60000.
2. RUN(S)
3. ATTACH, lfn*, CRL1, ID=GSGBNDLR , PW=CAD.
4. LOAD, lfn*.
5. LGO.

   eor

---

*logical file name

1. Main program

Write the main program as indicated below:

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

Dimension the following arrays:

```
DIMENSION  D(K), G(K), Y(K), PY(K), ASTRT(K), DUM1(K),
           DUM2(K), EPS(K), H(M), GRAD(K), XX(3,NINT),
           NUMB(N), X(N), X1(N), ERROR(N), EHELP(N),
           AP(N), INUMB(N)
```

where

K is the number of variable parameters

M = K(K+7)/2

NINT the total number of intervals including necessary ones for parameter constraints

N is the total number of sample points from all intervals

Call the subroutine CRL1 as follows:

```
CALL CRL1 (K, D, ASTRT, G, Y, PY, DUM1, DUM2,
          EPS, H, GRAD, NUMB, XX, X, X1, ERROR,
          EHELP, AP, INUMB)
```

Add

```
CALL EXIT
END
```

2. The CRL1 package

A listing is contained in this report.

3. Data deck

Parameters to be supplied as data are defined below:

MM	The number of blocks in the circuit.
NE	The total number of parameters in all blocks.
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.
IC(I), I=1,MM	A sequence of code numbers of blocks which specify the order in which blocks are connected. (See pp. 12-28 for code numbers.)
AA(I), I=1,NE	Values of each parameter in the circuit.
AB(I), I=1,J where J=2*ND+NC+1	Values of the parameters of the C and D sections and the d level. (See Kudsia [7]).
INUMB(I), I=1,NE	Indicates whether the parameters in the circuit are fixed or variable. Set to 1 if variable and 0 if fixed.
NINT	The total number of intervals including necessary ones for parameter constraints.
XX1(I), I=1,NINT	The lower frequency bounds corresponding to each interval.

XX2(I), I=1,NINT	The upper frequency bounds corresponding to each interval. Lower bound equals upper bound for single point specifications.
NUMB(I), I=1,NINT	The number of subintervals (equals sample points minus one) corresponding to each interval. Set to 0 for single point specifications.
FUN(I), I=1,NINT	A sequence of numbers to be used as specifications in each interval including parameter constraints.
XX3(I), I=1,NINT	Indicates whether a specification in any given interval is an upper or lower specification. Set to 1. for upper and to -1. for lower specification.
WT(I), I=1,NINT	The weighting factors (positive) in each interval. Set to 1. if unsure.
IOBJ(I), I=1,NINT	The approximating function in each interval. 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.
R	The load resistance.
FM	The center frequency.
WC	The cut-off frequency for C- and D-sections.
K	The number of variable parameters.
ASTRT(I), I=1,K	The starting values for the variable parameters.

MET	Optimization method to be used. Set to 1 for Fletcher method and to 2 for Fletcher-Powell method.
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 convergence in the Fletcher method (e.g., 10^{-4}).
EPS1	A small quantity for convergence in the Fletcher-Powell method (e.g., 10^{-4}).
EST	A realistic under-estimate 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.
KSI	A small quantity by which specifications could be shifted artificially. Set to 0. if not sure.
ITER	The number of complete optimizations desired. Each optimization starts from the previous optimum obtained.
IPA(I), I=1,ITER	Vector containing the values of p (positive integer, greater than one) to be used successively for each complete optimization.

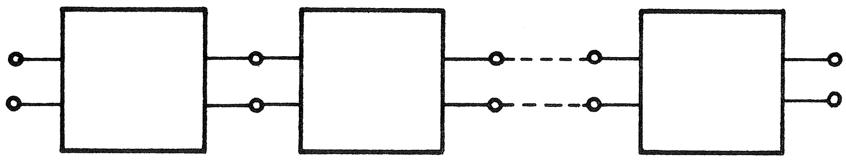
Table I shows the arrangement of the data deck.

TABLE I THE ARRANGEMENT OF THE DATA DECK

Condition	Number of cards	Parameters	Type	Format
-	1	MM,NE,NC,ND,KVR	INTEGER	5I10
-	1	IC(I), I=1,MM	INTEGER	8I10
-	As many as required by NE and J	(AA(I), I=1,NE), (AB(I), I=1,J(=2*ND+NC+1))	REAL	5E16.8
-	As many as required by NE	INUMB(I), I=1,NE	INTEGER	8I10
-	1	NINT	INTEGER	I10
-	As many as required by NINT	XX1(I), XX2(I), I=1,NINT	REAL	5E16.8
-	"	NUMB(I), I=1, NINT	INTEGER	8I10
-	"	FUN(I), I=1,NINT	REAL	5E16.8
-	"	XX3(I), I=1,NINT	REAL	5E16.8
-	"	WT(I), I=1,NINT	REAL	5E16.8
-	"	IOBJ(I), I=1,NINT	INTEGER	8I10
-	1	R, FM, WC	REAL	3E16.8
-	1	K	INTEGER	I10
-	As many as required by K	ASTRT(I), I=1,K	REAL	5E16.8
-	1	MET, MAX, IPRINT	INTEGER	3I10
IF MET=1	As many as required by K	EPS(I), I=1,K	REAL	5E16.8
IF MET=2	1	EPS1	REAL	E16.8
-	1	EST,DIF,KSI	REAL	3E16.8
-	1	ITER	INTEGER	I10
-	As many as required by ITER	IPA(I), I=1,ITER	INTEGER	8I10

CIRCUIT CONFIGURATION

AND BUILDING BLOCKS



CIRCUIT
CONFIGURATION

Possibilities

1. A cascade connection of two-port circuit blocks consisting of any of the elements depicted on the following pages in any order, and as many as required.
2. As many C- and D-sections as required.
3. Modification of program to accommodate new blocks is readily effected. See page 29.

Implementation*

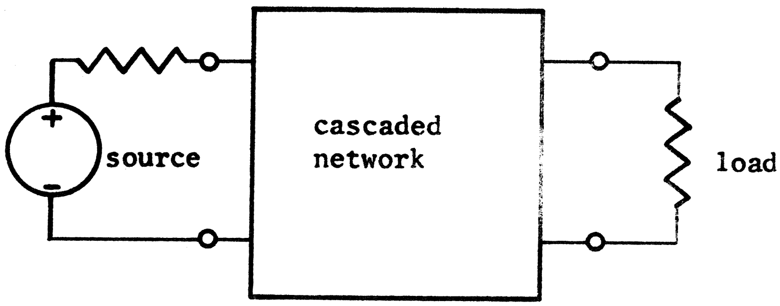
1. All blocks are numbered sequentially from left to right.
2. Each block has a code number associated with it defining the element it contains.

*Except allpass networks.

Parameters Required

Other than the parameters listed and defined together with the individual blocks, the following values must be supplied.

1. The total number of blocks (not including C- and D-sections).
2. The total number of parameters in these blocks.
3. The number of C-sections.
4. The number of D-sections.
5. The center frequency (e.g., in MHz, for normalization).
6. The cutoff frequency for C- and D-sections (e.g., in MHz).
7. The d-level for allpass networks (see Kudsia [7]). This parameter is treated like any other circuit parameter. It is the very last variable to be entered.



SOURCE
AND
LOAD
CONFIGURATION

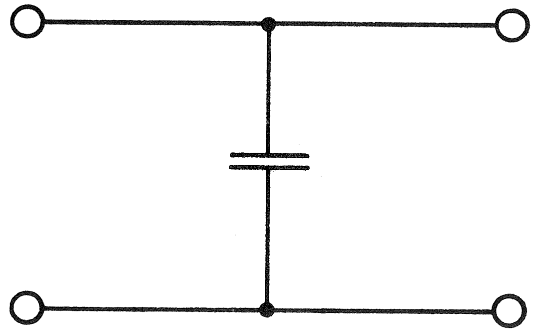
Possibilities

1. Complex (but constant) load impedance; will, therefore, usually be a resistance.

2. Modification of program needed to have frequency dependent source and load impedances (source is assumed to be unity).

Parameters Required

1. Load impedance.



SHUNT
CAPACITOR

Code

1

Parameters

1 2 3

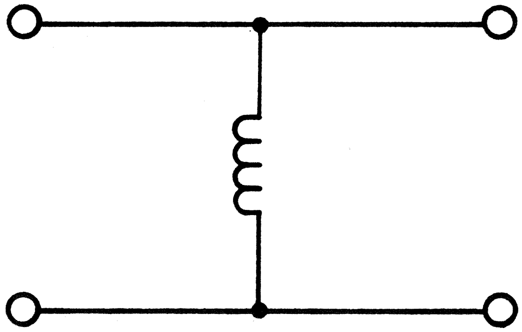
C

Parameter Definition

C = capacitance (normalized)

Comments

Upper and lower bounds or fixed values can be accommodated.



SHUNT
INDUCTOR

Code

2

Parameters

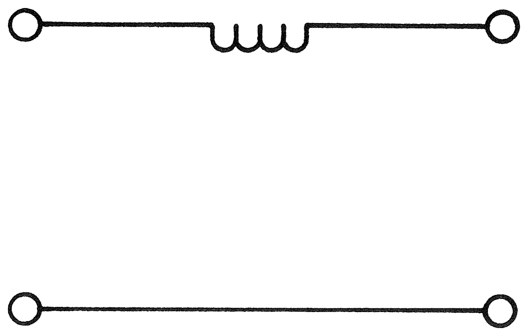
1	2	3
L		

Parameter Definition

L = inductance (normalized)

Comments

Upper and lower bounds or fixed values can be accommodated.



SERIES
INDUCTOR

Code

3

Parameters

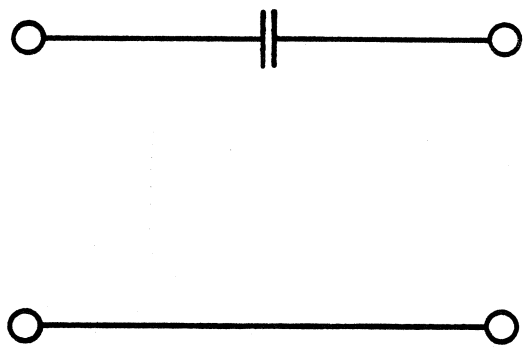
1	2	3
L		

Parameter Definition

L = inductance (normalized)

Comments

Upper and lower bounds or fixed values can be accommodated.



SERIES
CAPACITOR

Code

4

Parameters

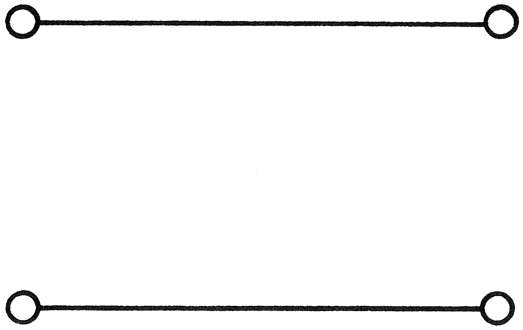
1	2	3
C		

Parameter Definition

C = capacitance (normalized)

Comments

Upper and lower bounds or fixed values can be accommodated.



LOSSLESS
TRANSMISSION
LINE

Code

5

Parameters

1	2	3
ℓ	Z_0	

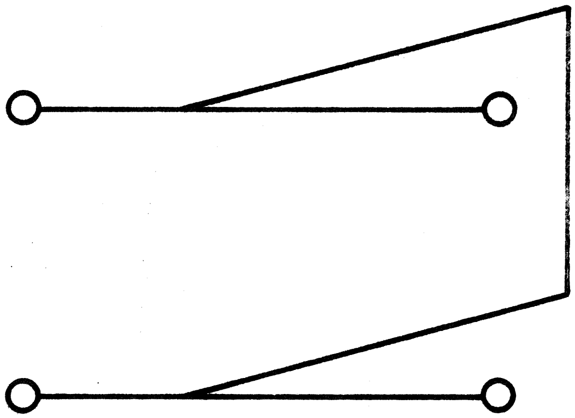
Parameter Definition

ℓ = length (normalized)

Z_0 = characteristic impedance (normalized)

Comments

Upper and lower bounds or fixed values can be accommodated.



SHUNT
SHORTED
LOSSLESS
TRANSMISSION
LINE

Code

6

Parameters

1	2	3
ℓ	Z_0	

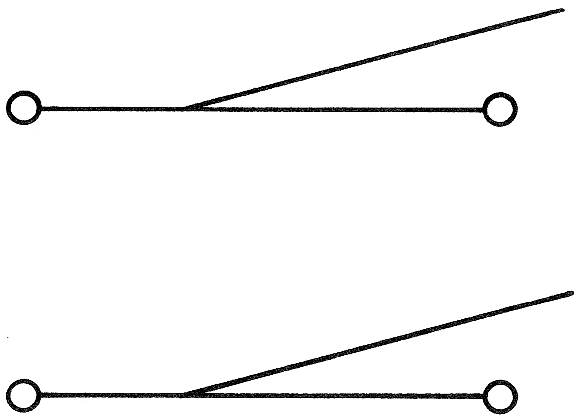
Parameter Definition

ℓ = length (normalized)

Z_0 = characteristic impedance (normalized)

Comments

Upper and lower bounds or fixed values can be accommodated.



SHUNT
OPEN
LOSSLESS
TRANSMISSION
LINE

Code

7

Parameters

1	2	3
ℓ	Z_0	

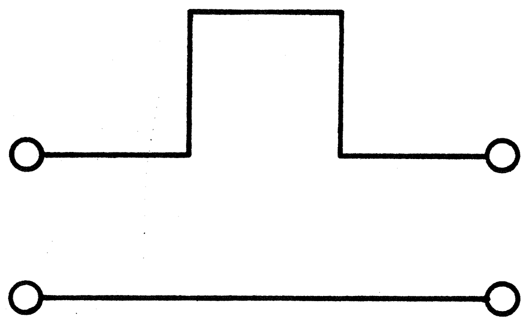
Parameter Definition

ℓ = length (normalized)

Z_0 = characteristic impedance (normalized)

Comments

Upper and lower bounds or fixed values can be accommodated.



SERIES
SHORTED
LOSSLESS
TRANSMISSION
LINE

Code

8

Parameters

1	2	3
ℓ	Z_0	

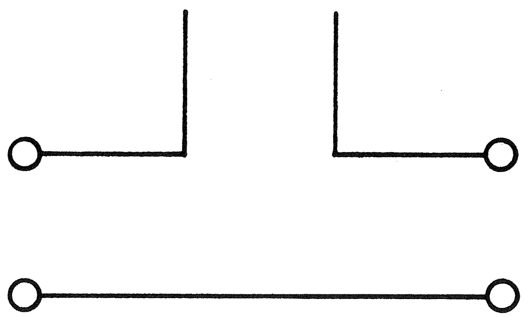
Parameter Definition

ℓ = length (normalized)

Z_0 = characteristic impedance (normalized)

Comments

Upper and lower bounds or fixed values can be accommodated.



SERIES
OPEN
LOSSLESS
TRANSMISSION
LINE

Code

9

Parameters

1	2	3
ℓ	Z_0	

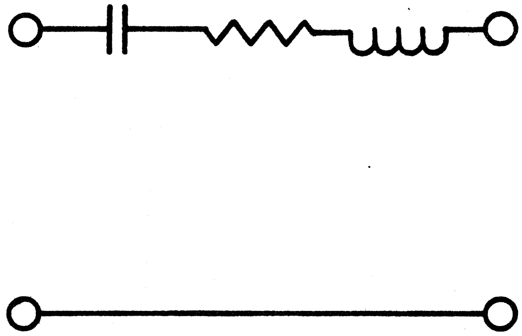
Parameter Definition

ℓ = length (normalized)

Z_0 = characteristic impedance (normalized)

Comments

Upper and lower bounds or fixed values can be accommodated.



SERIES
RESONANT
CIRCUIT

Code

10

Parameters

1	2	3
ω_R	Q	X'

Parameter Definition

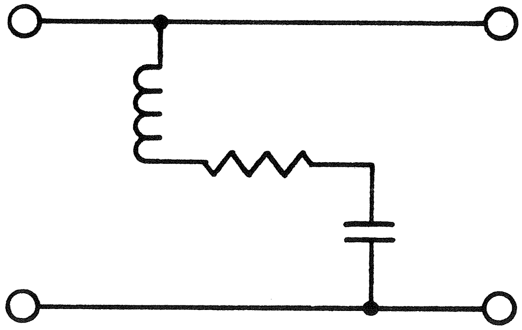
ω_R = resonant frequency (normalized)

Q = quality factor

X' = slope of reactance at resonance (normalized)

Comments

Upper and lower bounds or fixed values can be accommodated.



SHUNT
RESONANT
CIRCUIT

Code	Parameters		
	1	2	3
11	ω_R	Q	X'

Parameter Definition

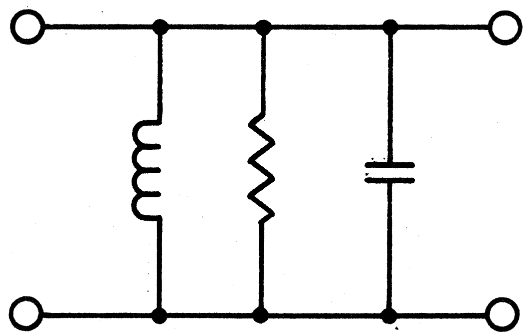
ω_R = resonant frequency (normalized)

Q = quality factor

X' = slope of reactance at resonance (normalized)

Comments

Upper and lower bounds or fixed values can be accommodated.



SHUNT
ANTIRESONANT
CIRCUIT

Code

12

Parameters

1	2	3
ω_R	Q	B'

Parameter Definition

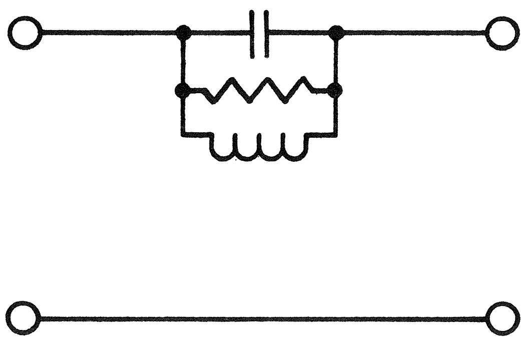
ω_R = antiresonant frequency (normalized)

Q = quality factor

B' = slope of susceptance at antiresonance (normalized)

Comments

Upper and lower bounds or fixed values can be accommodated.



SERIES
ANTIRESONANT
CIRCUIT

Code	Parameters		
	1	2	3
13	ω_R	Q	B'

Parameter Definition

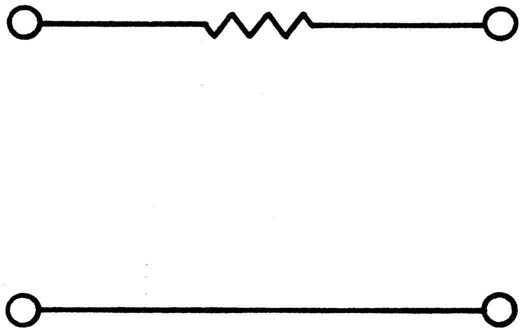
ω_R = antiresonant frequency (normalized)

Q = quality factor

B' = slope of susceptance at antiresonance (normalized)

Comments

Upper and lower bounds or fixed values can be accommodated.



SERIES
RESISTOR

Code

14

Parameters

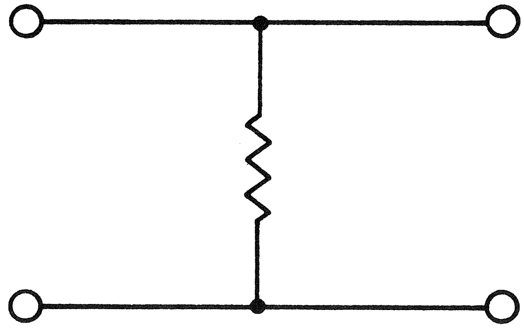
1	2	3
R		

Parameter Definition

R = resistance (normalized)

Comments

Upper and lower bounds or fixed values can be accommodated.



SHUNT
RESISTOR

Code

15

Parameters

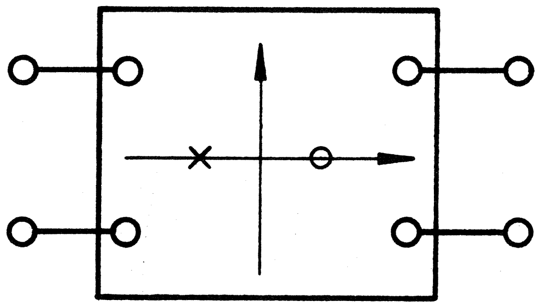
1	2	3
R		

Parameter Definition

R = resistance (normalized)

Comments

Upper and lower bounds or fixed values can be accommodated.



ALLPASS
C-SECTIONS
(Total number n_c)

Code

16 (not used)

Parameters

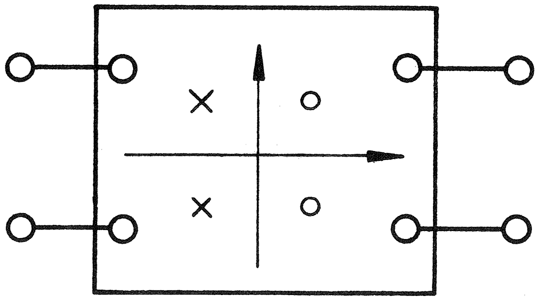
1	2	3 ...	n_c
σ_1	σ_2	σ_3...	σ_{n_c}

Parameter Definition

σ_i = location of ith real zero

Comments

1. The user specifies the number of C-sections required.
2. One cutoff frequency (fixed) and one d-level (variable) must be specified whenever any C- or D-section is used.
3. The user should consult theoretical concepts reviewed by Kudsia [7].
4. C- and D-section parameters are either all fixed or all variable.



ALLPASS
D-SECTIONS
(Total number n_d)

<u>Code</u>

17 (not used)

<u>Parameters</u>

$\underline{1}$	$\underline{2}$	$\underline{3}$ ...	$\underline{n_{d+1}}$	$\underline{n_{d+2}}$	$\underline{n_{d+3}}$	...
σ_1	σ_2	σ_3...	ω_1	ω_2	ω_3	...

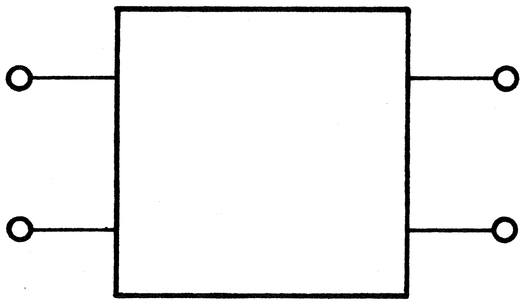
<u>Parameter Definition</u>

σ_i = location of real part of ith zero

ω_i = location of imaginary part of ith zero

<u>Comments</u>

1. The user specifies the number of D-sections required.
2. One cutoff frequency (fixed) and one d-level (variable) must be specified whenever any C- or D-section is used.
3. The user should consult theoretical concepts reviewed by Kudsia [7].
4. C- and D-section parameters are either all fixed or all variable.



TWO-PORT
SECTION

Possibilities

Addition of various new blocks is possible because of the modular approach which has been used in the development of the package. The following basic procedure has to be carried out.

Implementation

An analysis subroutine must be written to calculate input voltage and current given the output voltage and current (ABCD matrix analysis). The subroutine is called exactly as any other analysis subroutine in the package is called and sensitivity formulas obtained by the adjoint network method (see Bandler and Seviora [5]) if the parameters of the two-port are to be varied.

Comments

A wide variety of other two-ports can be added, e.g., distributed RC lines, transistor amplifier stages, operational amplifier stages, etc.

INPUT-OUTPUT EXAMPLE

It is desired to approximate a certain lowpass filter insertion loss specification using a ladder network consisting of lumped lossless inductors and capacitors. The first element is a shunt capacitor, followed by a series inductor and so on, with a total of three capacitors and three inductors. The source and load resistances are each taken as unity.

The input data and the results obtained by the Fletcher method are in Figs. 1 and 2, respectively.

```
         6         6         0         0         0
         1         3         1         3         1         3
      1.           1.           1.           1.           1.
    1.             1.
         1         1         1         1         1         1
         3
  1.E      -10      0.9         1.75         1.75          2.5
      2.5
        10         0         0
      0.           40.           60.
      1.            -1.          -1.
     5.              1.        1.
         2         2         2
     1.          1.          1.
         6
     1.            1.           1.           1.           1.
     1.
         1        100       20
  1.E      -4   1.E      -4   1.E      -4   1.E      -4   1.E      -4
  1.E      -4
    -10.          0.          0.
         1
        10
```

CD TOT 0023

Fig. 1. The data for the LC filter example.

RESPONSE AT THE STARTING POINT

FREQUENCY	INSERTION LOSS
9.998757377616E-11	0.
9.000000009001E-02	6.273254703084E-04
1.800000000800E-01	9.387443337173E-03
2.700000000700E-01	4.228551917297E-02
3.600000000600E-01	1.123638535239E-01
4.500000000500E-01	2.161346511038E-01
5.400000000400E-01	3.272044052784E-01
6.300000000300E-01	4.023235178967E-01
7.200000000200E-01	3.979612850876E-01
8.100000000100E-01	2.935268963817E-01
9.000000000000E-01	1.216171278175E-01

FREQUENCY	INSERTION LOSS
1.750000000000E+00	1.659962268589E-01

FREQUENCY	INSERTION LOSS
2.500000000000E+00	3.453938101504E+01

GRADIENTS CHECKING

GRADIENTS HAVE BEEN CHECKED AT THE FOLLOWING POINT

X(1)= 1.00000000E+00
X(2)= 1.00000000E+00
X(3)= 1.00000000E+00
X(4)= 1.00000000E+00
X(5)= 1.00000000E+00
X(6)= 1.00000000E+00

ANALYTICAL GRADIENTS	NUMERICAL GRADIENTS	PERCENTAGE ERROR
-4.75456254E-01	-4.75892991E-01	9.17721474E-02
-5.86685597E+00	-5.87271772E+00	9.98132450E-02
-1.08873021E+01	-1.09038185E+01	1.51473126E-01
-1.08873021E+01	-1.09038185E+01	1.51473126E-01
-5.86685597E+00	-5.87271772E+00	9.98132450E-02
-4.75456254E-01	-4.75892991E-01	9.17721474E-02

GRADIENTS ARE O. K.

Fig. 2. The results for the LC filter example.

```
INPUT DATA
----------

FOLLOWING METHOD HAVE BEEN CALLED

FLETCHER METHOD

NUMBER OF INDEPENDENT VARIABLES..................................N=    6

MAXIMUM NUMBER OF ALLOWABLE ITERATIONS.........................MAX=  100

INTERMEDIATE OUTPUT TO BE PRINTED EVERY IPRINT ITERATIONS.....IPRINT=   20

STARTING VALUE FOR VECTOR X(I)..........................XSTRT( 1)=  1.00000000E+00
                                                        XSTRT( 2)=  1.00000000E+00
                                                        XSTRT( 3)=  1.00000000E+00
                                                        XSTRT( 4)=  1.00000000E+00
                                                        XSTRT( 5)=  1.00000000E+00
                                                        XSTRT( 6)=  1.00000000E+00

TEST QUANTITIES TO BE USED IN FLETCHER METHOD...............EPS( 1)=  1.00000000E-04
                                                            EPS( 2)=  1.00000000E-04
                                                            EPS( 3)=  1.00000000E-04
                                                            EPS( 4)=  1.00000000E-04
                                                            EPS( 5)=  1.00000000E-04
                                                            EPS( 6)=  1.00000000E-04

ESTIMATE OF LOWER BOUND ON FUNCTION TO BE MINIMIZED.............EST= -1.00000000E+01
```

Fig. 2. [cont.]

OPTIMIZATION BY FLETCHER METHOD

ITERATION NUMBER	FUNCTION EVALUATIONS	TIME ELAPSED (SECONDS)	OBJECTIVE FUNCTION	VARIABLE VECTOR X(I)	GRADIENT VECTOR G(I)
0	1	1.15000000E-01	3.98791052E+01	1.00000000E+00	-4.75456254E-01
				1.00000000E+00	-5.86685597E+00
				1.00000000E+00	-1.08873021E+01
				1.00000000E+00	-1.08873021E+01
				1.00000000E+00	-5.86685597E+00
				1.00000000E+00	-4.75456254E-01
20	33	3.95900000E+00	3.02553915E-01	1.02885986E+00	6.15166812E-01
				1.63829385E+00	-5.67250488E-01
				1.91392608E+00	-1.19312372E+00
				1.90089580E+00	-1.39171271E+00
				1.63324024E+00	-7.79015170E-01
				1.03804420E+00	7.13553906E-01

IEXIT= 1CRITERION FOR OPTIMUM HAS BEEN SATISFIED

FOLLOWING IS THE OPTIMUM SOLUTION

F = 2.43263030E-01

X(1)= 1.01329556E+00
X(2)= 1.65330685E+00
X(3)= 1.91543883E+00
X(4)= 1.91549271E+00
X(5)= 1.65328478E+00
X(6)= 1.01333880E+00

EXECUTION TIME IN SECONDS= 5.80900

Fig. 2. [cont.]

P = 10

FINAL RESPONSE OF THE CIRCUIT

FREQUENCY	INSERTION LOSS
9.99375 7377616E-11	0.
9.000000009001E-02	5.458616179835E-04
1.800000000800E-01	7.101371031245E-03
2.700000000700E-01	2.470787263300E-02
3.600000000600E-01	4.271376270489E-02
4.500000000500E-01	3.913906022018E-02
5.400000000400E-01	1.206470209338E-02
6.300000000300E-01	2.087945390354E-03
7.200000000200E-01	4.011430930891E-02
8.100000000100E-01	4.187772189332E-02
9.000000000000E-01	4.154605395557E-02

FREQUENCY	INSERTION LOSS
1.750000000000E+00	3.982426353284E+01

FREQUENCY	INSERTION LOSS
2.500000000000E+00	6.033997021533E+01

Fig. 2. [cont.]

ACKNOWLEDGEMENTS

The assistance of Dr. C. Charalambous, now at the University of Waterloo, is gratefully acknowledged. Contributions by V.K. Jha, now with RCA Limited, St. Anne-de-Bellevue, Quebec, are also acknowledged.

REFERENCES

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

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

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

[4] J.R. Popović, "General programs for least pth and near minimax approximation", M.Eng. Thesis, McMaster University, Hamilton, Ontario, October 1972.

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

[6] J.W. Bandler and V.K. Jha, "Network optimization computer program package", Communications Research Laboratory, McMaster University, Hamilton, Canada, CRL Internal Report Series No. CRL-5, November 1972.

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

LISTING OF THE VERSION CRL1

```
      SUBROUTINE CRL1 (N1,A,XSTRT,G,Y,PY,DUM1,DUM2,EPS,H,GRAD,NUMB,XX,X,V1
     1X1,ERROR,EHELP,AP,INUMB)                                          V1
C                                                                       V1
C          SUBROUTINE WHICH COORDINATES                                 V1
C          THE OTHER SUBROUTINES                                        V1
C                                                                       V1
      EXTERNAL FUNCS,W,FCT,FUNCT                                        V1
      DIMENSION FX(100), IOBJ(100), XX1(100), XX2(100), XX3(100)        V1
      DIMENSION A(1), G(1), Y(1), PY(1), XSTRT(15), DUM1(1), DUM2(1), EPV1
     1S(14), H(1), GRAD(1), XX(3,1), X(1), X1(1), ERROR(1), EHELP(1), APV1
     2(1), INUMB(1)                                                     V1
      DIMENSION NUMB(100), IPA(100)                                     V1
      COMMON /BLK/ KO                                                   V1
      COMMON /SP/ FUN(50)                                               V1
      COMMON /SP1/ WT(50)                                               V1
      COMMON /BLACK/ IC(100),AA(100),B(100),MM,NE,RL,NC,ND,KVR,AB(100),FV1
     1M,WC                                                              V1
      LOGICAL B                                                         V1
      LOGICAL CONV,UNITH                                                V1
      COMPLEX RL                                                        V1
      ERR(Z)=EPSNP(Z,IINT,FCT,W,A,N1,GRAD,APP,PSI,XX,1)                 V1
      UNITH=.TRUE.                                                      V1
      READ (5,42) MM,NE,NC,ND,KVR                                       V1
      READ (5,42) (IC(L),L=1,MM)                                        V1
      J=2*ND+NC+1                                                       V1
      READ (5,43) (AA(I),I=1,NE),(AB(I),I=1,J)                          V1
      READ (5,42) (INUMB(I),I=1,NE)                                     V1
      DO 1 I=1,NE                                                       V1
      B(I)=.TRUE.                                                       V1
      IF (INUMB(I).EQ.0) B(I)=.FALSE.                                   V1
    1 CONTINUE                                                          V1
      READ (5,42) NINT                                                  V1
      READ (5,43) (XX1(I),XX2(I),I=1,NINT)                              V1
      READ (5,42) (NUMB(I),I=1,NINT)                                    V1
      READ (5,43) (FUN(I),I=1,NINT)                                     V1
      READ (5,43) (XX3(I),I=1,NINT)                                     V1
      READ (5,43) (WT(I),I=1,NINT)                                      V1
      READ (5,42) (IOBJ(I),I=1,NINT)                                    V1
      READ (5,43) R,FM,WC                                               V1
      RL=CMPLX(R,0.)                                                    V1
      READ (5,42) N1                                                    V1
      READ (5,43) (XSTRT(I),I=1,N1)                                     V1
      READ (5,42) MET,MAX,IPRINT                                        V1
      IF (MET.EQ.1) READ (5,43) (EPS(I),I=1,N1)                         V1
      IF (MET.EQ.2) READ (5,43) EPS1                                    V1
      READ (5,43) EST,DIF,PSI                                           V1
      READ (5,42) ITER                                                  V1
      READ (5,42) (IPA(I),I=1,ITER)                                     V1
      IOPT=1                                                            V1
      IREAD=0                                                           V1
      DO 2 I=1,NINT                                                     V1
      XX(1,I)=(XX1(I)/FM)+FLOAT(IOBJ(I)-1)*10.0                         V1
      XX(2,I)=(XX2(I)/FM)+FLOAT(IOBJ(I)-1)*10.                          V1
      IF (IOBJ(I).EQ.0) XX(1,I)=XX1(I)+30.                              V1
      IF (IOBJ(I).EQ.0) XX(2,I)=XX2(I)+30.                              V1
      XX(3,I)=XX3(I)                                                    V1
    2 CONTINUE                                                          V1
      DO 3 I=1,N1                                                       V1
      A(I)=XSTRT(I)                                                     V1
```

```
3     CONTINUE
      K=0
      WRITE (6,49)
      WRITE (6,50)
      IF (IREAD.EQ.0) IREAD=2
      GO TO (7,4), IREAD
4     DO 6 J=1,NINT
      IINT=J
      IF (XX(1,J).LE.10.) WRITE (6,46)
      IF (XX(1,J).LE.20.0.AND.XX(1,J).GT.10.) WRITE (6,47)
      IF (XX(1,J).LE.30.0.AND.XX(1,J).GT.20.) WRITE (6,48)
      L=NUMB(J)+1
      IF (NUMB(J).EQ.0) Z=XX(1,J)
      DO 6 I=1,L
      IF (NUMB(J).GT.0) Z=XX(1,J)+(XX(2,J)-XX(1,J))*(I-1)/NUMB(J)
      ER=ERR(Z)
      K=K+1
      ERROR(K)=ERR(Z)
      EHELP(K)=ERROR(K)*XX(3,J)
      X(K)=Z
      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 5
      WRITE (6,45) FX(K),ERT
5     CONTINUE
6     AP(K)=APP
      GO TO 11
7     DO 10 J=1,NINT
      IINT=J
      IF (XX(1,J).LE.10.) WRITE (6,46)
      IF (XX(1,J).LE.20.0.AND.XX(1,J).GT.10.) WRITE (6,47)
      IF (XX(1,J).LE.30.0.AND.XX(1,J).GT.20.) WRITE (6,48)
      K=K+1
      KL=K+NUMB(J)
C
      DO 9 I=K,KL
      ERROR(I)=ERR(X(I))
      L=I-K+1
      ERT=(ERROR(I)/WT(IINT))+FUN(IINT)
      Z=X(I)
      IF (IOBJ(J).EQ.1) FX(I)=Z*FM
      IF (IOBJ(J).EQ.2) FX(I)=(Z-10.)*FM
      IF (IOBJ(J).EQ.3) FX(I)=(Z-20.)*FM
      IF (IOBJ(J).EQ.0) GO TO 8
      WRITE (6,45) FX(I),ERT
8     CONTINUE
      EHELP(I)=ERROR(I)*XX(3,J)
      AP(I)=APP
9     CONTINUE
      K=KL
10    CONTINUE
11    EMAX=EHELP(1)
      DO 12 M=2,K
      EMAX=AMAX1(EMAX,EHELP(M))
12    CONTINUE
C     WRITE(6,6000)
```

```
C     WRITE(6,7000)
C     WRITE(6,5000)EMAX
      CALL ERRO (FCT,W,A,N1,K,GRAD,APP,PSI,2,NUMB,XX,X,X1,ERROR,EHELP,AP
     1,EMAX,N,INUMB,NINT,IP)
C     WRITE(6,4000)
C     WRITE(6,4100)
C     WRITE(6,8000)(J,X1(J),ERROR(J),J=1,N)
      IGRDCH=1
C
C          OPTIMIZATION
C
      DO 40 K=1,IOPT
      KR=1
      IF (K-1) 14,14,13
13    CONTINUE
14    DO 40 KK=1,ITER
      IP=IPA(KK)
      IF (KR.EQ.0) GO TO 15
C
C          DATA FOR THE OPTIMALITY
C          CRITERIA FOR THE OPTIMIZATION METHODS
C
      M=2
C
15    DO 16 I=1,N1
      A(I)=XSTRT(I)
16    CONTINUE
      IF (IGRDCH.GT.1) GO TO 17
      CALL GRDCHK (N1,A,G,PY,Y,GRAD,APP,PSI,NUMB,XX,X,X1,ERROR,EHELP,AP,
     1EMAX,N,INUMB,NINT,IP)
17    IF (KR.EQ.0) GO TO 18
      CALL INPUT (MET,M,MAX,N1,IPRINT,1,EPS1,EST,EPS,XSTRT)
18    IF (MET.EQ.0) MET=4
      INDEX=0
      GO TO (19,24,33,29), MET
19    IF (IPRINT.EQ.0) GO TO 20
      CALL WRITE1 (1)
20    CALL SECOND (T1)
      IF (KR.NE.0) GO TO 22
      DO 21 I=1,N1
      A(I)=DUM1(I)
21    CONTINUE
22    CALL VMM01 (N1,A,F,G,H,UNITH,EST,EPS,MAX,IPRINT,IEXIT,GRAD,APP,PSI
     1,NUMB,XX,X,X1,ERROR,EHELP,AP,EMAX,N,INUMB,NINT,IP)
      DO 23 I=1,N1
      DUM1(I)=A(I)
23    CONTINUE
      CALL SECOND (T2)
      CALL FINAL (A,F,N1)
      T=T2-T1
      WRITE (6,41) T
      GO TO 29
24    IF (IPRINT.EQ.0) GO TO 25
      CALL WRITE1 (2)
25    CALL SECOND (T1)
      IF (KR.NE.0) GO TO 27
      DO 26 I=1,N1
      A(I)=DUM2(I)
```

```
26      CONTINUE
27      CALL EMEP (FUNCT,N1,A,F,G,EST,EPS1,MAX,IER,H,IPRINT,GRAD,APP,PSI,N
       1UMB,XX,X,X1,ERROR,EHELP,AP,EMAX,N,INUMB,NINT,IP)
        DO 28 I=1,N1
        DUM2(I)=A(I)
28      CONTINUE
        CALL SECOND (T2)
        CALL FINAL (A,F,N1)
        T=T2-T1
        WRITE (6,41) T
29      INDEX=INDEX+1
        IF (M.EQ.1) GO TO 30
        GO TO 32
30      DO 31 I=1,N1
        A(I)=XSTRT(I)
31      CONTINUE
32      CONTINUE
C
33      KR=0
        WRITE (6,44) IP
        IF (KK-1) 36,34,34
34      IF (KK-ITER) 35,36,36
35      CONTINUE
36      CONTINUE
        KN=0
        KQ=0
        WRITE (6,49)
        WRITE (6,51)
        DO 39 J=1,NINT
        IINT=J
        IF (XX(1,J).LE.10.) WRITE (6,46)
        IF (XX(1,J).LE.20.0.AND.XX(1,J).GT.10.) WRITE (6,47)
        IF (XX(1,J).LE.30.0.AND.XX(1,J).GT.20.) WRITE (6,48)
        KQ=KQ+1
        KL=KQ+NUMB(J)
        DO 38 I=KQ,KL
        L=I-KQ+1
        ER=ERR(X(I))
        ERT=(ER/WT(IINT))+FUN(IINT)
        IF (IOBJ(J).EQ.0) GO TO 37
        WRITE (6,45) EX(I),ERT
37      CONTINUE
        KN=KN+1
38      CONTINUE
        KQ=KL
39      CONTINUE
C       WRITE(6,4000)
C       WRITE(6,4100)
C       WRITE(6,8000)(J,X1(J),ERROR(J),J=1,N)
C       WRITE(6,6000)
C       WRITE(6,7000)
C       WRITE(6,5000)EMAX
        IGRDCH=IGRDCH+1
40      CONTINUE
        RETURN
C
41      FORMAT (1HO,//25X,*EXECUTION TIME IN SECONDS=*,F10.5)
42      FORMAT (8I10)
```

```
43     FORMAT (5E16.8)
44     FORMAT (1H1,19X,*P  =*,I7/////)
45     FORMAT (12X,2E23.12)
46     FORMAT (/,20X,*FREQUENCY*,14X,*REFLECTION COEFF.*)
47     FORMAT (/,20X,*FREQUENCY*,14X,*INSERTION LOSS*)
48     FORMAT (/,20X,*FREQUENCY*,14X,*GROUP DELAY*)
49     FORMAT (20(/))
50     FORMAT (//23X,*RESPONSE AT THE STARTING POINT*,/23X,31(*-*))
51     FORMAT (//,23X,*FINAL RESPONSE OF THE CIRCUIT*,/23X,30(*-*))
       END
```

```
      FUNCTION FCT (Z,FUNCS,W,IINT,PSI,XX)
C
C         FUNCTION SUBROUTINE WHICH DEFINES
C         MODIFIED UPPER AND LOWER
C         SPECIFIED FUNCTION
C
      EXTERNAL FUNCS,W
      DIMENSION XX(3,1)
      FCT=FUNCS(Z,IINT)+PSI*XX(3,IINT)/W(Z,IINT)
      RETURN
      END
```

```
      FUNCTION FUNCS (FN,NINT)
C
C         FUNCTION SUBROUTINE WHICH DEFINES
C         UPPER AND LOWER
C         SPECIFIED FUNCTION
C
C
      COMMON /SP/ FUN(50)
      FUNCS=FUN(NINT)
      RETURN
      END
```

```
      FUNCTION W (X,IINT)
      COMMON /SP1/ WT(50)
C
C         FUNCTION SUBROUTINE WHICH DEFINES
C         UPPER AND LOWER
C         WEIGHTING FUNCTION
C
      W=WT(IINT)
      RETURN
      END
```

```
      FUNCTION EPSNP (Z,IINT,FCT,W,A,N1,GRAD,APP,PSI,XX,IPOINT)
C
C         FUNCTION SUBROUTINE WHICH CALCULATES
C         UPPER AND LOWER WEIGHTED ERROR FUNCTION
C
      EXTERNAL FUNCS,W,FCT
      DIMENSION A(1), GRAD(1), XX(3,1)
      IF (IPOINT) 1,2,1
1     CONTINUE
      CALL FCTAPP (Z,N1,A,APP,GRAD,IINT,1)
2     CONTINUE
      IF (PSI) 3,4,3
3     EPSNP=(APP-FCT(Z,FUNCS,W,IINT,PSI,XX))*W(Z,IINT)
      RETURN
4     EPSNP=(APP-FUNCS(Z,IINT))*W(Z,IINT)
      RETURN
      END
```

```
      SUBROUTINE ERRO (FCT,W,A,N1,K,GRAD,APP,PSI,INDIC,NUMB,XX,X,X1,ERRO
     1R,EHELP,AP,EMAX,N,INUMB,NINT,IP)
C
C        SUBROUTINE WHICH SELECTS THE WEIGHTED
C        ERROR FUNCTION OF INTEREST FOR
C        THE OBJECTIVE FUNCTION
C
      EXTERNAL FUNCS,W,FCT
      DIMENSION A(1), GRAD(1), NUMB(1), XX(3,1), X(1), X1(1), ERROR(1),
     1EHELP(1), AP(1), INUMB(1)
      ERR(Z)=EPSNP(Z,IINT,FCT,W,A,N1,GRAD,APP,PSI,XX,IPOINT)
      GO TO (1,9), INDIC
1     CONTINUE
      IPOINT=1
      K=0
      KL=0
      DO 7 J=1,NINT
      IINT=J
      IF (J.EQ.1) GO TO 2
      KL=KL+L
2     L=NUMB(J)+1
      DO 6 I=1,L
      K=K+1
      IF (J.EQ.1) GO TO 5
      DO 4 KK=1,KL
      IF (X(K)-X(KK)) 4,3,4
3     AP(K)=AP(KK)
      APP=AP(K)
      IPOINT=0
      GO TO 5
4     CONTINUE
5     ERROR(K)=ERR(X(K))
      EHELP(K)=ERROR(K)*XX(3,J)
      IF (IPOINT.NE.0) AP(K)=APP
      IPOINT=1
6     CONTINUE
7     CONTINUE
      EMAX=EHELP(1)
      DO 8 M=2,K
      EMAX=AMAX1(EMAX,EHELP(M))
8     CONTINUE
9     CONTINUE
      IF (EMAX) 10,11,11
10    IP=-IABS(IP)
      GO TO 12
11    IP=IABS(IP)
12    K=0
      N=0
      INUMB(1)=0
      DO 16 J=1,NINT
      IINT=J
      L=NUMB(J)+1
      DO 15 I=1,L
      K=K+1
      IF (IP) 14,13,13
13    IF (EHELP(K)) 15,14,14
14    N=N+1
      X1(N)=X(K)
      ERROR(N)=ERROR(K)
```

```
      EHELP(N)=AP(K)
15    CONTINUE
      INUMB(J+1)=N
16    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FUNCT (N1,A,OBJ,G,GRAD,APP,PSI,NUMB,XX,X,X1,ERROR,EHELP
     1,AP,EMAX,N,INUMB,NINT,IP)
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
      EXTERNAL FUNCS,W,FCT
      DIMENSION A(1), GRAD(1), NUMB(1), XX(2,1), X(1), X1(1), ERROR(1),
     1EHELP(1), AP(1), INUMB(1), G(1)
      OBJP=0.
      GRADP=0.
      DO 1 K=1,N1
      G(K)=0.
    1 CONTINUE
      CALL ERRO (FCT,W,A,N1,K,GRAD,APP,PSI,1,NUMB,XX,X,X1,ERROR,EHELP,AP
     1,EMAX,N,INUMB,NINT,IP)
      DO 7 I=1,N
      Z=X1(I)
      DEL=ERROR(I)/EMAX
      OBJI=DEL**IP
      GRADI=DEL**(IP-1)
      OBJP=OBJP+OBJI
      DO 4 J=1,NINT
      IF (I-INUMB(J+1)) 2,2,4
    2 IF (I-INUMB(J)) 4,4,3
    3 IINT=J
      GO TO 5
    4 CONTINUE
    5 CONTINUE
      APP=EHELP(I)
      CALL FCTAPP (Z,N1,A,APP,GRAD,IINT,2)
      DO 6 K=1,N1
      GRAD(K)=GRADI*W(Z,IINT)*GRAD(K)
      G(K)=G(K)+GRAD(K)
    6 CONTINUE
    7 CONTINUE
      PR=1./IP
      OBJ=EMAX*(OBJP**PR)
      GRP=OBJP**(PR-1.)
      DO 8 K=1,N1
      G(K)=GRP*G(K)
    8 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GRDCHK (N,X,G,PY,Y,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHE
     1LP,AP,EMAX,NP,INUMB,NINT,IP)
      DIMENSION X(1), G(1), PY(1), Y(1), GRAD(1), NUMB(1), XX(3,1), XP(1
     1), X1(1), ERROR(1), EHELP(1), AP(1), INUMB(1)
      CALL FUNCT (N,X,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAX
     1,NP,INUMB,NINT,IP)
      DO 1 I=1,N
      DELX=1.E-4*X(I)
      IF (ABS(DELX).LT.1.E-40) DELX=1.E-20
      X(I)=X(I)+DELX
      CALL FUNCT (N,X,FNEW,PY,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,
     1EMAX,NP,INUMB,NINT,IP)
      Y(I)=(FNEW-F)/DELX
      X(I)=X(I)-DELX
1     CONTINUE
      DO 2 I=1,N
      IF (ABS(Y(I)).LT.1.E-20) Y(I)=1.E-20
      IF (ABS(G(I)).LT.1.E-20) G(I)=1.E-20
      PY(I)=ABS((Y(I)-G(I))/Y(I))*100.
2     CONTINUE
      WRITE (6,6)
      WRITE (6,7)
      WRITE (6,8) (I,X(I),I=1,N)
      WRITE (6,9)
      DO 3 I=1,N
      WRITE (6,10) G(I),Y(I),PY(I)
3     CONTINUE
      DO 4 I=1,N
      IF (PY(I).GT.10.) GO TO 5
4     CONTINUE
      WRITE (6,11)
      RETURN
5     WRITE (6,12)
      CALL EXIT
C
6     FORMAT (20(/))
7     FORMAT (1H0,5X,*GRADIENTS CHECKING*,/,6X,18(*-*),//,6X,*GRADIENTS
     1HAVE BEEN CHECKED AT THE FOLLOWING POINT*/)
8     FORMAT (10X,*X(*,I2,*)=*,E16.8)
9     FORMAT (///,1H0,5X,*ANALYTICAL GRADIENTS*,5X,*NUMERICAL GRADIENTS*
     1,5X,*PERCENTAGE ERROR*,/)
10    FORMAT (1H0,5X,3(E16.8,9X))
11    FORMAT (1H0,//,6X,*GRADIENTS ARE O. K.*)
12    FORMAT (1H0,//,6X,*YOUR PROGRAM HAS BEEN TERMINATED BECAUSE GRADIE
     1NTS ARE INCORRECT*,/6X,*PLEASE CHECK IT AGAIN*)
      END
```

```
      SUBROUTINE INPUT (MET,M,MAX,N,IPRINT,IDATA,EPS1,EST,EPS,XSTRT)
      DIMENSION XSTRT(1), EPS(1)
      WRITE (6,5)
      IF (MET.EQ.0) MET=4
      INDEX=0
      GO TO (1,2,4,3), MET
1     WRITE (6,6)
      GO TO 3
2     WRITE (6,7)
3     CONTINUE
      WRITE (6,8) N
      WRITE (6,9) MAX
      WRITE (6,10) IPRINT
      WRITE (6,11) XSTRT(1)
      WRITE (6,12) (I,XSTRT(I),I=2,N)
      IF (MET.EQ.1) WRITE (6,13) EPS(1)
      IF (MET.EQ.1) WRITE (6,14) (I,EPS(I),I=2,N)
      IF (MET.EQ.2) WRITE (6,15) EPS1
      WRITE (6,16) EST
      RETURN
C
4     WRITE (6,17)
      CALL EXIT
C
5     FORMAT (20(/),1H0,*INPUT DATA*,/,1X,10(*-*),//,1X,*FOLLOWING METHO
     1D HAVE BEEN CALLED*/)
6     FORMAT (1H0,*FLETCHER METHOD*)
7     FORMAT (1H0,*FLETCHER-POWELL METHOD*)
8     FORMAT (1H0,/1X,*NUMBER OF INDEPENDENT VARIABLES*,36(*.*),*N=*,I5,
     1/)
9     FORMAT (1H0,*MAXIMUM NUMBER OF ALLOWABLE ITERATIONS*,27(*.*),*MAX=
     1*,I5,/)
10    FORMAT (1H0,*INTERMEDIATE OUTPUT TO BE PRINTED EVERY IPRINT ITERAT
     1IONS*,5(*.*),*IPRINT=*,I5,/)
11    FORMAT (1H0,*STARTING VALUE FOR VECTOR X(I)*,29(*.*),*XSTRT( 1)=*,
     1E16.8)
12    FORMAT (1H0,59X,*XSTRT(*,I2,*)=*,E16.8)
13    FORMAT (1H0,/,1H0,*TEST QUANTITIES TO BE USED IN FLETCHER METHOD*,
     116(*.*),*EPS( 1)=*,E16.8)
14    FORMAT (1H0,61X,*EPS(*,I2,*)=*,E16.8)
15    FORMAT (1H0,/,1H0,*TEST QUANTITY TO BE USED IN FLETCHER-POWELL MET
     1HOD*,14(*.*),*EPS1=*,E16.8)
16    FORMAT (1H0,/,1H0,*ESTIMATE OF LOWER BOUND ON FUNCTION TO BE MINIM
     1IZED*,14(*.*),*EST=*,E16.8)
17    FORMAT (1H0,*NONE OF THE OPTIMIZATION METHODS HAVE BEEN CALLED*,/,
     11X,*PLEASE CHECK THE VALUE OF MET*,/,1X,*REMAINDER*,/,1X,*MET=1
     2 FLETCHER METHOD WOULD BE CALLED*,/,1X,*MET=2    FLETCHER-POWELL M
     3ETHOD WOULD BE CALLED*)
      END
```

```
      SUBROUTINE WRITE1 (N)
      WRITE (6,4)
      GO TO (1,2), N
1     WRITE (6,5)
      GO TO 3
2     WRITE (6,6)
3     CONTINUE
      WRITE (6,7)
      RETURN
C
C
C
4     FORMAT (20(/))
5     FORMAT (* OPTIMIZATION BY FLETCHER METHOD*,/,1X,31(*-*))
6     FORMAT (* OPTIMIZATION BY FLETCHER-POWELL METHOD*,/,1HO,38(*-*))
7     FORMAT (1HO,*ITERATION*,2X,*FUNCTION*,6X,*TIME ELAPSED*,8X,*OBJECT
     1IVE *,13X,*VARIABLE VECTOR X(I)*,9X,*GRADIENT VECTOR G(I)*,/1HO,*N
     2UMBER*,5X,*EVALUATIONS*,3X,*(SECONDS)*,11X,*FUNCTION*,/)
      END
```

```
      SUBROUTINE WRITE2 (X,N,G,F,NUMF,ITER,TIME)
      DIMENSION X(1), G(1)
      WRITE (6,1) ITER,NUMF,TIME,F,((X(I),G(I)),I=1,N)
      RETURN
C
C
C
1     FORMAT (1HO,I5,7X,I5,5X,E16.8,3X,F16.8,12X,95(E16.8,13X,E16.8,/,70
     1X))
      END
```

```
      SUBROUTINE FINAL (X,F,N)
      DIMENSION X(1)
      COMMON /BLK/ KO
      WRITE (6,3)
      IF (KO.EQ.0) GO TO 1
      WRITE (6,4)
      GO TO 2
    1 WRITE (6,5)
    2 CONTINUE
      WRITE (6,6) F
      WRITE (6,7) (I,X(I),I=1,N)
      RETURN
C
C
C
    3 FORMAT (20(/))
    4 FORMAT (41X,*FOLLOWING IS THE OPTIMUM SOLUTION*,/,41X,33(*-*))
    5 FORMAT (45X,*RESULTS AT LAST ITERATION*/,45X,25(*-*))
    6 FORMAT (,//,48X,*F =*,E16.8,/)
    7 FORMAT (,44X,*X(*,I2,*)=*,E16.8)
      END
```

```
      SUBROUTINE VMMO1 (N,X,F,G,H,UNITH,FEST,EPS,MAXFN,IPRINT,IEXIT,GRAD
     1,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAX,NP,INUMB,NINT,IP)
      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), INUMB(1)
      LOGICAL CONV,UNITH
      COMMON /BLK/ KO
      CALL SECOND (T3)
      KO=0
      CALL FUNCT (N,X,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAX
     1,NP,INUMB,NINT,IP)
      IF (F.LT.FEST) GO TO 23
      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 7
      IF (MOD(ITN,IPRINT).NE.0) GO TO 7
      CALL SECOND (T4)
      TIME=T4-T3
      CALL WRITE2 (X,N,G,F,NFNS,ITN,TIME)
7     IEXIT=1
      IF (CONV) GO TO 24
      IEXIT=2
      IF (GDX.GE.0.) GO TO 24
      Z=1.
      IF (ITN.LT.N.AND.UNITH) Z=STEP
      W=2.*(FEST-F)/GDX
      IF (W.LT.Z) Z=W
      STEP=Z
8     GDX=GDX*Z
```

```
      DO 9 I=1,N
      H(IDX+I)=H(IDX+I)*Z
      X(I)=X(I)+H(IDX+I)
9     CONTINUE
      CALL FUNCT (N,X,FP,H,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMA
     1X,NP,INUMB,NINT,IP)
      IF (FP.LT.FEST) GO TO 23
      NFNS=NFNS+1
      IEXIT=3
      IF (ITN.EQ.MAXFN) GO TO 24
      GPDX=0.
      DO 10 I=1,N
      H(IDG+I)=H(I)-G(I)
      GPDX=GPDX+H(I)*H(IDX+I)
10    CONTINUE
      DGDX=GPDX-GDX
      IF (F.GT.FP-.0001*GDX) GO TO 12
      IEXIT=4
      IF (GPDX.LT.0..AND.ITN.GT.N) GO TO 24
      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 11 I=1,N
      X(I)=X(I)-H(IDX+I)
11    CONTINUE
      GO TO 14
12    F=FP
      DO 13 I=1,N
      G(I)=H(I)
13    CONTINUE
      IF (DGDX.GT.0.) GO TO 15
      GDX=GPDX
      Z=4.
14    STEP=Z*STEP
      GO TO 8
15    IF (GPDX.LT.0.5*GDX) STEP=2.*STEP
      DGHDG=0.
      DO 19 I=1,N
      Z=0.
      IJ=IH+I
      IF (I.EQ.1) GO TO 17
      II=I-1
      DO 16 J=1,II
      Z=Z+H(IJ)*H(IDG+J)
      IJ=IJ+N-J
16    CONTINUE
17    DO 18 J=I,N
      Z=Z+H(IJ)*H(IDG+J)
      IJ=IJ+1
18    CONTINUE
      DGHDG=DGHDG+Z*H(IDG+I)
      H(I)=Z
19    CONTINUE
      IF (DGHDG.LT.0.0) DGHDG=DGDX*0.01
      IF (DGDX.LT.DGHDG) GO TO 21
      W=1.0+DGHDG/DGDX
      DO 20 I=1,N
```

```
      H(IDX+I)=W*H(IDX+I)-H(I)
20    CONTINUE
      DGDX=DGDX+DGHDG
      DGHDG=DGDX
21    IJ=IH
      DO 22 I=1,N
      W=H(IDX+I)/DGDX
      Z=H(I)/DGHDG
      DO 22 J=I,N
      IJ=IJ+1
22    H(IJ)=H(IJ)+W*H(IDX+J)-Z*H(J)
      ITN=ITN+1
      GO TO 2
23    IEXIT=5
24    IF (IEXIT.EQ.1) KO=1
      IF (IPRINT.EQ.0) RETURN
      GO TO (25,26,27,26,28), IEXIT
25    WRITE (6,30) IEXIT
      GO TO 29
26    WRITE (6,31) IEXIT
      GO TO 29
27    WRITE (6,32) IEXIT
      GO TO 29
28    WRITE (6,33) IEXIT
29    CONTINUE
      RETURN
C
C
C
30    FORMAT (/,1HO,*IEXIT=*,I2,*CRITERION FOR OPTIMUM HAS BEEN SATISFIE
     1D*)
31    FORMAT (/,1HO,*IEXIT=*,I2,*EITHER OF THE FOLLOWING THINGS HAS HAPP
     1 ENED*,/,9X,*1. EPS CHOSEN IS TOO SMALL*,/,9X,*2. GRADIENTS ARE NO
     2  CORRECT*,/,9X,*3. MATRIX H GOES SINGULAR*)
32    FORMAT (/,1HO,*IEXIT=*,I2,*MAXIMUM NUMBER OF ALLOWABLE ITERATION H
     1 AS BEEN EXCEEDED*)
33    FORMAT (/,1HO,*IEXIT=*,I2,*FUNCTION VALUE LESS THAN MINIMUM ESTIMA
     1 TED HAS BEEN DETECTED*)
      END
```

```
      SUBROUTINE FMFP (FUNCT,N,X,F,G,EST,EPS,LIMIT,IER,H,IPRINT,GRAD,APP
     1,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAX,NP,INUMB,NINT,IP)
      COMMON /BLK/ KO
C
C        TO FIND A LOCAL MINIMUM OF A FUNCTION OF SEVERAL VARIABLES
C        BY THE METHOD OF FLETCHER AND POWELL
C
      DIMENSION H(1), X(1), G(1), GRAD(1), NUMB(1), XX(3,1), XP(1), X1(1
     1), ERROR(1), EHELP(1), AP(1), INUMB(1)
C
C     COMPUTE FUNCTION VALUE AND GRADIENT VECTOR FOR INITIAL ARGUMENT
      KO=0
      CALL SECOND (T3)
      CALL FUNCT (N,X,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAX
     1,NP,INUMB,NINT,IP)
      KOUNT=0
      NUMF=1
      CALL SECOND (T4)
      TIME=T4-T3
      IF (IPRINT.EQ.0) GO TO 1
      CALL WRITE2 (X,N,G,F,NUMF,KOUNT,TIME)
    1 CONTINUE
C
C     RESET ITERATION COUNTER AND GENERATE IDENTITY MATRIX
      IER=0
      KK=0
      N2=N+N
      N3=N2+N
      N31=N3+1
    2 K=N31
      DO 5 J=1,N
      H(K)=1.
      NJ=N-J
      IF (NJ) 6,6,3
    3 DO 4 L=1,NJ
      KL=K+L
      H(KL)=0.
    4 CONTINUE
      K=KL+1
    5 CONTINUE
C
C     START ITERATION LOOP
    6 IF (KOUNT.EQ.0) GO TO 7
      IF (KK.NE.IPRINT) GO TO 7
      KK=0
      CALL SECOND (T4)
      TIME=T4-T3
      CALL WRITE2 (X,N,G,F,NUMF,KOUNT,TIME)
    7 CONTINUE
      KOUNT=KOUNT+1
      KK=KK+1
C
C     SAVE FUNCTION VALUE, ARGUMENT VECTOR AND GRADIENT VECTOR
      OLDF=F
      DO 11 J=1,N
      K=N+J
      H(K)=G(J)
      K=K+N
      H(K)=X(J)
```

```
C
C        DETERMINE DIRECTION VECTOR H
      K=J+N3
      T=0.
      DO 10 L=1,N
      T=T-G(L)*H(K)
      IF (L-J) 8,9,9
8     K=K+N-L
      GO TO 10
9     K=K+1
10    CONTINUE
      H(J)=T
11    CONTINUE
C
C        CHECK WHETHER FUNCTION WILL DECREASE STEPPING ALONG H.
      DY=0.
      HNRM=0.
      GNRM=0.
C
C        CALCULATE DIRECTIONAL DERIVATIVE AND TESTVALUES FOR DIRECTION
C        VECTOR H AND GRADIENT VECTOR G.
      DO 12 J=1,N
      HNRM=HNRM+ABS(H(J))
      GNRM=GNRM+ABS(G(J))
      DY=DY+H(J)*G(J)
12    CONTINUE
C
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DIRECTIONAL
C        DERIVATIVE APPEARS TO BE POSITIVE OR ZERO.
      IF (DY) 13,57,57
C
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DIRECTION
C        VECTOR H IS SMALL COMPARED TO GRADIENT VECTOR G.
13    IF (HNRM/GNRM-EPS) 57,57,14
C
C        SEARCH MINIMUM ALONG DIRECTION H
C
C        SEARCH ALONG H FOR POSITIVE DIRECTIONAL DERIVATIVE
14    FY=F
      ALFA=2.*(EST-F)/DY
      AMBDA=1.
C
C        USE ESTIMATE FOR STEPSIZE ONLY IF IT IS POSITIVE AND LESS THAN
C        1. OTHERWISE TAKE 1. AS STEPSIZE
      IF (ALFA) 17,17,15
15    IF (ALFA-AMBDA) 16,17,17
16    AMBDA=ALFA
17    ALFA=0.
C
C        SAVE FUNCTION AND DERIVATIVE VALUES FOR OLD ARGUMENT
18    FX=FY
      DX=DY
C
C        STEP ARGUMENT ALONG H
      DO 19 I=1,N
      X(I)=X(I)+AMBDA*H(I)
19    CONTINUE
C
```

```
C        COMPUTE FUNCTION VALUE AND GRADIENT FOR NEW ARGUMENT
      CALL FUNCT (N,X,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAX
     1,NP,INUMB,NINT,IP)
      NUMF=NUMF+1
      FY=F
C
C        COMPUTE DIRECTIONAL DERIVATIVE DY FOR NEW ARGUMENT.  TERMINATE
C        SEARCH, IF DY IS POSITIVE. IF DY IS ZERO THE MINIMUM IS FOUND
      DY=0.
      DO 20 I=1,N
      DY=DY+G(I)*H(I)
20    CONTINUE
      IF (DY) 21,41,24
C
C        TERMINATE SEARCH ALSO IF THE FUNCTION VALUE INDICATES THAT
C        A MINIMUM HAS BEEN PASSED
21    IF (FY-FX) 22,24,24
C
C        REPEAT SEARCH AND DOUBLE STEPSIZE FOR FURTHER SEARCHES
22    AMBDA=AMBDA+ALFA
      ALFA=AMBDA
C        END OF SEARCH LOOP
C
C        TERMINATE IF THE CHANGE IN ARGUMENT GETS VERY LARGE
      IF (HNRM*AMBDA-1.E10) 18,18,23
C
C        LINEAR SEARCH TECHNIQUE INDICATES THAT NO MINIMUM EXISTS
23    IER=2
      GO TO 62
C
C        INTERPOLATE CUBICALLY IN THE INTERVAL DEFINED BY THE SEARCH
C        ABOVE AND COMPUTE THE ARGUMENT X FOR WHICH THE INTERPOLATION
C        POLYNOMIAL IS MINIMIZED
24    T=0.
25    IF (AMBDA) 26,41,26
26    Z=3.*(FX-FY)/AMBDA+DX+DY
      ALFA=AMAX1(ABS(Z),ABS(DX),ABS(DY))
      DALFA=Z/ALFA
      DALFA=DALFA*DALFA-DX/ALFA*DY/ALFA
      IF (DALFA) 57,27,27
27    W=ALFA*SQRT(DALFA)
      ALFA=DY-DX+W+W
      IF (ALFA) 28,29,28
28    ALFA=(DY-Z+W)/ALFA
      GO TO 30
29    ALFA=(Z+DY-W)/(Z+DX+Z+DY)
30    ALFA=ALFA*AMBDA
      DO 31 I=1,N
      X(I)=X(I)+(T-ALFA)*H(I)
31    CONTINUE
C
C        TERMINATE, IF THE VALUE OF THE ACTUAL FUNCTION AT X IS LESS
C        THAN THE FUNCTION VALUES AT THE INTERVAL ENDS. OTHERWISE REDUCE
C        THE INTERVAL BY CHOOSING ONE END-POINT EQUAL TO X AND REPEAT
C        THE INTERPOLATION.  WHICH END-POINT IS CHOOSEN DEPENDS ON THE
C        VALUE OF THE FUNCTION AND ITS GRADIENT AT X
C
      NUMF=NUMF+1
```

```
      CALL FUNCT (N,X,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAX
     1,NP,INUMB,NINT,IP)
      IF (F-FX) 32,32,33
32    IF (F-FY) 41,41,33
33    DALFA=0.
      DO 34 I=1,N
      DALFA=DALFA+G(I)*H(I)
34    CONTINUE
      IF (DALFA) 35,38,38
35    IF (F-FX) 37,36,38
36    IF (DX-DALFA) 37,41,37
37    FX=F
      DX=DALFA
      T=ALFA
      AMBDA=ALFA
      GO TO 25
38    IF (FY-F) 40,39,40
39    IF (DY-DALFA) 40,41,40
40    FY=F
      DY=DALFA
      AMBDA=AMBDA-ALFA
      GO TO 24
C
C        TERMINATE, IF FUNCTION HAS NOT DECREASED DURING LAST ITERATION
41    IF (OLDF-F+EPS) 57,42,42
C
C        COMPUTE DIFFERENCE VECTORS OF ARGUMENT AND GRADIENT FROM
C        TWO CONSECUTIVE ITERATIONS
42    DO 43 J=1,N
      K=N+J
      H(K)=G(J)-H(K)
      K=N+K
C        IF AT LEAST N ITERATIONS HAVE BEEN EXECUTED. TERMINATE, IF
      H(K)=X(J)-H(K)
43    CONTINUE
C
C        TEST LENGTH OF ARGUMENT DIFFERENCE VECTOR AND DIRECTION VECTOR
C        BOTH ARE LESS THAN  EPS
      IER=0
      IF (KOUNT-N) 47,44,44
44    T=0.
      Z=0.
      DO 45 J=1,N
      K=N+J
      W=H(K)
      K=K+N
      T=T+ABS(H(K))
      Z=Z+W*H(K)
45    CONTINUE
      IF (HNRM-EPS) 46,46,47
46    IF (T-EPS) 62,62,47
C
C        TERMINATE, IF NUMBER OF ITERATIONS WOULD EXCEED  LIMIT
47    IF (KOUNT-LIMIT) 48,55,55
C
C        PREPARE UPDATING OF MATRIX H
48    ALFA=0.
      DO 52 J=1,N
```

```
      K=J+N3
      W=0.
      DO 51 L=1,N
      KL=N+L
      W=W+H(KL)*H(K)
      IF (L-J) 49,50,50
49    K=K+N-L
      GO TO 51
50    K=K+1
51    CONTINUE
      K=N+J
      ALFA=ALFA+W*H(K)
      H(J)=W
52    CONTINUE
C
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF RESULTS
C        ARE NOT SATISFACTORY
      IF (Z*ALFA) 53,2,53
C
C        UPDATE MATRIX H
53    K=N31
      DO 54 L=1,N
      KL=N2+L
      DO 54 J=L,N
      NJ=N2+J
      H(K)=H(K)+H(KL)*H(NJ)/Z-H(L)*H(J)/ALFA
54    K=K+1
      GO TO 6
C        END OF ITERATION LOOP
C
C        NO CONVERGENCE AFTER  LIMIT  ITERATIONS
55    IER=1
      IF (KK.NE.IPRINT) GO TO 56
      CALL WRITE2 (X,N,G,F,NUMF,KOUNT)
56    CONTINUE
      GO TO 62
C
C        RESTORE OLD VALUES OF FUNCTION AND ARGUMENTS
57    DO 58 J=1,N
      K=N2+J
      X(J)=H(K)
58    CONTINUE
      CALL FUNCT (N,X,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAX
     1,NP,INUMB,NINT,IP)
      NUMF=NUMF+1
C
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DERIVATIVE
C        FAILS TO BE SUFFICIENTLY SMALL
      IF (GNRM-EPS) 61,61,59
C
C        TEST FOR REPEATED FAILURE OF ITERATION
59    IF (IER) 62,60,60
60    IER=-1
      GO TO 2
61    IER=0
62    II=IER+2
      IF (II.EQ.2) KO=1
      IF (IPRINT.EQ.0) RETURN
```

```
      GO TO (63,64,65,66), II
63    WRITE (6,68) IER
      GO TO 67
64    WRITE (6,69) IER
      GO TO 67
65    WRITE (6,70) IER
      GO TO 67
66    WRITE (6,71) IER
67    RETURN
C
C
C
68    FORMAT (1HO,*IER=*,I2,* ERROR IN GRADIENTS CALCULATIONS*)
69    FORMAT (1HO,*IER=*,I2,* CRITERION FOR OPTIMUM HAS BEEN SATISFIED*)
70    FORMAT (1HO,*IER=*,I2,* MAXIMUM NUMBER OF ALLOWABLE ITERATIONS HAS
     1  BEEN EXCEEDED*)
71    FORMAT (1HO,*IER=*,I2,* CHANGE IN ARGUMENTS GETS TOO LARGE, LINEAR
     1  SEARCH INDICATES THAT NO MINIMUM EXISTS*)
      END
```

```
      SUBROUTINE FCTAPP (OMEGN,N1,X,APP,GRAD,IINT,INDIC)
      DIMENSION AD(50), ADJJ(50), G(50), X(1), GRAD(1)
      COMPLEX AD,ADJJ,CONRHO,RHO,I,IHAT,V,VHAT,INEW,VNEW,IOLD,VOLD,G,V1,
     1V2,IHAT1,IHAT2,PSIL,RL,GL,GC,GLLTEL,GLLTZO,GOMC,GOML,GSCTEL,GSCTOM
     2,GSCTZO,GOCTEL,GOCTZO,GOCTOM,GLCPBP,GLCPQ,GLCPOM,GLCPOR,GLCSQ,GLCS
     3OM,GLCSOR,GLCSXP
      COMPLEX POLD,PHASE
      COMMON /BLACK/ IC(100),A(100),B(100),M,NE,RL,NC,ND,KVR,AB(100),FM,
     1WC
      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)=2.*ATAN(1.0)*OMEGA*EL
      GC(OMEGA,V,VHAT)=-CMPLX(0.,OMEGA)*V*VHAT
      GOMC(OMEGA,V,VHAT)=-CMPLX(0.,C)*V*VHAT
      GOML(AL,I,IHAT)=I*IHAT*AL*CMPLX(0.,1.)
      GL(OMEGA,I,IHAT)=I*IHAT*OMEGA*CMPLX(0.,1.)
      GLLTZO(V1,V2,IHAT1,IHAT2,ZO)=(V1*IHAT1-V2*IHAT2)/ZO
      GLLTEL(OMEGA,EL,V1,V2,IHAT1,IHAT2)=2.*ATAN(1.)*OMEGA*(V1*IHAT2-V2*
     1IHAT1)/SIN(THETA(EL))
      GSCTEL(OMEGA,EL,ZO,I,IHAT)=I*IHAT*ZO*2.*ATAN(1.)*OMEGA*(1./COS(THE
     1TA(EL)))**2*CMPLX(0.,1.)
      GSCTOM(EL,ZO,I,IHAT)=I*IHAT*ZO*EL*CMPLX(0.,1.)/COS(THETA(EL))
      GSCTZO(EL,I,IHAT)=I*IHAT*SIN(THETA(EL))*CMPLX(0.,1.)/COS(THETA(EL)
     1)
      GOCTEL(ZO,EL,OMEGA,I,IHAT)=CMPLX(0.,1.)*ZO*I*IHAT*(THETA(EL)/EL)*(
     11./SIN(THETA(EL)))**2
      GOCTZO(OMEGA,EL,I,IHAT)=CMPLX(0.,-1.)*COS(THETA(EL))*I*IHAT/SIN(TH
     1ETA(EL))
      GOCTOM(OMEGA,EL,ZO,I,IHAT)=I*IHAT*ZO*THETA(EL)/(OMEGA*SIN(THETA(EL
     1))**2)*CMPLX(0.,1.)
      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.)
      GLCPOM(BP,OMEGA,OMEGAR,V,VHAT)=-V*VHAT*CMPLX(0.,(BP/2.)*(1.+(OMEGA
     1/OMEGAR)**2))
      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
      GLCSOM(OMEGA,XP,OMEGAR,I,IHAT)=((XP/2.)*(1.+(OMEGAR/OMEGA)**2))*I*
     1IHAT*CMPLX(0.,1.)
      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.
      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 114
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 90
      DELO=1.E-7*OMEGA
      KKGD=0
      IF (KKGD.EQ.0) GO TO 3
2     DO 89 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
      DO 88 KV=1,2
4     CONTINUE
      J=0
      K=0
      VOLD=RL
      IOLD=1.0
      DO 83 L=1,M
      MM=M+1-L
      NN=IC(MM)
      GO TO (5,9,13,17,21,25,31,37,43,48,55,62,69,76,79), 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     CALL SUB1 (IOLD,VOLD,A(KKK),OMEGA,INEW,VNEW)
      IF (B(KKK).AND.OMEGN.GT.10.) GO TO 8
      IF (B(KKK)) G(JJ)=GC(OMEGA,VOLD,VOLD)
8     CONTINUE
      IF (B(KKK)) AD(JJ)=VOLD
      KN=K+1
      GO TO 82
9     KKK=NE-K
      IF (B(KKK)) GO TO 10
      GO TO 11
10    JJ=N-J
      A(KKK)=X(JJ)
      J=J+1
11    CALL SUB2 (IOLD,VOLD,A(KKK),OMEGA,INEW,VNEW)
      IOLD=INEW-IOLD
      IF (B(KKK).AND.OMEGN.GT.10.) GO TO 12
      IF (B(KKK)) G(JJ)=GL(OMEGA,IOLD,IOLD)
12    CONTINUE
      IF (B(KKK)) AD(JJ)=IOLD
      KN=K+1
      GO TO 82
13    KKK=NE-K
      IF (B(KKK)) GO TO 14
      GO TO 15
14    JJ=N-J
```

```
      A(KKK)=X(JJ)
      J=J+1
15    CALL SUB3 (IOLD,VOLD,A(KKK),OMEGA,INEW,VNEW)
      IF (B(KKK).AND.OMEGN.GT.10.) GO TO 16
      IF (B(KKK)) G(JJ)=GL(OMEGA,IOLD,IOLD)
16    CONTINUE
      IF (B(KKK)) AD(JJ)=IOLD
      KN=K+1
      GO TO 82
17    KKK=NE-J
      IF (B(KKK)) GO TO 18
      GO TO 19
18    JJ=N-J
      A(KKK)=X(JJ)
      J=J+1
19    CALL SUB4 (IOLD,VOLD,A(KKK),OMEGA,INEW,VNEW)
      VOLD=VNEW-VOLD
      IF (B(KKK).AND.OMEGN.GT.10.) GO TO 20
      IF (B(KKK)) G(JJ)=GC(OMEGA,VOLD,VOLD)
20    CONTINUE
      IF (B(KKK)) AD(JJ)=VOLD
      KN=K+1
      GO TO 82
21    KK=K
      JOLD=J
      DO 24 II=1,2
      KKK=NE-KK
      IF (B(KKK)) GO TO 22
      GO TO 23
22    JJ=N-J
      A(KKK)=X(JJ)
      J=J+1
23    KK=KK+1
24    CONTINUE
      JV=NE-K-1
      JH=NE-K
      CALL SUB5 (VOLD,IOLD,A(JV),OMEGA,A(JH),VNEW,INEW)
      NJ=N-JOLD
      IF (B(JH)) AD(NJ)=VOLD
      IF (B(JH)) ADJJ(NJ)=VNEW
      IF (B(JH)) G(NJ)=GLLTZO(VNEW,VOLD,INEW,IOLD,A(JH))
      IF (B(JH)) JOLD=JOLD+1
      NJ=N-JOLD
      IF (B(JV)) AD(NJ)=VOLD
      IF (B(JV)) G(NJ)=GLLTEL(OMEGA,A(JV),VNEW,VOLD,INEW,IOLD)
      IF (B(JV)) ADJJ(NJ)=VNEW
      KN=K+2
      GO TO 82
25    KK=K
      JOLD=J
      DO 28 II=1,2
      KKK=NE-K
      IF (B(KKK)) GO TO 26
      GO TO 27
26    JJ=N-J
      A(KKK)=X(JJ)
      J=J+1
27    KK=KK+1
```

```
28    CONTINUE
      JV=NE-K-1
      JH=NE-K
      CALL SUB6 (VOLD,IOLD,A(JV),OMEGA,A(JH),VNEW,INEW)
      IOLD=INEW-IOLD
      NJ=N-JOLD
      IF (B(JH).AND.OMEGN.GT.10.) GO TO 29
      IF (B(JH)) G(NJ)=GSCTZO(A(JV),IOLD,IOLD)
29    CONTINUE
      IF (B(JH)) JOLD=JOLD+1
      IF (B(JH)) AD(NJ)=IOLD
      NJ=N-JOLD
      IF (B(JV).AND.OMEGN.GT.10.) GO TO 30
      IF (B(JV)) G(NJ)=GSCTEL(OMEGA,A(JV),A(JH),IOLD,IOLD)
30    CONTINUE
      IF (B(JV)) AD(NJ)=IOLD
      KN=K+2
      GO TO 82
31    KK=K
      JOLD=J
      DO 34 II=1,2
      KKK=NE-KK
      IF (B(KKK)) GO TO 32
      GO TO 33
32    JJ=N-J
      A(KKK)=X(JJ)
      J=J+1
33    KK=KK+1
34    CONTINUE
      JV=NE-K-1
      JH=NE-K
      CALL SUB7 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW)
      IOLD=INEW-IOLD
      NJ=N-JOLD
      IF (B(JH).AND.OMEGN.GT.10.) GO TO 35
      IF (B(JH)) G(NJ)=GOCTZO(OMEGA,A(JV),IOLD,IOLD)
35    CONTINUE
      IF (B(JH)) JOLD=JOLD+1
      IF (B(JH)) AD(NJ)=IOLD
      NJ=N-JOLD
      IF (B(JV).AND.OMEGN.GT.10.) GO TO 36
      IF (B(JV)) G(NJ)=GOCTEL(A(JH),A(JV),OMEGA,IOLD,IOLD)
36    CONTINUE
      IF (B(JV)) AD(NJ)=IOLD
      KN=K+2
      GO TO 82
37    KK=K
      JOLD=J
      DO 40 II=1,2
      KKK=NE-KK
      IF (B(KKK)) GO TO 38
      GO TO 39
38    JJ=N-J
      A(KKK)=X(JJ)
      J=J+1
39    KK=KK+1
40    CONTINUE
      JV=NE-K-1
```

```
      JH=NE-K
      CALL SUB8 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW)
      NJ=N-JOLD
      IF (B(JH).AND.OMEGN.GT.10.) GO TO 41
      IF (B(JH)) G(NJ)=GSCTZO(A(JV),IOLD,IOLD)
41    CONTINUE
      IF (B(JH)) JOLD=JOLD+1
      IF (B(JH)) AD(NJ)=IOLD
      NJ=N-JOLD
      IF (B(JV).AND.OMEGN.GT.10.) GO TO 42
      IF (B(JV)) G(NJ)=GSCTEL(OMEGA,A(JV),A(JH),IOLD,IOLD)
42    CONTINUE
      IF (B(JV)) AD(NJ)=IOLD
      KN=K+2
      GO TO 82
43    KK=K
      JOLD=J
      DO 46 II=1,2
      KKK=NE-KK
      IF (B(KKK)) GO TO 44
      GO TO 45
44    JJ=N-J
      A(KKK)=X(JJ)
      J=J+1
45    KK=KK+1
46    CONTINUE
      JV=NE-K-1
      JH=NE-K
      CALL SUB9 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW)
      NJ=N-JOLD
      IF (B(JH).AND.OMEGN.GT.10.) GO TO 47
      IF (B(JH)) G(NJ)=GOCTZO(OMEGA,A(JV),IOLD,IOLD)
47    CONTINUE
      IF (B(JH)) JOLD=JOLD+1
      IF (B(JH)) AD(NJ)=IOLD
      NJ=N-JOLD
      IF (B(JV).AND.OMEGN.GT.10.) GO TO 42
      IF (B(JV)) G(NJ)=GOCTEL(A(JH),A(JV),OMEGA,IOLD,IOLD)
      IF (B(JV)) AD(NJ)=IOLD
      KN=K+2
      GO TO 82
48    KK=K
      JOLD=J
      DO 51 II=1,3
      KKK=NE-KK
      IF (B(KKK)) GO TO 49
      GO TO 50
49    JJ=N-J
      A(KKK)=X(JJ)
      J=J+1
50    KK=KK+1
51    CONTINUE
      JK=NE-K-2
      JV=NE-K-1
      JH=NE-K
      CALL SUB10 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      NJ=N-JOLD
      IF (B(JH).AND.OMEGN.GT.10.) GO TO 52
```

```
      IF (B(JH)) G(NJ)=GLCSXP(OMEGA,A(JK),A(JV),IOLD,IOLD)
52    CONTINUE
      IF (B(JH)) JOLD=JOLD+1
      IF (B(JH)) AD(NJ)=IOLD
      NJ=N-JOLD
      IF (B(JV).AND.OMEGN.GT.10.) GO TO 53
      IF (B(JV)) G(NJ)=GLCSQ(A(JH),A(JK),A(JV),IOLD,IOLD)
53    CONTINUE
      IF (B(JV)) AD(NJ)=IOLD
      IF (B(JV)) JOLD=JOLD+1
      NJ=N-JOLD
      IF (B(JK).AND.OMEGN.GT.10.) GO TO 54
      IF (B(JK)) G(NJ)=GLCSOR(OMEGA,A(JH),A(JK),A(JV),IOLD,IOLD)
54    CONTINUE
      IF (B(JK)) AD(NJ)=IOLD
      IF (B(JK)) JOLD=JOLD+1
      KN=K+3
      GO TO 82
55    KK=K
      JOLD=J
      DO 58 II=1,3
      KKK=NE-KK
      IF (B(KKK)) GO TO 56
      GO TO 57
56    JJ=N-J
      A(KKK)=X(JJ)
      J=J+1
57    KK=KK+1
58    CONTINUE
      JK=NE-K-2
      JV=NE-K-1
      JH=NE-K
      CALL SUB11 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      IOLD=INEW-IOLD
      NJ=N-JOLD
      IF (B(JH).AND.OMEGN.GT.10.) GO TO 59
      IF (B(JH)) G(NJ)=GLCSXP(OMEGA,A(JK),A(JV),IOLD,IOLD)
59    CONTINUE
      IF (B(JH)) AD(NJ)=IOLD
      IF (B(JH)) JOLD=JOLD+1
      NJ=N-JOLD
      IF (B(JV).AND.OMEGN.GT.10.) GO TO 60
      IF (B(JV)) G(NJ)=GLCSQ(A(JH),A(JK),A(JV),IOLD,IOLD)
60    CONTINUE
      IF (B(JV)) JOLD=JOLD+1
      IF (B(JV)) AD(NJ)=IOLD
      NJ=N-JOLD
      IF (B(JK).AND.OMEGN.GT.10.) GO TO 61
      IF (B(JK)) G(NJ)=GLCSOR(OMEGA,A(JH),A(JK),A(JV),IOLD,IOLD)
61    CONTINUE
      IF (B(JK)) AD(NJ)=IOLD
      KN=K+3
      GO TO 82
62    KK=K
      JOLD=J
      DO 65 II=1,3
      KKK=NE-KK
      IF (B(KKK)) GO TO 63
```

```
      GO TO 64
63    JJ=N-J
      A(KKK)=X(JJ)
      J=J+1
64    KK=KK+1
65    CONTINUE
      JK=NE-K-2
      JV=NE-K-1
      JH=NE-K
      CALL SUB12 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      NJ=N-JOLD
      IF (B(JH).AND.OMEGN.GT.10.) GO TO 66
      IF (B(JH)) G(NJ)=GLCPBP(A(JH),OMEGA,A(JV),A(JK),VOLD,VOLD)
66    CONTINUE
      IF (B(JH)) AD(NJ)=VOLD
      IF (B(JH)) JOLD=JOLD+1
      NJ=N-JOLD
      IF (B(JV).AND.OMEGN.GT.10.) GO TO 67
      IF (B(JV)) G(NJ)=GLCPQ(A(JH),A(JV),A(JK),VOLD,VOLD)
67    CONTINUE
      IF (B(JV)) JOLD=JOLD+1
      IF (B(JV)) AD(NJ)=VOLD
      NJ=N-JOLD
      IF (B(JK).AND.OMEGN.GT.10.) GO TO 68
      IF (B(JK)) G(NJ)=GLCPOR(A(JH),OMEGA,A(JK),A(JV),VOLD,VOLD)
68    CONTINUE
      IF (B(JK)) AD(NJ)=VOLD
      KN=K+3
      GO TO 82
69    KK=K
      JOLD=J
      DO 72 II=1,3
      KKK=NE-KK
      IF (B(KKK)) GO TO 70
      GO TO 71
70    JJ=N-J
      A(KKK)=X(JJ)
      J=J+1
71    KK=KK+1
72    CONTINUE
      JK=NE-K-2
      JV=NE-K-1
      JH=NE-K
      CALL SUB13 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      VOLD=VNEW-VOLD
      NJ=N-JOLD
      IF (B(JH).AND.OMEGN.GT.10.) GO TO 73
      IF (B(JH)) G(NJ)=GLCPBP(A(JH),OMEGA,A(JV),A(JK),VOLD,VOLD)
73    CONTINUE
      IF (B(JH)) JOLD=JOLD+1
      IF (B(JH)) AD(NJ)=VOLD
      IF (B(JV).AND.OMEGN.GT.10.) GO TO 74
      IF (B(JV)) G(NJ)=GLCPQ(A(JH),A(JV),A(JK),VOLD,VOLD)
74    CONTINUE
      IF (B(JV)) JOLD=JOLD+1
      IF (B(JV)) AD(NJ)=VOLD
      NJ=N-JOLD
      IF (B(JK).AND.OMEGN.GT.10.) GO TO 75
```

```
      IF (B(JK)) G(NJ)=GLCPOR(A(JH),OMEGA,A(JK),A(JV),VOLD,VOLD)
75    CONTINUE
      IF (B(JK)) AD(NJ)=VOLD
      NJ=N-JOLD
      KN=K+3
      GO TO 82
76    KKK=NE-K
      IF (B(KKK)) GO TO 77
      GO TO 78
77    JJ=N-J
      A(KKK)=X(JJ)
      J=J+1
78    CALL SUB14 (IOLD,VOLD,A(KKK),INEW,VNEW)
      IF (B(KKK)) G(JJ)=IOLD*IOLD
      IF (B(KKK)) AD(JJ)=IOLD
      KN=K+1
      GO TO 82
79    KKK=NE-K
      IF (B(KKK)) GO TO 80
      GO TO 81
80    JJ=N-J
      A(KKK)=X(JJ)
      J=J+1
81    CALL SUB15 (IOLD,VOLD,A(KKK),INEW,VNEW)
      IOLD=INEW-IOLD
      IF (B(KKK)) G(JJ)=IOLD*IOLD
      IF (B(KKK)) AD(JJ)=IOLD
      KN=K+1
      GO TO 82
82    VOLD=VNEW
      IOLD=INEW
      K=KN
83    CONTINUE
      IF (OMEGN.GT.20.) GO TO 87
      IF (OMEGN.GT.10.) GO TO 93
      RHO=1.-2.*INEW/(VNEW+INEW)
      CONRHO=CONJG(RHO)
      APP=CABS(RHO)
      DO 84 L=1,N
      GRAD(L)=REAL((CONRHO/APP)*2.*G(L)/((VNEW+INEW)**2))
84    CONTINUE
      IF (NC.GT.0.OR.ND.GT.0) GO TO 85
      RETURN
85    IF (KVR.EQ.0) RETURN
      NNN=N+1
      DO 86 L=NNN,N1
      GRAD(L)=0.
86    CONTINUE
      RETURN
87    CONTINUE
      PHASE=(1./(VNEW+INEW))
      IF (KGD.EQ.0) OMEGA=OMEGA+2*OMEGA
      IF (KGD.EQ.0) POLD=PHASE
      IF (KGD.GT.0) DELAY=-AIMAG((1./POLD)*((PHASE-POLD)/(2*DELO)))*(100
     10./(8.*ATAN(1.)*FM))
      KGD=KGD+1
88    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
99    CONTINUE
      APP=OLDEL
      IF (NC.GT.0.OR.ND.GT.0) GO TO 90
      RETURN
90    IF (KVR.EQ.0) GO TO 92
      NNN=N1-N
      DO 91 L=1,NNN
      LL=N+L
      AB(L)=X(LL)
91    CONTINUE
92    WCC=WC/FM
      IF (NC.GT.0) CALL SUB16 (NC,N,AB,WCC,OMEGA,GD1,GRAD,FM)
      IF (ND.GT.0) CALL SUB17 (ND,N,NC,AB,WCC,OMEGA,GD2,GRAD,FM)
      APP=OLDEL+GD1+GD2-AB(NNN)
      IF (KVR.EQ.1) GRAD(N1)=-1.0
      RETURN
93    APP=-20.*ALOG10((CABS(1./(VNEW+INEW)))*(1.+RL))
      K=1
      J=1
      VOLD=1.0
      IOLD=1.0
      DO 110 L=1,M
      NN=IC(L)
      GO TO (94,95,96,97,98,99,100,101,102,103,104,105,106,107,108), NN
94    CALL SUB1 (IOLD,VOLD,A(K),OMEGA,INEW,VNEW)
      IF (B(K)) G(J)=GC(OMEGA,AD(J),VOLD)
      IF (B(K)) J=J+1
      K=K+1
      GO TO 109
95    CALL SUB2 (IOLD,VOLD,A(K),OMEGA,INEW,VNEW)
      IOLD=INEW-IOLD
      IF (B(K)) G(J)=GL(OMEGA,AD(J),IOLD)
      IF (B(K)) J=J+1
      K=K+1
      GO TO 109
96    CALL SUB3 (IOLD,VOLD,A(K),OMEGA,INEW,VNEW)
      IF (B(K)) G(J)=-GL(OMEGA,AD(J),IOLD)
      IF (B(K)) J=J+1
      K=K+1
      GO TO 109
97    CALL SUB4 (IOLD,VOLD,A(K),OMEGA,INEW,VNEW)
      VOLD=VNEW-VOLD
      IF (B(K)) G(J)=-GC(OMEGA,AD(J),VOLD)
      IF (B(K)) J=J+1
      K=K+1
      GO TO 109
98    JV=K
      JH=K+1
      CALL SUB5 (VOLD,IOLD,A(JV),OMEGA,A(JH),VNEW,INEW)
      IF (B(JV)) G(J)=-GLLTEL(OMEGA,A(JV),ADJJ(J),AD(J),IOLD,INEW)
      IF (B(JV)) J=J+1
      IF (B(JH)) G(J)=-GLLTZO(ADJJ(J),AD(J),IOLD,INEW,A(JH))
      IF (B(JH)) J=J+1
      K=K+2
```

```
      GO TO 109
99    JV=K
      JH=K+1
      CALL SUB6 (VOLD,IOLD,A(JV),OMEGA,A(JH),VNEW,INEW)
      IOLD=INEW-IOLD
      IF (B(JV)) G(J)=GSCTEL(OMEGA,A(JV),A(JH),AD(J),IOLD)
      IF (B(JV)) J=J+1
      IF (B(JH)) G(J)=GSCTZO(A(JV),AD(J),IOLD)
      IF (B(JH)) J=J+1
      K=K+2
      GO TO 109
100   JV=K
      JH=K+1
      CALL SUB7 (VOLD,IOLD,A(JV),OMEGA,A(JH),VNEW,INEW)
      IOLD=INEW-IOLD
      IF (B(JV)) G(J)=GOCTEL(A(JH),A(JV),OMEGA,AD(J),IOLD)
      IF (B(JV)) J=J+1
      IF (B(JH)) G(J)=GOCTZO(OMEGA,A(JV),AD(J),IOLD)
      IF (B(JH)) J=J+1
      K=K+2
      GO TO 109
101   JV=K
      JH=K+1
      CALL SUB8 (VOLD,IOLD,A(JV),OMEGA,A(JH),VNEW,INEW)
      IF (B(JV)) G(J)=-GSCTEL(OMEGA,A(JV),A(JH),AD(J),IOLD)
      IF (B(JV)) J=J+1
      IF (B(JH)) G(J)=-GSCTZO(A(JV),AD(J),IOLD)
      IF (B(JH)) J=J+1
      K=K+2
      GO TO 109
102   JV=K
      JH=K+1
      CALL SUB9 (VOLD,IOLD,A(JV),OMEGA,A(JH),VNEW,INEW)
      IF (B(JV)) G(J)=-GOCTEL(A(JH),A(JV),OMEGA,AD(J),IOLD)
      IF (B(JV)) J=J+1
      IF (B(JH)) G(J)=-GOCTZO(OMEGA,A(JV),AD(J),IOLD)
      IF (B(JH)) J=J+1
      K=K+2
      GO TO 109
103   JK=K
      JV=K+1
      JH=K+2
      CALL SUB10 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      IF (B(JK)) G(J)=-GLCSOR(OMEGA,A(JH),A(JK),A(JV),AD(J),IOLD)
      IF (B(JK)) J=J+1
      IF (B(JV)) G(J)=-GLCSQ(A(JH),A(JK),A(JV),AD(J),IOLD)
      IF (B(JV)) J=J+1
      IF (B(JH)) G(J)=-GLCSXP(OMEGA,A(JK),A(JV),AD(J),IOLD)
      IF (B(JH)) J=J+1
      K=K+3
      GO TO 109
104   JK=K
      JV=K+1
      JH=K+2
      CALL SUB11 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      IOLD=INEW-IOLD
      IF (B(JK)) G(J)=GLCSOR(OMEGA,A(JH),A(JK),A(JV),AD(J),IOLD)
      IF (B(JK)) J=J+1
```

```
      IF (B(JV)) G(J)=GLCSQ(A(JH),A(JK),A(JV),AD(J),IOLD)
      IF (B(JV)) J=J+1
      IF (B(JH)) G(J)=GLCSXP(OMEGA,A(JK),A(JV),AD(J),IOLD)
      IF (B(JH)) J=J+1
      K=K+3
      GO TO 109
105   JK=K
      JV=K+1
      JH=K+2
      CALL SUB12 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      IF (B(JK)) G(J)=GLCPOR(A(JH),OMEGA,A(JK),A(JV),AD(J),VOLD)
      IF (B(JK)) J=J+1
      IF (B(JV)) G(J)=GLCPQ(A(JH),A(JV),A(JK),AD(J),VOLD)
      IF (B(JV)) J=J+1
      IF (B(JH)) G(J)=GLCPBP(A(JH),OMEGA,A(JV),A(JK),AD(J),VOLD)
      IF (B(JH)) J=J+1
      K=K+3
      GO TO 109
106   JK=K
      JV=K+1
      JH=K+2
      CALL SUB13 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      VOLD=VNEW-VOLD
      IF (B(JK)) G(J)=-GLCPOR(A(JH),OMEGA,A(JK),A(JV),AD(J),VOLD)
      IF (B(JK)) J=J+1
      IF (B(JV)) G(J)=-GLCPQ(A(JH),A(JV),A(JK),AD(J),VOLD)
      IF (B(JV)) J=J+1
      IF (B(JH)) G(J)=-GLCPBP(A(JH),OMEGA,A(JV),A(JK),AD(J),VOLD)
      IF (B(JH)) J=J+1
      K=K+3
      GO TO 109
107   CALL SUB14 (IOLD,VOLD,A(K),INEW,VNEW)
      IF (B(K)) G(J)=-IOLD*AD(J)
      IF (B(K)) J=J+1
      K=K+1
      GO TO 109
108   CALL SUB15 (IOLD,VOLD,A(K),INEW,VNEW)
      IOLD=INEW-IOLD
      IF (B(K)) G(J)=IOLD*AD(J)
      IF (B(K)) J=J+1
      K=K+1
      GO TO 109
109   IOLD=INEW
      VOLD=VNEW
110   CONTINUE
      PSIL=VNEW+INEW*RL
      CONSTN=20./ALOG(10.)
      DO 111 L=1,N
      GRAD(L)=-REAL(G(L)/PSIL)*CONSTN
111   CONTINUE
      IF (NC.GT.0.OR.ND.GT.0) GO TO 112
      RETURN
112   IF (KVR.EQ.0) RETURN
      NNN=N+1
      DO 113 L=NNN,N1
      GRAD(L)=0.
113   CONTINUE
      RETURN
```

```
 114    INT=IFIX(OMEGN-30.)
        APP=X(INT)
        DO 115 L=1,N1
        GRAD(L)=0.
 115    CONTINUE
        GRAD(INT)=1.0
        RETURN
        END
```

```
SUBROUTINE SUB1 (IOLD,VOLD,C,OMEGA,INEW,VNEW)
COMPLEX IOLD,VOLD,INEW,VNEW,Y
Y=CMPLX(0.,OMEGA*C)
VNEW=VOLD
INEW=IOLD+VOLD*Y
RETURN
END
```

```
SUBROUTINE SUB2 (IOLD,VOLD,AL,OMEGA,INEW,VNEW)
COMPLEX IOLD,VOLD,INEW,VNEW,Y
Y=CMPLX(0.,(-1./(OMEGA*AL)))
VNEW=VOLD
INEW=IOLD+VOLD*Y
RETURN
END
```

```
SUBROUTINE SUB3 (IOLD,VOLD,AL,OMEGA,INEW,VNEW)
COMPLEX IOLD,VOLD,INEW,VNEW,Z
Z=CMPLX(0.,OMEGA*AL)
VNEW=VOLD+Z*IOLD
INEW=IOLD
RETURN
END
```

```
SUBROUTINE SUB4 (IOLD,VOLD,C,OMEGA,INEW,VNEW)
COMPLEX IOLD,VOLD,INEW,VNEW,Z
Z=CMPLX(0.,-(1./(OMEGA*C)))
INEW=IOLD
VNEW=VOLD+Z*IOLD
RETURN
END
```

```
      SUBROUTINE SUB5 (VOLD,IOLD,EL,OMEGA,ZO,VNEW,INEW)                 V1  1689
      COMPLEX VOLD,IOLD,INEW,VNEW,JSINE                                 V1  1690
      THETA=EL*OMEGA*2.*ATAN(1.0)                                       V1  1691
      CT=COS(THETA)                                                     V1  1692
      JSINE=CMPLX(0.,1.)*SIN(THETA)                                     V1  1693
      VNEW=CT*VOLD+ZO*JSINE*IOLD                                        V1  1694
      INEW=JSINE*VOLD/ZO+CT*IOLD                                        V1  1695
      RETURN                                                            V1  1696
      END                                                               V1  1697
```

```
SUBROUTINE SUB6 (IOLD,VOLD,EL,OMEGA,ZO,INEW,VNEW)
COMPLEX IOLD,VOLD,INEW,VNEW
THETA=2.*ATAN(1.0)*OMEGA*EL
VNEW=VOLD
INEW=IOLD-VOLD*COS(THETA)/(ZO*SIN(THETA))*CMPLX(0.,1.)
RETURN
END
```

```
SUBROUTINE SUB7 (IOLD,VOLD,EL,OMEGA,ZO,INEW,VNEW)
COMPLEX IOLD,VOLD,INEW,VNEW
THETA=2.*ATAN(1.0)*OMEGA*EL
VNEW=VOLD
INEW=IOLD+VOLD*SIN(THETA)*CMPLX(0.,1.)/(ZO*COS(THETA))
RETURN
END
```

```
SUBROUTINE SUB8 (IOLD,VOLD,EL,OMEGA,ZO,INEW,VNEW)
COMPLEX IOLD,VOLD,INEW,VNEW
THETA=2.*ATAN(1.0)*OMEGA*EL
INEW=IOLD
VNEW=VOLD+CMPLX(0.,1.0)*ZO*IOLD*SIN(THETA)/COS(THETA)
RETURN
END
```

```
SUBROUTINE SUB9 (IOLD,VOLD,EL,OMEGA,ZO,INEW,VNEW)
COMPLEX IOLD,VOLD,INEW,VNEW
THETA=2.*ATAN(1.0)*OMEGA*EL
INEW=IOLD
VNEW=VOLD-INEW*ZO*COS(THETA)*CMPLX(0.,1.)/SIN(THETA)
RETURN
END
```

```
SUBROUTINE SUB10 (IOLD,VOLD,Q,OMEGAR,OMEGA,XP,INEW,VNEW)
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 SUB11 (IOLD,VOLD,Q,OMEGAR,OMEGA,XP,INEW,VNEW)
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 SUB12 (IOLD,VOLD,Q,OMEGAR,OMEGA,BP,INEW,VNEW)
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 SUB13 (IOLD,VOLD,Q,OMEGAR,OMEGA,BP,INEW,VNEW)
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 SUB14 (IOLD,VOLD,R,INEW,VNEW)
COMPLEX IOLD,VOLD,INEW,VNEW,Z
Z=CMPLX(R,0.)
VNEW=VOLD+IOLD*Z
INEW=IOLD
RETURN
END
```

```
      SUBROUTINE SUB15 (IOLD,VOLD,R,INEW,VNEW)
      COMPLEX IOLD,VOLD,INEW,VNEW,Y
      Y=CMPLX(1./R,0.)
      VNEW=VOLD
      INEW=IOLD+VOLD*Y
      RETURN
      END
```

```
      SUBROUTINE SUB16 (K,N,A,WC,OMEGA,GD,GRAD,FM)
      DIMENSION A(1), GRAD(1)
      PIE=4.0*ATAN(1.)
      FC=(PIE/2.)*SQRT((OMEGA*OMEGA-WC*WC)/(1.-WC*WC))
      CAPO=SIN(FC)/COS(FC)
      GD=0
      DO 1 I=1,K
      J=I
      S=OMEGA/SQRT((OMEGA**2-WC**2)*(1.-WC*WC))
      QUA1=(A(J)**2+CAPO*CAPO)**2
      QUA2=(1.+CAPO*CAPO)/2.
      GD=GD+(A(J)/SQRT(QUA1))*QUA2*S/(FM*0.001)
      GRAD(J)=-S*QUA2*((A(J)**2-CAPO*CAPO)/QUA1)/(FM*0.001)
1     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SUB17 (M,N,K,A,WC,OMEGA,GD,GRAD,FM)
      DIMENSION A(1), GRAD(1)
      CONST=CMPLX(0.,1.)
      PIF=4.*ATAN(1.)
      FC=(PIF/2.0)*SQRT((OMEGA*OMEGA-WC*WC)/(1.-WC*WC))
      S=OMEGA/SQRT((OMEGA**2-WC**2)*(1.-WC*WC))
      CAPO=SIN(FC)/COS(FC)
      GD=0
      DO 1 I=1,M
      J=I+K
      JJ=J+M
      SAM=SQRT(A(J)**2+A(JJ)**2)
      GD=GD+A(J)*(1.+CAPO*CAPO)*((CAPO**2+SAM**2)/(CAPO**4+SAM**4+2.*CAP
     1O*CAPO*(A(J)**2-A(JJ)**2)))
      GD=GD*1000./FM
    1 CONTINUE
      GD=S*GD
      DO 2 I=1,M
      J=I+K
      JJ=J+M
      DEN=CAPO**4+2.*CAPO*CAPO*(A(J)**2-A(JJ)**2)+(A(J)**2+A(JJ)**2)**2
      DEN1=CAPO*CAPO+3.*A(J)*A(J)+A(JJ)*A(JJ)
      AK=1.+CAPO*CAPO
      DEN2=CAPO*CAPO*A(J)+A(J)**3+A(JJ)*A(JJ)*A(J)
      DEN3=4.*A(J)*(CAPO*CAPO+A(J)*A(J)+A(JJ)*A(JJ))
      GRAD(J)=S*(DEN*AK*DEN1-AK*DEN2*DEN3)/(DEN**2)
      DEN4=4.*A(JJ)*(A(JJ)**2+A(J)**2-CAPO*CAPO)
      GRAD(JJ)=S*(DEN*AK*2.*A(J)*A(JJ)-AK*DEN2*DEN4)/(DEN**2)
      GRAD(J)=GRAD(J)/(FM*0.001)
      GRAD(JJ)=GRAD(JJ)/(FM*0.001)
    2 CONTINUE
      RETURN
      END
```

ΤΑ ΠΑΝΤΑ ΕΝ ΧΡΙΣΤΩΙ ΣΥΝΕΣΤΗΚΕΝ