

**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**

CHI INTERNAL REPORT SERIES

No. CHI-6

Network Optimization Program

J. W. Hamblin and J. P. Lippitt

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 Seviara [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.

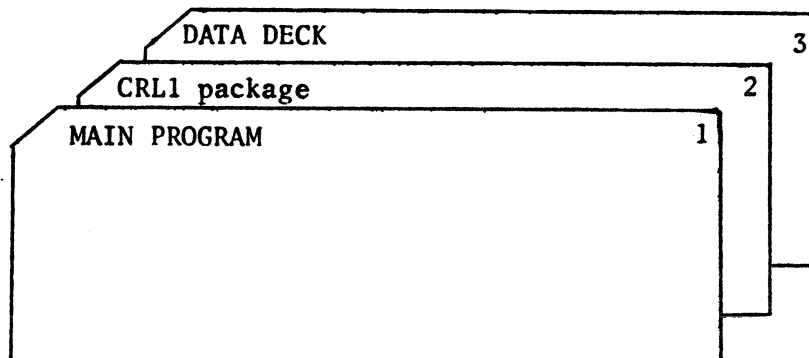
Gradients are automatically checked before optimization. Responses before and after optimization are printed out. Much other useful information which can be used to check on the progress of the optimization process and to diagnose errors is printed out at the user's discretion.

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, 1fn\*, CRL1, ID=GSGBNDLR , PW=CAD.
4. LOAD, 1fn\*.
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.

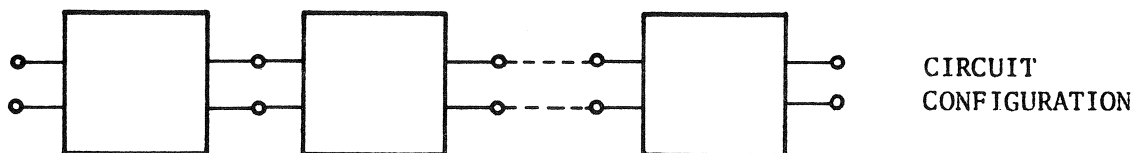
MET	Optimization method to be used. Set to 1 for Fletcher method and to 2 for Fletcher-Powell method.
MAX	The maximum number of allowable iterations (e.g., 100).
IPRINT	Intermediate output is printed out after every specified number of iterations. Set to 0 if no intermediate output is desired.
EPS(I), 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



### 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.

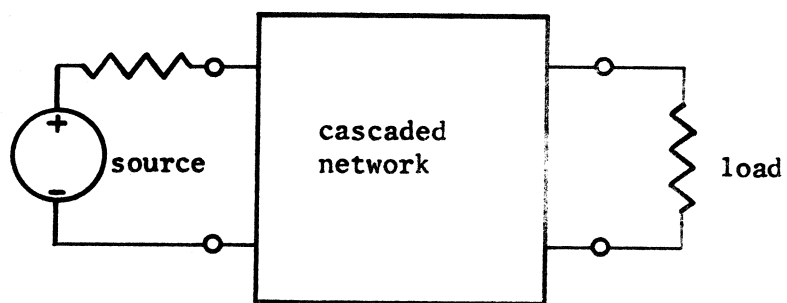
\*Except allpass networks.

### 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.





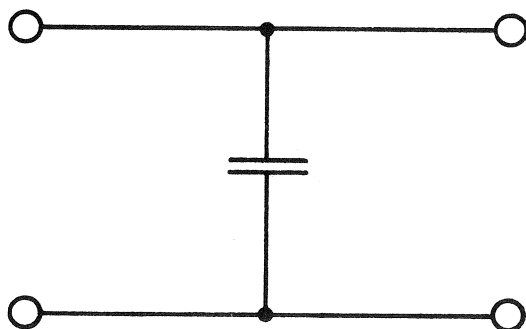
SOURCE  
AND  
LOAD  
CONFIGURATION

### 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

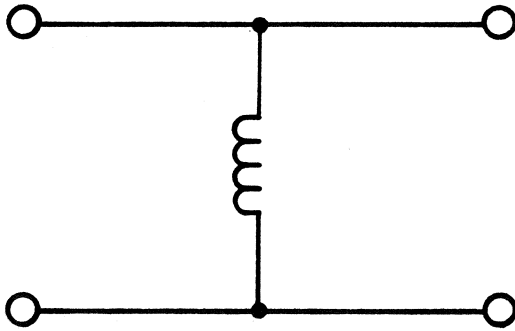
C

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

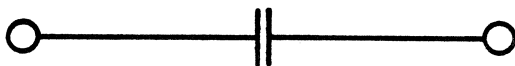
L

Parameter Definition

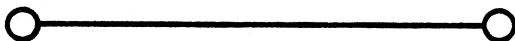
L = inductance (normalized)

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

<u>1</u>	<u>2</u>	<u>3</u>
$\ell$	$Z_0$	

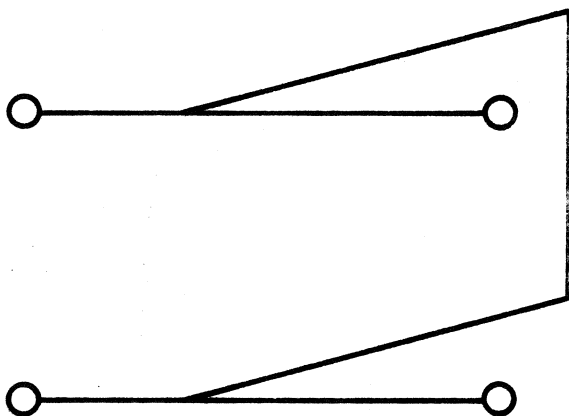
Parameter Definition

$\ell$  = 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

$l$     $Z_0$

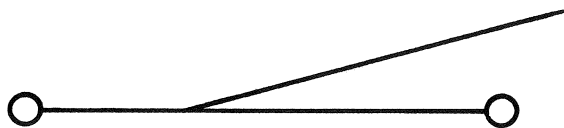
Parameter Definition

$l$  = length (normalized)

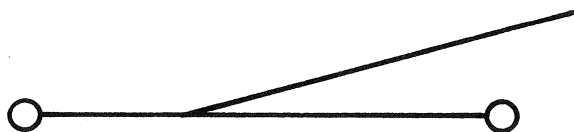
$Z_0$  = characteristic impedance (normalized)

Comments

Upper and lower bounds or fixed values can be accommodated.



SHUNT  
OPEN  
LOSSLESS  
TRANSMISSION  
LINE



Code

7

Parameters

1   2   3

$\ell$     $Z_0$

Parameter Definition

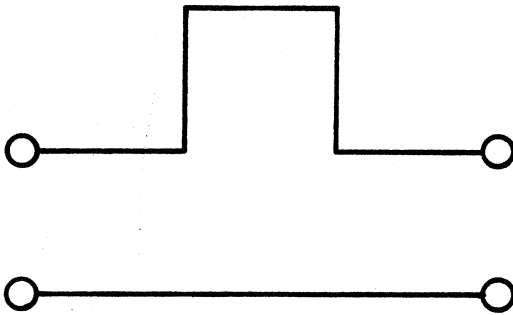
$\ell$  = length (normalized)

$Z_0$  = 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

$\ell$   $Z_0$

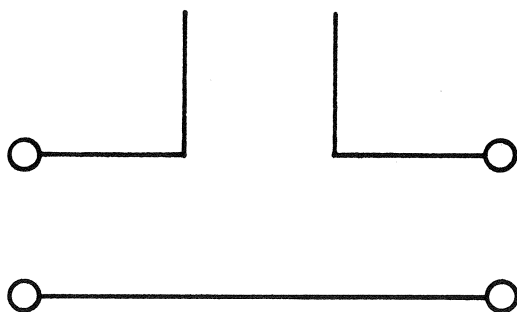
Parameter Definition

$\ell$  = 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

<u>1</u>	<u>2</u>	<u>3</u>
$\ell$	$Z_0$	

Parameter Definition

$\ell$  = length (normalized)

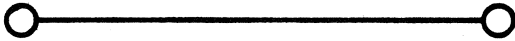
$Z_0$  = characteristic impedance (normalized)

Comments

Upper and lower bounds or fixed values can be accommodated.



SERIES  
RESONANT  
CIRCUIT



Code

10

Parameters

<u>1</u>	<u>2</u>	<u>3</u>
$\omega_R$	Q	$X'$

Parameter Definition

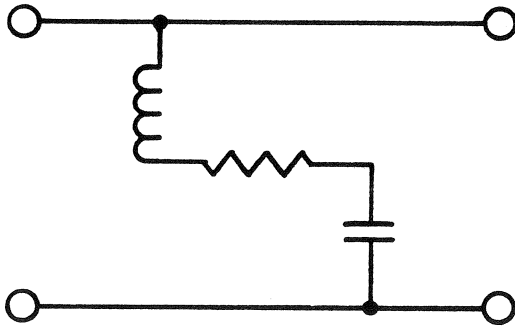
$\omega_R$  = resonant frequency (normalized)

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

11

Parameters

<u>1</u>	<u>2</u>	<u>3</u>
$\omega_R$	Q	$X'$

Parameter Definition

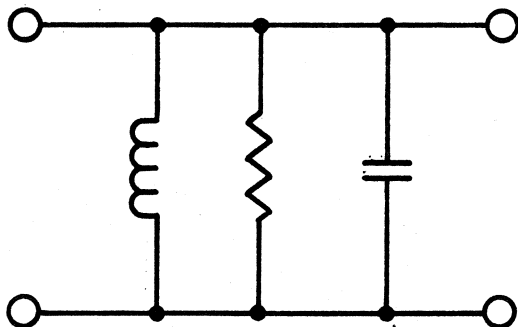
$\omega_R$  = resonant frequency (normalized)

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

<u>1</u>	<u>2</u>	<u>3</u>
$\omega_R$	Q	B'

Parameter Definition

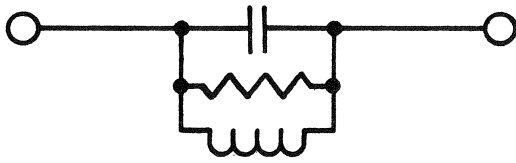
$\omega_R$  = antiresonant frequency (normalized)

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

13

Parameters

<u>1</u>	<u>2</u>	<u>3</u>
$\omega_R$	Q	B'

Parameter Definition

$\omega_R$  = antiresonant frequency (normalized)

Q = quality factor

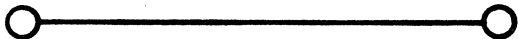
B' = slope of susceptance at antiresonance (normalized)

Comments

Upper and lower bounds or fixed values can be accommodated.



SERIES  
RESISTOR



Code

14

Parameters

1 2 3

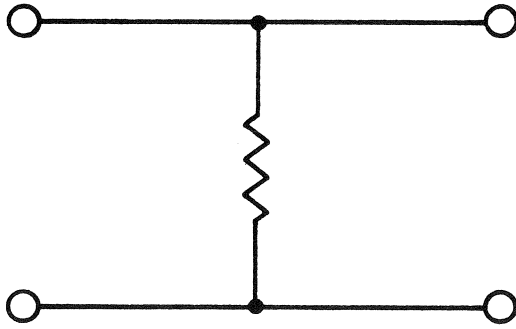
R

Parameter Definition

R = resistance (normalized)

Comments

Upper and lower bounds or fixed values can be accommodated.

SHUNT  
RESISTORCode

15

Parameters1   2   3

R

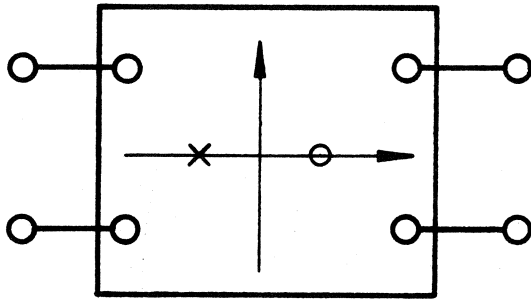
Parameter Definition

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

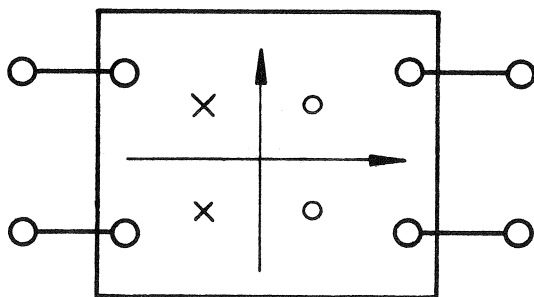
$\underline{1}$     $\underline{2}$     $\underline{3}$  ...  $\underline{n_c}$   
 $\sigma_1$     $\sigma_2$     $\sigma_3$  ...  $\sigma_{n_c}$

Parameter Definition

$\sigma_i$  = location of  $i$ th 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  $n_d$ )

Code

17 (not used)

Parameters

1   2   3   ...    $n_{d+1}$     $n_{d+2}$     $n_{d+3}$    ...

$\sigma_1$     $\sigma_2$     $\sigma_3$  ...  $\omega_1$     $\omega_2$     $\omega_3$    ...

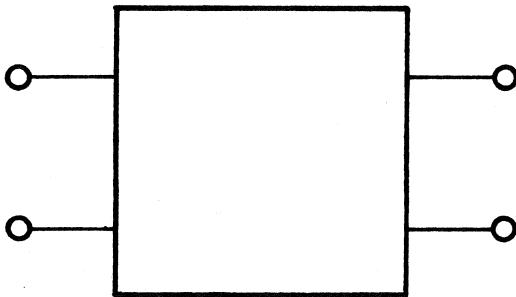
Parameter Definition

$\sigma_i$  = location of real part of  $i$ th zero

$\omega_i$  = location of imaginary part of  $i$ th zero

Comments

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 Seviara [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	1		
3								
1.F	-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.F	-4	1.E	-4	1.E
1.E	-4							-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
1.80000000008000E-01	9.387443337173E-03
2.70000000007000E-01	4.228551917297E-02
3.60000000006000E-01	1.123638535239E-01
4.50000000005000E-01	2.161346511038E-01
5.40000000004000E-01	3.272044052784E-01
6.30000000003000E-01	4.073623517897E-01
7.20000000002000E-01	3.97366128550897E-01
8.10000000001000E-01	2.935268963817E-01
9.00000000000000E-01	1.216171278175E-01

FREQUENCY	INSERTION LOSS
1.75000000000000E+00	1.659962268589E-01

FREQUENCY	INSERTION LOSS
2.50000000000000E+00	3.453938101504E+01

GRADIENTS CHECKING

GRADIENTS HAVE BEEN CHECKED AT THE FOLLOWING POINT

- X( 1) = 1.000000000E+00
- X( 2) = 1.000000000E+00
- X( 3) = 1.000000000E+00
- X( 4) = 1.000000000E+00
- X( 5) = 1.000000000E+00
- X( 6) = 1.000000000E+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
-5.86685597E+00	-1.09038185E+01	1.51473126E-01
-4.75456254E-01	-5.87271772E+00	9.98132450E-02
-4.75456254E-01	-4.75892991E-01	9.17721474E-02

GRADIENTS ARE 0. 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= 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 1.00000000E+00 1.00000000E+00 1.00000000E+00 1.00000000E+00	-4.75456254E-01 -5.86682597E+00 -1.08873021E+01 -1.08873021E+01 -5.86682597E+00
20	33	3.95900000E+00	3.02553915E-01	1.02885986E+00 1.63822938E+00 1.91392260E+00 1.90089580E+00 1.63324024E+00 1.03804420E+00	6.15166812E-01 -5.67250488E-01 -1.19312372E+00 -1.39171271E+00 -7.79015170E-01 7.13553906E-01

ITERATION CRITERION 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) = 1.65532847E+00  
 X( 6) = 1.01333880E+00

EXECUTION TIME IN SECONDS = 5.80900

Fig. 2. [cont.]



FINAL RESPONSE OF THE CIRCUIT

FREQUENCY	INSERTION LOSS
9.993757377616E-11	0.458616179835E-04
9.000000009001E-02	7.101371031245E-03
1.8000000000800E-01	2.470787263300E-02
2.7000000000700E-01	4.271376270489E-02
3.6000000000600E-01	3.913906022018E-02
4.5000000000500E-01	1.206470209338E-02
5.4000000000400E-01	2.087945390354E-03
6.3000000000300E-01	4.011430930891E-02
7.2000000000200E-01	4.18772189332E-02
8.1000000000100E-01	4.154605395557E-02
9.0000000000000E-01	

FREQUENCY	INSERTION LOSS
1.750000000000E+00	3.982426353284E+01
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 minimization", 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. Seviara, "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



3	CONTINUE	V1	60
	K=0	V1	61
	WRITE (6,49)	V1	62
	WRITE (6,50)	V1	63
	IF (IREAD.EQ.0) IREAD=2	V1	64
	GO TO (7,4), IREAD	V1	65
4	DO 6 J=1,NINT	V1	66
	IINT=J	V1	67
	IF (XX(1,J).LE.10.) WRITE (6,46)	V1	68
	IF (XX(1,J).LE.20.0.AND.XX(1,J).GT.10.) WRITE (6,47)	V1	69
	IF (XX(1,J).LE.30.0.AND.XX(1,J).GT.20.) WRITE (6,48)	V1	70
	L=NUMB(J)+1	V1	71
	IF (NUMB(J).EQ.0) Z=XX(1,J)	V1	72
	DO 6 I=1,L	V1	73
	IF (NUMB(J).GT.0) Z=XX(1,J)+(XX(2,J)-XX(1,J))*(I-1)/NUMB(J)	V1	74
	FRR=FRR(Z)	V1	75
	K=K+1	V1	76
	FRROR(K)=FRR(Z)	V1	77
	FHFLP(K)=FRROR(K)*XX(3,J)	V1	78
	X(K)=Z	V1	79
	FRT=(FRR/WT(IINT))+FUN(IINT)	V1	80
	IF (IORJ(J).EQ.1) FX(K)=Z*FM	V1	81
	IF (IORJ(J).EQ.2) FX(K)=(Z-10.)*FM	V1	82
	IF (IORJ(J).EQ.3) FX(K)=(Z-20.)*FM	V1	83
	IF (IORJ(J).EQ.0) GO TO 5	V1	84
	WRITE (6,45) FX(K),FRT	V1	85
5	CONTINUE	V1	86
6	AP(K)=APP	V1	87
	GO TO 11	V1	88
7	DO 10 J=1,NINT	V1	89
	IINT=J	V1	90
	IF (XX(1,J).LE.10.) WRITE (6,46)	V1	91
	IF (XX(1,J).LE.20.0.AND.XX(1,J).GT.10.) WRITE (6,47)	V1	92
	IF (XX(1,J).LE.30.0.AND.XX(1,J).GT.20.) WRITE (6,48)	V1	93
	K=K+1	V1	94
	KL=K+NUMB(J)	V1	95
C		V1	96
	DO 9 I=K,KL	V1	97
	FRROR(I)=FRR(X(I))	V1	98
	L=I-K+1	V1	99
	FRT=(FRROR(I)/WT(IINT))+FUN(IINT)	V1	100
	Z=X(I)	V1	101
	IF (IORJ(J).EQ.1) FX(I)=Z*FM	V1	102
	IF (IORJ(J).EQ.2) FX(I)=(Z-10.)*FM	V1	103
	IF (IORJ(J).EQ.3) FX(I)=(Z-20.)*FM	V1	104
	IF (IORJ(J).EQ.0) GO TO 8	V1	105
	WRITE (6,45) FX(I),FRT	V1	106
8	CONTINUE	V1	107
	FHFLP(I)=FRROR(I)*XX(3,J)	V1	108
	AP(I)=APP	V1	109
9	CONTINUE	V1	110
	K=KL	V1	111
10	CONTINUE	V1	112
11	EMAX=FHFLP(1)	V1	113
	DO 12 M=2,K	V1	114
	EMAX=AMAX1(EMAX,FHFLP(M))	V1	115
12	CONTINUE	V1	116
C	WRITE(6,6000)	V1	117

C	WRITE(6,7000)	V1	118
C	WRITE(6,5000)EMAX	V1	119
	CALL ERRO (FCT,W,A,N1,K,GRAD,APP,PSI,2,NUMB,XX,X,X1,ERROR,EHELP,APV1	V1	120
	1,EMAX,N,INUMB,NINT,IP)	V1	121
C	WRITE(6,4000)	V1	122
C	WRITE(6,4100)	V1	123
C	WRITE(6,8000)(J,X1(J),ERROR(J),J=1,N)	V1	124
	IGRDCH=1	V1	125
C		V1	126
C	OPTIMIZATION	V1	127
C		V1	128
	DO 40 K=1,IOPT	V1	129
	KR=1	V1	130
	IF (K-1) 14,14,13	V1	131
13	CONTINUE	V1	132
14	DO 40 KK=1,ITER	V1	133
	IP=IPA(KK)	V1	134
	IF (KR.EQ.0) GO TO 15	V1	135
C		V1	136
C	DATA FOR THE OPTIMALITY	V1	137
C	CRITERIA FOR THE OPTIMIZATION METHODS	V1	138
C		V1	139
	M=2	V1	140
C		V1	141
15	DO 16 I=1,N1	V1	142
	A(I)=XSTRT(I)	V1	143
16	CONTINUE	V1	144
	IF (IGRDCH.GT.1) GO TO 17	V1	145
	CALL GRDCHK (N1,A,G,PY,Y,GRAD,APP,PSI,NUMB,XX,X,X1,ERROR,EHELP,AP,	V1	146
	1EMAX,N,INUMB,NINT,IP)	V1	147
17	IF (KR.EQ.0) GO TO 18	V1	148
	CALL INPUT (MET,M,MAX,N1,IPRINT,1,EPS1,EST,EPS,XSTRT)	V1	149
18	IF (MET.EQ.0) MET=4	V1	150
	INDEX=0	V1	151
	GO TO (19,24,33,29), MET	V1	152
19	IF (IPRINT.EQ.0) GO TO 20	V1	153
	CALL WRITE1 (1)	V1	154
20	CALL SECOND (T1)	V1	155
	IF (KR.NE.0) GO TO 22	V1	156
	DO 21 I=1,N1	V1	157
	A(I)=DUM1(I)	V1	158
21	CONTINUE	V1	159
22	CALL VMM01 (N1,A,F,G,H,UNITH,EST,EPS,MAX,IPRINT,IEXIT,GRAD,APP,PSI	V1	160
	1,NUMB,XX,X,X1,ERROR,EHELP,AP,EMAX,N,INUMB,NINT,IP)	V1	161
	DO 23 I=1,N1	V1	162
	DUM1(I)=A(I)	V1	163
23	CONTINUE	V1	164
	CALL SECOND (T2)	V1	165
	CALL FINAL (A,F,N1)	V1	166
	T=T2-T1	V1	167
	WRITE (6,41) T	V1	168
	GO TO 29	V1	169
24	IF (IPRINT.EQ.0) GO TO 25	V1	170
	CALL WRITE1 (2)	V1	171
25	CALL SECOND (T1)	V1	172
	IF (KR.NE.0) GO TO 27	V1	173
	DO 26 I=1,N1	V1	174
	A(I)=DUM2(I)	V1	175

26	CONTINUE	V1	176
27	CALL FMFP (FUNCT,N1,A,F,G,EST,EPS1,MAX,IFR,H,IPRINT,GRAD,APP,PSI,NV1	V1	177
	NUMB,XX,X,X1,ERROR,EHELP,AP,EMAX,N,INUMB,NINT,IP)	V1	178
	DO 28 I=1,N1	V1	179
	DUM2(I)=A(I)	V1	180
28	CONTINUE	V1	181
	CALL SECOND (T2)	V1	182
	CALL FINAL (A,F,N1)	V1	183
	T=T2-T1	V1	184
	WRITE (6,41) T	V1	185
29	INDEX=INDEX+1	V1	186
	IF (M.EQ.1) GO TO 30	V1	187
	GO TO 32	V1	188
30	DO 31 I=1,N1	V1	189
	A(I)=XSTRT(I)	V1	190
31	CONTINUE	V1	191
32	CONTINUE	V1	192
C		V1	193
33	KR=0	V1	194
	WRITE (6,44) IP	V1	195
	IF (KK-1) 36,34,34	V1	196
34	IF (KK-ITER) 35,36,36	V1	197
35	CONTINUE	V1	198
36	CONTINUE	V1	199
	KN=0	V1	200
	KQ=0	V1	201
	WRITE (6,49)	V1	202
	WRITE (6,51)	V1	203
	DO 39 J=1,NINT	V1	204
	IINT=J	V1	205
	IF (XX(1,J).LF.10.) WRITE (6,46)	V1	206
	IF (XX(1,J).LF.20.0.AND.XX(1,J).GT.10.) WRITE (6,47)	V1	207
	IF (XX(1,J).LF.30.0.AND.XX(1,J).GT.20.) WRITE (6,48)	V1	208
	KQ=KQ+1	V1	209
	KL=KQ+NUMB(J)	V1	210
	DO 38 I=KQ,KL	V1	211
	L=I-KQ+1	V1	212
	ER=ERR(X(I))	V1	213
	FRT=(ER/WT(IINT))+FUN(IINT)	V1	214
	IF (IORJ(J).EQ.0) GO TO 37	V1	215
	WRITE (6,45) FX(I),FRT	V1	216
37	CONTINUE	V1	217
	KN=KN+1	V1	218
38	CONTINUE	V1	219
	KQ=KL	V1	220
39	CONTINUE	V1	221
C	WRITE(6,4000)	V1	222
C	WRITE(6,4100)	V1	223
C	WRITE(6,8000)(J,X1(J),FRROR(J),J=1,N)	V1	224
C	WRITE(6,6000)	V1	225
C	WRITE(6,7000)	V1	226
C	WRITE(6,5000)EMAX	V1	227
	IGRDCH=IGRDCH+1	V1	228
40	CONTINUE	V1	229
	RETURN	V1	230
C		V1	231
41	FORMAT (1H0, //25X,*EXECUTION TIME IN SECONDS=*,F10.5)	V1	232
42	FORMAT (8I10)	V1	233



43	FORMAT (5E16.8)	V1	234
44	FORMAT (1H1,19X,*P =*,I7/////)	V1	235
45	FORMAT (12X,2E23.12)	V1	236
46	FORMAT (/,20X,*FREQUENCY*,14X,*REFLECTION COEFF.*)	V1	237
47	FORMAT (/,20X,*FREQUENCY*,14X,*INSERTION LOSS*)	V1	238
48	FORMAT (/,20X,*FREQUENCY*,14X,*GROUP DELAY*)	V1	239
49	FORMAT (20(/))	V1	240
50	FORMAT (//23X,*RESPONSE AT THE STARTING POINT*,/23X,31(*-*))	V1	241
51	FORMAT (//,23X,*FINAL RESPONSE OF THE CIRCUIT*,/23X,30(*-*))	V1	242
	END	V1	243

```
FUNCTION FCT (Z,FUNCS,W,IINT,PSI,XX)          V1  244
C                                             V1  245
C           FUNCTION SUBROUTINE WHICH DEFINES  V1  246
C           MODIFIED UPPER AND LOWER         V1  247
C           SPECIFIED FUNCTION               V1  248
C                                             V1  249
EXTERNAL FUNCS,W                             V1  250
DIMENSION XX(3,1)                            V1  251
FCT=FUNCS(Z,IINT)+PSI*XX(3,IINT)/W(Z,IINT)  V1  252
RETURN                                       V1  253
END                                           V1  254
```

C  
C  
C  
C  
C  
C  
FUNCTION FUNCS (FN,NINT)

V1 255

FUNCTION SUBROUTINE WHICH DEFINES  
UPPER AND LOWER  
SPECIFIED FUNCTION

V1 256

V1 257

V1 258

V1 259

V1 260

V1 261

COMMON /SP/ FUN(50)

V1 262

FUNCS=FUN(NINT)

V1 263

RETURN

V1 264

END

V1 265

```
FUNCTION W (X,IINT)
COMMON /SP1/ WT(50)
```

```
C
C     FUNCTION SUBROUTINE WHICH DEFINES
C     UPPER AND LOWER
C     WEIGHTING FUNCTION
```

```
W=WT(IINT)
RETURN
END
```

```
V1 266
V1 267
V1 268
V1 269
V1 270
V1 271
V1 272
V1 273
V1 274
V1 275
```

	FUNCTION EPSNP (Z,IINT,FCT,W,A,N1,GRAD,APP,PSI,XX,IPOINT)	V1	276
C		V1	277
C	FUNCTION SUBROUTINE WHICH CALCULATES	V1	278
C	UPPER AND LOWER WEIGHTED ERROR FUNCTION	V1	279
C		V1	280
	EXTERNAL FUNCS,W,FCT	V1	281
	DIMENSION A(1), GRAD(1), XX(3,1)	V1	282
	IF (IPOINT) 1,2,1	V1	283
1	CONTINUE	V1	284
	CALL FCTAPP (Z,N1,A,APP,GRAD,IINT,1)	V1	285
2	CONTINUE	V1	286
	IF (PSI) 3,4,3	V1	287
3	EPSNP=(APP-FCT(Z,FUNCS,W,IINT,PSI,XX))*W(Z,IINT)	V1	288
	RETURN	V1	289
4	EPSNP=(APP-FUNCS(Z,IINT))*W(Z,IINT)	V1	290
	RETURN	V1	291
	END	V1	292

	SUBROUTINE ERRO (FCT,W,A,N1,K,GRAD,APP,PSI,INDIC,NUMB,XX,X,X1,ERROV1	293
	1R,EHELP,AP,EMAX,N,INUMB,NINT,IP)	V1 294
C		V1 295
C	SUBROUTINE WHICH SELECTS THE WEIGHTED	V1 296
C	ERROR FUNCTION OF INTEREST FOR	V1 297
C	THE OBJECTIVE FUNCTION	V1 298
C		V1 299
	EXTERNAL FUNCS,W,FCT	V1 300
	DIMENSION A(1), GRAD(1), NUMB(1), XX(3,1), X(1), X1(1), ERROR(1),	V1 301
	1EHELP(1), AP(1), INUMB(1)	V1 302
	FRR(Z)=EPSNP(Z,IINT,FCT,W,A,N1,GRAD,APP,PSI,XX,IPOINT)	V1 303
	GO TO (1,9), INDIC	V1 304
1	CONTINUE	V1 305
	IPOINT=1	V1 306
	K=0	V1 307
	KL=0	V1 308
	DO 7 J=1,NINT	V1 309
	IINT=J	V1 310
	IF (J.EQ.1) GO TO 2	V1 311
	KL=KL+L	V1 312
2	L=NUMB(J)+1	V1 313
	DO 6 I=1,L	V1 314
	K=K+1	V1 315
	IF (J.EQ.1) GO TO 5	V1 316
	DO 4 KK=1,KL	V1 317
	IF (X(K)-X(KK)) 4,3,4	V1 318
3	AP(K)=AP(KK)	V1 319
	APP=AP(K)	V1 320
	IPOINT=0	V1 321
	GO TO 5	V1 322
4	CONTINUE	V1 323
5	ERROR(K)=ERR(X(K))	V1 324
	EHELP(K)=ERROR(K)*XX(3,J)	V1 325
	IF (IPOINT.NE.0) AP(K)=APP	V1 326
	IPOINT=1	V1 327
6	CONTINUE	V1 328
7	CONTINUE	V1 329
	FMAX=EHELP(1)	V1 330
	DO 8 M=2,K	V1 331
	EMAX=AMAX1(EMAX,EHELP(M))	V1 332
8	CONTINUE	V1 333
9	CONTINUE	V1 334
	IF (FMAX) 10,11,11	V1 335
10	IP=-IABS(IP)	V1 336
	GO TO 12	V1 337
11	IP=IABS(IP)	V1 338
12	K=0	V1 339
	N=0	V1 340
	INUMB(1)=0	V1 341
	DO 16 J=1,NINT	V1 342
	IINT=J	V1 343
	L=NUMB(J)+1	V1 344
	DO 15 I=1,L	V1 345
	K=K+1	V1 346
	IF (IP) 14,13,13	V1 347
13	IF (EHELP(K)) 15,14,14	V1 348
14	N=N+1	V1 349
	X1(N)=X(K)	V1 350
	ERROR(N)=ERROR(K)	V1 351

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

```
V1 352  
V1 353  
V1 354  
V1 355  
V1 356  
V1 357
```

	SUBROUTINE FUNCT (N1,A,OBJ,G,GRAD,APP,PSI,NUMB,XX,X,X1,ERROR,EHELPV1	358
	1,AP,EMAX,N,INUMB,NINT,IP)	V1 359
C		V1 360
C	SUBROUTINE WHICH COMPUTES THE OBJECTIVE FUNCTION	V1 361
C	AND ITS GRADIENTS W.R.T. THE VARIABLE PARAMETERS	V1 362
C	IN THE LEAST P-TH SENSE	V1 363
		V1 364
	EXTERNAL FUNCS,W,FCT	V1 365
	DIMENSION A(1), GRAD(1), NUMB(1), XX(2,1), X(1), X1(1), ERROR(1),	V1 366
1	1EHELP(1), AP(1), INUMB(1), G(1)	V1 367
	OBJP=0.	V1 368
	GRADP=0.	V1 369
	DO 1 K=1,N1	V1 370
	G(K)=0.	V1 371
1	CONTINUE	V1 372
	CALL ERRO (FCT,W,A,N1,K,GRAD,APP,PSI,1,NUMB,XX,X,X1,ERROR,EHELP,APV1	373
1	1,EMAX,N,INUMB,NINT,IP)	V1 374
	DO 7 I=1,N	V1 375
	Z=X1(I)	V1 376
	DEL=ERROR(I)/EMAX	V1 377
	ORJI=DEL**IP	V1 378
	GRADI=DEL**(IP-1)	V1 379
	OBJP=OBJP+OBJI	V1 380
	DO 4 J=1,NINT	V1 381
	IF (I-INUMB(J+1)) 2,2,4	V1 382
2	IF (I-INUMB(J)) 4,4,3	V1 383
3	IINT=J	V1 384
	GO TO 5	V1 385
4	CONTINUE	V1 386
5	CONTINUE	V1 387
	APP=FHELP(I)	V1 388
	CALL FCTAPP (Z,N1,A,APP,GRAD,IINT,2)	V1 389
	DO 6 K=1,N1	V1 390
	GRAD(K)=GRADI*W(Z,IINT)*GRAD(K)	V1 391
	G(K)=G(K)+GRAD(K)	V1 392
6	CONTINUE	V1 393
7	CONTINUE	V1 394
	PR=1./IP	V1 395
	ORJ=EMAX*(OBJP**PR)	V1 396
	GRP=OBJP**(PR-1.)	V1 397
	DO 8 K=1,N1	V1 398
	G(K)=GRP*G(K)	V1 399
8	CONTINUE	V1 400
	RETURN	V1 401
	END	V1 402



```

SUBROUTINE GRDCHK (N,X,G,PY,Y,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHEV1 403
1LP,AP,EMAX,NP,INUMB,NINT,IP) V1 404
  DIMENSION X(1), G(1), PY(1), Y(1), GRAD(1), NUMB(1), XX(3,1), XP(1V1 405
1), X1(1), ERROR(1), EHELP(1), AP(1), INUMB(1) V1 406
  CALL FUNCT (N,X,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAXV1 407
1,NP,INUMB,NINT,IP) V1 408
  DO 1 I=1,N V1 409
  DELX=1.F-4*X(I) V1 410
  IF (ABS(DE LX).LT.1.E-40) DELX=1.F-20 V1 411
  X(I)=X(I)+DELX V1 412
  CALL FUNCT (N,X,FNEW,PY,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,V1 413
1EMAX,NP,INUMB,NINT,IP) V1 414
  Y(I)=(FNEW-F)/DELX V1 415
  X(I)=X(I)-DELX V1 416
1 CONTINUE V1 417
  DO 2 I=1,N V1 418
  IF (ABS(Y(I)).LT.1.E-20) Y(I)=1.E-20 V1 419
  IF (ABS(G(I)).LT.1.E-20) G(I)=1.E-20 V1 420
  PY(I)=ABS((Y(I)-G(I))/Y(I))*100. V1 421
2 CONTINUE V1 422
  WRITE (6,6) V1 423
  WRITE (6,7) V1 424
  WRITE (6,8) (I,X(I),I=1,N) V1 425
  WRITE (6,9) V1 426
  DO 3 I=1,N V1 427
  WRITE (6,10) G(I),Y(I),PY(I) V1 428
3 CONTINUE V1 429
  DO 4 I=1,N V1 430
  IF (PY(I).GT.10.) GO TO 5 V1 431
4 CONTINUE V1 432
  WRITE (6,11) V1 433
  RETURN V1 434
5 WRITE (6,12) V1 435
  CALL EXIT V1 436
C V1 437
6 FORMAT (20(/)) V1 438
7 FORMAT (1H0,5X,*GRADIENTS CHECKING*,/,6X,18(*-*),//,6X,*GRADIENTS V1 439
1HAVE BEEN CHECKED AT THE FOLLOWING POINT*/) V1 440
8 FORMAT (10X,**X(*,I2,*)=*,E16.8) V1 441
9 FORMAT (///,1H0,5X,*ANALYTICAL GRADIENTS*,5X,*NUMERICAL GRADIENTS*V1 442
1,5X,*PERCENTAGE ERROR*,/) V1 443
10 FORMAT (1H0,5X,3(E16.8,9X)) V1 444
11 FORMAT (1H0,///,6X,*GRADIENTS ARE O. K.*) V1 445
12 FORMAT (1H0,///,6X,*YOUR PROGRAM HAS BEEN TERMINATED BECAUSE GRADIEV1 446
1NTS ARE INCORRECT*,/6X,*PLEASE CHECK IT AGAIN*) V1 447
  END V1 448

```

```

SUBROUTINE INPUT (MET,M,MAX,N,IPRINT,IDATA,EPS1,EPS,XSTRT)  V1  449
DIMENSION XSTRT(1), EPS(1)  V1  450
WRITE (6,5)  V1  451
IF (MET.EQ.0) MET=4  V1  452
INDEX=0  V1  453
GO TO (1,2,4,3), MET  V1  454
1 WRITE (6,6)  V1  455
GO TO 3  V1  456
2 WRITE (6,7)  V1  457
CONTINUE  V1  458
WRITE (6,8) N  V1  459
WRITE (6,9) MAX  V1  460
WRITE (6,10) IPRINT  V1  461
WRITE (6,11) XSTRT(1)  V1  462
WRITE (6,12) (I,XSTRT(I),I=2,N)  V1  463
IF (MET.EQ.1) WRITE (6,13) EPS(1)  V1  464
IF (MET.EQ.1) WRITE (6,14) (I,EPS(I),I=2,N)  V1  465
IF (MET.EQ.2) WRITE (6,15) EPS1  V1  466
WRITE (6,16) EST  V1  467
RETURN  V1  468
C  V1  469
4 WRITE (6,17)  V1  470
CALL EXIT  V1  471
C  V1  472
5 FORMAT (20(/),1H0,*INPUT DATA*,/,1X,10(*-*),//,1X,*FOLLOWING METHOV1  473
ID HAVE BEEN CALLED*/ )  V1  474
6 FORMAT (1H0,*FLETCHER METHOD*)  V1  475
7 FORMAT (1H0,*FLETCHER-POWELL METHOD*)  V1  476
8 FORMAT (1H0,/,1X,*NUMBER OF INDEPENDENT VARIABLES*,36(*.*),*N=*,I5,V1  477
1/)  V1  478
9 FORMAT (1H0,*MAXIMUM NUMBER OF ALLOWABLE ITERATIONS*,27(*.*),*MAX=V1  479
1*,I5,/)  V1  480
10 FORMAT (1H0,*INTERMEDIATE OUTPUT TO BE PRINTED EVERY IPRINT ITERATV1  481
IONS*,5(*.*),*IPRINT=*,I5,/)  V1  482
11 FORMAT (1H0,*STARTING VALUE FOR VECTOR X(I)*,29(*.*),*XSTRT( 1)=*,V1  483
1E16.8)  V1  484
12 FORMAT (1H0,59X,*XSTRT(*,I2,*)=*,E16.8)  V1  485
13 FORMAT (1H0,/,1H0,*TEST QUANTITIES TO BE USED IN FLETCHER METHOD*,V1  486
116(*.*),*EPS( 1)=*,E16.8)  V1  487
14 FORMAT (1H0,61X,*FPS(*,I2,*)=*,E16.8)  V1  488
15 FORMAT (1H0,/,1H0,*TEST QUANTITY TO BE USED IN FLETCHER-POWELL METV1  489
1HOD*,14(*.*),*EPS1=*,E16.8)  V1  490
16 FORMAT (1H0,/,1H0,*ESTIMATE OF LOWER BOUND ON FUNCTION TO BE MINIMV1  491
1IZED*,14(*.*),*FST=*,F16.8)  V1  492
17 FORMAT (1H0,*NONE OF THE OPTIMIZATION METHODS HAVE BEEN CALLED*,/,V1  493
11X,*PLEASE CHECK THE VALUE OF MET*,/,1X,*REMAINDER*,/,1X,*MET=1  V1  494
2 FLETCHER METHOD WOULD BE CALLED*,/,1X,*MET=2  FLETCHER-POWELL MV1  495
3ETHOD WOULD BE CALLED*)  V1  496
END  V1  497

```

	SUBROUTINE WRITE1 (N)	V1	498
	WRITE (6,4)	V1	499
	GO TO (1,2), N	V1	500
1	WRITE (6,5)	V1	501
	GO TO 3	V1	502
2	WRITE (6,6)	V1	503
3	CONTINUE	V1	504
	WRITE (6,7)	V1	505
	RETURN	V1	506
C		V1	507
C		V1	508
C		V1	509
4	FORMAT (20(/))	V1	510
5	FORMAT (* OPTIMIZATION BY FLETCHER METHOD*,/,1X,31(*-*))	V1	511
6	FORMAT (* OPTIMIZATION BY FLETCHER-POWELL METHOD*,/,1H0,38(*-*))	V1	512
7	FORMAT (1H0,*ITERATION*,2X,*FUNCTION*,6X,*TIME ELAPSED*,8X,*OBJECTIVE	V1	513
	NUMBER*,13X,*VARIABLE VECTOR X(I)*,9X,*GRADIENT VECTOR G(I)*,/1H0,*NV1	V1	514
	NUMBER*,5X,*EVALUATIONS*,3X,* (SECONDS)*,11X,*FUNCTION*,/)	V1	515
	END	V1	516



	SUBROUTINE FINAL (X,F,N)	V1	527
	DIMENSION X(1)	V1	528
	COMMON /BLK/ KO	V1	529
	WRITE (6,3)	V1	530
	IF (KO.EQ.0) GO TO 1	V1	531
	WRITE (6,4)	V1	532
	GO TO 2	V1	533
1	WRITE (6,5)	V1	534
2	CONTINUE	V1	535
	WRITE (6,6) F	V1	536
	WRITE (6,7) (I,X(I),I=1,N)	V1	537
	RETURN	V1	538
C		V1	539
C		V1	540
C		V1	541
3	FORMAT (20(/))	V1	542
4	FORMAT (41X,*FOLLOWING IS THE OPTIMUM SOLUTION*,/,41X,33(*-*))	V1	543
5	FORMAT (45X,*RESULTS AT LAST ITERATION*/,45X,25(*-*))	V1	544
6	FORMAT (,/,48X,*F =*,E16.8,/) )	V1	545
7	FORMAT (,44X,*X(*,I2,* )=*,E16.8)	V1	546
	END	V1	547

	SUBROUTINE VMM01 (N,X,F,G,H,UNITH,FEST,EPS,MAXFN,IPRINT,IEXIT,GRADV1	548
	1,APP,PSI,NUMR,XX,XP,X1,ERROR,EHELP,AP,EMAX,NP,INUMB,NINT,IP)	V1 549
	DIMENSION X(1), G(1), H(1), EPS(1), GRAD(1), NUMB(1), XX(3,1), XP(V1	550
	11), X1(1), ERROR(1), EHELP(1), AP(1), INUMB(1)	V1 551
	LOGICAL CONV,UNITH	V1 552
	COMMON /BLK/ KO	V1 553
	CALL SECOND (T3)	V1 554
	KO=0	V1 555
	CALL FUNCT (N,X,F,G,GRAD,APP,PSI,NUMR,XX,XP,X1,FRROR,EHELP,AP,EMAXV1	556
	1,NP,INUMB,NINT,IP)	V1 557
	IF (F.LT.FEST) GO TO 23	V1 558
	NFNS=1	V1 559
	ITN=0	V1 560
	STEP=1.	V1 561
	IDX=N	V1 562
	IDG=N+N	V1 563
	IH=IDG+N	V1 564
	IF (.NOT.UNITH) GO TO 2	V1 565
	IJ=IH+1	V1 566
	DO 1 I=1,N	V1 567
	DO 1 J=I,N	V1 568
	H(IJ)=0.	V1 569
	IF (I.EQ.J) H(IJ)=1.0	V1 570
1	IJ=IJ+1	V1 571
2	CONV=.TRUE.	V1 572
	GDX=0.	V1 573
	DO 6 I=1,N	V1 574
	Z=0.	V1 575
	IJ=IH+I	V1 576
	IF (I.EQ.1) GO TO 4	V1 577
	II=I-1	V1 578
	DO 3 J=1,II	V1 579
	Z=Z-H(IJ)*G(J)	V1 580
	IJ=IJ+N-J	V1 581
3	CONTINUE	V1 582
4	DO 5 J=I,N	V1 583
	Z=Z-H(IJ)*G(J)	V1 584
	IJ=IJ+1	V1 585
5	CONTINUE	V1 586
	IF (ABS(Z).GT.EPS(I)) CONV=.FALSE.	V1 587
	H(IDX+I)=Z	V1 588
	GDX=GDX+G(I)*Z	V1 589
6	CONTINUE	V1 590
C		V1 591
	IF (IPRINT.EQ.0) GO TO 7	V1 592
	IF (MOD(ITN,IPRINT).NE.0) GO TO 7	V1 593
	CALL SECOND (T4)	V1 594
	TIME=T4-T3	V1 595
	CALL WRITE2 (X,N,G,F,NFNS,ITN,TIME)	V1 596
7	IEXIT=1	V1 597
	IF (CONV) GO TO 24	V1 598
	IEXIT=2	V1 599
	IF (GDX.GF.0.) GO TO 24	V1 600
	Z=1.	V1 601
	IF (ITN.LT.N.AND.UNITH) Z=STEP	V1 602
	W=2.*(FEST-F)/GDX	V1 603
	IF (W.LT.Z) Z=W	V1 604
	STEP=Z	V1 605
8	GDX=GDX*Z	V1 606

	DO 9 I=1,N	V1	607
	H(IDX+I)=H(IDX+I)*Z	V1	608
	X(I)=X(I)+H(IDX+I)	V1	609
9	CONTINUE	V1	610
	CALL FUNCT (N,X,FP,H,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAV1	V1	611
	1X,NP,INUMB,NINT,IP)	V1	612
	IF (FP.LT.FEST) GO TO 23	V1	613
	NFNS=NFNS+1	V1	614
	IEXIT=3	V1	615
	IF (ITN.EQ.MAXFN) GO TO 24	V1	616
	GPD $X=0.$	V1	617
	DO 10 I=1,N	V1	618
	H(IDG+I)=H(I)-G(I)	V1	619
	GPD $X=GPDX+H(I)*H(IDX+I)$	V1	620
10	CONTINUE	V1	621
	DGD $X=GPDX-GDX$	V1	622
	IF (F.GT.FP-.0001*GDX) GO TO 12	V1	623
	IFXIT=4	V1	624
	IF (GPD $X.LT.0.$ ..AND.ITN.GT.N) GO TO 24	V1	625
	Z=3.*(F-FP)+GPD $X+GDX$	V1	626
	W=SQRT(1.-GDX/Z*GPD $X/Z$ )*ABS(Z)	V1	627
	Z=1.-(GPD $X+W-Z$ )/(DGD $X+2.*W$ )	V1	628
	IF (Z.LT.0.1) Z=0.1	V1	629
	DO 11 I=1,N	V1	630
	X(I)=X(I)-H(IDX+I)	V1	631
11	CONTINUE	V1	632
	GO TO 14	V1	633
12	F=FP	V1	634
	DO 13 I=1,N	V1	635
	G(I)=H(I)	V1	636
13	CONTINUE	V1	637
	IF (DGD $X.GT.0.$ ) GO TO 15	V1	638
	GDX=GPD $X$	V1	639
	Z=4.	V1	640
14	STEP=Z*STEP	V1	641
	GO TO 8	V1	642
15	IF (GPD $X.LT.0.5*GDX$ ) STEP=2.*STEP	V1	643
	DGH $DG=0.$	V1	644
	DO 19 I=1,N	V1	645
	Z=0.	V1	646
	IJ=IH+I	V1	647
	IF (I.EQ.1) GO TO 17	V1	648
	II=I-1	V1	649
	DO 16 J=1,II	V1	650
	Z=Z+H(IJ)*H(IDG+J)	V1	651
	IJ=IJ+N-J	V1	652
16	CONTINUE	V1	653
17	DO 18 J=1,N	V1	654
	Z=Z+H(IJ)*H(IDG+J)	V1	655
	IJ=IJ+1	V1	656
18	CONTINUE	V1	657
	DGH $DG=DGHDG+Z*H(IDG+I)$	V1	658
	H(I)=Z	V1	659
19	CONTINUE	V1	660
	IF (DGH $DG.LT.0.0$ ) DGH $DG=DGDX*0.01$	V1	661
	IF (DGD $X.LT.DGHDG) GO TO 21$	V1	662
	W=1.0+DGH $DG/DGDX$	V1	663
	DO 20 I=1,N	V1	664

	H(IDX+I)=W*H(IDX+I)-H(I)	V1	665
20	CONTINUE	V1	666
	DGDX=DGDX+DGHDG	V1	667
	DGHDG=DGDX	V1	668
21	IJ=IH	V1	669
	DO 22 I=1,N	V1	670
	W=H(IDX+I)/DGDX	V1	671
	Z=H(I)/DGHDG	V1	672
	DO 22 J=I,N	V1	673
	IJ=IJ+1	V1	674
22	H(IJ)=H(IJ)+W*H(IDX+J)-Z*H(J)	V1	675
	ITN=ITN+1	V1	676
	GO TO 2	V1	677
23	IFXIT=5	V1	678
24	IF (IFXIT.FQ.1) KO=1	V1	679
	IF (IPRINT.FQ.0) RETURN	V1	680
	GO TO (25,26,27,26,28), IEXIT	V1	681
25	WRITE (6,30) IFXIT	V1	682
	GO TO 29	V1	683
26	WRITE (6,31) IEXIT	V1	684
	GO TO 29	V1	685
27	WRITE (6,32) IFXIT	V1	686
	GO TO 29	V1	687
28	WRITE (6,33) IEXIT	V1	688
29	CONTINUE	V1	689
	RETURN	V1	690
C		V1	691
C		V1	692
C		V1	693
20	FORMAT (/ ,1H0,*IFXIT=*,I2,*CRITERION FOR OPTIMUM HAS BEEN SATISFIED*)	V1	694
		V1	695
21	FORMAT (/ ,1H0,*IFXIT=*,I2,*EITHER OF THE FOLLOWING THINGS HAS HAPPENED*)	V1	696
	1 FNEED*,/,9X,*1. FPS CHOSEN IS TOO SMALL*,/,9X,*2. GRADIENTS ARE NOT	V1	697
	2 CORRECT*,/,9X,*3. MATRIX H GOES SINGULAR*)	V1	698
22	FORMAT (/ ,1H0,*IFXIT=*,I2,*MAXIMUM NUMBER OF ALLOWABLE ITERATION HAS	V1	699
	1 AS BEEN EXCEEDED*)	V1	700
23	FORMAT (/ ,1H0,*IEXIT=*,I2,*FUNCTION VALUE LESS THAN MINIMUM ESTIMATED	V1	701
	1 TED HAS BEEN DETECTED*)	V1	702
	END	V1	703



	SUBROUTINE FMFP (FUNCT,N,X,F,G,EST,EPS,LIMIT,IER,H,IPRINT,GRAD,APPV1	V1	704
	1,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAX,NP,INUMB,NINT,IP)	V1	705
	COMMON /RLK/ KO	V1	706
C		V1	707
C	TO FIND A LOCAL MINIMUM OF A FUNCTION OF SEVERAL VARIABLES	V1	708
C	BY THE METHOD OF FLETCHER AND POWELL	V1	709
C		V1	710
	DIMENSION H(1), X(1), G(1), GRAD(1), NUMB(1), XX(3,1), XP(1), X1(1V1	V1	711
	1), ERROR(1), EHELP(1), AP(1), INUMB(1)	V1	712
		V1	713
C	COMPUTE FUNCTION VALUE AND GRADIENT VECTOR FOR INITIAL ARGUMENTV1	V1	714
C	KO=0	V1	715
	CALL SECOND (T3)	V1	716
	CALL FUNCT (N,X,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAXV1	V1	717
	1,NP,INUMB,NINT,IP)	V1	718
	KOUNT=0	V1	719
	NUMF=1	V1	720
	CALL SECOND (T4)	V1	721
	TIMF=T4-T3	V1	722
	IF (IPRINT.EQ.0) GO TO 1	V1	723
	CALL WRITE2 (X,N,G,F,NUMF,KOUNT,TIME)	V1	724
1	CONTINUE	V1	725
C		V1	726
C	RESET ITERATION COUNTER AND GENERATE IDENTITY MATRIX	V1	727
	IER=0	V1	728
	KK=0	V1	729
	N2=N+N	V1	730
	N3=N2+N	V1	731
	N31=N3+1	V1	732
2	K=N31	V1	733
	DO 5 J=1,N	V1	734
	H(K)=1.	V1	735
	NJ=N-J	V1	736
	IF (NJ) 6,6,3	V1	737
3	DO 4 L=1,NJ	V1	738
	KL=K+L	V1	739
	H(KL)=0.	V1	740
4	CONTINUE	V1	741
	K=KL+1	V1	742
5	CONTINUE	V1	743
C		V1	744
C	START ITERATION LOOP	V1	745
6	IF (KOUNT.EQ.0) GO TO 7	V1	746
	IF (KK.NE.IPRINT) GO TO 7	V1	747
	KK=0	V1	748
	CALL SECOND (T4)	V1	749
	TIMF=T4-T3	V1	750
	CALL WRITE2 (X,N,G,F,NUMF,KOUNT,TIME)	V1	751
7	CONTINUE	V1	752
	KOUNT=KOUNT+1	V1	753
	KK=KK+1	V1	754
C		V1	755
C	SAVE FUNCTION VALUE, ARGUMENT VECTOR AND GRADIENT VECTOR	V1	756
	OLDF=F	V1	757
	DO 11 J=1,N	V1	758
	K=N+J	V1	759
	H(K)=G(J)	V1	760
	K=K+N	V1	761
	H(K)=X(J)	V1	762

C		V1	763
C	DETERMINE DIRECTION VECTOR H	V1	764
	K=J+N3	V1	765
	T=0.	V1	766
	DO 10 L=1,N	V1	767
	T=T-G(L)*H(K)	V1	768
	IF (L-J) 8,9,9	V1	769
R	K=K+N-L	V1	770
	GO TO 10	V1	771
9	K=K+1	V1	772
10	CONTINUE	V1	773
	H(J)=T	V1	774
11	CONTINUE	V1	775
C		V1	776
C	CHECK WHETHER FUNCTION WILL DECREASE STEPPING ALONG H.	V1	777
	DY=0.	V1	778
	HNRM=0.	V1	779
	GHRM=0.	V1	780
C		V1	781
C	CALCULATE DIRECTIONAL DERIVATIVE AND TESTVALUES FOR DIRECTION	V1	782
C	VECTOR H AND GRADIENT VECTOR G.	V1	783
	DO 12 J=1,N	V1	784
	HNRM=HNRM+ABS(H(J))	V1	785
	GHRM=GHRM+ABS(G(J))	V1	786
	DY=DY+H(J)*G(J)	V1	787
12	CONTINUE	V1	788
		V1	789
C		V1	790
C	REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DIRECTIONAL	V1	791
C	DERIVATIVE APPEARS TO BE POSITIVE OR ZERO.	V1	792
	IF (DY) 13,57,57	V1	793
C		V1	794
C	REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DIRECTION	V1	795
C	VECTOR H IS SMALL COMPARED TO GRADIENT VECTOR G.	V1	796
13	IF (HNRM/GHRM-EPS) 57,57,14	V1	797
		V1	798
C	SEARCH MINIMUM ALONG DIRECTION H	V1	799
		V1	800
C	SEARCH ALONG H FOR POSITIVE DIRECTIONAL DERIVATIVE	V1	801
14	FY=F	V1	802
	ALFA=2.*(EST-F)/DY	V1	803
	AMBDA=1.	V1	804
C		V1	805
C	USE ESTIMATE FOR STEPSIZE ONLY IF IT IS POSITIVE AND LESS THAN	V1	806
C	1. OTHERWISE TAKE 1. AS STEPSIZE	V1	807
	IF (ALFA) 17,17,15	V1	808
15	IF (ALFA-AMBDA) 16,17,17	V1	809
16	AMBDA=ALFA	V1	810
17	ALFA=0.	V1	811
C		V1	812
C	SAVE FUNCTION AND DERIVATIVE VALUES FOR OLD ARGUMENT	V1	813
18	FX=FY	V1	814
	DX=DY	V1	815
C		V1	816
C	STEP ARGUMENT ALONG H	V1	817
	DO 19 I=1,N	V1	818
	X(I)=X(I)+AMBDA*H(I)	V1	819
19	CONTINUE	V1	820
C			

```

C      COMPUTE FUNCTION VALUE AND GRADIENT FOR NEW ARGUMENT          V1      821
CALL FUNCT (N,X,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAXV1 822
1,NP,INUMR,NINT,IP)          V1      823
NUMF=NUMF+1                  V1      824
FY=F                          V1      825
C      COMPUTE DIRECTIONAL DERIVATIVE DY FOR NEW ARGUMENT.  TERMINATE V1      826
C      SEARCH, IF DY IS POSITIVE. IF DY IS ZERO THE MINIMUM IS FOUND V1      827
DY=0.                          V1      828
DO 20 I=1,N                    V1      829
DY=DY+G(I)*H(I)                V1      830
CONTINUE                       V1      831
IF (DY) 21,41,24                V1      832
C      TERMINATE SEARCH ALSO IF THE FUNCTION VALUE INDICATES THAT V1      833
C      A MINIMUM HAS BEEN PASSED V1      834
21 IF (FY-FX) 22,24,24          V1      835
C      REPEAT SEARCH AND DOUBLE STEPSIZE FOR FURTHER SEARCHES V1      836
C      AMBDA=AMBDA+ALFA V1      837
22 ALFA=AMBDA                   V1      838
END OF SEARCH LOOP             V1      839
C      TERMINATE IF THE CHANGE IN ARGUMENT GETS VERY LARGE V1      840
IF (HNRM*AMBDA-1.E10) 18,18,23 V1      841
C      LINEAR SEARCH TECHNIQUE INDICATES THAT NO MINIMUM EXISTS V1      842
23 IER=2                         V1      843
GO TO 62                       V1      844
C      INTERPOLATE CUBICALLY IN THE INTERVAL DEFINED BY THE SEARCH V1      845
C      ABOVE AND COMPUTE THE ARGUMENT X FOR WHICH THE INTERPOLATION V1      846
C      POLYNOMIAL IS MINIMIZED V1      847
24 T=0.                          V1      848
25 IF (AMRDA) 26,41,26          V1      849
26 Z=3.*(FX-FY)/AMRDA+DX+DY      V1      850
ALFA=AMAX1(ABS(Z),ABS(DX),ABS(DY)) V1      851
DALFA=Z/ALFA                   V1      852
DALFA=DALFA*DALFA-DX/ALFA*DY/ALFA V1      853
IF (DALFA) 57,27,27            V1      854
27 W=ALFA*SQRT(DALFA)          V1      855
ALFA=DY-DX+W+W                  V1      856
IF (ALFA) 28,29,28             V1      857
28 ALFA=(DY-Z+W)/ALFA          V1      858
GO TO 30                        V1      859
29 ALFA=(Z+DY-W)/(Z+DX+Z+DY)    V1      860
30 ALFA=ALFA*AMBDA              V1      861
DO 31 I=1,N                     V1      862
X(I)=X(I)+(T-ALFA)*H(I)        V1      863
CONTINUE                        V1      864
C      TERMINATE, IF THE VALUE OF THE ACTUAL FUNCTION AT X IS LESS V1      865
C      THAN THE FUNCTION VALUES AT THE INTERVAL ENDS. OTHERWISE REDUCE V1      866
C      THE INTERVAL BY CHOOSING ONE END-POINT EQUAL TO X AND REPEAT V1      867
C      THE INTERPOLATION. WHICH END-POINT IS CHOOSEN DEPENDS ON THE V1      868
C      VALUE OF THE FUNCTION AND ITS GRADIENT AT X V1      869
C      V1      870
C      V1      871
C      V1      872
C      V1      873
C      V1      874
C      V1      875
C      V1      876
C      V1      877
C      V1      878
NUMF=NUMF+1

```

	CALL FUNCT (N,X,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAXV1	879
	1,NP,INUMB,NINT,IP)	V1 880
	IF (F-FX) 32,32,33	V1 881
32	IF (F-FY) 41,41,33	V1 882
33	DALFA=0.	V1 883
	DO 34 I=1,N	V1 884
	DALFA=DALFA+G(I)*H(I)	V1 885
34	CONTINUE	V1 886
	IF (DALFA) 35,38,38	V1 887
35	IF (F-FX) 37,36,38	V1 888
36	IF (DX-DALFA) 37,41,37	V1 889
37	FX=F	V1 890
	DX=DALFA	V1 891
	T=ALFA	V1 892
	AMBDA=ALFA	V1 893
	GO TO 25	V1 894
38	IF (FY-F) 40,39,40	V1 895
39	IF (DY-DALFA) 40,41,40	V1 896
40	FY=F	V1 897
	DY=DALFA	V1 898
	AMBDA=AMBDA-ALFA	V1 899
	GO TO 24	V1 900
C		V1 901
C	TERMINATE, IF FUNCTION HAS NOT DECREASED DURING LAST ITERATION	V1 902
41	IF (OLDF-F+EPS) 57,42,42	V1 903
C		V1 904
C	COMPUTE DIFFERENCE VECTORS OF ARGUMENT AND GRADIENT FROM	V1 905
C	TWO CONSECUTIVE ITERATIONS	V1 906
42	DO 43 J=1,N	V1 907
	K=N+J	V1 908
	H(K)=G(J)-H(K)	V1 909
	K=N+K	V1 910
C	IF AT LEAST N ITERATIONS HAVE BEEN EXECUTED. TERMINATE, IF	V1 911
	H(K)=X(J)-H(K)	V1 912
43	CONTINUE	V1 913
C		V1 914
C	TEST LENGTH OF ARGUMENT DIFFERENCE VECTOR AND DIRECTION VECTOR	V1 915
C	BOTH ARE LESS THAN EPS	V1 916
	IFR=0	V1 917
	IF (KOUNT-N) 47,44,44	V1 918
44	T=0.	V1 919
	Z=0.	V1 920
	DO 45 J=1,N	V1 921
	K=N+J	V1 922
	W=H(K)	V1 923
	K=K+N	V1 924
	T=T+ABS(H(K))	V1 925
	Z=Z+W*H(K)	V1 926
45	CONTINUE	V1 927
	IF (HNRM-EPS) 46,46,47	V1 928
46	IF (T-EPS) 62,62,47	V1 929
C		V1 930
C	TERMINATE, IF NUMBER OF ITERATIONS WOULD EXCEED LIMIT	V1 931
47	IF (KOUNT-LIMIT) 48,55,55	V1 932
C		V1 933
C	PREPARE UPDATING OF MATRIX H	V1 934
48	ALFA=0.	V1 935
	DO 52 J=1,N	V1 936

	K=J+N3	V1	937
	W=0.	V1	938
	DO 51 L=1,N	V1	939
	KL=N+L	V1	940
	W=W+H(KL)*H(K)	V1	941
	IF (L-J) 49,50,50	V1	942
49	K=K+N-L	V1	943
	GO TO 51	V1	944
50	K=K+1	V1	945
51	CONTINUE	V1	946
	K=N+J	V1	947
	ALFA=ALFA+W*H(K)	V1	948
	H(J)=W	V1	949
52	CONTINUE	V1	950
C		V1	951
C	REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF RESULTS	V1	952
C	ARE NOT SATISFACTORY	V1	953
C	IF (Z*ALFA) 53,2,53	V1	954
C		V1	955
C	UPDATE MATRIX H	V1	956
53	K=N31	V1	957
	DO 54 L=1,N	V1	958
	KL=N2+L	V1	959
	DO 54 J=L,N	V1	960
	NJ=N2+J	V1	961
	H(K)=H(K)+H(KL)*H(NJ)/Z-H(L)*H(J)/ALFA	V1	962
54	K=K+1	V1	963
	GO TO 6	V1	964
C	END OF ITERATION LOOP	V1	965
C		V1	966
C	NO CONVERGENCE AFTER LIMIT ITERATIONS	V1	967
55	IER=1	V1	968
	IF (KK.NE.IPRINT) GO TO 56	V1	969
	CALL WRITE2 (X,N,G,F,NUMF,KOUNT)	V1	970
56	CONTINUE	V1	971
	GO TO 62	V1	972
C		V1	973
C	RESTORE OLD VALUES OF FUNCTION AND ARGUMENTS	V1	974
57	DO 58 J=1,N	V1	975
	K=N2+J	V1	976
	X(J)=H(K)	V1	977
58	CONTINUE	V1	978
	CALL FUNCT (N,X,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAXV1	V1	979
	1,NP,INUMB,NINT,IP)	V1	980
	NUMF=NUMF+1	V1	981
C		V1	982
C	REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DERIVATIVE	V1	983
C	FAILS TO BE SUFFICIENTLY SMALL	V1	984
C	IF (GNRM-EPS) 61,61,59	V1	985
C		V1	986
C	TEST FOR REPEATED FAILURE OF ITERATION	V1	987
59	IF (IER) 62,60,60	V1	988
60	IFR=-1	V1	989
	GO TO 2	V1	990
61	IER=0	V1	991
62	II=IFR+2	V1	992
	IF (II.EQ.2) KO=1	V1	993
	IF (IPRINT.EQ.0) RETURN	V1	994

	GO TO (63,64,65,66), II	V1	995
63	WRITE (6,68) IER	V1	996
	GO TO 67	V1	997
64	WRITE (6,69) IER	V1	998
	GO TO 67	V1	999
65	WRITE (6,70) IER	V1	1000
	GO TO 67	V1	1001
66	WRITE (6,71) IFR	V1	1002
67	RETURN	V1	1003
C		V1	1004
C		V1	1005
C		V1	1006
68	FORMAT (1H0,*IER=*,I2,* ERROR IN GRADIENTS CALCULATIONS*)	V1	1007
69	FORMAT (1H0,*IER=*,I2,* CRITERION FOR OPTIMUM HAS BEEN SATISFIED*)	V1	1008
70	FORMAT (1H0,*IER=*,I2,* MAXIMUM NUMBER OF ALLOWABLE ITERATIONS HAS	V1	1009
	1 BEEN EXCEEDED*)	V1	1010
71	FORMAT (1H0,*IFR=*,I2,* CHANGE IN ARGUMENTS GETS TOO LARGE, LINEAR	V1	1011
	1 SEARCH INDICATES THAT NO MINIMUM EXISTS*)	V1	1012
	END	V1	1013

```

SUBROUTINE FCTAPP (OMEGN,N1,X,APP,GRAD,IINT,INDIC)          V1 1014
DIMENSION AD(50),ADJJ(50),G(50),X(1),GRAD(1)             V1 1015
COMPLEX AD,ADJJ,CONRHO,RHO,I,IHAT,V,VHAT,INFW,VNEW,IOLD,VOLD,G,V1,V1 1016
1V2,IHAT1,IHAT2,PSIL,RL,GL,GC,GLLTFL,GLLTZO,GOMC,GOML,GSCTEL,GSCTOMV1 1017
2,GSCTZO,GOCTFL,GOCTZO,GOCTOM,GLCPBP,GLCPQ,GLCPOM,GLCPOR,GLCSQ,GLCSV1 1018
3OM,GLCSOR,GLCSXP                                         V1 1019
COMPLEX POLD,PHASE                                         V1 1020
COMMON /BLACK/ IC(100),A(100),B(100),M,NE,RL,NC,ND,KVR,AB(100),FM,V1 1021
1WC                                                         V1 1022
LOGICAL B                                                  V1 1023
C
C M IS THE TOTAL NUMBER OF ELEMENTS IN THE CIRCUIT        V1 1024
C NF IS THE TOTAL NUMBER OF ELEMENTS IN THE CIRCUIT      V1 1025
C A ARRAY CONTAINS PARAMETER VALUFS                       V1 1026
C B ARRAY CONTAINS LOGICAL VARIABLES                      V1 1027
C IC ARRAY CONTAINS CODE NUMBER GIVING ORDER IN WHICH BLOCKS ARE CONV1 1029
C NECTED                                                  V1 1030
C
C FOLLOWING FUNCTION STATEMENTS DEFINE SENSITIVITIES      V1 1032
C
C THETA(EL)=2.*ATAN(1.0)*OMEGA*EL                         V1 1034
GC(OMEGA,V,VHAT)=-CMPLX(0.,OMEGA)*V*VHAT                V1 1035
GOMC(OMEGA,V,VHAT)=-CMPLX(0.,C)*V*VHAT                 V1 1036
GOML(AL,I,IHAT)=I*IHAT*AL*CMPLX(0.,1.)                V1 1037
GL(OMEGA,I,IHAT)=I*IHAT*OMEGA*CMPLX(0.,1.)            V1 1038
GLLTZO(V1,V2,IHAT1,IHAT2,ZO)=(V1*IHAT1-V2*IHAT2)/ZO    V1 1039
GLLTFL(OMEGA,FL,V1,V2,IHAT1,IHAT2)=2.*ATAN(1.)*OMEGA*(V1*IHAT2-V2*V1 1040
1IHAT1)/SIN(THETA(EL))                                   V1 1041
GSCTEL(OMEGA,FL,ZO,I,IHAT)=I*IHAT*ZO*2.*ATAN(1.)*OMEGA*(1./COS(THETA(EL)) 1042
1TA(FL))**2*CMPLX(0.,1.)                                 V1 1043
GSCTOM(EL,ZO,I,IHAT)=I*IHAT*ZO*EL*CMPLX(0.,1.)/COS(THETA(EL)) V1 1044
GSCTZO(EL,I,IHAT)=I*IHAT*SIN(THETA(EL))*CMPLX(0.,1.)/COS(THETA(EL)) V1 1045
1)                                                        V1 1046
GOCTFL(ZO,EL,OMEGA,I,IHAT)=CMPLX(0.,1.)*ZO*I*IHAT*(THETA(EL)/EL)*(V1 1047
11./SIN(THETA(EL)))**2                                   V1 1048
GOCTZO(OMEGA,FL,I,IHAT)=(CMPLX(0.,-1.)*COS(THETA(EL))*I*IHAT/SIN(THETA(EL)) 1049
1FTA(FL))                                               V1 1050
GOCTOM(OMEGA,FL,ZO,I,IHAT)=I*IHAT*ZO*THETA(FL)/(OMEGA*SIN(THETA(EL)) 1051
1)**2)*CMPLX(0.,1.)                                     V1 1052
GLCPBP(RP,OMEGA,Q,OMEGAR,V,VHAT)=-V*VHAT*CMPLX((OMEGAR/Q),((OMEGA*V1 1053
1OMEGA-OMEGAR*OMEGAR)/OMEGA))/2.0                       V1 1054
GLCPQ(BP,Q,OMEGAR,V,VHAT)=V*VHAT*CMPLX(BP*OMEGAR/(2.*Q*Q),0.) V1 1055
GLCPOM(RP,OMEGA,OMEGAR,V,VHAT)=-V*VHAT*CMPLX(0.,(BP/2.)*(1.+(OMEGA*V1 1056
1/OMEGAR)**2))                                          V1 1057
GLCPOR(RP,OMEGA,OMEGAR,Q,V,VHAT)=-RP*V*VHAT*CMPLX(1./(2.*Q),-OMEGA*V1 1058
1R/OMEGA)                                               V1 1059
GLCSQ(XP,OMEGAR,Q,I,IHAT)=-CMPLX((XP*OMEGAR)/(2.*Q*Q),0.)*I*IHAT V1 1060
GLCSOM(OMEGA,XP,OMEGAR,I,IHAT)=-((XP/2.)*(1.+(OMEGAR/OMEGA)**2))*I*V1 1061
1IHAT*CMPLX(0.,1.)                                       V1 1062
GLCSOR(OMEGA,XP,OMEGAR,Q,I,IHAT)=CMPLX((XP/(2.*Q)),-OMEGAR*XP/OMEGA V1 1063
1A)*I*IHAT                                               V1 1064
GLCSXP(OMEGA,OMEGAR,Q,I,IHAT)=CMPLX(OMEGAR/Q,((OMEGA*OMEGA-OMEGAR*V1 1065
1OMEGAR)/OMEGA))*I*IHAT/2.                               V1 1066
OLDFL=0.                                                  V1 1067
GD1=0.                                                    V1 1068
GD2=0.                                                    V1 1069
N=N1                                                       V1 1070
IF (KVR.EQ.1) N=N1-NC-2*ND-1                             V1 1071
IF (OMEGN.LE.30.) GO TO 1                                V1 1072

```

	GO TO 114	V1	1073
1	CONTINUE	V1	1074
	OMEGA=OMEGN	V1	1075
	IF (OMEGN.GT.10..AND.OMEGN.LE.20.) OMEGA=OMFGN-10.	V1	1076
	IF (OMEGN.GT.20.) OMEGA=OMEGN-20.	V1	1077
	IF (M.EQ.0) GO TO 90	V1	1078
	DELO=1.E-7*OMEGA	V1	1079
	KKGD=0	V1	1080
	IF (KKGD.EQ.0) GO TO 3	V1	1081
2	DO 89 KKV=1,N	V1	1082
	KKGD=KKGD+1	V1	1083
	DELX=1.E-4*X(KKV)	V1	1084
	X(KKV)=X(KKV)+DELX	V1	1085
3	CONTINUE	V1	1086
	KGD=0	V1	1087
	IF (OMEGN.LE.20.) GO TO 4	V1	1088
	OMEGA=OMEGA-DELO	V1	1089
	DO 88 KV=1,2	V1	1090
4	CONTINUE	V1	1091
	J=0	V1	1092
	K=0	V1	1093
	VOLD=RL	V1	1094
	IOLD=1.0	V1	1095
	DO 83 L=1,M	V1	1096
	MM=M+1-L	V1	1097
	NN=IC(MM)	V1	1098
	GO TO (5,9,13,17,21,25,31,37,43,48,55,62,69,76,79), NN	V1	1099
5	KKK=NE-K	V1	1100
	IF (B(KKK)) GO TO 6	V1	1101
	GO TO 7	V1	1102
6	JJ=N-J	V1	1103
	A(KKK)=X(JJ)	V1	1104
	J=J+1	V1	1105
7	CALL SUB1 (IOLD,VOLD,A(KKK),OMEGA,INEW,VNEW)	V1	1106
	IF (B(KKK).AND.OMEGN.GT.10.) GO TO 8	V1	1107
	IF (B(KKK)) G(JJ)=GC(OMEGA,VOLD,VOLD)	V1	1108
8	CONTINUE	V1	1109
	IF (B(KKK)) AD(JJ)=VOLD	V1	1110
	KN=K+1	V1	1111
	GO TO 82	V1	1112
9	KKK=NE-K	V1	1113
	IF (B(KKK)) GO TO 10	V1	1114
	GO TO 11	V1	1115
10	JJ=N-J	V1	1116
	A(KKK)=X(JJ)	V1	1117
	J=J+1	V1	1118
11	CALL SUB2 (IOLD,VOLD,A(KKK),OMEGA,INEW,VNEW)	V1	1119
	IOLD=INFW-IOLD	V1	1120
	IF (B(KKK).AND.OMEGN.GT.10.) GO TO 12	V1	1121
	IF (B(KKK)) G(JJ)=GL(OMEGA,IOLD,IOLD)	V1	1122
12	CONTINUE	V1	1123
	IF (B(KKK)) AD(JJ)=IOLD	V1	1124
	KN=K+1	V1	1125
	GO TO 82	V1	1126
13	KKK=NE-K	V1	1127
	IF (B(KKK)) GO TO 14	V1	1128
	GO TO 15	V1	1129
14	JJ=N-J	V1	1130



	A(KKK)=X(JJ)	V1	1131
	J=J+1	V1	1132
15	CALL SUB3 (IOLD,VOLD,A(KKK),OMEGA,INEW,VNEW)	V1	1133
	IF (B(KKK).AND.OMEGN.GT.10.) GO TO 16	V1	1134
	IF (B(KKK)) G(JJ)=GL(OMEGA,IOLD,IOLD)	V1	1135
16	CONTINUE	V1	1136
	IF (R(KKK)) AD(JJ)=IOLD	V1	1137
	KN=K+1	V1	1138
	GO TO 82	V1	1139
17	KKK=NE-J	V1	1140
	IF (B(KKK)) GO TO 18	V1	1141
	GO TO 19	V1	1142
18	JJ=N-J	V1	1143
	A(KKK)=X(JJ)	V1	1144
	J=J+1	V1	1145
19	CALL SUB4 (IOLD,VOLD,A(KKK),OMEGA,INEW,VNEW)	V1	1146
	VOLD=VNEW-VOLD	V1	1147
	IF (B(KKK).AND.OMEGN.GT.10.) GO TO 20	V1	1148
	IF (B(KKK)) G(JJ)=GC(OMEGA,VOLD,VOLD)	V1	1149
20	CONTINUE	V1	1150
	IF (B(KKK)) AD(JJ)=VOLD	V1	1151
	KN=K+1	V1	1152
	GO TO 82	V1	1153
21	KK=K	V1	1154
	JOLD=J	V1	1155
	DO 24 II=1,2	V1	1156
	KKK=NE-KK	V1	1157
	IF (B(KKK)) GO TO 22	V1	1158
	GO TO 23	V1	1159
22	JJ=N-J	V1	1160
	A(KKK)=X(JJ)	V1	1161
	J=J+1	V1	1162
23	KK=KK+1	V1	1163
24	CONTINUE	V1	1164
	JV=NE-K-1	V1	1165
	JH=NE-K	V1	1166
	CALL SUB5 (VOLD,IOLD,A(JV),OMEGA,A(JH),VNEW,INEW)	V1	1167
	NJ=N-JOLD	V1	1168
	IF (B(JH)) AD(NJ)=VOLD	V1	1169
	IF (B(JH)) ADJJ(NJ)=VNEW	V1	1170
	IF (B(JH)) G(NJ)=GLLTZO(VNEW,VOLD,INEW,IOLD,A(JH))	V1	1171
	IF (B(JH)) JOLD=JOLD+1	V1	1172
	NJ=N-JOLD	V1	1173
	IF (B(JV)) AD(NJ)=VOLD	V1	1174
	IF (B(JV)) G(NJ)=GLLTEL(OMEGA,A(JV),VNEW,VOLD,INEW,IOLD)	V1	1175
	IF (B(JV)) ADJJ(NJ)=VNEW	V1	1176
	KN=K+2	V1	1177
	GO TO 82	V1	1178
25	KK=K	V1	1179
	JOLD=J	V1	1180
	DO 28 II=1,2	V1	1181
	KKK=NE-K	V1	1182
	IF (B(KKK)) GO TO 26	V1	1183
	GO TO 27	V1	1184
26	JJ=N-J	V1	1185
	A(KKK)=X(JJ)	V1	1186
	J=J+1	V1	1187
27	KK=KK+1	V1	1188

28	CONTINUE	V1	1189
	JV=NE-K-1	V1	1190
	JH=NE-K	V1	1191
	CALL SUB6 (VOLD,IOLD,A(JV),OMEGA,A(JH),VNEW,INEW)	V1	1192
	IOLD=INEW-IOLD	V1	1193
	NJ=N-JOLD	V1	1194
	IF (B(JH).AND.OMEGN.GT.10.) GO TO 29	V1	1195
	IF (B(JH)) G(NJ)=GSCTZO(A(JV),IOLD,IOLD)	V1	1196
29	CONTINUE	V1	1197
	IF (B(JH)) JOLD=JOLD+1	V1	1198
	IF (B(JH)) AD(NJ)=IOLD	V1	1199
	NJ=N-JOLD	V1	1200
	IF (B(JV).AND.OMEGN.GT.10.) GO TO 30	V1	1201
	IF (B(JV)) G(NJ)=GSCTEL(OMEGA,A(JV),A(JH),IOLD,IOLD)	V1	1202
30	CONTINUE	V1	1203
	IF (B(JV)) AD(NJ)=IOLD	V1	1204
	KN=K+2	V1	1205
	GO TO 82	V1	1206
31	KK=K	V1	1207
	JOLD=J	V1	1208
	DO 34 II=1,2	V1	1209
	KKK=NE-KK	V1	1210
	IF (B(KKK)) GO TO 32	V1	1211
	GO TO 33	V1	1212
32	JJ=N-J	V1	1213
	A(KKK)=X(JJ)	V1	1214
	J=J+1	V1	1215
33	KK=KK+1	V1	1216
34	CONTINUE	V1	1217
	JV=NE-K-1	V1	1218
	JH=NE-K	V1	1219
	CALL SUB7 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW)	V1	1220
	IOLD=INEW-IOLD	V1	1221
	NJ=N-JOLD	V1	1222
	IF (B(JH).AND.OMEGN.GT.10.) GO TO 35	V1	1223
	IF (B(JH)) G(NJ)=GOCTZO(OMEGA,A(JV),IOLD,IOLD)	V1	1224
35	CONTINUE	V1	1225
	IF (B(JH)) JOLD=JOLD+1	V1	1226
	IF (B(JH)) AD(NJ)=IOLD	V1	1227
	NJ=N-JOLD	V1	1228
	IF (B(JV).AND.OMEGN.GT.10.) GO TO 36	V1	1229
	IF (B(JV)) G(NJ)=GOCTFL(A(JH),A(JV),OMEGA,IOLD,IOLD)	V1	1230
36	CONTINUE	V1	1231
	IF (B(JV)) AD(NJ)=IOLD	V1	1232
	KN=K+2	V1	1233
	GO TO 82	V1	1234
37	KK=K	V1	1235
	JOLD=J	V1	1236
	DO 40 II=1,2	V1	1237
	KKK=NF-KK	V1	1238
	IF (B(KKK)) GO TO 38	V1	1239
	GO TO 39	V1	1240
38	JJ=N-J	V1	1241
	A(KKK)=X(JJ)	V1	1242
	J=J+1	V1	1243
39	KK=KK+1	V1	1244
40	CONTINUE	V1	1245
	JV=NF-K-1	V1	1246

	JH=NE-K	V1	1247
	CALL SUB8 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW)	V1	1248
	NJ=N-JOLD	V1	1249
	IF (B(JH).AND.OMEGN.GT.10.) GO TO 41	V1	1250
	IF (B(JH)) G(NJ)=GSCTZO(A(JV),IOLD,IOLD)	V1	1251
41	CONTINUE	V1	1252
	IF (B(JH)) JOLD=JOLD+1	V1	1253
	IF (B(JH)) AD(NJ)=IOLD	V1	1254
	NJ=N-JOLD	V1	1255
	IF (B(JV).AND.OMEGN.GT.10.) GO TO 42	V1	1256
	IF (B(JV)) G(NJ)=GSCTEL(OMEGA,A(JV),A(JH),IOLD,IOLD)	V1	1257
42	CONTINUE	V1	1258
	IF (B(JV)) AD(NJ)=IOLD	V1	1259
	KN=K+2	V1	1260
	GO TO 82	V1	1261
43	KK=K	V1	1262
	JOLD=J	V1	1263
	DO 46 II=1,2	V1	1264
	KKK=NE-KK	V1	1265
	IF (B(KKK)) GO TO 44	V1	1266
	GO TO 45	V1	1267
44	JJ=N-J	V1	1268
	A(KKK)=X(JJ)	V1	1269
	J=J+1	V1	1270
45	KK=KK+1	V1	1271
46	CONTINUE	V1	1272
	JV=NE-K-1	V1	1273
	JH=NE-K	V1	1274
	CALL SUB9 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW)	V1	1275
	NJ=N-JOLD	V1	1276
	IF (B(JH).AND.OMEGN.GT.10.) GO TO 47	V1	1277
	IF (B(JH)) G(NJ)=GOCTZO(OMEGA,A(JV),IOLD,IOLD)	V1	1278
47	CONTINUE	V1	1279
	IF (B(JH)) JOLD=JOLD+1	V1	1280
	IF (B(JH)) AD(NJ)=IOLD	V1	1281
	NJ=N-JOLD	V1	1282
	IF (B(JV).AND.OMEGN.GT.10.) GO TO 42	V1	1283
	IF (B(JV)) G(NJ)=GOCTEL(A(JH),A(JV),OMEGA,IOLD,IOLD)	V1	1284
	IF (B(JV)) AD(NJ)=IOLD	V1	1285
	KN=K+2	V1	1286
	GO TO 82	V1	1287
48	KK=K	V1	1288
	JOLD=J	V1	1289
	DO 51 II=1,3	V1	1290
	KKK=NE-KK	V1	1291
	IF (B(KKK)) GO TO 49	V1	1292
	GO TO 50	V1	1293
49	JJ=N-J	V1	1294
	A(KKK)=X(JJ)	V1	1295
	J=J+1	V1	1296
50	KK=KK+1	V1	1297
51	CONTINUE	V1	1298
	JK=NE-K-2	V1	1299
	JV=NE-K-1	V1	1300
	JH=NE-K	V1	1301
	CALL SUB10 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)	V1	1302
	NJ=N-JOLD	V1	1303
	IF (B(JH).AND.OMEGN.GT.10.) GO TO 52	V1	1304

	IF (B(JH)) G(NJ)=GLCSXP(OMEGA,A(JK),A(JV),IOLD,IOLD)	V1	1305
52	CONTINUE	V1	1306
	IF (B(JH)) JOLD=JOLD+1	V1	1307
	IF (B(JH)) AD(NJ)=IOLD	V1	1308
	NJ=N-JOLD	V1	1309
	IF (B(JV).AND.OMEGN.GT.10.) GO TO 53	V1	1310
	IF (B(JV)) G(NJ)=GLCSQ(A(JH),A(JK),A(JV),IOLD,IOLD)	V1	1311
53	CONTINUE	V1	1312
	IF (B(JV)) AD(NJ)=IOLD	V1	1313
	IF (B(JV)) JOLD=JOLD+1	V1	1314
	NJ=N-JOLD	V1	1315
	IF (B(JK).AND.OMEGN.GT.10.) GO TO 54	V1	1316
	IF (B(JK)) G(NJ)=GLCSOR(OMEGA,A(JH),A(JK),A(JV),IOLD,IOLD)	V1	1317
54	CONTINUE	V1	1318
	IF (B(JK)) AD(NJ)=IOLD	V1	1319
	IF (B(JK)) JOLD=JOLD+1	V1	1320
	KN=K+3	V1	1321
	GO TO 82	V1	1322
55	KK=K	V1	1323
	JOLD=J	V1	1324
	DO 58 II=1,3	V1	1325
	KKK=NE-KK	V1	1326
	IF (B(KKK)) GO TO 56	V1	1327
	GO TO 57	V1	1328
56	JJ=N-J	V1	1329
	A(KKK)=X(JJ)	V1	1330
	J=J+1	V1	1331
57	KK=KK+1	V1	1332
58	CONTINUE	V1	1333
	JK=NE-K-2	V1	1334
	JV=NE-K-1	V1	1335
	JH=NE-K	V1	1336
	CALL SUR11 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)	V1	1337
	IOLD=INEW-IOLD	V1	1338
	NJ=N-JOLD	V1	1339
	IF (B(JH).AND.OMEGN.GT.10.) GO TO 59	V1	1340
	IF (B(JH)) G(NJ)=GLCSXP(OMEGA,A(JK),A(JV),IOLD,IOLD)	V1	1341
59	CONTINUE	V1	1342
	IF (B(JH)) AD(NJ)=IOLD	V1	1343
	IF (B(JH)) JOLD=JOLD+1	V1	1344
	NJ=N-JOLD	V1	1345
	IF (B(JV).AND.OMEGN.GT.10.) GO TO 60	V1	1346
	IF (B(JV)) G(NJ)=GLCSQ(A(JH),A(JK),A(JV),IOLD,IOLD)	V1	1347
60	CONTINUE	V1	1348
	IF (B(JV)) JOLD=JOLD+1	V1	1349
	IF (B(JV)) AD(NJ)=IOLD	V1	1350
	NJ=N-JOLD	V1	1351
	IF (B(JK).AND.OMEGN.GT.10.) GO TO 61	V1	1352
	IF (B(JK)) G(NJ)=GLCSOR(OMEGA,A(JH),A(JK),A(JV),IOLD,IOLD)	V1	1353
61	CONTINUE	V1	1354
	IF (B(JK)) AD(NJ)=IOLD	V1	1355
	KN=K+3	V1	1356
	GO TO 82	V1	1357
62	KK=K	V1	1358
	JOLD=J	V1	1359
	DO 65 II=1,3	V1	1360
	KKK=NE-KK	V1	1361
	IF (B(KKK)) GO TO 63	V1	1362

	GO TO 64	V1	1363
63	JJ=N-J	V1	1364
	A(KKK)=X(JJ)	V1	1365
	J=J+1	V1	1366
64	KK=KK+1	V1	1367
65	CONTINUE	V1	1368
	JK=NE-K-2	V1	1369
	JV=NE-K-1	V1	1370
	JH=NE-K	V1	1371
	CALL SUB12 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)	V1	1372
	NJ=N-JOLD	V1	1373
	IF (B(JH).AND.OMEGN.GT.10.) GO TO 66	V1	1374
	IF (B(JH)) G(NJ)=GLCPBP(A(JH),OMEGA,A(JV),A(JK),VOLD,VOLD)	V1	1375
66	CONTINUE	V1	1376
	IF (B(JH)) AD(NJ)=VOLD	V1	1377
	IF (B(JH)) JOLD=JOLD+1	V1	1378
	NJ=N-JOLD	V1	1379
	IF (B(JV).AND.OMEGN.GT.10.) GO TO 67	V1	1380
	IF (B(JV)) G(NJ)=GLCPQ(A(JH),A(JV),A(JK),VOLD,VOLD)	V1	1381
67	CONTINUE	V1	1382
	IF (B(JV)) JOLD=JOLD+1	V1	1383
	IF (B(JV)) AD(NJ)=VOLD	V1	1384
	NJ=N-JOLD	V1	1385
	IF (B(JK).AND.OMEGN.GT.10.) GO TO 68	V1	1386
	IF (B(JK)) G(NJ)=GLCPOR(A(JH),OMEGA,A(JK),A(JV),VOLD,VOLD)	V1	1387
68	CONTINUE	V1	1388
	IF (B(JK)) AD(NJ)=VOLD	V1	1389
	KN=K+3	V1	1390
	GO TO 82	V1	1391
69	KK=K	V1	1392
	JOLD=J	V1	1393
	DO 72 II=1,3	V1	1394
	KKK=NE-KK	V1	1395
	IF (B(KKK)) GO TO 70	V1	1396
	GO TO 71	V1	1397
70	JJ=N-J	V1	1398
	A(KKK)=X(JJ)	V1	1399
	J=J+1	V1	1400
71	KK=KK+1	V1	1401
72	CONTINUE	V1	1402
	JK=NE-K-2	V1	1403
	JV=NE-K-1	V1	1404
	JH=NE-K	V1	1405
	CALL SUB13 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)	V1	1406
	VOLD=VNEW-VOLD	V1	1407
	NJ=N-JOLD	V1	1408
	IF (B(JH).AND.OMEGN.GT.10.) GO TO 73	V1	1409
	IF (B(JH)) G(NJ)=GLCPBP(A(JH),OMEGA,A(JV),A(JK),VOLD,VOLD)	V1	1410
73	CONTINUE	V1	1411
	IF (B(JH)) JOLD=JOLD+1	V1	1412
	IF (B(JH)) AD(NJ)=VOLD	V1	1413
	IF (B(JV).AND.OMEGN.GT.10.) GO TO 74	V1	1414
	IF (B(JV)) G(NJ)=GLCPQ(A(JH),A(JV),A(JK),VOLD,VOLD)	V1	1415
74	CONTINUE	V1	1416
	IF (B(JV)) JOLD=JOLD+1	V1	1417
	IF (B(JV)) AD(NJ)=VOLD	V1	1418
	NJ=N-JOLD	V1	1419
	IF (B(JK).AND.OMEGN.GT.10.) GO TO 75	V1	1420

	IF (B(JK)) G(NJ)=GLCPOR(A(JH),OMEGA,A(JK),A(JV),VOLD,VOLD)	V1	1421
75	CONTINUE	V1	1422
	IF (B(JK)) AD(NJ)=VOLD	V1	1423
	NJ=N-JOLD	V1	1424
	KN=K+3	V1	1425
	GO TO 82	V1	1426
76	KKK=NE-K	V1	1427
	IF (B(KKK)) GO TO 77	V1	1428
	GO TO 78	V1	1429
77	JJ=N-J	V1	1430
	A(KKK)=X(JJ)	V1	1431
	J=J+1	V1	1432
78	CALL SUB14 (IOLD,VOLD,A(KKK),INEW,VNEW)	V1	1433
	IF (B(KKK)) G(JJ)=IOLD*IOLD	V1	1434
	IF (B(KKK)) AD(JJ)=IOLD	V1	1435
	KN=K+1	V1	1436
	GO TO 82	V1	1437
79	KKK=NE-K	V1	1438
	IF (B(KKK)) GO TO 80	V1	1439
	GO TO 81	V1	1440
80	JJ=N-J	V1	1441
	A(KKK)=X(JJ)	V1	1442
	J=J+1	V1	1443
81	CALL SUB15 (IOLD,VOLD,A(KKK),INEW,VNEW)	V1	1444
	IOLD=INEW-IOLD	V1	1445
	IF (B(KKK)) G(JJ)=IOLD*IOLD	V1	1446
	IF (B(KKK)) AD(JJ)=IOLD	V1	1447
	KN=K+1	V1	1448
	GO TO 82	V1	1449
82	VOLD=VNEW	V1	1450
	IOLD=INEW	V1	1451
	K=KN	V1	1452
83	CONTINUE	V1	1453
	IF (OMEGN.GT.20.) GO TO 87	V1	1454
	IF (OMEGN.GT.10.) GO TO 93	V1	1455
	RHO=1.-2.*INEW/(VNEW+INEW)	V1	1456
	CONRHO=CONJG(RHO)	V1	1457
	APP=CARS(RHO)	V1	1458
	DO 84 L=1,N	V1	1459
	GRAD(L)=REAL(((CONRHO/APP)*2.*G(L)/((VNEW+INEW)**2))	V1	1460
84	CONTINUE	V1	1461
	IF (NC.GT.0.OR.ND.GT.0) GO TO 85	V1	1462
	RETURN	V1	1463
85	IF (KVR.EQ.0) RETURN	V1	1464
	NNN=N+1	V1	1465
	DO 86 L=NNN,N1	V1	1466
	GRAD(L)=0.	V1	1467
86	CONTINUE	V1	1468
	RETURN	V1	1469
87	CONTINUE	V1	1470
	PHASE=(1./(VNEW+INEW))	V1	1471
	IF (KGD.EQ.0) OMEGA=OMEGA+2*OMEGA	V1	1472
	IF (KGD.EQ.0) POLD=PHASE	V1	1473
	IF (KGD.GT.0) DFLAY=-AIMAG((1./POLD)*((PHASE-POLD)/(2*DELO)))*(100V1	V1	1474
	10./(8.*ATAN(1.)*FM))	V1	1475
	KGD=KGD+1	V1	1476
88	CONTINUE	V1	1477
	OMEGA=OMEGA-DELO	V1	1478

	IF (KKGD.EQ.0) OLDEL=DELAY	V1	1479
	IF (KKGD.EQ.0) GO TO 2	V1	1480
	GRAD(KKV)=(DFLAY-OLDEL)/DELX	V1	1481
	X(KKV)=X(KKV)-DFLX	V1	1482
90	CONTINUE	V1	1483
	APP=OLDEL	V1	1484
	IF (NC.GT.0.OR.ND.GT.0) GO TO 90	V1	1485
	RETURN	V1	1486
90	IF (KVR.EQ.0) GO TO 92	V1	1487
	NNN=N1-N	V1	1488
	DO 91 L=1,NNN	V1	1489
	LL=N+L	V1	1490
	AR(L)=X(LL)	V1	1491
91	CONTINUE	V1	1492
92	WCC=WC/FM	V1	1493
	IF (NC.GT.0) CALL SUB16 (NC,N,AB,WCC,OMEGA,GD1,GRAD,FM)	V1	1494
	IF (ND.GT.0) CALL SUB17 (ND,N,NC,AB,WCC,OMEGA,GD2,GRAD,FM)	V1	1495
	APP=OLDEL+GD1+GD2-AR(NNN)	V1	1496
	IF (KVR.FQ.1) GRAD(N1)=-1.0	V1	1497
	RETURN	V1	1498
92	APP=-20.*ALOG10((CABS(1./(VNEW+INEW)))*(1.+RL))	V1	1499
	K=1	V1	1500
	J=1	V1	1501
	VOLD=1.0	V1	1502
	IOLD=1.0	V1	1503
	DO 110 L=1,M	V1	1504
	NN=IC(L)	V1	1505
	GO TO (94,95,96,97,98,99,100,101,102,103,104,105,106,107,108), NN	V1	1506
94	CALL SUB1 (IOLD,VOLD,A(K),OMEGA,INEW,VNEW)	V1	1507
	IF (B(K)) G(J)=GC(OMEGA,AD(J),VOLD)	V1	1508
	IF (B(K)) J=J+1	V1	1509
	K=K+1	V1	1510
	GO TO 109	V1	1511
95	CALL SUB2 (IOLD,VOLD,A(K),OMEGA,INEW,VNEW)	V1	1512
	IOLD=INEW-IOLD	V1	1513
	IF (B(K)) G(J)=GL(OMEGA,AD(J),IOLD)	V1	1514
	IF (B(K)) J=J+1	V1	1515
	K=K+1	V1	1516
	GO TO 109	V1	1517
96	CALL SUB3 (IOLD,VOLD,A(K),OMEGA,INEW,VNEW)	V1	1518
	IF (B(K)) G(J)=-GL(OMEGA,AD(J),IOLD)	V1	1519
	IF (B(K)) J=J+1	V1	1520
	K=K+1	V1	1521
	GO TO 109	V1	1522
97	CALL SUB4 (IOLD,VOLD,A(K),OMEGA,INEW,VNEW)	V1	1523
	VOLD=VNEW-VOLD	V1	1524
	IF (B(K)) G(J)=-GC(OMEGA,AD(J),VOLD)	V1	1525
	IF (B(K)) J=J+1	V1	1526
	K=K+1	V1	1527
	GO TO 109	V1	1528
98	JV=K	V1	1529
	JH=K+1	V1	1530
	CALL SUB5 (VOLD,IOLD,A(JV),OMEGA,A(JH),VNEW,INEW)	V1	1531
	IF (B(JV)) G(J)=-GLLTEL(OMEGA,A(JV),ADJJ(J),AD(J),IOLD,INEW)	V1	1532
	IF (B(JV)) J=J+1	V1	1533
	IF (B(JH)) G(J)=-GLLTZO(ADJJ(J),AD(J),IOLD,INEW,A(JH))	V1	1534
	IF (B(JH)) J=J+1	V1	1535
	K=K+2	V1	1536

	GO TO 109	V1	1537
99	JV=K	V1	1538
	JH=K+1	V1	1539
	CALL SUB6 (VOLD,IOLD,A(JV),OMEGA,A(JH),VNEW,INEW)	V1	1540
	IOLD=INEW-IOLD	V1	1541
	IF (B(JV)) G(J)=GSCTEL(OMEGA,A(JV),A(JH),AD(J),IOLD)	V1	1542
	IF (B(JV)) J=J+1	V1	1543
	IF (B(JH)) G(J)=GSCTZO(A(JV),AD(J),IOLD)	V1	1544
	IF (B(JH)) J=J+1	V1	1545
	K=K+2	V1	1546
	GO TO 109	V1	1547
100	JV=K	V1	1548
	JH=K+1	V1	1549
	CALL SUB7 (VOLD,IOLD,A(JV),OMEGA,A(JH),VNEW,INEW)	V1	1550
	IOLD=INEW-IOLD	V1	1551
	IF (B(JV)) G(J)=GOCTEL(A(JH),A(JV),OMEGA,AD(J),IOLD)	V1	1552
	IF (B(JV)) J=J+1	V1	1553
	IF (B(JH)) G(J)=GOCTZO(OMEGA,A(JV),AD(J),IOLD)	V1	1554
	IF (B(JH)) J=J+1	V1	1555
	K=K+2	V1	1556
	GO TO 109	V1	1557
101	JV=K	V1	1558
	JH=K+1	V1	1559
	CALL SUB8 (VOLD,IOLD,A(JV),OMEGA,A(JH),VNEW,INEW)	V1	1560
	IF (B(JV)) G(J)=-GSCTEL(OMEGA,A(JV),A(JH),AD(J),IOLD)	V1	1561
	IF (B(JV)) J=J+1	V1	1562
	IF (B(JH)) G(J)=-GSCTZO(A(JV),AD(J),IOLD)	V1	1563
	IF (B(JH)) J=J+1	V1	1564
	K=K+2	V1	1565
	GO TO 109	V1	1566
102	JV=K	V1	1567
	JH=K+1	V1	1568
	CALL SUB9 (VOLD,IOLD,A(JV),OMEGA,A(JH),VNEW,INEW)	V1	1569
	IF (B(JV)) G(J)=-GOCTEL(A(JH),A(JV),OMEGA,AD(J),IOLD)	V1	1570
	IF (B(JV)) J=J+1	V1	1571
	IF (B(JH)) G(J)=-GOCTZO(OMEGA,A(JV),AD(J),IOLD)	V1	1572
	IF (B(JH)) J=J+1	V1	1573
	K=K+2	V1	1574
	GO TO 109	V1	1575
103	JK=K	V1	1576
	JV=K+1	V1	1577
	JH=K+2	V1	1578
	CALL SUB10 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)	V1	1579
	IF (B(JK)) G(J)=-GLCSOR(OMEGA,A(JH),A(JK),A(JV),AD(J),IOLD)	V1	1580
	IF (B(JK)) J=J+1	V1	1581
	IF (B(JV)) G(J)=-GLCSQ(A(JH),A(JK),A(JV),AD(J),IOLD)	V1	1582
	IF (B(JV)) J=J+1	V1	1583
	IF (B(JH)) G(J)=-GLCSXP(OMEGA,A(JK),A(JV),AD(J),IOLD)	V1	1584
	IF (B(JH)) J=J+1	V1	1585
	K=K+3	V1	1586
	GO TO 109	V1	1587
104	JK=K	V1	1588
	JV=K+1	V1	1589
	JH=K+2	V1	1590
	CALL SUB11 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)	V1	1591
	IOLD=INEW-IOLD	V1	1592
	IF (B(JK)) G(J)=GLCSOR(OMEGA,A(JH),A(JK),A(JV),AD(J),IOLD)	V1	1593
	IF (B(JK)) J=J+1	V1	1594



	IF (B(JV)) G(J)=GLCSQ(A(JH),A(JK),A(JV),AD(J),IOLD)	V1	1595
	IF (B(JV)) J=J+1	V1	1596
	IF (B(JH)) G(J)=GLCSXP(OMEGA,A(JK),A(JV),AD(J),IOLD)	V1	1597
	IF (B(JH)) J=J+1	V1	1598
	K=K+3	V1	1599
	GO TO 109	V1	1600
105	JK=K	V1	1601
	JV=K+1	V1	1602
	JH=K+2	V1	1603
	CALL SUB12 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)	V1	1604
	IF (B(JK)) G(J)=GLCPOR(A(JH),OMEGA,A(JK),A(JV),AD(J),VOLD)	V1	1605
	IF (B(JK)) J=J+1	V1	1606
	IF (B(JV)) G(J)=GLCPQ(A(JH),A(JV),A(JK),AD(J),VOLD)	V1	1607
	IF (B(JV)) J=J+1	V1	1608
	IF (B(JH)) G(J)=GLCPBP(A(JH),OMEGA,A(JV),A(JK),AD(J),VOLD)	V1	1609
	IF (B(JH)) J=J+1	V1	1610
	K=K+3	V1	1611
	GO TO 109	V1	1612
106	JK=K	V1	1613
	JV=K+1	V1	1614
	JH=K+2	V1	1615
	CALL SUB13 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)	V1	1616
	VOLD=VNEW-VOLD	V1	1617
	IF (B(JK)) G(J)=-GLCPOR(A(JH),OMEGA,A(JK),A(JV),AD(J),VOLD)	V1	1618
	IF (B(JK)) J=J+1	V1	1619
	IF (B(JV)) G(J)=-GLCPQ(A(JH),A(JV),A(JK),AD(J),VOLD)	V1	1620
	IF (B(JV)) J=J+1	V1	1621
	IF (B(JH)) G(J)=-GLCPBP(A(JH),OMEGA,A(JV),A(JK),AD(J),VOLD)	V1	1622
	IF (B(JH)) J=J+1	V1	1623
	K=K+3	V1	1624
	GO TO 109	V1	1625
107	CALL SUB14 (IOLD,VOLD,A(K),INEW,VNEW)	V1	1626
	IF (B(K)) G(J)=-IOLD*AD(J)	V1	1627
	IF (B(K)) J=J+1	V1	1628
	K=K+1	V1	1629
	GO TO 109	V1	1630
108	CALL SUB15 (IOLD,VOLD,A(K),INEW,VNEW)	V1	1631
	IOLD=INEW-IOLD	V1	1632
	IF (B(K)) G(J)=IOLD*AD(J)	V1	1633
	IF (B(K)) J=J+1	V1	1634
	K=K+1	V1	1635
	GO TO 109	V1	1636
109	IOLD=INEW	V1	1637
	VOLD=VNEW	V1	1638
110	CONTINUE	V1	1639
	PSIL=VNEW+INEW*RL	V1	1640
	CONSTN=20./ALOG(10.)	V1	1641
	DO 111 L=1,N	V1	1642
	GRAD(L)=-REAL(G(L)/PSIL)*CONSTN	V1	1643
111	CONTINUE	V1	1644
	IF (NC.GT.0.OR.ND.GT.0) GO TO 112	V1	1645
	RETURN	V1	1646
112	IF (KVR.FQ.0) RRETURN	V1	1647
	NNN=N+1	V1	1648
	DO 113 L=NNN,N1	V1	1649
	GRAD(L)=0.	V1	1650
113	CONTINUE	V1	1651
	RETURN	V1	1652

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

```
V1 1653  
V1 1654  
V1 1655  
V1 1656  
V1 1657  
V1 1658  
V1 1659  
V1 1660
```

```
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
```

```
V1 1661  
V1 1662  
V1 1663  
V1 1664  
V1 1665  
V1 1666  
V1 1667
```

```
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
```

```
V1 1668  
V1 1669  
V1 1670  
V1 1671  
V1 1672  
V1 1673  
V1 1674
```

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

```
V1 1675
V1 1676
V1 1677
V1 1678
V1 1679
V1 1680
V1 1681
```

```
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
```

```
V1 1682  
V1 1683  
V1 1684  
V1 1685  
V1 1686  
V1 1687  
V1 1688
```



SUBROUTINE SUB6 (IOLD,VOLD,EL,OMEGA,ZO,INEW,VNEW)	V1	1698
COMPLEX IOLD,VOLD,INEW,VNEW	V1	1699
THETA=2.*ATAN(1.0)*OMEGA*EL	V1	1700
VNEW=VOLD	V1	1701
INEW=IOLD-VOLD*COS(THETA)/(ZO*SIN(THETA))*CMPLX(0.,1.)	V1	1702
RETURN	V1	1703
END	V1	1704





SUBROUTINE SUBR (IOLD,VOLD,EL,OMEGA,ZO,INEW,VNEW)	V1	1712
COMPLX IOLD,VOLD,INEW,VNEW	V1	1713
THETA=2.*ATAN(1.0)*OMEGA*EL	V1	1714
INEW=IOLD	V1	1715
VNEW=VOLD+CMPLX(0.,1.0)*ZO*IOLD*SIN(THETA)/COS(THETA)	V1	1716
RETURN	V1	1717
END	V1	1718











```
SUBROUTINE SUB14 (IOLD,VOLD,R,INFW,VNEW)  
COMPLEX IOLD,VOLD,INFW,VNEW,Z  
Z=CMPLX(R,0.)  
VNEW=VOLD+IOLD*Z  
INFW=IOLD  
RETURN  
END
```

```
V1 1754  
V1 1755  
V1 1756  
V1 1757  
V1 1758  
V1 1759  
V1 1760
```



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

```
V1 1761  
V1 1762  
V1 1763  
V1 1764  
V1 1765  
V1 1766  
V1 1767
```

SUBROUTINE SUB16 (K,N,A,WC,OMEGA,GD,GRAD,FM)	V1	1768
DIMENSION A(1), GRAD(1)	V1	1769
PIF=4.0*ATAN(1.)	V1	1770
FC=(PIE/2.)*SQRT((OMEGA*OMEGA-WC*WC)/(1.-WC*WC))	V1	1771
CAPO=SIN(FC)/COS(FC)	V1	1772
GD=0	V1	1773
DO 1 I=1,K	V1	1774
J=I	V1	1775
S=OMEGA/SQRT((OMEGA**2-WC**2)*(1.-WC*WC))	V1	1776
QUA1=(A(J)**2+CAPO*CAPO)**2	V1	1777
QUA2=(1.+CAPO*CAPO)/2.	V1	1778
GD=GD+(A(J)/SQRT(QUA1))*QUA2*S/(FM*0.001)	V1	1779
GRAD(J)=-S*QUA2*((A(J)**2-CAPO*CAPO)/QUA1)/(FM*0.001)	V1	1780
CONTINUE	V1	1781
RETURN	V1	1782
END	V1	1783

1

SUBROUTINE SUB17 (M,N,K,A,WC,OMEGA,GD,GRAD,FM)	V1	1784
DIMENSION A(1), GRAD(1)	V1	1785
CONST=CMPLX(0.,1.)	V1	1786
PIF=4.*ATAN(1.)	V1	1787
FC=(PIF/2.0)*SQRT((OMEGA*OMEGA-WC*WC)/(1.-WC*WC))	V1	1788
S=OMEGA/SQRT((OMEGA**2-WC**2)*(1.-WC*WC))	V1	1789
CAPO=SIN(FC)/COS(FC)	V1	1790
GD=0	V1	1791
DO 1 I=1,M	V1	1792
J=I+K	V1	1793
JJ=J+M	V1	1794
SAM=SQRT(A(J)**2+A(JJ)**2)	V1	1795
GD=GD+A(J)*(1.+CAPO*CAPO)*((CAPO**2+SAM**2)/(CAPO**4+SAM**4+2.*CAPO	V1	1796
10*CAPO*(A(J)**2-A(JJ)**2)))	V1	1797
GD=GD*1000./FM	V1	1798
CONTINUE	V1	1799
GD=S*GD	V1	1800
DO 2 I=1,M	V1	1801
J=I+K	V1	1802
JJ=J+M	V1	1803
DEN=CAPO**4+2.*CAPO*CAPO*(A(J)**2-A(JJ)**2)+(A(J)**2+A(JJ)**2)**2	V1	1804
DEN1=CAPO*CAPO+3.*A(J)*A(J)+A(JJ)*A(JJ)	V1	1805
AK=1.+CAPO*CAPO	V1	1806
DEN2=CAPO*CAPO*A(J)+A(J)**3+A(JJ)*A(JJ)*A(J)	V1	1807
DEN3=4.*A(J)*(CAPO*CAPO+A(J)*A(J)+A(JJ)*A(JJ))	V1	1808
GRAD(J)=S*(DEN*AK*DEN1-AK*DEN2*DEN3)/(DEN**2)	V1	1809
DEN4=4.*A(JJ)*(A(JJ)**2+A(J)**2-CAPO*CAPO)	V1	1810
GRAD(JJ)=S*(DEN*AK*2.*A(J)*A(JJ)-AK*DEN2*DEN4)/(DEN**2)	V1	1811
GRAD(J)=GRAD(J)/(FM*0.001)	V1	1812
GRAD(JJ)=GRAD(JJ)/(FM*0.001)	V1	1813
CONTINUE	V1	1814
RETURN	V1	1815
END	V1	1816





