CRL INTERNAL REPORT SERIES
No. CRL-12

CANOPT - CASCADED NETWORK OPTIMIZATION PACKAGE

J.W. Bandler and J.R. Popović

COMMUNICATIONS RESEARCH LABORATORY

FACULTY OF ENGINEERING

McMASTER UNIVERSITY

HAMILTON, ONTARIO, CANADA

CRL INTERNAL REPORT SERIES
No. CRL-12

CANOPT - CASCADED NETWORK OPTIMIZATION PACKAGE

J.W. Bandler and J.R. Popović

NOVEMBER 1973

CANOPT - CASCADED NETWORK OPTIMIZATION PACKAGE

PURPOSE: The program analyzes and optimizes certain cascaded linear, time-invariant networks in the frequency domain made up of the two-port elements such as resistors, inductors, capacitors, lossless transmission lines, lossless short-circuited and open-circuited transmission line stubs, series and parallel RLC resonant circuits and microwave allpass C- and D-sections.

AUTHORS: J.W. Bandler and J.R. Popović
Communications Research Laboratory
and Department of Electrical Engineering
McMaster University, Hamilton, Ontario, Canada

LANGUAGE: FORTRAN IV, 1578 cards.

CORE REQUIREMENT: Around $18K_{10}$ or around 45000 words in octal.

AVAILABILITY: A user's manual with an example and program listing is appended.

DESCRIPTION: At the user's discretion, a least pth objective function [1] incorporating simultaneously input reflection coefficient, insertion loss, relative group delay and parameter constraints (if any) is automatically created and minimized by gradient optimization methods such as Fletcher-Powell [2] or Fletcher [3]. The adjoint method of gradient evaluation for circuit elements in the frequency domain is employed [4]. Table I is a summary of the features and options of the program and the parameters which must be specified by the user.

The package CANOPT will analyze and optimize a cascade connection of the two-port elements listed in Tables II and III. Elements 1 to 15 may be connected in any order (sequentially from the source to the load) using as many as required or as many as the computer being used can accommodate.

The first six elements are one-parameter lumped elements. Their parameter values should be normalized by the user to his center frequency and source resistance, appropriately, as outlined in the companion paper [5].

The next four elements are three-parameter tuned circuits. They are characterized by resonant frequency, quality factor, and slope reactance or susceptance, as appropriate. Normalization as before must again be carried out by the user.

Elements 11 to 15 are two-parameter lossless transmission-line components. All are characterized by normalized length and characteristic impedance (see reference [5]).

The allpass sections (Table III) are treated in the same way as, for example, Kudsia [6]. Group delay relative to delay level in nsec is calculated.

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

Before attempting to use the program the user should familiarize himself with the paper describing the theory and organization of the package [5].

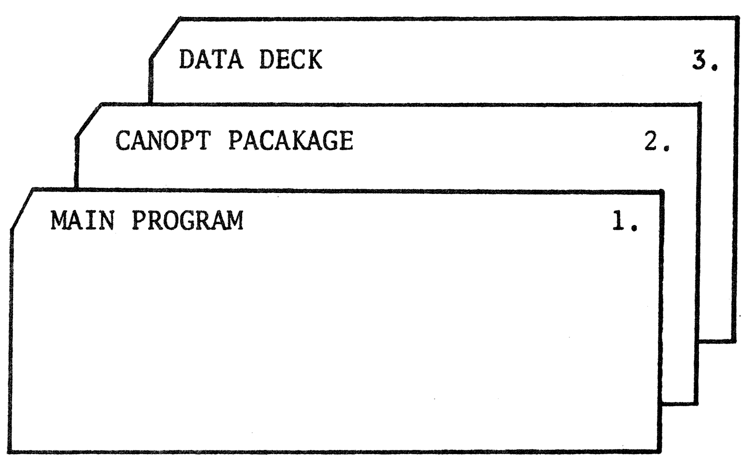
HOW TO USE THE PROGRAM

The package is written in FORTRAN IV and tested on a CDC 6400 digital computer and requires the CDC system routine SECOND which keeps track of elapsed time. For a different system the cards 230, 247, 264, 518, 558, 687, 692 and 723 should be replaced by cards appropriate to the system or removed together with cards 274, 559, 693 and 724.

Data cards No. 20 and 21 may also have different forms for different systems, and should be replaced by appropriate cards.

A core requirement of about $18K_{10}$ is sufficient to optimize, e.g., a seven parameter problem using 25 sample points.

Set up the input deck as follows:



1. Main program

Write the main program as indicated below:

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

Dimension the following arrays:

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

where

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

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

Call the subroutine CANOPT as follows:

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

2. The CANOPT package

A listing is contained in this report.

3. Data deck

Parameters to be supplied as data are defined below:

MM	The number of elements in the circuit, not including C and D sections. Set to 0 if you do not want any.
IC(I),I=1,MM	A sequence of code numbers of elements which specify the order in which the elements are sequentially connected from ~~load~~ source to ~~source~~ load. (See Table II for element numbers).
AA(I),I=1,NE	Values of parameters in the circuit including starting values for variables. (The total number of parameters is NE).
IC(MM+I),I=1,NE	Indicates whether the parameters in the circuit are fixed or variable. Set to 1 if variable and 0 if fixed.
NC	The number of C sections. Set to 0 if you do not want any.
ND	The number of D sections. Set to 0 if you do not want any.
KVR	Denotes whether C and D section parameters are variable or fixed. Set to 1 if variable and 0 if fixed.
AB(I),I=1,J	Values of the parameters of the C and D sections and d level. (See Kudsia [6]). The d level is treated like any other circuit parameter.
R	The load resistance.
NINTD	The number of frequency bands or intervals.
NINTS	The number of other frequency points and constraints.
XX(1,I),I=1,NINT where $NINT \triangleq NINTD+NINTS$	The lower frequency bounds (band edges) for bands, single frequency points, parameters to be constrained (artifical frequency points).

XX(2,J),J=1,NINTD	The upper frequency bounds (band edges) for bands.
NUMB(I),I=1,NINTD	The number of subintervals (equals sample points minus one).
FUN(I),I=1,NINT	A sequence of numbers to be used as specifications or constraints.
XX(3,I),I=1,NINT	Indicates whether a specification or constraint for any given I is an upper or lower one. Set to 1. for upper, to -1. for lower and to 0. for single.
WT(I),I=1,NINT	The weighting factors (positive). Set to 1. if unsure.
IOBJ(I),I=1,NINT	The approximating function: For reflection coefficient set to 1. For insertion loss (dB) set to 2. For group delay (nsec) set to 3. For parameter constraints set to 0.
FM	The center frequency (e.g., in MHz, for normalization).
WC	The cut-off frequency for C- and D-sections (e.g., in MHz).
MET	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 testing convergence in the Fletcher method (e.g., 10^{-4}). K is the number of variable parameters.
EPS1	A small quantity for testing convergence in the Fletcher-Powell method (e.g., 10^{-4}).
EST	A realistic under-estimate (lower bound) of the value of the objective function.
DIF	The value of the difference between objective function values in successive optimizations. Set to 0. if not sure.
KSI	A small quantity by which specifications could be shifted artificially [1]. Set to 0. if not sure.
IPA(I),I=1,ITER	Vector containing the values of p (positive integer, greater than one) to be used successively in a total of ITER complete optimizations.

Table IV shows the arrangement of the data deck.

COMMENTS

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

The program terminates when stopping criteria for the Fletcher-Powell or Fletcher method are satisfied, or when the relative change in the objective function in two successive iterations is less than a small prescribed quantity.

From the experimental results the Fletcher-Powell method was found to be reliable. The Fletcher method, however, was found much quicker and more suitable for network optimization. If the Fletcher-Powell method is completely removed from the package (take out cards no. 261, 262 and 674 to 998), there is a saving in core memory of about $1K_{10}$.

The background theory and possibility for additional features is described in reference [5].

The package has been tested on a CDC 6400 digital computer. A test example will be presented here to illustrate the approach. Examples of input and output as well as actual execution times will be given.

INPUT-OUTPUT EXAMPLE

A seven-section equal-ripple band-pass filter of 3 to 1 bandwidth (ratio of upper band edge to lower band edge) consisting of two unit elements and five stubs which has been considered by Horton and Wenzel [7], as represented in Fig. 1., will show how to set up the package CANOPT. The terminations of the filter are unity. The filter will be designed to have 0.1 dB ripple in the pass band, from 1.0875 to 3.2625 GHz, and the attenuation above 50 dB at the frequency points .6 and 3.75 GHz in the stopband. All section lengths were kept fixed at normalized values 1, and the normalized characteristic impedances are used as variables. Only 21 uniformly spaced sample points were used in the passband to demonstrate the work of the package, although there is actaully a need for a larger number of discrete points to obtain better results. The artificial margin is set to be zero and the weighting function is set to be 1 everywhere. The starting value of the variable vector (See Fig. 1) was

$$\underset{\sim}{Z}_0 = [.63 \quad .33 \quad 1.27 \quad .26 \quad 1.27 \quad .33 \quad .63]^T$$

Although for physical reasons the symmetrical results for the variables are expected, symmetry was not assumed.

Ihe user's written main program and data deck are shown in Fig. 2. Core requirement was 42 331 words in octal. The typical output of CANOPT for the example is shown in Fig. 3. Results are obtained by the Fletcher method for p=2 and $p=10^3$.

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. Suggestions for improvement of the package made by W.Y. Chu of McMaster University are gratefully acknowledged. The continuing support of this work by the Communications Research Laboratory of McMaster University is appreciated.

REFERENCES

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

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

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

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

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

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

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

TABLE I

SUMMARY OF FEATURES, OPTIONS AND PARAMETERS REQUIRED

Features	Type	Options	Parameters
Objective Functions	Least pth	$1 < p < \infty$	Value of p for each of a specified number of optimizations Artificial margin Difference in objective functions for termination
Performance Specifications and Parameter Constraints	Upper (+1.) Lower (-1.) Single (0.)	Reflection coefficient (1) Insertion loss (2) Group delay (3) Parameter value (0)	Normalization frequency Number of points and constraints; Number of bands or intervals For each: Specification/constraint Weighting factor Type Option Frequency (sample point) or parameter; Lower and upper frequencies (band edges) Number of subintervals
Optimization Methods	Gradient	Fletcher (1) Fletcher-Powell (2)	Option Number of iterations allowed Estimate of lower bound on objective function Test quantities for termination
Circuit Elements	Cascaded Two-port	See Tables II and III	Number of elements Sequence of code numbers Parameter values Indicator for fixed or variable parameters Load resistance See Table III for C- and D-sections

TABLE II

ELEMENTS AND CODE NUMBERS

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

TABLE III

ALLPASS SECTIONS

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

TABLE IV - THE ARRANGEMENT OF THE DATA DECK

Condition	Number of cards	Parameters	Type	Format
-	1	MM	INTEGER	I10
MM>0	As many as required	IC(I),I=1,MM	INTEGER	8I10
MM>0	As many as required	AA(I),I=1,NE	REAL	5E16.8
MM>0	As many as required	IC(MM+I),I=1,NE	INTEGER	8I10
-	1	NC,ND	INTEGER	2I10
NC≠0 & ND≠0	1	KVR	INTEGER	I10
NC≠0 & ND≠0	As many as required	AB(I),I=1,J	REAL	5E16.8
-	1	R	REAL	E16.8
-	1	NINTD,NINTS	INTEGER	2I10
NINTD≠0	As many as required by NINTD	XX(1,I),XX(2,I),FUN(I),WT(I),XX(3,I), IOBJ(I),NUMB(I)	5 REALS and 2 INTEGERS	4E16.8,F6.0, 2I5
NINTS≠0	As many as required by NINTS	XX(1,I),FUN(I),WT(I),XX(3,I), IOBJ(I)	4 REALS and 1 INTEGER	4E16.8, I6
-	1	FM,WC	REAL	2E16.8
-	1	MET,MAX,IPRINT	INTEGER	3I10
MET=1	As many as required	EPS(I),I=1,K	REAL	5E16.8
MET=2	1	EPS1	REAL	E16.8
-	1	EST,DIF,KSI	REAL	3E16.8
-	1	ITER	INTEGER	I10
-	As many as required	IPA(I),I=1,ITER	INTEGER	8I10

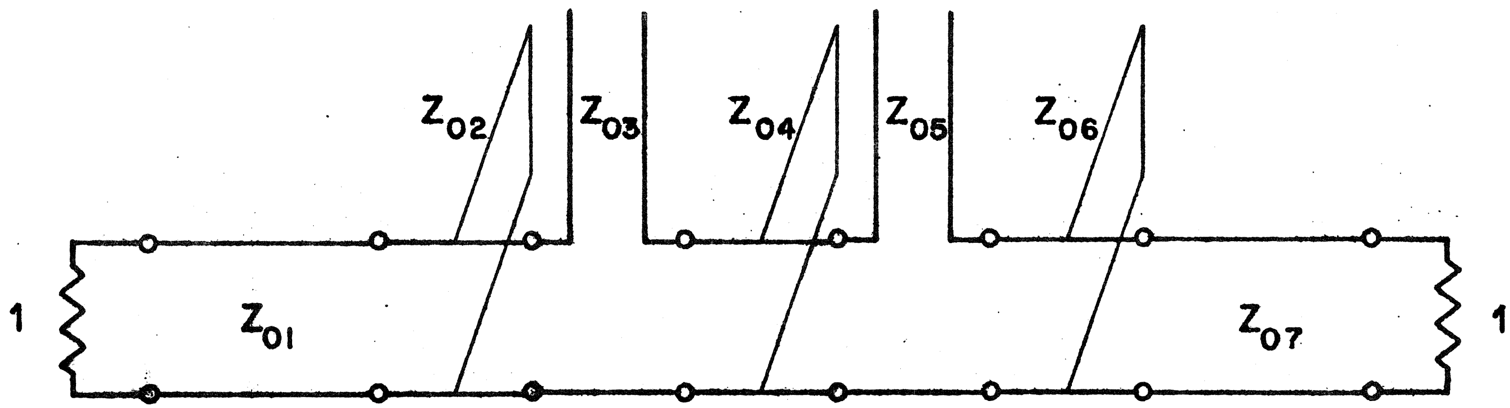


Fig. 1. Seven-section band-pass filter example.

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

               M A I N   P R O G R A M

      DIMENSION A1(14),A2(14),A3(1),A4(3),A5(3),A6(3,3),A7(25),A8(25),
     1 A9(25),A10(25),A11(25),A12(25),A13(7),A14(7),A15(7),A16(7),
     2 A17(175),A18(49),IA1(21),IA2(3),IA3(14)
      CALL CANOPT (A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12,A13,A14,A15,
     + A16,A17,A18,IA1,IA2,IA3)
      CALL EXIT
      END
```

```
                         CD TOT    0011
```

```
        7
       15         14          13          14          13          14          15
       1.             0.63                1.             0.33                1.
  1.27                 1.         0.26                       1.               1.27
  1.                  0.33                1.                  0.63
        0          1           0           1           0          1           0          1
        0          1           0           1           0          1
        0          0
      1.
        1          2
       1087.5         3262.5            0.1                     1.               1.      2     20
  600.                50.              1.                      -1.                2
  3750.               50.              1.                      -1.                2
   2175.
        1        200           20
  1.E         -6      1.E         -6      1.E         -6      1.E        -6       1.E       -6
  1.E         -6      1.E         -6
     -0.1
        2
        2       1000
```

```
                         CD TOT    0020
```

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

INPUT DATA

NUMBER OF ELEMENTS 7

THE CALCULATED NUMBER OF PARAMETERS 14

CODE NUMBER	PARAMETER NUMBER	PARAMETER VALUE	PARAMETER CONDITION
15	1	1.00000000E+00	FIXED
15	2	6.30000000E-01	VARIABLE
14	3	1.00000000E+00	FIXED
14	4	3.30000000E-01	VARIABLE
13	5	1.00000000E+00	FIXED
13	6	1.27000000E+00	VARIABLE
14	7	1.00000000E+00	FIXED
14	8	2.60000000E-01	VARIABLE
13	9	1.00000000E+00	FIXED
13	10	1.27000000E+00	VARIABLE
14	11	1.00000000E+00	FIXED
14	12	3.30000000E-01	VARIABLE
15	13	1.00000000E+00	FIXED
15	14	6.30000000E-01	VARIABLE

NUMBER OF C SECTIONS 0

NUMBER OF D SECTIONS 0

LOAD RESISTANCE 1.00000000E+00

NUMBER OF FREQUENCY INTERVALS 1

NUMBER OF FREQUENCY POINTS 2

LOWER FREQUENCY	UPPER FREQUENCY	NUMBER OF SUBINTERVALS	SPECIFICATION		TYPE	WEIGHTING FACTOR
1.08750000E+03	3.26250000E+03	20	1.00000000E-01	INSERTION LOSS	UPPER	1.00000000E+00

FREQUENCY	SPECIFICATION		TYPE	WEIGHTING FACTOR
6.00000000E+02	5.00000000E+01	INSERTION LOSS	LOWER	1.00000000E+00
3.75000000E+03	5.00000000E+01	INSERTION LOSS	LOWER	1.00000000E+00

THE CALCULATED TOTAL NUMBER OF INTERVALS 3

CENTER FREQUENCY 2.17500000E+03

CUT-OFF FREQUENCY -0.

FLETCHER METHOD WILL BE USED

TEST QUANTITIES TO BE USED IN FLETCHER METHOD
1.00000000E-06
1.00000000E-06
1.00000000E-06
1.00000000E-06
1.00000000E-06
1.00000000E-06
1.00000000E-06

ESTIMATE OF LOWER BOUND ON FUNCTION TO BE MINIMIZED -1.00000000E-01

DIFFERENCE IN THE OBJECTIVE FUNCTION
IN SUCCESSIVE OPTIMIZATIONS -0.

ARTIFICIAL MARGIN -0.

NUMBER OF COMPLETE OPTIMIZATIONS 2

VALUES OF P
2
1000

MAXIMUM NUMBER OF ALLOWABLE ITERATIONS 200

INTERMEDIATE OUTPUT TO BE PRINTED EVERY 20 ITERATIONS

Fig. 3. Printout for the example.

RESPONSE AT THE STARTING POINT

FREQUENCY	INSERTION LOSS
1.08750000E+03	1.35249553E+01
1.19625000E+03	3.77723729E+00
1.30500000E+03	2.45398088E-01
1.41375000E+03	5.11378932E-04
1.52250000E+03	7.81659925E-02
1.63125000E+03	2.54449657E-01
1.74000000E+03	4.43708021E-01
1.84875000E+03	4.96993400E-01
1.95750000E+03	3.52689270E-01
2.06625000E+03	1.16871592E-01
2.17500000E+03	-6.17284002E-14
2.28375000E+03	1.16871592E-01
2.39250000E+03	3.52689270E-01
2.50125000E+03	4.96993400E-01
2.61000000E+03	4.43708021E-01
2.71875000E+03	2.54449657E-01
2.82750000E+03	7.81659925E-02
2.93625000E+03	5.11378932E-04
3.04500000E+03	2.45398088E-01
3.15375000E+03	3.77723729E+00
3.26250000E+03	1.35249553E+01
6.00000000E+02	5.88821079E+01
3.75000000E+03	5.88821079E+01

Fig. 3. [Continued]

OPTIMIZATION BY FLETCHER METHOD

ITERATION NUMBER	FUNCTION EVALUATIONS	TIME ELAPSED (SECONDS)	OBJECTIVE FUNCTION	VARIABLE VECTOR	GRADIENT VECTOR
0	1	3.41000000E-01	1.97046376E+01	6.30000000E-01	1.12444514E+01
				3.30000000E-01	-3.91212547E+01
				1.27000000E+00	2.12917557E+01
				2.60000000E-01	-1.15888062E+02
				1.27000000E+00	2.12917557E+01
				3.30000000E-01	-3.91212547E+01
				6.30000000E-01	1.12444514E+01
20	27	9.14400000E+00	3.43654871E-01	6.19923771E-01	1.77929528E+00
				3.13756640E-01	-8.70075302E+00
				8.34654473E-01	4.81676407E+00
				2.62674781E-01	-5.07034974E+00
				8.04375173E-01	1.16237093E+00
				3.49312256E-01	-4.57788572E+00
				7.53269462E-01	3.04020868E+00
40	47	1.60310000E+01	-9.59251794E-03	5.98060225E-01	-3.68631053E-02
				2.94395008E-01	-4.66075966E-03
				7.13552857E-01	4.48206834E-02
				2.33524121E-01	-2.63201978E-01
				7.19806526E-01	1.23202499E-02
				3.07862727E-01	6.07176145E-02
				6.14193037E-01	-3.00170446E-02
60	67	2.29740000E+01	-9.86591183E-03	6.07402988E-01	9.25178875E-04
				3.01887138E-01	-1.47570822E-03
				7.19251801E-01	-3.53137280E-04
				2.34548712E-01	2.71734223E-03
				7.19309281E-01	-3.00951949E-04
				3.01950112E-01	-4.82251428E-04
				6.07303700E-01	1.87904959E-04

IEXIT= 1CRITERION FOR OPTIMUM HAS BEEN SATISFIED

OPTIMUM SOLUTION

ITERATION NUMBER	FUNCTION EVALUATIONS	EXECUTION TIME (SECONDS)	OBJECTIVE FUNCTION	VARIABLE VECTOR	GRADIENT VECTOR
67	74	2.54540000E+01	-9.86595387E-03	6.07349080E-01	-8.79707218E-06
				3.01921452E-01	1.59307560E-05
				7.19285626E-01	5.09747766E-06
				2.34547743E-01	-2.99932472E-05
				7.19284754E-01	2.77107985E-06
				3.01920804E-01	7.97388286E-06
				6.07349938E-01	-1.86984012E-06

VALUE OF Q . -2

Fig. 3. [Continued]

FINAL RESPONSE OF THE CIRCUIT

FREQUENCY	INSERTION LOSS
1.08750000E+03	6.17463201E-02
1.19625000E+03	7.03446199E-02
1.30500000E+03	3.41361594E-03
1.41375000E+03	6.79762275E-02
1.52250000E+03	4.95938179E-02
1.63125000E+03	2.42384214E-03
1.74000000E+03	1.80744928E-02
1.84875000E+03	5.90357652E-02
1.95750000E+03	5.99070842E-02
2.06625000E+03	2.26651675E-02
2.17500000E+03	8.51674287E-12
2.28375000E+03	2.26651675E-02
2.39250000E+03	5.99070842E-02
2.50125000E+03	5.90357652E-02
2.61000000E+03	1.80744928E-02
2.71875000E+03	2.42384214E-03
2.82750000E+03	4.95938179E-02
2.93625000E+03	6.79762275E-02
3.04500000E+03	3.41361594E-03
3.15375000E+03	7.03446199E-02
3.26250000E+03	6.17463201E-02
6.00000000E+02	5.00813259E+01
3.75000000E+03	5.00813259E+01

Fig. 3. [Continued]

OPTIMIZATION BY FLETCHER METHOD

ITERATION NUMBER	FUNCTION EVALUATIONS	TIME ELAPSED (SECONDS)	OBJECTIVE FUNCTION	VARIABLE VECTOR	GRADIENT VECTOR
0	1	3.58000000E-01	-2.96348316E-02	6.07349080E-01	1.45026020E+00
				3.01921452E-01	-3.06112067E+00
				7.19285626E-01	-2.24927701E-01
				2.34547743E-01	5.34792898E+00
				7.19284754E-01	-2.24952824E-01
				3.01920804E-01	-3.06111629E+00
				6.07349938E-01	1.45025937E+00
20	71	2.43940000E+01	-3.45065393E-02	6.07046955E-01	2.43146435E-01
				3.04364419E-01	5.41284048E+00
				7.22751577E-01	-3.47088422E+00
				2.35604612E-01	1.35298967E+01
				7.21277265E-01	-3.47277562E+00
				3.01718167E-01	5.44080315E+00
				6.05880965E-01	2.52467931E-01
40	91	3.13620000E+01	-3.46265192E-02	6.06598455E-01	1.78940279E-02
				3.03152536E-01	-2.69389413E-01
				7.22121246E-01	1.20367588E-01
				2.35607520E-01	-4.62619715E-01
				7.22022075E-01	1.22083140E-01
				3.02962093E-01	-2.71899636E-01
				6.06316253E-01	1.74612187E-02

IEXIT= 1CRITERION FOR OPTIMUM HAS BEEN SATISFIED

OPTIMUM SOLUTION

ITERATION NUMBER	FUNCTION EVALUATIONS	EXECUTION TIME (SECONDS)	OBJECTIVE FUNCTION	VARIABLE VECTOR	GRADIENT VECTOR
49	100	3.45040000E+01	-3.46268632E-02	6.06458532E-01	1.84695275E-04
				3.03058477E-01	-8.90842022E-05
				7.22074333E-01	-1.34024023E-04
				2.35608560E-01	4.85415313E-04
				7.22074511E-01	-1.31204514E-04
				3.03058564E-01	-9.28834544E-05
				6.06458761E-01	1.86216649E-04

VALUE OF Q -1000

Fig. 3. [Continued]

FINAL RESPONSE OF THE CIRCUIT

FREQUENCY	INSERTION LOSS
1.08750000E+03	6.52816674E-02
1.19625000E+03	6.53110538E-02
1.30500000E+03	3.73619248E-03
1.41375000E+03	6.53085874E-02
1.52250000E+03	4.48375357E-02
1.63125000E+03	1.23804683E-03
1.74000000E+03	2.21519195E-02
1.84875000E+03	6.52929956E-02
1.95750000E+03	6.45114012E-02
2.06625000E+03	2.41661428E-02
2.17500000E+03	4.93827201E-13
2.28375000E+03	2.41661428E-02
2.39250000E+03	6.45114012E-02
2.50125000E+03	6.52929956E-02
2.61000000E+03	2.21519195E-02
2.71875000E+03	1.23804683E-03
2.82750000E+03	4.48375357E-02
2.93625000E+03	6.53085874E-02
3.04500000E+03	3.73619248E-03
3.15375000E+03	6.53110538E-02
3.26250000E+03	6.52816674E-02
6.00000000E+02	5.00347973E+01
3.75000000E+03	5.00347973E+01

Fig. 3. [Continued]

```
      SUBROUTINE CANOPT (AA,B,AB,FUN,WT,XX,X,X1,FX,ERROR,EHELP,AP,A,G,GR
     1AD,EPS,GGRAD,H,IC,IOBJ,NUMB)
C
C          SUBROUTINE WHICH COORDINATES
C          THE OTHER SUBROUTINES
C
      EXTERNAL OBJECT
      DIMENSION TEXT(11)
      DIMENSION A(1), G(1), EPS(1), AA(1), B(1), AB(1), IC(1), IOBJ(1),
     1H(1), GRAD(1), XX(3,1), X(1), X1(1), ERROR(1), EHELP(1), AP(1), GG
     2RAD(1), FUN(1), WT(1), NUMB(1), FX(1)
      DIMENSION IPA(25)
      COMMON /BLK/ KO,T1,KOUNT,NUMF
      COMMON /BLACK/ MM,NE,RL,NC,ND,KVR,EM,WC
      COMMON /EXT/ APP,PSI,EMAX,N,NINT,IP
      COMMON /S16/ PIE,WCC,WCSQ,WCS1,EMC
      LOGICAL B
      LOGICAL CONV,UNITH
      COMPLEX RL
      DATA F,V/10HFIXED     ,10HVARIABLE  /
      DATA TEXT/10HPARAMETER ,10HREFLECTION,10HINSERTION ,10HGROUP DELA,
     17HVALUE   ,7H COEFF.,7HLOSS   ,7HY      ,6HLOWER ,6HSINGLE,6HUPPER
     2/
      ERR(Z)=ERROX (Z,IINT,FUN,WT,A,N1,GRAD,APP,PSI,XX,1,IC,AA,B,AB)
      UNITH=.TRUE.
      T1=0.
      K=0
      KK=0
      NE=0
      NINT=0
      WRITE (6,51)
      READ (5,71) MM
      WRITE (6,82) MM
      IF (MM.EQ.0) GO TO 10
      READ (5,71) (IC(L),L=1,MM)
      DO 6 I=1,MM
      IF (IC(I).LE.6) 1,2
1     NUMB(NE+1)=IC(I)
      NE=NE+1
2     IF (IC(I).GE.7.AND.IC(I).LE.10) 3,4
3     NUMB(NE+1)=IC(I)
      NUMB(NE+2)=IC(I)
      NUMB(NE+3)=IC(I)
      NE=NE+3
4     IF (IC(I).GE.11.AND.IC(I).LE.15) 5,6
5     NUMB(NE+1)=IC(I)
      NUMB(NE+2)=IC(I)
      NE=NE+2
6     CONTINUE
      WRITE (6,83) NE
      READ (5,72) (AA(I),I=1,NE)
      READ (5,71) (IC(MM+I),I=1,NE)
      WRITE (6,103)
      DO 9 I=1,NE
      B(I)=.TRUE.
      IF (IC(MM+I).EQ.0) B(I)=.FALSE.
      IF (B(I)) 7,8
7     K=K+1
      A(K)=AA(I)
```

```
      WRITE (6,104) NUMB(I),I,AA(I),V
      GO TO 9
8     WRITE (6,104) NUMB(I),I,AA(I),F
9     CONTINUE
10    READ (5,71) NC,ND
      WRITE (6,85) NC
      WRITE (6,86) ND
      IF (NC.EQ.0.AND.ND.EQ.0) GO TO 14
      READ (5,71) KVR
      J=2*ND+NC+1
      READ (5,72) (AB(I),I=1,J)
      WRITE (6,87)
      IF (KVR.EQ.0) GO TO 13
      DO 11 I=1,J
      KK=KK+1
      WRITE (6,84) AB(I),V
      AA(NE+I)=AB(I)
11    CONTINUE
      KK=NE+KK
      NE1=NE+1
      DO 12 I=NE1,KK
      K=K+1
      A(I)=AA(I)
12    CONTINUE
      GO TO 14
13    WRITE (6,84) AB(I),F
14    CONTINUE
      N1=K
      READ (5,72) R
      WRITE (6,88) R
      RL=CMPLX(R,0.)
      READ (5,71) NINTD,NINTS
      WRITE (6,89) NINTD
      WRITE (6,90) NINTS
      IF (NINTD.EQ.0) GO TO 16
      WRITE (6,91)
      DO 15 I=1,NINTD
      NINT=NINT+1
      READ (5,80) XX(1,NINT),XX(2,NINT),FUN(NINT),WT(NINT),XX(3,NINT),IO
     1BJ(NINT),NUMB(NINT)
      IND1=IOBJ(NINT)+1
      IND2=IND1+4
      IND3=XX(3,NINT)+10
      WRITE (6,93) XX(1,NINT),XX(2,NINT),NUMB(NINT),FUN(NINT),TEXT(IND1)
     1,TEXT(IND2),TEXT(IND3),WT(NINT)
      IF (XX(3,NINT).NE.0) GO TO 15
      XX(3,NINT)=1.
      XX(3,NINT+1)=-1.
      XX(1,NINT+1)=XX(1,NINT)
      XX(2,NINT+1)=XX(2,NINT)
      FUN(NINT+1)=FUN(NINT)
      WT(NINT+1)=WT(NINT)
      NUMB(NINT+1)=NUMB(NINT)
      IOBJ(NINT+1)=IOBJ(NINT)
      NINT=NINT+1
15    CONTINUE
16    IF (NINTS.EQ.0) GO TO 18
      WRITE (6,92)
      DO 17 I=1,NINTS
```

```
      NINT=NINT+1
      NUMB(NINT)=0
      READ (5,81) XX(1,NINT),FUN(NINT),WT(NINT),XX(3,NINT),IOBJ(NINT)
      IND1=IOBJ(NINT)+1
      IND2=IND1+4
      IND3=XX(3,NINT)+10
      WRITE (6,94) XX(1,NINT),FUN(NINT),TEXT(IND1),TEXT(IND2),TEXT(IND3)
     1,WT(NINT)
      XX(2,NINT)=XX(1,NINT)
      IF (XX(3,NINT).NE.0) GO TO 17
      XX(3,NINT)=1.
      XX(3,NINT+1)=-1.
      XX(1,NINT+1)=XX(1,NINT)
      XX(2,NINT+1)=XX(2,NINT)
      FUN(NINT+1)=FUN(NINT)
      WT(NINT+1)=WT(NINT)
      NUMB(NINT+1)=0
      IOBJ(NINT+1)=IOBJ(NINT)
      NINT=NINT+1
 17   CONTINUE
 18   WRITE (6,102) NINT
      READ (5,72) FM,WC
      WRITE (6,95) FM
      WRITE (6,96) WC
      READ (5,71) MET,MAX,IPRINT
      IF (MET.EQ.1) GO TO 19
      GO TO 20
 19   WRITE (6,52)
      READ (5,72) (EPS(I),I=1,N1)
      WRITE (6,56)
      WRITE (6,57) (EPS(I),I=1,N1)
 20   CONTINUE
      IF (MET.EQ.2) GO TO 21
      GO TO 22
 21   WRITE (6,53)
      READ (5,72) EPS1
      WRITE (6,58) EPS1
 22   IF (MET.NE.1.AND.MET.NE.2) GO TO 23
      READ (5,72) EST,DIF,PSI
      WRITE (6,59) EST
      WRITE (6,97) DIF
      WRITE (6,98) PSI
      READ (5,71) ITER
      WRITE (6,99) ITER
      RE+D (5,71) (IPA(I),I=1,ITER)
      WRITE (6,100)
      WRITE (6,101) (IPA(I),I=1,ITER)
      WRITE (6,54) MAX
      WRITE (6,55) IPRINT
 23   CONTINUE
      DO 24 I=1,NINT
      IF (IOBJ(I).EQ.0) GO TO 200
      XX(1,I)=XX(1,I)/FM+(IOBJ(I)-1)*10.
      XX(2,I)=XX(2,I)/FM+(IOBJ(I)-1)*10.
      GO TO 24
 200  XX(1,I)=XX(1,I)+30.
      XX(2,I)=XX(1,I)
 24   CONTINUE
      K=0
```

```
      PIE=4.*ATAN(1.)
      WCC=WC/FM
      WCSQ=WCC*WCC
      WCS1=1.-WCSQ
      FMC=FM*0.001
      WRITE (6,78)
      DO 29 J=1,NINT
      IINT=J
      IF (J-1) 27,26,25
25    IF (IOBJ(J)-IOBJ(J-1)) 26,27,26
26    IF (XX(1,J).LE.10.) WRITE (6,75)
      IF (XX(1,J).LE.20.0.AND.XX(1,J).GT.10.) WRITE (6,76)
      IF (XX(1,J).LE.30.0.AND.XX(1,J).GT.20.) WRITE (6,77)
27    L=NUMB(J)+1
      IF (NUMB(J).EQ.0) Z=XX(1,J)
      DO 29 I=1,L
      IF (NUMB(J).GT.0) Z=XX(1,J)+(XX(2,J)-XX(1,J))*(I-1)/NUMB(J)
      K=K+1
      X(K)=Z
      ERROR(K)=ERR(Z)
      EHELP(K)=ERROR(K)*XX(3,J)
      ER=ERROR(K)
      IF (PSI.NE.0.) ER=ER+PSI*XX(3,IINT)
      ERT=(ER/WT(IINT))+FUN(IINT)
      IF (IOBJ(J).EQ.1) FX(K)=Z*FM
      IF (IOBJ(J).EQ.2) FX(K)=(Z-10.)*FM
      IF (IOBJ(J).EQ.3) FX(K)=(Z-20.)*FM
      IF (IOBJ(J).EQ.0) GO TO 28
      WRITE (6,74) FX(K),ERT
28    CONTINUE
29    AP(K)=APP
      EMAX=EHELP(1)
      DO 30 M=2,K
      EMAX=AMAX1(EMAX,EHELP(M))
30    CONTINUE
      IF (ITER.GT.25) GO TO 50
C
C         OPTIMIZATION
C
      DO 49 KK=1,ITER
      IP=IPA(KK)
C
C         PRINTS THE INPUT DATA
C         FOR THE OPTIMIZATION PROCESS
      IF (MET.NE.1.AND.MET.NE.2) GO TO 31
      GO TO 32
31    WRITE (6,60)
      CALL EXIT
C
C
32    GO TO (33,36), MET
33    CONTINUE
      CALL SECOND (T1)
      WRITE (6,61)
      WRITE (6,62)
      IF (IPRINT.EQ.0) GO TO 35
C
C         PRINTS THE INTERMEDIATE RESULTS
C         FOR THE FLETCHER METHOD
```

```
C
      IF (T1.EQ.0.) GO TO 34
      WRITE (6,64)
      GO TO 35
34    WRITE (6,65)
35    CONTINUE
      CALL OPTIM1 (N1,A,F,G,H,UNITH,FST,EPS,MAX,IPRINT,IEXIT,GRAD,NUMB,X
     1X,X,X1,ERROR,FHELP,AP,GGRAD,FUN,WT,IC,AA,B,AB)
      GO TO 39
36    CONTINUE
      CALL SECOND (T1)
      WRITE (6,61)
      WRITE (6,63)
      IF (IPRINT.EQ.0) GO TO 38
C
C          PRINTS THE INTERMEDIATE RESULTS
C          FOR THE FLETCHER-POWELL METHOD
C
      IF (T1.EQ.0.) GO TO 37
      WRITE (6,64)
      GO TO 38
37    WRITE (6,65)
      GO TO 38
38    CONTINUE
      CALL OPTIM2 (OBJECT,N1,A,F,G,FST,EPS1,MAX,IER,H,IPRINT,GRAD,NUMB,X
     1X,X,X1,ERROR,FHELP,AP,GGRAD,FUN,WT,IC,AA,B,AB)
39    CONTINUE
      CALL SECOND (T2)
C
C          PRINTS THE RESULTS
C          FOR THE OPTIMIZATION PROCESS
C
      IF (KO.EQ.0) GO TO 40
      WRITE (6,66)
      GO TO 41
40    WRITE (6,68)
41    CONTINUE
      T=T2-T1
      IF (T1.5Q.0.) GO TO 42
      WRITE (6,67)
      WRITE (6,69) KOUNT,NUMF,T,F,((A(I),G(I)),I=1,N1)
      GO TO 43
42    WRITE (6,65)
      WRITE (6,70) KOUNT,NUMF,F,((A(I),G(I)),I=1,N1)
43    WRITE (6,73) IP
      KQ=0
      WRITE (6,79)
      DO 48 J=1,NINT
      IINT=J
      IF (J-1) 46,45,44
44    IF (IOBJ(J)-IOBJ(J-1)) 45,46,45
45    IF (XX(1,J).LE.10.) WRITE (6,75)
      IF (XX(1,J).LE.20.0.AND.XX(1,J).GT.10.) WRITE (6,76)
      IF (XX(1,J).LE.30.0.AND.XX(1,J).GT.20.) WRITE (6,77)
46    KQ=KQ+1
      KL=KQ+NUMB(J)
      DO 47 I=KQ,KL
      L=I-KQ+1
      ER=ERR(X(I))
```

```
      IF (PSI.NE.0.) ER=ER+PSI*XX(3,IINT)
      ERT=(ER/WT(IINT))+FUN(IINT)
      IF (IOBJ(J).EQ.0) GO TO 47
      WRITE (6,74) FX(I),ERT
47    CONTINUE
      KO=KL
48    CONTINUE
49    CONTINUE
      RETURN
50    WRITE (6,105)
      CALL EXIT
C
C
C
51    FORMAT (1H1,10HINPUT DATA,/,1X,10(1H-),//)
52    FORMAT (1X,28HFLETCHER METHOD WILL BE USED//)
53    FORMAT (1X,35HFLETCHER-POWELL METHOD WILL BE USED//)
54    FORMAT (///1H0,38HMAXIMUM NUMBER OF ALLOWABLE ITERATIONS,18X,I5//)
55    FORMAT (1H0,39HINTERMEDIATE OUTPUT TO BE PRINTED EVERY,I8,5X,10HIT
     1ERATIONS)
56    FORMAT (1H0,/,1X,45HTEST QUANTITIES TO BE USED IN FLETCHER METHOD)
57    FORMAT (1H0,51X,E16.8)
58    FORMAT (1H0,/,1X,50HTEST QUANTITY TO BE USED IN FLETCHER-POWELL ME
     1THOD,1X,E16.8)
59    FORMAT (//1X,51HESTIMATE OF LOWER BOUND ON FUNCTION TO BE MINIMIZE
     1D,E16.8//)
60    FORMAT (1H0,49HNONE OF THE OPTIMIZATION METHODS HAVE BEEN CALLED,/
     1,1X,29HPLEASE CHECK THE VALUE OF MET,/,1X,9HREMAINDER,/,1X,40HMET=
     21    FLETCHER METHOD WOULD BE CALLED,/,1X,47HMET=2    FLETCHER-POW
     3ELL METHOD WOULD BE CALLED)
61    FORMAT (1H1)
62    FORMAT (1H0,31HOPTIMIZATION BY FLETCHER METHOD,/,1H0,31(1H-))
63    FORMAT (1H0,38HOPTIMIZATION BY FLETCHER-POWELL METHOD,/,1H0,38(1H-
     1))
64    FORMAT (1H0,9HITERATION,2X,8HFUNCTION,6X,12HTIME ELAPSED,8X,9HOBJE
     1CTIVE,16X,16HVARIABLE VECTOR ,13X,16HGRADIENT VECTOR ,/1X,6HNUMBER
     2,5X,11HEVALUATIONS,3X,9H(SECONDS),11X,8HFUNCTION,/)
65    FORMAT (1H0,9HITERATION,2X,8HFUNCTION,8X,9HOBJECTIVE,16X,16HVARIAB
     1LE VECTOR ,13X,16HGRADIENT VECTOR ,/1X,6HNUMBER,5X,11HEVALUATIONS,
     25X,8HFUNCTION,/)
66    FORMAT (////1X,16HOPTIMUM SOLUTION,/,1X,16(1H-)/)
67    FORMAT (1H0,9HITERATION,2X,8HFUNCTION,6X,14HEXECUTION TIME,6X,9HOB
     1JECTIVE,16X,16HVARIABLE VECTOR ,13X,16HGRADIENT VECTOR ,/1X,6HNUMB
     2ER,5X,11HEVALUATIONS,3X,9H(SECONDS),11X,8HFUNCTION,/)
68    FORMAT (////1X,25HRESULTS AT LAST ITERATION/,1X,25(1H-))
69    FORMAT (1H0,I5,7X,I5,5X,E16.8,3X,E16.8,12X,95(E16.8,13X,E16.8,/,70
     1X))
70    FORMAT (1H0,I5,7X,I5,8X,E16.8,7X,95(E16.8,13X,E16.8,/,49X))
71    FORMAT (8I10)
72    FORMAT (5E16.8)
73    FORMAT (5(/),1X,10HVALUE OF Q,I12)
74    FORMAT (15X,E16.8,10X,E16.8)
75    FORMAT (/20X,9HFREQUENCY,14X,17HREFLECTION COEFF.)
76    FORMAT (/20X,9HFREQUENCY,14X,14HINSERTION LOSS)
77    FORMAT (/20X,9HFREQUENCY,14X,11HGROUP DELAY)
78    FORMAT (1H1,23X,30HRESPONSE AT THE STARTING POINT,/23X,31(1H-))
79    FORMAT (1H1,23X,29HFINAL RESPONSE OF THE CIRCUIT,/23X,30(1H-))
80    FORMAT (4F16.8,F6.0,2I5)
81    FORMAT (4F16.8,I6)
```

```
82    FORMAT (1X,18HNUMBER OF ELEMENTS,40X,I3//)
83    FORMAT (//1X,35HTHE CALCULATED NUMBER OF PARAMETERS,20X,I5///)
84    FORMAT (52X,E16.8,3X,A10)
85    FORMAT (///1X,20HNUMBER OF C SECTIONS,36X,I5//)
86    FORMAT (1X,20HNUMBER OF D SECTIONS,36X,I5//)
87    FORMAT (1X,37HPARAMETERS OF THE C AND/OR D SECTIONS)
88    FORMAT (/1X,15HLOAD RESISTANCE,36X,E16.8//)
89    FORMAT (1X,29HNUMBER OF FREQUENCY INTERVALS,27X,I5//)
90    FORMAT (1X,26HNUMBER OF FREQUENCY POINTS,30X,I5//)
91    FORMAT (/6X,5HLOWER,13X,5HUPPER,15X,9HNUMBER OF,20X,13HSPECIFICATI
     1ON,14X,4HTYPE,10X,9HWEIGHTING,/4X,9HFREQUENCY,9X,9HFREQUENCY,11X,1
     22HSUBINTERVALS,61X,6HFACTOR/)
92    FORMAT (///4X,9HFREQUENCY,24X,13HSPECIFICATION,16X,4HTYPE,7X,16HWE
     1IGHTING FACTOR/)
93    FORMAT (2(E16.8,4X),I10,10X,E16.8,3X,A10,A7,4X,A6,4X,E16.8)
94    FORMAT (E16.8,10X,E16.8,3X,A10,A7,4X,A6,4X,E16.8)
95    FORMAT (/1X,16HCENTER FREQUENCY,35X,E16.8//)
96    FORMAT (1X,17HCUT-OFF FREQUENCY,34X,E16.8//)
97    FORMAT (1X,36HDIFFERENCE IN THE OBJECTIVE FUNCTION,/1X,27HIN SUCCE
     1SSIVE OPTIMIZATIONS,24X,E16.8//)
98    FORMAT (1X,17HARTIFICIAL MARGIN,34X,E16.8//)
99    FORMAT (1X,32HNUMBER OF COMPLETE OPTIMIZATIONS,24X,I5//)
100   FORMAT (1X,11HVALUES OF P)
101   FORMAT (57X,I5)
102   FORMAT (////1X,40HTHE CALCULATED TOTAL NUMBER OF INTERVALS,16X,I5/
     1/)
103   FORMAT (6X,4HCODE,12X,9HPARAMETER,13X,9HPARAMETER,13X,9HPARAMETER,
     1/5X,6HNUMBER,12X,6HNUMBER,17X,5HVALUE,15X,9HCONDITION/)
104   FORMAT (7X,I2,15X,I3,13X,E16.8,11X,A10)
105   FORMAT (/////5X,55HNUMBER OF COMPLETE OPTIMIZATIONS ITER EXCEEDS V
     1ALUE 25.,//5X,65HPLEASE CHANGE THE LENGTH OF ARRAY IPA(ITER) IN SU
     2BROUTINE CANOPT.)
      END
```

CD TOT 0387

```
      FUNCTION ERROX  (Z,IINT,FUNCS,W,A,N1,GRAD,APP,PSI,XX,IPOINT,IC,AA,
     1B,AB)
C
C         FUNCTION SUBROUTINE WHICH CALCULATES
C         UPPER AND LOWER WEIGHTED ERROR FUNCTION
C
      DIMENSION A(1), GRAD(1), XX(3,1), FUNCS(1), W(1), AA(1), B(1), AB(
     11), IC(1)
      IF (IPO9NT) 1,2,1
1     CONTINUE
      CALL APPROX (Z,N1,A,APP,GRAD,IC,AA,B,AB)
2     CONTINUE
      IF (PSI) 3,4,3
3     ERROX =(APP-FUNCS(IINT))*W(IINT)-PSI*XX(3,IINT)
      RETURN
4     ERROX =(APP-FUNCS(IINT))*W(IINT)
      RETURN
      END
```

CD TOT   0018

```
      SUBROUTINE OBJECT (N1,A,OBJ,G,GRAD,NUMB,XX,X,X1,ERROR,EHELP,AP,GGR
     1AD,FUN,W,IC,AA,B,AB)
C
C         SUBROUTINE WHICH COMPUTES THE OBJECTIVE FUNCTION
C         AND ITS GRADIENTS W.R.T. THE VARIABLE PARAMETERS
C         IN THE LEAST P-TH SENSE
C
      DIMENSION A(1), GRAD(1), NUMB(1), XX(3,1), X(1), X1(1), ERROR(1),
     1EHELP(1), AP(1), GGRAD(1), G(1), FUN(1), W(1), IC(1), AA(1), B(1),
     2 AB(1)
      COMMON /EXT/ APP,PSI,EMAX,N,NINT,IP
      ERR(Z)=ERROX (Z,IINT,FUN,W,A,N1,GRAD,APP,PSI,XX,IPOINT,IC,AA,B,AB)
      OBJP=0.
      GRADP=0.
      DO 1 K=1,N1
      G(K)=0.
    1 CONTINUE
      INDIC=1
      IPOINT=1
      K=0
      KL=0
      KG=0
      DO 8 J=1,NINT
      IINT=J
      IF (J.EQ.1) GO TO 2
      KL=KL+L
    2 L=NUMB(J)+1
      DO 7 I=1,L
      K=K+1
      IF (J.EQ.1) GO TO 5
      DO 4 KK=1,KL
      IF (ABS(X(K)-X(KK)).GT.10E-6) GO TO 4
      AP(K)=AP(KK)
      APP=AP(K)
      KGK=(K-1)*N1+1
      KGKK=(KK-1)*N1+1
      DO 3 KN1=1,N1
      GGRAD(KGK)=GGRAD(KGKK)
      GRAD(KN1)=GGRAD(KGKK)
      KGK=KGK+1
      KGKK=KGKK+1
    3 CONTINUE
      IPOINT=0
      GO TO 5
    4 CONTINUE
    5 ERROR(K)=ERR(X(K))
      EHELP(K)=ERROR(K)*XX(3,J)
      DO 6 KN1=1,N1
      KG=KG+1
      GGRAD(KG)=GRAD(KN1)
    6 CONTINUE
      IF (IPOINT.NE.0) AP(K)=APP
      IPOINT=1
    7 CONTINUE
    8 CONTINUE
      EMAX=EHELP(1)
      DO 9 M=2,K
      EMAX=AMAX1(EMAX,EHELP(M))
    9 CONTINUE
```

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

CD TOT 0105

```
      SUBROUTINE OPTIM1 (N,X,F,G,H,UNITH,FEST,EPS,MAXFN,IPRINT,IEXIT,GRA
     1D,NUMB,XX,XP,X1,ERROR,EHELP,AP,GGRAD,FUN,WT,IC,AA,B,AB)
      DIMENSION X(1), G(1), H(1), EPS(1), GRAD(1), NUMB(1), XX(3,1), XP(
     11), X1(1), ERROR(1), EHELP(1), AP(1), GGRAD(1), FUN(1), WT(1), AA(
     21), B(1), AB(1), IC(1)
      LOGICAL CONV,UNITH
      COMMON /BLK/ KO,T1,ITN,NFNS
      CALL SECOND (T3)
      KO=0
      CALL OBJECT (N,X,F,G,GRAD,NUMB,XX,XP,X1,ERROR,EHELP,AP,GGRAD,FUN,W
     1T,IC,AA,B,AB)
      IF (F.LT.FEST) GO TO 24
      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 8
      IF (MOD(ITN,IPRINT).NE.0) GO TO 8
      CALL SECOND (T4)
      TIME=T4-T3
      IF (T1.EQ.0.) GO TO 7
      WRITE (6,31) ITN,NFNS,TIME,F,((X(I),G(I)),I=1,N)
      GO TO 8
7     WRITE (6,32) ITN,NFNS,F,((X(I),G(I)),I=1,N)
8     IEXIT=1
      IF (CONV) GO TO 25
      IEXIT=2
      IF (GDX.GE.0.) GO TO 25
      Z=1.
      IF (ITN.LT.N.AND.UNITH) Z=STEP
```

```
      W=2.*(FEST-F)/GDX
      IF (W.LT.Z) Z=W
      STEP=Z
9     GDX=GDX*Z
      DO 10 I=1,N
      H(IDX+I)=H(IDX+I)*Z
      X(I)=X(I)+H(IDX+I)
10    CONTINUE
      CALL OBJECT (N,X,FP,H,GRAD,NUMB,XX,XP,X1,ERROR,FHELP,AP,GGRAD,FUN,
     1WT,IC,AA,B,AB)
      IF (FP.LT.FEST) GO TO 24
      NFNS=NFNS+1
      IFXIT=3
      IF (IT5.EQ.MAXFN) GO TO 25
      GPDX=0.
      DO 11 I=1,N
      H(IDG+I)=H(I)-G(I)
      GPDX=GPDX+H(I)*H(IDX+I)
11    CONTINUE
      DGDX=GPDX-GDX
      IF (F.GT.FP-.0001*GDX) GO TO 13
      IFXIT=4
      IF (GPDX.LT.0..AND.ITN.GT.N) GO TO 25
      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 12 I=1,N
      X(I)=X(I)-H(IDX+I)
12    CONTINUE
      GO TO 15
13    F=FP
      DO 14 I=1,N
      G(I)=H(I)
14    CONTINUE
      IF (DGDX.GT.0.) GO TO 16
      GDX=GPDX
      Z=4.
15    STEP=Z*STEP
      GO TO 9
16    IF (GPDX.LT.0.5*GDX) STEP=2.*STEP
      DGHDG=0.
      DO 20 I=1,N
      Z=0.
      IJ=IH+I
      IF (I.EQ.1) GO TO 18
      II=I-1
      DO 17 J=1,II
      Z=Z+H(IJ)*H(IDG+J)
      IJ=IJ+N-J
17    CONTINUE
18    DO 19 J=I,N
      Z=Z+H(IJ)*H(IDG+J)
      IJ=IJ+1
19    CONTINUE
      DGHDG=DGHDG+Z*H(IDG+I)
      H(I)=Z
20    CONTINUE
      IF (DGHDG.LT.0.0) DGHDG=DGDX*0.01
```

```
      IF (DGDX.LT.DGHDG) GO TO 22
      W=1.0+DGHDG/DGDX
      DO 21 I=1,N
      H(IDX+I)=W*H(IDX+I)-H(I)
21    CONTINUE
      DGDX=DGDX+DGHDG
      DGHDG=DGDX
22    IJ=IH
      DO 23 I=1,N
      W=H(IDX+I)/DGDX
      Z=H(I)/DGHDG
      DO 23 J=I,N
      IJ=IJ+1
23    H(IJ)=H(IJ)+W*H(IDX+J)-Z*H(J)
      ITN=ITN+1
      GO TO 2
24    IEXIT=5
25    IF (IEXIT.EQ.1) KO=1
      IF (IPRINT.EQ.0) RETURN
      GO TO (26,27,28,27,29), IEXIT
26    WRITE (6,33) IEXIT
      GO TO 30
27    WRITE (6,34) IEXIT
      GO TO 30
28    WRITE (6,35) IEXIT
      GO TO 30
29    WRITE (6,36) IEXIT
30    CONTINUE
      RETURN
C
C
C
31    FORMAT (1HO,I5,7X,I5,5X,F16.8,3X,F16.8,12X,95(F16.8,13X,F16.8,/,70
     1X))
32    FORMAT (1HO,I5,7X,I5,8X,F16.8,7X,95(F16.8,13X,F16.8,/,49X))
33    FORMAT (/,1HO,6HIEXIT=,I2,40HCRITERION FOR OPTIMUM HAS BEEN SATISF
     1IED)
34    FORMAT (/,1HO,6HIEXIT=,I2,43HEITHER OF THE FOLLOWING THINGS HAS HA
     1PPENED,/,9X,26H1. EPS CHOSEN IS TOO SMALL,/,9X,25H2. MATRIX H GOES
     2 SINGULAR)
35    FORMAT (/,1HO,6HIEXIT=,I2,56HMAXIMUM NUMBER OF ALLOWABLE ITERATION
     1S HAS BEEN EXCEEDED)
36    FORMAT (/,1HO,6HIEXIT=,I2,61HFUNCTION VALUE LESS THAN MINIMUM ESTI
     1MATED HAS BEEN DETECTED )
      END
```

CD TOT 0163

```
      SUBROUTINE OPTIM2 (FUNCT,N,X,F,G,EST,EPS,LIMIT,IER,H,IPRINT,GRAD,N
     1UMB,XX,XP,X1,ERROR,EHELP,AP,GGRAD,FUN,WT,IC,AA,B,AB)
      COMMON /BLK/ KO,T1,KOUNT,NUMF
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), GGRAD(1), FUN(1), WT(1), IC(1), AA(1
     2), B(1), AB(1)
C
C        COMPUTE FUNCTION VALUE AND GRADIENT VECTOR FOR INITIAL ARGUMENT
      KO=0
      CALL SECOND (T3)
      CALL OBJECT (N,X,F,G,GRAD,NUMB,XX,XP,X1,ERROR,EHELP,AP,GGRAD,FUN,W
     1T,IC,AA,B,AB)
      KOUNT=0
      NUMF=1
      CALL SECOND (T4)
      TIME=T4-T3
      IF (IPRINT.EQ.0) GO TO 2
      IF (T1.EQ.0.) GO TO 1
      WRITE (6,71) KOUNT,NUMF,TIME,F,((X(I),G(I)),I=1,N)
      GO TO 2
1     WRITE (6,72) KOUNT,NUMF,F,((X(I),G(I)),I=1,N)
2     CONTINUE
C
C        RESET ITERATION COUNTER AND GENERATE IDENTITY MATRIX
      IER=0
      KK=0
      N2=N+N
      N3=N2+N
      N31=N3+1
3     K=N31
      DO 6 J=1,N
      H(K)=1.
      NJ=N-J
      IF (NJ) 7,7,4
4     DO 5 L=1,NJ
      KL=K+L
      H(KL)=0.
5     CONTINUE
      K=KL+1
6     CONTINUE
C
C        START ITERATION LOOP
7     IF (KOUNT.EQ.0) GO TO 9
      IF (KK.NE.IPRINT) GO TO 9
      KK=0
      CALL SECOND (T4)
      TIME=T4-T3
      IF (T1.EQ.0.) GO TO 8
      WRITE (6,71) KOUNT,NUMF,TIME,F,((X(I),G(I)),I=1,N)
      GO TO 9
8     WRITE (6,72) KOUNT,NUMF,F,((X(I),G(I)),I=1,N)
9     CONTINUE
      KOUNT=KOUNT+1
      KK=KK+1
C
```

```
C        SAVE FUNCTION VALUE, ARGUMENT VECTOR AND GRADIENT VECTOR
      OLDF=F
      DO 13 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 12 L=1,N
      T=T-G(L)*H(K)
      IF (L-J) 10,11,11
10    K=K+N-L
      GO TO 12
11    K=K+1
12    CONTINUE
      H(J)=T
13    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 14 J=1,N
      HNRM=HNRM+ABS(H(J))
      GNRM=GNRM+ABS(G(J))
      DY=DY+H(J)*G(J)
14    CONTINUE
C
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DIRECTIONAL
C        DERIVATIVE APPEARS TO BE POSITIVE OR ZERO.
      IF (DY) 15,60,60
C
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DIRECTION
C        VECTOR H IS SMALL COMPARED TO GRADIENT VECTOR G.
15    IF (HNRM/GNRM-EPS) 60,60,16
C
C        SEARCH MINIMUM ALONG DIRECTION H
C
C        SEARCH ALONG H FOR POSITIVE DIRECTIONAL DERIVATIVE
16    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) 19,19,17
17    IF (ALFA-AMBDA) 18,19,19
18    AMBDA=ALFA
19    ALFA=0.
C
C        SAVE FUNCTION AND DERIVATIVE VALUES FOR OLD ARGUMENT
20    FX=FY
      DX=DY
```

```
C
C        STEP ARGUMENT ALONG H
      DO 21 I=1,N
      X(I)=X(I)+AMBDA*H(I)
21    CONTINUE
C
C        COMPUTE FUNCTION VALUE AND GRADIENT FOR NEW ARGUMENT
      CALL OBJECT (N,X,F,G,GRAD,NUMB,XX,XP,X1,ERROR,EHELP,AP,GGRAD,FUN,W
     1T,IC,AA,B,AB)
      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 22 I=1,N
      DY=DY+G(I)*H(I)
22    CONTINUE
      IF (DY) 23,43,26
C
C        TERMINATE SEARCH ALSO IF THE FUNCTION VALUE INDICATES THAT
C        A MINIMUM HAS BEEN PASSED
23    IF (FY-FX) 24,26,26
C
C        REPEAT SEARCH AND DOUBLE STEPSIZE FOR FURTHER SEARCHES
24    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) 20,20,25
C
C        LINEAR SEARCH TECHNIQUE INDICATES THAT NO MINIMUM EXISTS
25    IER=2
      GO TO 65
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
26    T=0.
27    IF (AMBDA) 28,43,28
28    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) 60,29,29
29    W=ALFA*SQRT(DALFA)
      ALFA=DY-DX+W+W
      IF (ALFA) 30,31,30
30    ALFA=(DY-Z+W)/ALFA
      GO TO 32
31    ALFA=(Z+DY-W)/(Z+DX+Z+DY)
32    ALFA=ALFA*AMBDA
      DO 33 I=1,N
      X(I)=X(I)+(T-ALFA)*H(I)
33    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 OBJECT (N,X,F,G,GRAD,NUMB,XX,XP,X1,ERROR,EHELP,AP,GGRAD,FUN,W
     1T,IC,AA,B,AB)
      IF (F-FX) 34,34,35
   34 IF (F-FY) 43,43,35
   35 DALFA=0.
      DO 36 I=1,N
      DALFA=DALFA+G(I)*H(I)
   36 CONTINUE
      IF (DALFA) 37,40,40
   37 IF (F-FX) 39,38,40
   38 IF (DX-DALFA) 39,43,39
   39 FX=F
      DX=DALFA
      T=ALFA
      AMBDA=ALFA
      GO TO 27
   40 IF (FY-F) 42,41,42
   41 IF (DY-DALFA) 42,43,42
   42 FY=F
      DY=DALFA
      AMBDA=AMBDA-ALFA
      GO TO 26
C
C         TERMINATE, IF FUNCTION HAS NOT DECREASED DURING LAST ITERATION
   43 IF (OLDF-F+EPS) 60,44,44
C
C         COMPUTE DIFFERENCE VECTORS OF ARGUMENT AND GRADIENT FROM
C         TWO 3ONSECUTIVE ITERATIONS
   44 DO 45 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)
   45 CONTINUE
C
C         TEST LENGTH OF ARGUMENT DIFFERENCE VECTOR AND DIRECTION VECTOR
C         BOTH ARE LESS THAN  EPS
      IER=0
      IF (KOUNT-N) 49,46,46
   46 T=0.
      DO 47 J=1,N
      K=N+J
      W=H(K)
      K=K+N
      T=T+ABS(H(K))
   47 CONTINUE
      IF (HNRM-EPS) 48,48,49
   48 IF (T-EPS) 65,65,49
C
C         TERMINATE, IF NUMBER OF ITERATIONS WOULD EXCEED  LIMIT
   49 IF (KOUNT-LIMIT) 50,57,57
C
C         PREPARE UPDATING OF MATRIX H
```

```
50    ALFA=0.
      Z=0.
      DO 54 J=1,N
      K=J+N3
      W=0.
      DO 53 L=1,N
      KL=N+L
      W=W+H(KL)*H(K)
      IF (L-J) 51,52,52
51    K=K+N-L
      GO TO 53
52    K=K+1
53    CONTINUE
      K=N+J
      KN=K+N
      Z=Z+H(K)*H(KN)
      ALFA=ALFA+W*H(K)
      H(J)=W
54    CONTINUE
C
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF RESULTS
C        ARE NOT SATISFACTORY
      IF (Z*ALFA) 55,3,55
C
C        UPDATE MATRIX H
55    K=N31
      DO 56 L=1,N
      KL=N2+L
      DO 56 J=L,N
      NJ=N2+J
      H(K)=H(K)+H(KL)*H(NJ)/Z-H(L)*H(J)/ALFA
56    K=K+1
      GO TO 7
C        END OF ITERATION LOOP
C
C        NO CONVERGENCE AFTER  LIMIT  ITERATIONS
57    IER=1
      IF (KK.NE.IPRINT) GO TO 59
      IF (T1.EQ.0.) GO TO 58
      WRITE (6,71) KOUNT,NUMF,TIME,F,((X(I),G(I)),I=1,N)
      GO TO 59
58    WRITE (6,72) KOUNT,NUMF,F,((X(I),G(I)),I=1,N)
59    CONTINUE
      GO TO 65
C
C        RESTORE OLD VALUES OF FUNCTION AND ARGUMENTS
60    DO 61 J=1,N
      K=N2+J
      X(J)=H(K)
61    CONTINUE
      CALL OBJECT (N,X,F,G,GRAD,NUMB,XX,XP,X1,ERROR,EHELP,AP,GGRAD,FUN,W
     1T,IC,AA,B,AB)
      NUMF=NUMF+1
C
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DERIVATIVE
C        FAILS TO BE SUFFICIENTLY SMALL
      IF (GNRM-EPS) 64,64,62
C
C        TEST FOR REPEATED FAILURE OF ITERATION
```

```
62    IF (IER) 65,63,63
63    IER=-1
      GO TO 3
64    IER=0
65    II=IER+2
      IF (II.EQ.2) KO=1
      IF (IPRINT.EQ.0) RETURN
      GO TO (66,67,68,69), II
66    WRITE (6,73) IER
      GO TO 70
67    WRITE (6,74) IER
      GO TO 70
68    WRITE (6,75) IER
      GO TO 70
69    WRITE (6,76) IER
70    RETURN
C
C
C
71    FORMAT (1H0,I5,7X,I5,5X,E16.8,3X,E16.8,12X,95(E16.8,13X,E16.8,/,70
     1X))
72    FORMAT (1H0,I5,7X,I5,8X,E16.8,7X,95(E16.8,13X,E16.8,/,49X))
73    FORMAT (1H0,4HIER=,I2,32H ERROR IN GRADIENTS CALCULATIONS)
74    FORMAT (1H0,4HIER=,I2,41H CRITERION FOR OPTIMUM HAS BEEN SATISFIED
     1)
75    FORMAT (1H0,4HIER=,I2,57H MAXIMUM NUMBER OF ALLOWABLE ITERATIONS H
     1AS BEEN EXCEEDED)
76    FORMAT (1H0,4HIER=,I2,83H CHANGE IN ARGUMENTS GETS TOO LARGE, LINE
     1AR SEARCH INDICATES THAT NO MINIMUM EXISTS)
      END
```

CD TOT 0325

```
      SUBROUTINE APPROX (OMEGN,N1,X,APP,GRAD,IC,A,B,AB)
      DIMENSION X(1), GRAD(1), IC(1), A(1), B(1), AB(1)
      DIMENSION AD(50), ADJJ(50), G(50)
      COMPLEX AD,ADJJ,CONRHO,RHO,I,IHAT,V,VHAT,INEW,VNEW,IOLD,VOLD,G,V1,
     1V2,IHAT1,IHAT2,PSIL,RL,GL,GC,GLLTEL,GLLTZO,GSCTEL,GSCTZO,GOCTEL,GO
     2CTZO,GLCPBP,GLCPQ,GLCPOR,GLCSQ,GLCSOR,GLCSXP
      COMPLEX POLD,PHASE
      COMMON /BLACK/ M,NE,RL,NC,ND,KVR,FM,WC
      COMMON /S16/ PIE,WCC,WCSQ,WCS1,FMC
      LOGICAL B
C
C     M IS THE TOTAL NUMBER OF ELEMENTS IN THE CIRCUIT
C      NE IS THE TOTA3 NUMBER OF ELEMENTS IN THE CIRCUIT
C      A ARRAY CONTAINS PARAMETER VALUES
C     B ARRAY CONTAINS LOGICAL VARIABLES
C     IC ARRAY CONTAINS CODE NUMBER GIVING ORDER IN WHICH BLOCKS ARE CON
C     NECTED
C
C     FOLLOWING FUNCTION STATEMENTS DEFINE SENSITIVITIES
C
      THETA(EL)=PIE/2.*OMEGA*EL
      GC(OMEGA,V,VHAT)=-CMPLX(0.,OMEGA)*V*VHAT
      GL(OMEGA,I,IHAT)=CMPLX(0.,OMEGA)*I*IHAT
      GLLTZO(V1,V2,IHAT1,IHAT2,ZO)=(V1*IHAT1-V2*IHAT2)/ZO
      GLLTEL(OMEGA,EL,V1,V2,IHAT1,IHAT2)=PIE/2.*OMEGA/SIN(THETA(EL))*(V1
     1*IHAT2-V2*IHAT1)
      GSCTEL(OMEGA,EL,ZO,I,IHAT)=PIE/2.*(1./COS(THETA(EL)))**2*CMPLX(0.,
     1ZO*OMEGA)*I*IHAT
      GSCTZO(EL,I,IHAT)=CMPLX(0.,SIN(THETA(EL))/COS(THETA(EL)))*I*IHAT
      GOCTEL(ZO,EL,OMEGA,I,IHAT)=CMPLX(0.,ZO*THETA(EL)/EL*(1./SIN(THETA(
     1EL)))**2)*I*IHAT
      GOCTZO(OMEGA,EL,I,IHAT)=CMPLX(0.,-COS(THETA(EL))/SIN(THETA(EL)))*I
     1*IHAT
      GLCPBP(BP,OMEGA,Q,OMEGAR,V,VHAT)=-V*VHAT*CMPLX((OMEGAR/Q),((OMEGA*
     1OMEGA-OMEGAR*OMEGAR)/OMEGA))/2.0
      GLCPQ(BP,Q,OMEGAR,V,VHAT)=V*VHAT*CMPLX(BP*OMEGAR/(2.*Q*Q),0.)
      GLCPOR(BP,OMEGA,OMEGAR,Q,V,VHAT)=-BP*V*VHAT*CMPLX(1./(2.*Q),-OMEGA
     1R/OMEGA)
      GLCSQ(XP,OMEGAR,Q,I,IHAT)=-CMPLX((XP*OMEGAR)/(2.*Q*Q),0.)*I*IHAT
      GLCSOR(OMEGA,XP,OMEGAR,Q,I,IHAT)=CMPLX((XP/(2.*Q)),-OMEGAR*XP/OMEG
     1A)*I*IHAT
      GLCSXP(OMEGA,OMEGAR,Q,I,IHAT)=CMPLX(OMEGAR/Q,((OMEGA*OMEGA-OMEGAR*
     1OMEGAR)/OMEGA))*I*IHAT/2.
      IF (N1.GT.50) GO TO 127
      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 125
    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 69
      DELO=1.E-7*OMEGA
      KKGD=0
      IF (KKGD.EQ.0) GO TO 3
```

```
2     DO 68 KKV=1,N
      KKGD=KKGD+1
      DFLX=1.E-4*X(KKV)
      X(KKV)=X(KKV)+DFLX
3     CONTINUE
      KGD=0
      IF (OMEGN.LE.20.) GO TO 4
      OMEGA=OMEGA-DELO
4     CONTINUE
      DO 67 KV=1,2
      J=0
      K=0
      VOLD=RL
      IOLD=1.0
      DO 60 L=1,M
      MM=M+1-L
      NN=IC(MM)
      GO TO (5,5,5,5,5,5,20,20,20,20,39,39,39,39,39), NN
5     KKK=NE-K
      IF (B(KKK)) GO TO 6
      GO TO 7
6     JJ=N-J
      A(KKK)=X(JJ)
      J=J+1
7     GO TO (8,9,10,11,15,16), NN
8     CALL CODE1 (IOLD,VOLD,A(KKK),OMEGA,INEW,VNEW)
      GO TO 12
9     CALL CODE2 (IOLD,VOLD,A(KKK),OMEGA,INEW,VNEW)
      GO TO 12
10    CALL CODE3 (IOLD,VOLD,A(KKK),OMEGA,INEW,VNEW)
      VOLD=VNEW-VOLD
      GO TO 12
11    CALL CODE4 (IOLD,VOLD,A(KKK),OMEGA,INEW,VNEW)
      IOLD=INEW-IOLD
12    IF (B(KKK).AND.OMEGN.GT.10.) GO TO 14
      IF (B(KKK)) GO TO 13
      GO TO 19
13    IF (NN.EQ.1.OR.NN.EQ.4) G(JJ)=GL(OMEGA,IOLD,IOLD)
      IF (NN.EQ.2.OR.NN.EQ.3) G(JJ)=GC(OMEGA,VOLD,VOLD)
14    IF (NN.EQ.1.OR.NN.EQ.4) AD(JJ)=IOLD
      IF (NN.EQ.2.OR.NN.EQ.3) AD(JJ)=VOLD
      GO TO 19
15    CALL CODE5 (IOLD,VOLD,A(KKK),INEW,VNEW)
      GO TO 17
16    CALL CODE6 (IOLD,VOLD,A(KKK),INEW,VNEW)
      IOLD=INEW-IOLD
17    IF (B(KKK)) GO TO 18
      GO TO 19
18    G(JJ)=IOLD*IOLD
      AD(JJ)=IOLD
19    KN=K+1
      GO TO 59
20    KK=K
      JOLD=J
      DO 23 II=1,3
      KKK=NE-KK
      IF (B(KKK)) GO TO 21
      GO TO 22
21    JJ=N-J
```

```
      A(KKK)=X(JJ)
      J=J+1
22    KK=KK+1
23    CONTINUE
      JK=NE-K-2
      JV=NE-K-1
      JH=NE-K
      NI=NN-6
      GO TO (24,25,26,27), NI
24    CALL CODE7 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      GO TO 28
25    CALL CODE8 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      GO TO 28
26    CALL CODE9 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      VOLD=VNEW-VOLD
      GO TO 28
27    CALL CODE10 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      IOLD=INEW-IOLD
28    CONTINUE
      IF (NN.EQ.7.OR.NN.EQ.10) NI=7
      IF (NN.EQ.8.OR.NN.EQ.9) NI=8
      DO 38 II=1,3
      GO TO (29,30,31), II
29    KVH=JH
      GO TO 32
30    KVH=JV
      GO TO 32
31    KVH=JK
32    CONTINUE
      NJ=N-JOLD
      IF (B(KVH).AND.OMEGN.GT.10.) GO TO 37
      IF (B(KVH)) GO TO 33
      GO TO 38
33    IF (KVH.EQ.JH) GO TO 34
      IF (KVH.EQ.JV) GO TO 35
      IF (KVH.EQ.JK) GO TO 36
34    IF (NI.EQ.7) G(NJ)=GLCSXP(OMEGA,A(JK),A(JV),IOLD,IOLD)
      IF (NI.EQ.8) G(NJ)=GLCPBP(A(JH),OMEGA,A(JV),A(JK),VOLD,VOLD)
      GO TO 37
35    IF (NI.EQ.7) G(NJ)=GLCSQ(A(JH),A(JK),A(JV),IOLD,IOLD)
      IF (NI.EQ.8) G(NJ)=GLCPQ(A(JH),A(JV),A(JK),VOLD,VOLD)
      GO TO 37
36    IF (NI.EQ.7) G(NJ)=GLCSOR(OMEGA,A(JH),A(JK),A(JV),IOLD,IOLD)
      IF (NI.EQ.8) G(NJ)=GLCPOR(A(JH),OMEGA,A(JK),A(JV),VOLD,VOLD)
37    IF (NI.EQ.7) AD(NJ)=IOLD
      IF (NI.EQ.8) AD(NJ)=VOLD
      JOLD=JOLD+1
38    CONTINUE
      KN=K+3
      GO TO 59
39    KK=K
      JOLD=J
      DO 42 II=1,2
      KKK=NE-KK
      IF (B(KKK)) GO TO 40
      GO TO 41
40    JJ=N-J
      A(KKK)=X(JJ)
      J=J+1
```

```
41      KK=KK+1
42      CONTINUE
        JV=NF-K-1
        JH=NF-K
        NI=NN-10
        GO TO (43,44,45,46,54), NI
43      CALL CODE11 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
        GO TO 48
44      CALL CODE12 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
        GO TO 47
45      CALL CODE13 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
        GO TO 48
46      CALL CODE14 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
        GO TO 47
47      IOLD=INEW-IOLD
48      NJ=N-JOLD
        IF (NN.EQ.11.OR.NN.EQ.14) NI=11
        IF (NN.EQ.12.OR.NN.EQ.13) NI=12
        IF (B(JH)) GO TO 49
        GO TO 51
49      IF (OMEGN.GT.10.) GO TO 50
        IF (NI.EQ.11) G(NJ)=GSCTZO(A(JV),IOLD,IOLD)
        IF (NI.EQ.12) G(NJ)=GOCTZO(OMEGA,A(JV),IOLD,IOLD)
50      CONTINUE
        JOLD=JOLD+1
        AD(NJ)=IOLD
51      NJ=N-JOLD
        IF (B(JV)) GO TO 52
        GO TO 58
52      IF (OMEGN.GT.10.) GO TO 53
        IF (NI.EQ.11) G(NJ)=GSCTEL(OMEGA,A(JV),A(JH),IOLD,IOLD)
        IF (NI.EQ.12) G(NJ)=GOCTEL(A(JH),A(JV),OMEGA,IOLD,IOLD)
53      CONTINUE
        AD(NJ)=IOLD
        GO TO 58
54      CALL CODE15 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
        NJ=N-JOLD
        IF (B(JH)) GO TO 55
        GO TO 56
55      AD(NJ)=VOLD
        ADJJ(NJ)=VNEW
        G(NJ)=GLLTZO(VNEW,VOLD,INEW,IOLD,A(JH))
        JOLD=JOLD+1
56      NJ=N-JOLD
        IF (B(JV)) GO TO 57
        GO TO 58
57      AD(NJ)=VOLD
        G(NJ)=GLLTEL(OMEGA,A(JV),VNEW,VOLD,INEW,IOLD)
        ADJJ(NJ)=VNEW
58      KN=K+2
59      VOLD=VNEW
        IOLD=INEW
        K=KN
60      CONTINUE
        IF (OMEGN.GT.20.) GO TO 64
        IF (OMEGN.GT.10.) GO TO 72
        RHO=1.-2.*INEW/(VNEW+INEW)
        CONRHO=CONJG(RHO)
        APP=CABS(RHO)
```

```
      DO 61 L=1,N
      GRAD(L)=REAL((CONRHO/APP)*2.*G(L)/((VNEW+INEW)**2))
 61   CONTINUE
      IF (NC.GT.0.OR.ND.GT.0) GO TO 62
      RETURN
 62   IF (KVR.EQ.0) RETURN
      NNN=N+1
      DO 63 L=NNN,N1
      GRAD(L)=0.
 63   CONTINUE
      RETURN
 64   CONTINUE
      PHASE=(1./(VNEW+INEW))
      IF (KGD.EQ.0) GO TO 65
      GO TO 66                              (IF KGD.GT.0)
 65   OMEGA=OMEGA+2.*OMEGA DELØ
      POLD=PHASE
 66   DELAY=-AIMAG((1./POLD)*((PHASE-POLD)/(2*DELO)))*(1000./(2.*PIE*FM)
     1)
 66   KGD=KGD+1
 67   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
 68   CONTINUE
      APP=OLDEL
      IF (NC.GT.0.OR.ND.GT.0) GO TO 69
      RETURN
 69   IF (KVR.EQ.0) GO TO 71
      NNN=N1-N
      DO 70 L=1,NNN
      LL=N+L
      AB(L)=X(LL)
 70   CONTINUE
 71   WCC=WC/FM
      IF (NC.GT.0) CALL CODEC (NC,N,AB,WCC,OMEGA,GD1,GRAD,FM)
      IF (ND.GT.0) CALL CODED (ND,N,NC,AB,WCC,OMEGA,GD2,GRAD,FM)
      APP=OLDEL+GD1+GD2-AB(NNN)
      IF (KVR.EQ.1) GRAD(N1)=-1.0
      RETURN
 72   APP=-20.*ALOG10((CABS(1./(VNEW+INEW)))*(1.+RL))
      K=1
      J=1
      VOLD=1.0
      IOLD=1.0
      DO 121 L=1,M
      NN=IC(L)
      GO TO (73,74,75,76,77,78,84,84,84,84,102,102,102,102,102), NN
 73   CALL CODE1 (IOLD,VOLD,A(K),OMEGA,INEW,VNEW)
      GO TO 79
 74   CALL CODE2 (IOLD,VOLD,A(K),OMEGA,INEW,VNEW)
      GO TO 79
 75   CALL CODE3 (IOLD,VOLD,A(K),OMEGA,INEW,VNEW)
      VOLD=VNEW-VOLD
      GO TO 79
 76   CALL CODE4 (IOLD,VOLD,A(K),OMEGA,INEW,VNEW)
      IOLD=INEW-IOLD
```

```
      GO TO 79
77    CALL CODE5 (IOLD,VOLD,A(K),INEW,VNEW)
      GO TO 79
78    CALL CODE6 (IOLD,VOLD,A(K),INEW,VNEW)
      IOLD=INEW-IOLD
79    IF (B(K)) GO TO 80
      GO TO 83
80    IF (NN.EQ.1.OR.NN.EQ.4) G(J)=GL(OMEGA,AD(J),IOLD)
      IF (NN.EQ.2.OR.NN.EQ.3) G(J)=GC(OMEGA,AD(J),VOLD)
      IF (NN.EQ.5.OR.NN.EQ.6) G(J)=IOLD*AD(J)
      GO TO (81,82,81,82,81,82), NN
81    G(J)=-G(J)
82    J=J+1
83    K=K+1
      GO TO 120
84    JK=K
      JV=K+1
      JH=K+2
      NII=NN-6
      GO TO (85,86,87,88), NII
85    CALL CODE7 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      GO TO 89
86    CALL CODE8 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      GO TO 89
87    CALL CODE9 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      VOLD=VNEW-VOLD
      GO TO 89
88    CALL CODE10 (IOLD,VOLD,A(JV),A(JK),OMEGA,A(JH),INEW,VNEW)
      IOLD=INEW-IOLD
89    IF (NN.EQ.7.OR.NN.EQ.10) NI=7
      IF (NN.EQ.8.OR.NN.EQ.9) NI=8
      DO 101 II=1,3
      GO TO (90,91,92), II
90    KVH=JK
      GO TO 93
91    KVH=JV
      GO TO 93
92    KVH=JH
93    IF (B(KVH)) GO TO 94
      GO TO 101
94    IF (KVH.EQ.JK) GO TO 95
      IF (KVH.EQ.JV) GO TO 96
      IF (KVH.EQ.JH) GO TO 97
95    IF (NI.EQ.7) G(J)=GLCSOR(OMEGA,A(JH),A(JK),A(JV),AD(J),IOLD)
      IF (NI.EQ.8) G(J)=GLCPOR(A(JH),OMEGA,A(JK),A(JV),AD(J),VOLD)
      GO TO 98
96    IF (NI.EQ.7) G(J)=GLCSQ(A(JH),A(JK),A(JV),AD(J),IOLD)
      IF (NI.EQ.8) G(J)=GLCPQ(A(JH),A(JV),A(JK),AD(J),VOLD)
      GO TO 98
97    IF (NI.EQ.7) G(J)=GLCSXP(OMEGA,A(JK),A(JV),AD(J),IOLD)
      IF (NI.EQ.8) G(J)=GLCPBP(A(JH),OMEGA,A(JV),A(JK),AD(J),VOLD)
98    GO TO (99,100,99,100), NII
99    G(J)=-G(J)
100   J=J+1
101   CONTINUE
      K=K+3
      GO TO 120
102   JV=K
      JH=K+1
```

```
      NII=NN-10
      GO TO (103,104,105,106,108), NII
103   CALL CODE11 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
      GO TO 107
104   CALL CODE12 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
      IOLD=INEW-IOLD
      GO TO 107
105   CALL CODE13 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
      GO TO 107
106   CALL CODE14 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
      IOLD=INEW-IOLD
107   IF (NN.EQ.11.OR.NN.EQ.14) NI=11
      IF (NN.EQ.12.OR.NN.EQ.13) NI=12
      GO TO 109
108   CALL CODE15 (IOLD,VOLD,A(JV),OMEGA,A(JH),INEW,VNEW,PIE)
109   DO 119 II=1,2
      GO TO (110,111), II
110   KVH=JV
      GO TO 112
111   KVH=JH
112   IF (B(KVH)) GO TO 113
      GO TO 119
113   IF (KVH.EQ.JV) GO TO 114
      IF (KVH.EQ.JH) GO TO 115
114   IF (NI.EQ.11) G(J)=GSCTEL(OMEGA,A(JV),A(JH),AD(J),IOLD)
      IF (NI.EQ.12) G(J)=GOCTEL(A(JH),A(JV),OMEGA,AD(J),IOLD)
      IF (NN.EQ.15) G(J)=-GLLTEL(OMEGA,A(JV),ADJJ(J),AD(J),IOLD,INEW)
      GO TO 116
115   IF (NI.EQ.11) G(J)=GSCTZO(A(JV),AD(J),IOLD)
      IF (NI.EQ.12) G(J)=GOCTZO(OMEGA,A(JV),AD(J),IOLD)
      IF (NN.EQ.15) G(J)=-GLLTZO(ADJJ(J),AD(J),IOLD,INEW,A(JH))
116   GO TO (117,118,117,118,118), NII
117   G(J)=-G(J)
118   J=J+1
119   CONTINUE
      K=K+2
120   IOLD=INEW
      VOLD=VNEW
121   CONTINUE
      PSIL=VNEW+INEW*RL
      CONSTN=20./ALOG(10.)
      DO 122 L=1,N
      GRAD(L)=-REAL(G(L)/PSIL)*CONSTN
122   CONTINUE
      IF (NC.GT.0.OR.ND.GT.0) GO TO 123
      RETURN
123   IF (KVR.EQ.0) RETURN
      NNN=N+1
      DO 124 L=NNN,N1
      GRAD(L)=0.
124   CONTINUE
      RETURN
125   INT=IFIX(OMEGN-30.)
      APP=X(INT)
      DO 126 L=1,N1
      GRAD(L)=0.
126   CONTINUE
      GRAD(INT)=1.0
      RETURN
```

```
127   WRITE (6,128)
      CALL EXIT
C
128   FORMAT (/////5X,69HNUMBER OF VARIABLE PARAMETERS N EXCEEDS 50. PLE
     1ASE CHANGE THE LENGTHS,/5X,55HOF ARRAYS AD(N), ADJJ(N) AND G(N) IN
     2 SUBROUTINE APPROX.)
      END
```

CD TOT 0420

```
SUBROUTINE CODEI (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
```

CD TOT 0007

```
SUBROUTINE CODE2 (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
```

CD TOT 0007

```
SUBROUTINE CODE3 (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
```

CD TOT   0007

```
SUBROUTINE CODE4 (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
```

CD TOT 0007

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

CD TOT 0006

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

CD TOT 0006

```
SUBROUTINE CODE7 (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
```

CD TOT 0007

```
SUBROUTINE CODE8 (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
```

CD TOT 0007

```
SUBROUTINE CODE9 (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
```

CD TOT 0007

```
SUBROUTINE CODE10 (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
```

CD TOT 0007

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

CD TOT 0007

```
SUBROUTINE CODE12 (IOLD,VOLD,EL,OMEGA,ZO,INEW,VNEW,PIE)
COMPLEX IOLD,VOLD,INEW,VNEW
THETA=PIE/2.*OMEGA*EL
VNEW=VOLD
INEW=IOLD+C4PLX(0.,SIN(THETA)/(ZO*COS(THETA)))*VOLD
RETURN
END
```

CD TOT 0007

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

CD TOT   0007

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

CD TOT 0007

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

CD TOT 0009

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

CD TOT 0020

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

CD TOT 0035

ΤΑ ΠΑΝΤΑ ΕΝ ΧΡΙΣΤΩΙ ΣΥΝΕΣΤΗΚΕΝ