

OSA

DESIGN CENTERING USING TOLCAD,

PART III: LISTING

OSA-86-TC-8-L

November 12, 1986

Optimization Systems Associates

163 Watson's Lane, Dundas, Ontario, Canada

©1983 J. W. Bandler, President

DESIGN CENTERING USING TOLCAD,

PART III: LISTING

OSA-86-TC-8-L

November 12, 1986

SUBROUTINES OF THE TOLCAD PACKAGE

Only subroutines modified or new to Raytheon are listed in full in this report. However, for completeness, all subroutines of COMPSHT are named and their current status indicated as follows.

PROGRAM TOLCAD	MODIFIED	LISTING-3
SUBROUTINE ABCDM	NOT ALTERED	
SUBROUTINE PIMP	NOT ALTERED	
SUBROUTINE SIMP	NOT ALTERED	
SUBROUTINE RPOLAR	NOT ALTERED	
SUBROUTINE MODSIN	NOT ALTERED	
SUBROUTINE COPY	NOT ALTERED	
SUBROUTINE PFUR	NOT ALTERED	
SUBROUTINE TAKEIN	MODIFIED	LISTING-8
SUBROUTINE PRGINT	MODIFIED	LISTING-14
SUBROUTINE CASCAD	NOT ALTERED	
SUBROUTINE ANGLE	NOT ALTERED	
SUBROUTINE EEFF1	NOT ALTERED	
SUBROUTINE EEFF2	NOT ALTERED	
SUBROUTINE ZOFRQ2	NOT ALTERED	
SUBROUTINE ANLDAT	MODIFIED	LISTING-16
SUBROUTINE ADUMP	NOT ALTERED	
SUBROUTINE SEARCH	MODIFIED	LISTING-37
SUBROUTINE ANALYZ	MODIFIED	LISTING-45
SUBROUTINE SNOISE	NOT ALTERED	
SUBROUTINE NODAL	NOT ALTERED	
SUBROUTINE STRIP	NOT ALTERED	
SUBROUTINE WIDTH	NOT ALTERED	

SUBROUTINE SWRIT	NOT ALTERED	
SUBROUTINE ANAERR	NOT ALTERED	
SUBROUTINE SNOIS1	NOT ALTERED	
SUBROUTINE SNOIS2	NOT ALTERED	
SUBROUTINE GRID	NOT ALTERED	
SUBROUTINE FTP	NEW	LISTING-75
SUBROUTINE FTPAQ	NEW	LISTING-77
SUBROUTINE FTPBQ	NEW	LISTING-80
SUBROUTINE FTPCQ	NEW	LISTING-84
SUBROUTINE FTPDQ	NEW	LISTING-88
SUBROUTINE FTPEQ	NEW	LISTING-91
SUBROUTINE MODEL4	[1]	LISTING-94
SUBROUTINE QPE	[1]	LISTING-97
SUBROUTINE VERT	[1]	LISTING-100
SUBROUTINE ACFUNS	NEW	LISTING-101

Reference

[1] J.W.Bandler and H.L.Abdel-Malek,"Optimal centering, tolerancing and yield determination via updated approximations and cuts", IEEE Trans. Circuits and Systems, vol. CAS-25, 1978, pp.853-871.

```
*************************************************************************
*   PRESENT VERSION:  TOLCAD V1.3
*************************************************************************
      PROGRAM TOLCAD (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE9,TAPE98
     *   ,TAPE99,TAPE15=512,TAPE16=512)
      COMMON KODE,IFLAG,XITIME,ITRAP,JTRAP,ISENS,IPRIN,MKODE,EX,W,FR,NT,
     *   NF,INDFF,NCO,IXX,MCODE,FF,IMA,IMA1,NOISE,IBANKF,IDCLOC
      COMMON /COMB/ U11,U22,ZIN,ATTN,CAB,IND,I,IJ,IJJ,IND3R,INDTR,J,KIMP
     *   ,M,N,NBAND,NCB,NEWEX,NFOPT,PPRD
      COMMON /HVPLOT/ HORI(7),VERT(7),EPLOT(7,7),SENDE(15),BBB(30,2,2)
      COMMON /ANALQ/ ERRD,AOPT,INRJX,JSTK,JSOK,KLM,KYM,MODIM,MODEL,NOOPT
     *   ,IPRI,ITHR,IN1,IN2,IN3,NOIT,IDIV,ITER,IPRAM,ERSET,ERRAN,ERNO,
     *   SAVEC,ILTGT(3),S3P(3,6),VAX(15),IP1(15),IP2(15)
      DOUBLE PRECISION ERRD
      COMMON /DATAS/ PERT,PI,DPR,TWOPI,RPD
      COMMON /ANALZ/ IBAND(3),RIND(70),INC1(70),INC2(70),INC3(70),KX(15)
     *   ,KY(15)
      COMMON /TAKE/ ARRAY(28),IARRAY(28)
      COMMON /TAKE1/ IBUF(136),LDC
      COMMON /TAKE2/ PERUP(3,15)
      COMMON /CO2/ SEND(15),GR(15),ERR,NOPT,CONV,INEXP,IGR(15)          12-16-85
      COMMON /CO3/ CARAY(30,2,2),DATX(70,6)
      COMMON /CO4/ VIMP(30,2),FRST(45),DC0(100),DC1(70),DC2(70),DC3(70),
     *   FIG(3,10,5),FIN(3,10),OUT(2,4),WI(3,6),S21(3),FIGG(2,10,5)
      COMMON /CO5/ SLIMP(30,4)
      COMMON /CO6/ S3R(16,3,3)
      COMMON /CO7/ YMAT(15,2,2)
      COMMON /CO8/ PS(30,2,4)
      COMMON /CO9/ AAA(30,2,2),A(2,2),AB1(2,2),AB2(2,2),AB3(2,2),AB4(2,2
     *   )
      COMMON /KSIO/ KSIN,KSOUT
      COMMON /CSERCH/ ALAMDA,GO(15),G1T(15),GA(15),H(15,15),KOUNT,LIMIT,
     *   MAT,NITER,V0,V1T,V01,V02,V03,X0(15),X1(15),X1T(15),XA(15),XST1(
     *   15),XST2(15),LDSERC,NLIST,ILIST
      COMMON /MODIFY/ FN
      COMMON /CNAERR/ IFTLER,AMET,ALOSS
      COMMON /PPLOT/ NPLOT,PARAY(30,2,5),NPOINT,IPNOPR,MIXPR,OLDT
      COMMON /DATANA/ DUMANA(200)
      COMMON /LAY10/ ISTOP,KOPT,IBYP
      COMMON /IOS/ IOT,INPET,INP,IFR,ITO,IED,IOPT,ISCR
      COMMON /BOMB/ JERRX
      COMMON /BUGS/ ABUGS(100),IABFLG
      COMMON /CNOISE/ DX(64),REFR,GN,RN,IN22,INOIS,PAV(10),IA
      COMMON /CGRID/ WRST(15),IFUNCT,IFUX,ICOUNT,IIII,CC,XPER
      COMMON /IVG/ IVG
      COMMON /NFILE/ IPINT
      COMMON /DPLOT/ NPTY
      COMMON /PRTOPT/ NSTAR
      COMMON /CODLAY/ DELAY,FPERT
      COMMON /IMPONE/ NIMP
C
C     INSERT NEW COMMON BLOCK FOR OPTIMIZATION W.R.T.                   12-16-85
C
      COMMON/TOLER/NVERT,TOL(15),NRANK,NCOUNT,NCALL,TRES,WIT(65)
      COMMON /QUAMOD/IQUAD,IFTP,IAPPR,MAJOR
```

```
      COMMON/MIM/B(253,1),IV(6,64)
      COMMON/LAST/XNOMLST(15),NRFLST(64),FERRLST(64),ZTLST(64,6)
      COMMON/BLKMRE/RMU(6,64),NRAND
      COMMON/BLKMRF/INTOL,KXTOL(6),KYTOL(6),IRATOL(6),PERTOL(2,6)
      COMMON/BLKMRG/NSPACE,KXMOD(21),KYMOD(21)
      COMPLEX CARAY
      COMPLEX AAA,A,AB1,AB2,AB3,AB4,BBB,S3R,U11,U22,YMAT,ZIN
      DIMENSION REMAINX(4)
      DOUBLE PRECISION ERRD1,ERRSD,PERT
      EQUIVALENCE (AR1,ARRAY(1))
      DATA REMAINX / "+0600",3*"+0000"/
      DATA AY,AYES,AN,ANO,REP,SAV / "Y","YES","N","NO","REP","SAV"/
C
C *****BEGIN PROGRAM
C
      IBYP = 0
      DPR = 57.29578
      PERT = 1.0D-7
      PI = 3.14159
      RPD = 1.7453292E-2
      TWOPI = 6.28318
      GO TO 30
   10 CONTINUE
      CALL ANAERR
   20 CONTINUE
      IMA = 0
      IMA1 = 0
      WRITE (IOT,310)
      REWIND IFR
      REWIND ITO
   30 IOT = 6
      INPET = 5
      IFR = 15
      ITO = 16
      ISCR = 9
      INP = 15
      IED = 16
      IOPT = 16
C
C***** INITIALIZE SYSTEM SWITCHES
C
   40 NPLOT = 0
      NPOINT = 0
      IBANKF = 0
      IMA = 0
      IMA1 = 0
      ISENS = 0
C
C***** INITIALIZE PROGRAM FILES
C
      CALL PRGINT
      IF (ISTOP.EQ.1) GO TO 300
C
C***** DETERMINE LINK TO BE CALLED
C
```

```
    IF (KOPT.EQ.44) GO TO 60
 50 IF (KOPT.EQ.13) GO TO 140
    IF (KOPT.EQ.27) GO TO 110
    GO TO (90,130,150,70,80,100,120,145), KOPT
 60 IMA = 1
 70 KODE = 5
    GO TO 260
 80 KODE = 6
    GO TO 260
 90 KODE = 9
    GO TO 260
100 KODE = 8
    GO TO 260
110 MKODE = 3
120 KODE = 10
    GO TO 260
130 ISENS = 1
140 MKODE = 1
145 NCOUNT=0
    NCALL=1
    IQUAD=0
150 LDSERC = 0
    KODE = 1
160 IF (KODE.NE.1) GO TO 170
    CALL ANLDAT
    IF (JERRX.NE.0) GO TO 20
    REWIND IFR
    REWIND ITO
    IF (IBANKF.EQ.0) GO TO 170
    IHOL = IFR
    IFR = ITO
    ITO = IHOL
    REWIND IFR
    REWIND ITO
    CALL ADUMP
    CALL PFUR (REP,IFR,FN,0,ISTA)
170 IF(KOPT.NE.8)GO TO 175

    IF(INTOL.GT.0.AND.NOPT.GT.0) GOTO 172
    PRINT(6,800)
800 FORMAT(/" THE FIXED-TOLERANCE PROBLEM IS ILL-DEFINED."/,
   +" EITHER NO TOLERANCED PARAMETERS EXIST OR NO VARIABLE"/,
   +" PARAMETERS EXIST.  BOTH ARE REQUIRED."/)
    JERRX=1
    GOTO 20

172 CALL FTP
    GO TO 176
175 CALL ANALYZ
176 CONTINUE
    IF (JERRX.EQ.0) GO TO 180
    IF (IFTLER.NE.23) GO TO 10
    JERRX = 0
    IFTLER = 0
    KODE = 1
```

```
      LDSERC = 0
      WRITE (IOT,320) IXX
      GO TO 170
  180 CONTINUE
      IF (KOPT.NE.13) GO TO 190
      IF (NPLOT.LE.0.OR.KODE.NE.1) GO TO 190
  190 CALL SEARCH
      IF (JERRX.NE.0) GO TO 20
      IF (KODE.EQ.2.AND.ISENS.EQ.2) GO TO 290
      IF (LDSERC.GT.0) GO TO 160
      IF (LDSERC.EQ.0) GO TO 210
      IF (IPRIN.EQ.1.OR.MKODE.EQ.1) GO TO 200
      CALL ANLDAT
  200 CALL ANALYZ
  210 IF (KOPT.EQ.2) GO TO 280
  220 PRINT *,"SAVE OPTIMIZED RESULTS (Y/N)     ",
      CALL TAKEIN (INPET,NOITEM)
      IF (ARRAY(1).EQ.AN) GO TO 280
      IF (ARRAY(1).NE.AY) GO TO 220
  230 PRINT *,"OPTIMIZED FILE NAME      ",
      CALL TAKEIN (INPET,NOITEM)
      IF (NOITEM.LE.0) GO TO 230
      IF (IARRAY(1).LT.1) GO TO 230
      IF (IARRAY(1).GT.7) WRITE (IOT,330)
      IF (IARRAY(1).GT.7) GO TO 230
      OPTFIL = AR1
      WRITE (ITO) IOT
      REWIND ITO
      CALL PFUR (SAV,ITO,OPTFIL,0,ISTAT)
      IF (ISTAT.EQ.0) GO TO 250
  240 PRINT *,"FILE IS PERMANENT - REPLACE IT (Y/N)?    "
      CALL TAKEIN (INPET,NOITEM)
      IF (ARRAY(1).EQ.AN.OR.ARRAY(1).EQ.ANO) GO TO 230
      IF (ARRAY(1).NE.AY.AND.ARRAY(1).NE.AYES) GO TO 240
  250 CALL ADUMP
      CALL PFUR (REP,IFR,OPTFIL,0,ISTAT)
      GO TO 280
  260 CALL ANLDAT
      IF (JERRX.NE.0) GO TO 20
      REWIND IFR
      REWIND ITO
      IF (IBANKF.EQ.0) GO TO 270
      IHOL = IFR
      IFR = ITO
      ITO = IHOL
      REWIND IFR
      REWIND ITO
      CALL ADUMP
      CALL PFUR (REP,IFR,FN,0,ISTA)
  270 CALL ANALYZ
      IF (JERRX.EQ.1) GO TO 10
      IF (IMA.EQ.0) GO TO 280
      IF (KODE.EQ.0) GO TO 280
C
C  **FOLLOW-ON TO COARSE SEARCH**
```

```
C
      IMA = 0
      IMA1 = 1
      KOPT = KODE
      GO TO 50
  280 IF (NPLOT.LE.0) GO TO 290
      NPLOT = INC3(NPLOT)
  290 IF (IMA1.EQ.1) IMA = 0
      GO TO 40
  300 CONTINUE
      STOP
C
  310 FORMAT (" EXECUTION STOPPED DUE TO FILE ERROR")
  320 FORMAT (/" EVEN- AND ODD-MODE IMPEDANCES RESET IN LINE",I3,
     *   ", BECAUSE OPTIMIZATION"/" FORCED ZOO>ZOE -- OPTIMIZATION",
     *   " RESTARTS USING NEW VALUES OF ZOE AND ZOO"/" AND",
     *   " CURRENT VALUES OF OTHER VARIABLES, IF ANY"/)
  330 FORMAT (" FILENAME TOO LONG")
C
      END
```

```
      SUBROUTINE TAKEIN (INP,NOITEMS)
      COMMON /TAKE/ ARRAY(28),IARRAY(28)
      COMMON /TAKE1/ IBUF(136),IN2
      COMMON /TAKE2/ PERUP(3,15)
      DIMENSION JBUF(17),INBUF(3,15),INUMS(14)
      DATA INUMS / "0","1","2","3","4","5","6","7","8","9",".","+","-",
     *   "E"/
      DATA IBL,ICOM,IPL,IMIN,IE,IPER / " ",",","+","-","E","."/
      DATA IPRL,IPRR / "(",")"/
C
C     READ THE INPUT FILE
C
      IPRNFLG = 0
      INCLFLG = 0
      LLEN = 136
      DO 10 IN1 = 1, 28
         ARRAY(IN1) = 0.0
         IARRAY(IN1) = 0
   10 CONTINUE
   20 READ (INP,730) IBUF
      IF (EOF(INP)) 710,30
   30 DO 40 IN1 = 1, LLEN
         IN2 = LLEN+1-IN1
         IF (IBUF(IN2).NE." ") GO TO 50
   40 CONTINUE
      GO TO 20
   50 IN3 = IN2
      GO TO 60
C
C     EXPLODE INPUT BUFFER INTO ONE CHARACTER PER WORD
C
   60 CONTINUE
C
C     GET READY TO SEARCH INPUT STRING
C
      ICOUNT = 0
      JCOUNT = 0
      ICURR = 0
      DO 70 IN1 = 1, IN3
         IN2 = IN3+1-IN1
         IF (IBUF(IN2).NE." ") GO TO 80
   70 CONTINUE
      IBUF(1) = " "
   80 IN2 = IN2+1
      IBUF(IN2) = ","
      IF (NOITEMS.EQ.-999) GO TO 720
C
C     PASS BY DELIMETERS
C
   90 ICURR = ICURR+1
      IF (ICURR.GT.IN2) GO TO 700
      IHOLD = IBUF(ICURR)
      IF (IHOLD.EQ.ICOM.OR.IHOLD.EQ.IBL) GO TO 90
      IF (IHOLD.NE.IPRL.AND.IHOLD.NE.IPRR) GO TO 110
      IF (IHOLD.NE.IPRL) GO TO 100
```

```
      IF (IPRNFLG.EQ.1) GO TO 110
      IPRNFLG = 1
      GO TO 90
  100 IF (IPRNFLG.EQ.0) GO TO 110
      IPRNFLG = 0
      INCLFLG = 0
      GO TO 90
  110 INCLFLG = IPRNFLG
C
C     PICK UP INPUT SUBSTRING
C
      IBEG = ICURR
  120 ICURR = ICURR+1
      IHOLD = IBUF(ICURR)
      IF (IHOLD.EQ.ICOM.OR.IHOLD.EQ.IBL) GO TO 140
      IF (IHOLD.NE.IPRL.AND.IHOLD.NE.IPRR) GO TO 120
      IF (IHOLD.NE.IPRL) GO TO 130
      IF (IPRNFLG.EQ.1) GO TO 120
      IPRNFLG = 1
      GO TO 140
  130 IF (IPRNFLG.EQ.0) GO TO 120
      IPRNFLG = 0
C
C     IS THE SUBSTRING NUMERIC OR ALPHA - IF NUMERIC, COUNT
C     THE OCCURANCES OF "+" "-" "." AND "E"
C
  140 IEND = ICURR-1
      IEFLAG = 0
      IPINT = 0
      IPDEC = 0
      IPEXP = 0
      DO 190 IN1 = IBEG, IEND
         IHOLD = IBUF(IN1)
         DO 180 IN3 = 1, 14
            IF (IHOLD.NE.INUMS(IN3)) GO TO 180
            IF (IHOLD.NE.IE) GO TO 150
            IEFLAG = IEFLAG+1
            GO TO 190
  150       IF (IHOLD.NE.IPER) GO TO 160
            IPINT = IPINT+1
            GO TO 190
  160       IF (IHOLD.NE.IMIN) GO TO 170
            IPDEC = IPDEC+1
            GO TO 190
  170       IF (IHOLD.NE.IPL) GO TO 190
            IPEXP = IPEXP+1
            GO TO 190
  180    CONTINUE
         GO TO 690
  190 CONTINUE
      IF (IEFLAG-1) 200 , 200 , 690
  200 IF (IPINT-1) 210 , 210 , 690
  210 IF (IPDEC-2) 220 , 220 , 690
  220 IF (IPEXP-2) 230 , 230 , 690
C
```

```
C     BEGIN DECODING THE NUMERIC SUBSTRING
C
  230 IPINT = 0
      IPDEC = 0
      IPEXP = 0
      ISMAN = 0
      ISEXP = 0
      DO 250 IN1 = 1, 3
         DO 240 IN3 = 1, 15
            INBUF(IN1,IN3) = IBL
  240    CONTINUE
  250 CONTINUE
C
C     EXAMINE EACH CHARACTER
C
      IGO = 1
      DO 510 IN1 = IBEG, IEND
         IHOLD = IBUF(IN1)
         DO 260 IN3 = 1, 14
            IF (IHOLD.EQ.INUMS(IN3)) GO TO 270
  260    CONTINUE
  270    JHOLD = IN3+26
         GO TO (280,350,390,430,470,500), IGO
C
C     ACCEPT SIGN OF MANTISSA AND FIRST NUMERIC OF INTEGER
C     PORTION - TERMINATE WITH "." OR "E"
C     "+" SIGN
C
  280    IF (IHOLD.NE.IPL) GO TO 290
         IF (ISMAN.NE.0) GO TO 690
         ISMAN = 1
         GO TO 510
C
C     "-" SIGN
C
  290    IF (IHOLD.NE.IMIN) GO TO 300
         IF (ISMAN.NE.0) GO TO 690
         ISMAN = -1
         GO TO 510
C
C     NUMERIC
C
  300    IF (JHOLD-27) 330 , 310 , 310
  310    IF (JHOLD-36) 320 , 320 , 330
  320    IPINT = 1
         INBUF(1,IPINT) = JHOLD-27
         IGO = 2
         GO TO 510
C
C     PERIOD
C
  330    IF (IHOLD.NE.IPER) GO TO 690
  340    IGO = 3
         GO TO 510
C
```

```
C     ACCEPT NUMERICS FOR INTEGER PART - TERMINATE WITH "."
C     OR "E"
C     NUMERIC
C
  350    IF (JHOLD-27) 380 , 360 , 360
  360    IF (JHOLD-36) 370 , 370 , 380
  370    IPINT = IPINT+1
         IF (IPINT.GT.15) GO TO 690
         INBUF(1,IPINT) = JHOLD-27
         GO TO 510
C
C     PERIOD
C
  380    IF (IHOLD.EQ.IPER) GO TO 340
C
C     "E"
C
         IF (IHOLD.NE.IE) GO TO 690
         IGO = 5
         GO TO 510
C
C     ACCEPT FIRST NUMERIC FOR DECIMAL PORTION - TERMINATE
C     WITH "E"
C     NUMERIC
C
  390    IF (JHOLD-27) 420 , 400 , 400
  400    IF (JHOLD-36) 410 , 410 , 420
  410    IPDEC = 1
         INBUF(2,IPDEC) = JHOLD-27
         IGO = 4
         GO TO 510
C
C     "E"
C
  420    IF (IHOLD.NE.IE) GO TO 690
C
         IGO = 5
         GO TO 510
C     ACCEPT REST OF NUMERICS FOR DECIMAL PORTION - TERMINATE
C     WITH "E"
C     NUMERIC
C
  430    IF (JHOLD-27) 460 , 440 , 440
  440    IF (JHOLD-36) 450 , 450 , 460
  450    IPDEC = IPDEC+1
         IF (IPDEC.GT.15) GO TO 690
         INBUF(2,IPDEC) = JHOLD-27
         GO TO 510
C
C     "E"
C
  460    IF (IHOLD.NE.IE) GO TO 690
C
         IGO = 5
         GO TO 510
```

```
C     ACCEPT SIGN AND NUMERICS IN EXPONENT PORTION - TERMINATE
C     WITH END-OF-SUBSTRING
C     "+"
  470    IF (IHOLD.NE.IPL) GO TO 480
         IF (ISEXP.NE.0) GO TO 690
         ISEXP = 1
         GO TO 510
C
C     "-"
C
  480    IF (IHOLD.NE.IMIN) GO TO 490
         IF (ISEXP.NE.0) GO TO 690
         ISEXP = -1
         GO TO 510
C
C     NUMERIC
C
  490    IF (JHOLD.LT.27.OR.JHOLD.GT.36) GO TO 690
         IPEXP = 1
         IGO = 6
         INBUF(3,IPEXP) = JHOLD-27
         GO TO 510
C
C     ACCEPT REST OF NUMERICS IN EXPONENT PORTION - TERMINATE
C     WITH END-OF-SUBSTRING
C
  500    IF (JHOLD.LT.27.OR.JHOLD.GT.36) GO TO 690
         IPEXP = IPEXP+1
         IF (IPEXP.GT.3) GO TO 690
         INBUF(3,IPEXP) = JHOLD-27
  510 CONTINUE
C
C     ASSEMBLE INTEGER PORTION
C
      XINT = 0.0
      IF (IPINT) 540 , 540 , 520
  520 DO 530 IN1 = 1, IPINT
         IN3 = IPINT+1-IN1
         XINT = XINT+INBUF(1,IN3)*10.**(IN1-1)
  530 CONTINUE
C
C     ASSEMBLE DECIMAL PORTION
C
  540 XDEC = 0.0
      IF (IPDEC) 570 , 570 , 550
  550 DO 560 IN1 = 1, IPDEC
         XDEC = XDEC+INBUF(2,IN1)*10.**(-IN1)
  560 CONTINUE
C
C     ASSEMBLE EXPONENT PORTION
C
  570 XEXP = 0.0
      IF (IPEXP) 600 , 600 , 580
  580 DO 590 IN1 = 1, IPEXP
         IN3 = IPEXP+1-IN1
```

```
          XEXP = XEXP+INBUF(3,IN3)*10.**(IN1-1)
  590 CONTINUE
C
C     CORRECT SIGNS OF MANTISSA AND EXPONENT IF NECESSARY
C
  600 IF (ISMAN) 620 , 610 , 620
  610 ISMAN = 1
  620 IF (ISEXP) 640 , 630 , 640
  630 ISEXP = 1
C
C     MERGE THE PORTIONS
C
  640 IF (XEXP) 650 , 680 , 650
  650 IF (XINT) 680 , 660 , 680
  660 IF (XDEC) 680 , 670 , 680
  670 XINT = 1.0
  680 ICOUNT = ICOUNT+1
      JCOUNT = JCOUNT+1
      ARRAY(ICOUNT) = ((XINT+XDEC)*ISMAN)*10.**(XEXP*ISEXP)
      IARRAY(JCOUNT) = 0
      IF (INCLFLG.EQ.1) IARRAY(JCOUNT) = -1
      GO TO 90
C     ASSEMBLE THE ALPHA SUBSTRING
  690 ISCR = IEND-IBEG+1
      JSCR = (ISCR-1)/10
      ICOUNT = ICOUNT+1
      JCOUNT = JCOUNT+1
      IARRAY(JCOUNT) = ISCR
      IF (INCLFLG.EQ.1) IARRAY(JCOUNT) = -1-ISCR
      KSCR = 1+JSCR
      ENCODE (ISCR,730,ARRAY(ICOUNT)) (IBUF(III),III=IBEG,IEND)
      ICOUNT = ICOUNT+JSCR
      GO TO 90
C
C     PREPARE THE RETURN CODES
C
  700 NOITEMS = JCOUNT
      GO TO 720
C
C     END-OF-FILE RETURN CODES
C
  710 NOITEMS = -1
  720 RETURN
  730 FORMAT (136A1)
      END
```

```
      SUBROUTINE PRGINT
C
C *****INITIALIZE THE RUN OR RERUN
C
      COMMON KODE,IFLAG,XITIME,ITRAP,JTRAP,ISENS,IPRIN,MKODE,EX,W,FR,NT,
     *   NF,INDFF,NCO,IXX,MCODE,FF,IMA,IMA1,NOISE,IBANKF,IDCLOC
      COMMON /BOMB/ JERRX
      COMMON /BUGS/ ABUGS(100),IABFLG
      COMMON /IOS/ IOT,INPET,INP,IFR,ITO,IED,IOPT,ISCR
      COMMON /LAY10/ ISTOP,KOPT,IBYP
      COMMON /MODIFY/ FN
      COMMON /TAKE/ ARRAY(28),IARRAY(28)
      COMMON /TAKE1/ IBUF(136),IN2
      DIMENSION AR(2)
      EQUIVALENCE (AR1,AR2,ARRAY(1))
      EQUIVALENCE (AR3,AR(1))
      DATA AP,BL1,BW,RET,XNEW,XNO,YES / "A"," ","BW","RET","NEWS","N",
     *   "Y"/
      DATA QUI,QUI1,INOCH / "QUIT","QUI",10 /
C
C *****SET PROGRAM VARIABLES, WRITE INTRO MESSAGES
C *****BYPASS AFTER FIRST RUN
C
      IF (IBYP.EQ.1) GO TO 10
      IBYP = 1
      ISTOP = 0
      KODE = 0
      JERRX = 0
      KOPT = 0
      FN = BL
      MKODE = 0
      IPRIN = 0
      ITRAP = 0
      IABFLG = 0
      GO TO 20
C
C *****REINITIALIZE PROGRAM VARIABLES FOR SUBSEQUENT PASSES
C
   10 ISTOP = 0
      FN = BL
      KOPT = 0
      MKODE = 0
      KODE = 0
      IPRIN = 0
      JERRX = 0
      ITRAP = 0
      IABFLG = 0
      REWIND 99
C
C *****WHICH OPTION - FILE TO RUN OR QUIT?
C
   20 PRINT *," "
      PRINT *,"'FILE NAME' OR 'QUIT'      ",
      CALL TAKEIN (INPET,NOITEM)
      ILOC = 1
```

```
      JLOC = 1
      IF (AR1.EQ.QUI.OR.AR1.EQ.QUI1) GO TO 80
      AR3 = AR2
      IF (IARRAY(1).LE.INOCH) AR(2) = IBL
      FN = AR3
      CALL PFUR (RET,INP,FN,0,ISTA)
      IF (ISTA.EQ.0) GO TO 30
      WRITE (IOT,100)
      GO TO 20
C
C *****WHICH FUNCTION - 1, 2, 3, 4, 5, 6, 7, 13, 44
C *****SET EXPERT MODE IF THERE ARE/WERE MULTIPLE INPUTS
C
   30 IF (NOITEM.LE.ILOC) GO TO 40
      ILOC = ILOC+1
      IF (IARRAY(JLOC).GT.INOCH) ILOC = ILOC+1
      IPRIN = 1
      GO TO 50
   40 PRINT *,"AN(1),SEN(2),OPT(3),SW(4),MAP(5),VAR(6),MC(7),TOL(8),ANOP
     +T(13),RND(44)   ",
      CALL TAKEIN (INPET,NOITEM)
      ILOC = 1
   50 KOPT = ARRAY(ILOC)
      IF (KOPT.GE.1.AND.KOPT.LE.8) GO TO 60
      IF (KOPT.EQ.13) GO TO 60
      IF (KOPT.NE.44) GO TO 40
C
C *****TEST FOR LINE SEQUENCE NUMBERS AND EMPTY FILE
C
   60 REWIND INP
      CALL TAKEIN (INP,NOITEM)
      IF (NOITEM.GT.0) GO TO 70
      WRITE (IOT,110)
      REWIND INP
      GO TO 20
   70 IF (IARRAY(1).EQ.0) ITRAP = 1
      REWIND INP
C
C *****TRANSFER THE FILE TO SCRATCH STORAGE
C
      CALL COPY (INP,IFR)
      GO TO 90
C
C *****QUIT
C
   80 ISTOP = 1
   90 RETURN
  100 FORMAT (" FILE NOT FOUND")
  110 FORMAT (" FILE IS EMPTY")
      END
```

```
      SUBROUTINE ANLDAT
C
C  **SUBROUTINE TO READ AND STORE DATA FILE**
C
      COMMON KODE,IFLAG,XITIME,ITRAP,JTRAP,ISENS,IPRIN,MKODE,EX,W,FR,NT,
     *   NF,INDFF,NCO,IXX,MCODE,FF,IMA,IMA1,NOISE,IBANKF,IDCLOC
      COMMON /COMB/ U11,U22,ZIN,ATTN,CAIND,IND,I,IJ,IJJ,IND3R,INDTR,J,
     *   KIMP,M,N,NBAND,NCB,NEWEX,NFOPT,PPRD
      COMMON /HVPLOT/ HORI(7),VERT(7),EPLOT(7,7),SENDE(15),BBB(30,2,2)
      COMMON /ANALQ/ ERRD,AOPT,INRJX,JSTK,JSOK,KLM,KYM,MODIM,MODEL,NOOPT
     *   ,IPRI,ITHR,IN1,IN2,IN3,NOIT,IDIV,ITER,IPRAM,ERSET,ERRAN,ERNO,
     *   SAVEC,ILTGT(3),S3P(3,6),VAX(15),IP1(15),IP2(15)
      COMMON /DATAS/ PERT,PI,DPR,TWOPI,RPD
      COMMON /ANALZ/ IBAND(3),RIND(70),INC1(70),INC2(70),INC3(70),KX(15)
     *   ,KY(15)
      COMMON /TAKE/ ARRAY(28),IARRAY(28)
      COMMON /TAKE1/ IBUF(136),LDC
      COMMON /TAKE2/ PERUP(3,15)
CBOB  COMMON /CO2/ SEND(15),GR(15),ERR,NOPT,CONV,INEXP
      COMMON /CO2/ SEND(15),GR(15),ERR,NOPT,CONV,INEXP,IGR(15)          12-16-85
      COMMON /CO3/ CARAY(30,2,2),DATX(70,6)
      COMMON /CO4/ VIMP(30,2),FRST(45),DC0(100),DC1(70),DC2(70),DC3(70),
     *   FIG(3,10,5),FIN(3,10),OUT(2,4),WI(3,6),S21(3),FIGG(2,10,5)
      COMMON /CO5/ SLIMP(30,4)
      COMMON /CO6/ S3R(16,3,3)
      COMMON /CO7/ YMAT(15,2,2)
      COMMON /CO8/ PS(30,2,4)
      COMMON /CO9/ AAA(30,2,2),A(2,2),AB1(2,2),AB2(2,2),AB3(2,2),AB4(2,2
     *   )
      COMMON /KSIO/ KSIN,KSOUT
      COMMON /PPLOT/ NPLOT,PARAY(30,2,5),NPOINT,IPNOPR,MIXPR,OLDT
      COMMON /IHOLD/ IHOLINE
      COMMON /IOS/ IOT,INPET,INP,IFR,ITO,IED,IOPT,ISCR
      COMMON /CNAERR/ IFTLER,AMET,ALOSS
      COMMON /BOMB/ JERRX
      COMMON /IVG/ IVG
      COMMON /NFILE/ IPINT
      COMMON /CGRID/ WRST(15),CGRID1(5),XPER
      COMMON /MODIFY/ FN
      COMMON /PRTOPT/ NSTAR
      COMMON /IMPONE/ NIMP
      COMMON /CODLAY/ DELAY,FPERT
      COMMON /WAVG/ IWAV1,IWAV2
      COMMON/BLKMRF/INTOL,KXTOL(6),KYTOL(6),IRATOL(6),PERTOL(2,6)
* WHERE:
*   INTOL    COUNTS THE NUMBER OF TOLERANCED PARAMETERS
*   KXTOL()  STORES THE FIRST INDEX OF DATX(,) FOR THE TOLERANCED
*            PARAMETER
*   KYTOL()  STORES THE SECOND INDEX OF DATX(,) FOR THE TOLERANCED
*            PARAMETER
*   IRATOL() FLAG 0/1 FOR RELATIVE/ABSOLUTE TOLERANCE
*   PERTOL(1,)  UPPER TOLERANCE SPECIFICATION
*   PERTOL(2,)  LOWER TOLERANCE SPECIFICATION
      DIMENSION COL1(80),COL2(60),KBANK(15)
      DIMENSION AMFG(17)
```

```
      COMPLEX AAA,A,AB1,AB2,AB3,AB4,BBB,S3R,U11,U22,YMAT,ZIN
      COMPLEX CARAY
      DOUBLE PRECISION PERT,ERRD
      REAL IR
      DATA AAGT / "GT"/ ,AALT / "LT"/ ,ANOF / "NOF"/ ,AONE / "ONE"/ ,
     *   END1 / "END"/ ,PRI / "PRI"/ ,RJX / "RJX"/ ,S1 / "S1"/ ,S2 /
     *   "S2"/ ,S3 / "S3"/ ,SEOI / "SEO"/ ,SETI / "SET"/ ,SMOI / "SMO"/
     *   ,SMTI / "SMT"/ ,THRI / "THR"/ ,TWO / "TWO"/ ,YM / "YM"/ ,ZIMP /
     *    "IMP"/ ,ANOP / "NOP"/ ,ANOS / "NOS"/ ,S8 / "S8"/
      DATA COL1 / "RES","IND","CAP","CAS","PAR","SER","PRI","SRL","PRL",
     *   "CAX","SRC","PRC","SLC","PLC","SRX","PRX","TWO","TRF","TRL",
     *   "OST","SST","CCS","CVS","OPA","SBR","PBR","RJX","CAB","QDC",
     *   "IMP","INV","CST","QLA","QLB","QLC","QLD","QLE","EQU","THP",
     *   "CHA","MUI","SET","MUC","NOF","ONE","SEO","SMT","SMO","THR",
     *   "MET","NEG","CON","DEF","X1G","X2G","NOS","NOP","TAP","CPL",
     *   "SUM","PRD","DIF","RAT","WAV","GEN","VGB","VPB",13*"   "/
      DATA AMFG / "OWN","AMP","AVA","CTC","DXL","FAI","FUJ","HPM","MBI",
     *   "MSC","NEC","PLE","RTN","SIX","TRW","VAR","MAI"/
      DATA NMFG / 17 /
      DATA COL2 / "PA","SE","AA","BB","CC","DD","EE","FF","GG","HH","II"
     *   ,"JJ","KK","LL","MM","NN","OO","PP","QQ","RR","SS","TT","UU",
     *   "VV","WW","XX","YY","ZZ","S1","S2","YM","OR","IR","VG","TR",
     *   "CU","VO","IM","RN","NC","PO","NE","S3","S4","S5","S6","T2",
     *   "T3","T4","SL","MS","CV","VC","B1","B2","S7","S8",3*"  "/
      DATA WAV / "WAV"/
      DATA PLO / "PLO"/
      DATA S4,IR,XMET,APCT,OWN / "S4","IR","MET","@B","OWN"/
C
      IF (KODE.GT.1.AND.KODE.LT.5) GO TO 1510
      IF (IMA1.EQ.1) GO TO 1500
      CUNIT = 1.0
      IPNOPR = 0
      NBANK = 0
      IPRI = 0
      NBAND = 0
      NFOPT = 0
      NF = 0
      DO 10 I = 1, 2
         DO 10 J = 1, 4
            OUT(I,J) = 0.0
   10 CONTINUE
      DO 20 I = 1, 15
         GR(I) = 0.0
         KBANK(I) = 0
         IP1(I) = 1
         IP2(I) = 1
   20 CONTINUE
      DO 30 I = 1, 30
         AAA(I,2,2) = CMPLX(0.,0.)
         AAA(I,1,1) = CMPLX(0.,0.)
   30 CONTINUE
      DO 40 I = 1, 3
         IBAND(I) = 0
         ILTGT(I) = 1
         DO 40 J = 1, 6
```

```
            WI(I,J) = 0.0
   40 CONTINUE
      DO 50 I = 1, 70
         RIND(I) = 0.0
         DO 50 J = 1, 6
            DATX(I,J) = 0.0
   50 CONTINUE
      DO 60 I = 1, 15
         DO 60 J = 1, 3
            PERUP(J,I) = 0.0
   60 CONTINUE
      NBB = 0
      AOPT = 0.0
C
C *****DELAYP = DELAY VALUE OF PREVIOUS FREQUENCY
C
      DELAYP = 0.0
      ERRD = 0.0
      ERSET = 0.0
      ERNO = 0.0
      NOLINE = 1
      NOOPT = 0
      ITER = 0
      MIXPR = 0
      MODEL = 0
      MOD = 0
      CAIND = 0
      MCODE = 0
      NCO = 0
      INDF = 0
      INDTR = 0
      NSTAR = 0
      NT = 0
      KLM = 0
      KYM = 0
      NOIT = 0
      MODE = 0
      NIMP = 0
      KIMP = 0
      INRJX = 0
      JSTI = 0
      JSOI = 0
      ITHR = 0
      IND3R = 0
      MODIM = 0
      INEXP = 0
      NEWEX = 2
      IDIV = 0
      ERRAN = 0.0
      JERRX = 0
      KSIN = 0
      KSOUT = 0
      IRJXFL = 0
      NOISE = 0
      ERINI = 1.0E12
```

```
      AMET = 1.
      IHOLINE = 0
      IFTLER = 0
      ERR = 0.
      KKSIN = 0
      NOPT = 0
      IVG = 0
      IPINT = 0
      ISTRIP = 0
      IWAV1 = 0
      IWAV2 = 0
      IOPTCNT = 0
      IBUF2 = 0
      INFNEG = 0
      FPERT = 0.0
      INTOL=0
      DO 63 I=1,6
      KXTOL(I)=0
      KYTOL(I)=0
      PERTOL(1,I)=0.0
      PERTOL(2,I)=0.0
63    CONTINUE
C
C******READ CIRCUIT DATA FROM FILE - "END" TERMINATES
C
      DO 460 KI = 1, 70
   70    I = KI
         INONV=0
         CALL MODSIN (IFR,NOITEM)
         IF (NOITEM.LE.0) GO TO 470
C
C******SET UP FOR PLOT
C
         IF (ARRAY(ITRAP+1).NE.PLO) GO TO 80
         ARRAY(ITRAP+1) = PRI
         NPLOT = I
         IPNOPR = 1
C
C*****MOVE DATA INTO "DATX" AND "PERUP" ARRAYS
C
   80    ILOC = 3+ITRAP
         IOPTFLG = 0
         KO = 0
   90    ILOC = ILOC+1
         IF (ILOC.GT.NOITEM) GO TO 220
         IF (IARRAY(ILOC)) 120 , 100 , 170
C
C-----ACCEPT DATA FOR "DATX"
C
  100    IF (IOPTFLG.EQ.2) GO TO 180
         KO = KO+1
         IF (ARRAY(ILOC).GE.0.0) GO TO 110
         MODE = MODE+1
         IF (MODE.GT.15) GO TO 190
         KX(MODE) = I
```

```
          KY(MODE) = KO
          DATX(I,KO) = -ARRAY(ILOC)
          IOPTFLG = 1
          GO TO 90
  110     DATX(I,KO) = ARRAY(ILOC)
          IF (DATX(I,KO).EQ.0.0) DATX(I,KO) = 1.E-20
          IOPTFLG = 0
          GO TO 90
C
C-----ACCEPT DATA FOR "PERUP"
C

  120     IF (IOPTFLG.NE.0) GO TO 122
          INONV=1
          IOPTFLG=1
  122     GO TO (130,140,150) IOPTFLG

  130     IOPTCNT = IOPTCNT+1
          PERUP1 = IOPTCNT
          PERUP3 = ARRAY(ILOC)
          IOPTFLG = 2
          GO TO 90

  140     IF (IARRAY(ILOC).LT.-1) GO TO 170
          PERUP2 = ARRAY(ILOC)
          IOPTFLG = 3
          INXT=ILOC+1
          IF (INXT.LE.NOITEM.AND.IARRAY(INXT).LT.0) GOTO 90
* MR: AT THIS POINT THE DATA FILE LINE HAS EITHER:
*     1) ENDED WITH THE LAST 2 BRACKETED VALUES, OR
*     2) HAS NOT ENDED BUT THE NEXT VALUE IS NOT BRACKETED
          IF (INONV.EQ.1) GOTO 200
          PERUP(1,MODE) = PERUP1
          PERUP(2,MODE) = PERUP2
          PERUP(3,MODE) = PERUP3
          GOTO 90

  150     IF (ARRAY(ILOC).NE."@B".AND.ARRAY(ILOC).NE."B") GO TO 162
          IF (INONV.EQ.1) GOTO 200
          PERUP(1,MODE) = PERUP1
          PERUP(2,MODE) = PERUP2
          PERUP(3,MODE) = PERUP3
  160     PERUP(2,MODE) = (1.+.01*PERUP(2,MODE))*DATX(I,KO)
          PERUP(3,MODE) = (1.-.01*PERUP(3,MODE))*DATX(I,KO)
          GOTO 168

  162     IF (ARRAY(ILOC).NE."%T".AND.ARRAY(ILOC).NE."T") GOTO 210
          IOPTCNT = IOPTCNT-1
          INTOL = INTOL+1
          IF (INTOL.GT.6) GOTO 195
          KXTOL(INTOL) = I
          KYTOL(INTOL) = KO
          IRATOL(INTOL) = 0
          IF (ARRAY(ILOC).EQ."T") IRATOL(INTOL) = 1
          PERTOL(1,INTOL) = PERUP2
```

```
          PERTOL(2,INTOL) = PERUP3

  168     IOPTFLG = 0
          GO TO 90
C
C------ERROR STATEMENTS
C
  170     WRITE (IOT,1650) (IBUF(II),II=1,LDC)
          GO TO 1640
  180     WRITE (IOT,1660) (IBUF(II),II=1,LDC)
          GO TO 1640
  190     WRITE (IOT,1670)
          GO TO 1640
  195     WRITE (IOT,1675)
          GO TO 1640
  200     WRITE (IOT,1680) (IBUF(II),II=1,LDC)
          GO TO 1640
  210     WRITE (IOT,1690) (IBUF(II),II=1,LDC)
          GO TO 1640
  220     NOPT = MODE
          DC1(I) = ARRAY(ITRAP+1)
          DC2(I) = ARRAY(ITRAP+2)
          DC3(I) = ARRAY(ITRAP+3)
          IF (ITRAP.EQ.1) DCO(I) = ARRAY(ITRAP)
          IF (ARRAY(ITRAP+1).EQ.PRI) GO TO 230
C       IF(ARRAY(ITRAP+1).EQ.PRI.AND.NPLOT.EQ.I)GOTO 96006
          GO TO 270
  230     DO 260 J = 1, NCO
             IF (DC1(J).NE.PRI) GO TO 260
             IF (DC2(J).NE.ARRAY(ITRAP+2)) GO TO 250
             IF (NPLOT.NE.I.AND.NPLOT.NE.J) GO TO 270
             NPLOT = J
             IPNOPR = 0
             DO 240 K = 1, 6
                DATX(I,K) = 0.0
  240        CONTINUE
             GO TO 70
  250        MIXPR = 1
  260     CONTINUE
          GO TO 270
C
C******SET FLAGS AND COUNTERS DEPENDING ON CIRCUIT ELEMENT
C
  270     IF (DC1(I).EQ.PRI) GO TO 340
          IF (DC1(I).EQ.THRI) ITHR = ITHR+1
          IF (DC1(I).EQ.ANOF.OR.DC1(I).EQ.ANOS.OR.DC1(I).EQ.ANOP) NOISE =
     *       I
          IF (DC1(I).EQ.SETI.OR.DC1(I).EQ.SMTI) MODEL = 1
          IF (DC1(I).EQ.SETI.OR.DC1(I).EQ.SMTI) JSTI = I
          IF (DC1(I).EQ.SEOI.OR.DC1(I).EQ.SMOI) MODEL = 1
          IF (DC1(I).EQ.ANOF) NOOPT = NOOPT+1
          IF (DC1(I).EQ.SETI.OR.DC1(I).EQ.SMTI) MOD = 1
          IF (DC1(I).EQ.SEOI.OR.DC1(I).EQ.SMOI) JSOI = I
          IF (DC1(I).EQ.ZIMP.OR.DC1(I).EQ.AONE) NIMP = NIMP+1
          IF (DC1(I).EQ.SEOI) MODIM = 1
```

```
         IF (DC1(I).EQ.SMOI) MODIM = 2
         IF (DC1(I).EQ.RJX) IRJXFL = 1
         IF (DC1(I).EQ.RJX) INRJX = I
         IF (DC1(I).EQ.XMET) AMET = 1./0.0254
         IF (DC1(I).EQ.WAV.AND.IWAV1.EQ.0) IWAV1 = I
         IF (DC1(I).EQ.WAV.AND.IWAV1.NE.0) IWAV2 = I
         IF (DC1(I).NE.TWO) GO TO 300
C
C  **TWO-PORT DATA WILL BE GIVEN FOR ELEMENT "TWO".  INCREASE
C    TRANSISTOR INDEX NT**
C
         KKSIN = KSIN
         IF (DC3(I).EQ.S1) KSIN = 1
         IF (DC3(I).EQ.S2) KSIN = 2
         IF (DC3(I).EQ.S3) KSIN = 3
  280    IF (KKSIN.EQ.KSIN.OR.KKSIN.EQ.0) GO TO 290
         WRITE (IOT,1700)
         GO TO 1640
  290    IF (DC3(I).EQ.YM) KYM = 1
         GO TO 330
C
C  **IS COL. 1 CODE "OWN" OR A MANUFACTURER"S CODE\**
C  **IF SO, ASSIGN VALUE 1 TO IMFG**
C
  300    CONTINUE
         IMFG = 0
         TEST = DC1(I)
         DO 310 II = 1, NMFG
            IF (TEST.EQ.AMFG(II)) GO TO 320
  310    CONTINUE
         GO TO 340
  320    IMFG = 1
         IBANKF = 1
         KSIN = 1
         GO TO 280
  330    NT = NT+1
         KBANK(NT) = I
  340    IF (DC1(I).EQ.PRI) IPRI = I
         IF (DC1(I).EQ.END1) GO TO 480
         IF (IPRI.EQ.0) GO TO 350
         IF (DATX(IPRI,3).NE.0.0.AND.DC3(IPRI).EQ.IR) AOPT = DATX(IPRI,3
     *       )
  350    NCO = NCO+1
         IF (IMFG.EQ.1) J = 17
         IF (IMFG.EQ.1) GO TO 370
         DO 360 J = 1, 80
C
C   **CONVERT COLUMN ONE TO NUMBER CODES AND STORE IN INC1**
C
            IF (DC1(I).EQ.COL1(J)) GO TO 370
  360    CONTINUE
         WRITE (IOT,1990) (IBUF(II),II=1,LDC)
         WRITE (IOT,1710)
         GO TO 1640
  370    INC1(I) = J
```

```
         IMFG = 0
         IF (J.EQ.51.AND.INFNEG.EQ.0) WRITE (IOT,1720)
         IF (J.EQ.51) INFNEG = 1
C
C   **CONVERT COLUMN TWO TO NUMBER CODES AND STORE IN INC2**
C
         DO 380 J = 1, 60
            IF (DC2(I).EQ.COL2(J)) GO TO 390
  380    CONTINUE
         WRITE (IOT,1990) (IBUF(II),II=1,LDC)
         WRITE (IOT,1730)
         GO TO 1640
  390    INC2(I) = J
         IF (J.EQ.55) IBUF2 = 1
C
C    **CONVERT COLUMN THREE TO NUMBER CODES AND STORE IN INC3**
C
         DO 400 J = 1, 60
            IF (DC3(I).EQ.COL2(J)) GO TO 410
  400    CONTINUE
         WRITE (IOT,1990) (IBUF(II),II=1,LDC)
         WRITE (IOT,1740)
         GO TO 1640
  410    INC3(I) = J
C
C *****CHECK FOR VALID "DEF"
C
         IF (INC1(I).NE.53) GO TO 420
         IF (INC3(I).NE.48.AND.INC3(I).NE.49) GO TO 420
         IF (INC2(I).EQ.54.OR.INC2(I).EQ.55) GO TO 420
         WRITE (IOT,1990) (IBUF(II),II=1,LDC)
         WRITE (IOT,1750)
         GO TO 1640
C
C *****CHECK FOR LOSS DATA IN TRL OST SST
C *****ISTRIP=1 FOR DIE. ISTRIP=2 FREQ.
C
  420    IF (ISTRIP.NE.0.) GO TO 460
         IF (INC1(I).LT.19.OR.INC1(I).GT.21) GO TO 460
         IF (INC3(I).GT.3) GO TO 440
         IF (DATX(I,3).GT.30.) GO TO 430
         IF (DATX(I,4).LE.0.) GO TO 460
         ISTRIP = 1
         FREQ = FRST(1)
         GO TO 460
  430    IF (DATX(I,4).LE.0.) GO TO 460
         ISTRIP = 2
         FREQ = DATX(I,3)
         GO TO 460
  440    IF (DATX(I,3).GT.30.) GO TO 450
         IF (DATX(I,5).LE.0.) GO TO 460
         ISTRIP = 1
         FREQ = FRST(1)
         GO TO 460
  450    IF (DATX(I,6).LE.0.) GO TO 460
```

```
         ISTRIP = 2
         FREQ = DATX(I,3)
  460 CONTINUE
  470 WRITE (IOT,1780)
      GO TO 1640
C
C    **UPDATE "DATX" WITH RESULTS OF COARSE SEARCH**
C
  480 IDCLOC = I
C
C  **IF "B2" IS USED, BE SURE THAT "WW-ZZ" ARE NOT USED**
C
      IF (IBUF2.EQ.0) GO TO 500
      DO 490 II = 1, I
         IF (INC2(II).LT.25.OR.INC2(II).GT.28) GO TO 490
         WRITE (IOT,1760)
         GO TO 1640
  490 CONTINUE
  500 CONTINUE
      IF (IMA.EQ.1.AND.KODE.NE.5) GO TO 510
      GO TO 530
  510 DO 520 I = 1, NOPT
         DATX(KX(I),KY(I)) = VAX(I)
  520 CONTINUE
  530 IF (JSOI.EQ.0) GO TO 540
      JSOK = INC3(JSOI)
  540 IF (JSTI.EQ.0) GO TO 550
      JSTK = INC3(JSTI)
  550 IF (KODE.EQ.6.AND.IRJXFL.EQ.0) GO TO 590
      IF (MODEL.EQ.1.AND.KODE.EQ.6) GO TO 580
      IF (KODE.NE.6.AND.IRJXFL.EQ.1) GO TO 590
      IF (IPRI.NE.0) GO TO 600
      WRITE (IOT,1770)
      GO TO 1640
  560 WRITE (IOT,1790)
      GO TO 1640
  570 WRITE (IOT,1800)
      GO TO 1640
  580 WRITE (IOT,1810)
      GO TO 1640
  590 WRITE (IOT,1820)
      GO TO 1640
C
C    ** READ INITIAL, FINAL AND STEP FREQUENCIES
C
  600 IF (KSIN.NE.0.AND.KYM.EQ.1) GO TO 560
      NF = NF+1
      L = NF+14
      CALL MODSIN (IFR,NOITEM)
      IF (NOITEM.LE.0) GO TO 710
      IDCLOC = IDCLOC+1
      IF (ITRAP.EQ.1) DCO(IDCLOC) = ARRAY(ITRAP)
      IF (NOITEM.LE.ITRAP+15) GO TO 610
      WRITE (IOT,1830)
      GO TO 1640
```

```
  610 IF (ARRAY(ITRAP+1).EQ.END1) GO TO 730
      IF (NBAND.GT.3) GO TO 710
      IILOOP = 0
      DO 620 ILOOP = NF, L
         IILOOP = IILOOP+1
         FRST(ILOOP) = ARRAY(ITRAP+IILOOP)
  620 CONTINUE
      NBAND = NBAND+1
      IF (FRST(NF).EQ.0.0) GO TO 700
      IF (FRST(NF+1).EQ.0.) GO TO 690
      IF (FRST(NF+2).EQ.0.) GO TO 670
      IF (FRST(NF+3).EQ.0.0) GO TO 650
      M = NF+4
      DO 630 KI = M, L
         IF (FRST(KI).EQ.0.0) GO TO 640
  630 CONTINUE
      NF = 15
      GO TO 690
  640 NF = KI-1
      GO TO 690
  650 IF (FRST(NF+2).GT.FRST(NF+1)) GO TO 680
      FI = FRST(NF)
      FF = FRST(NF+1)
      FS = FRST(NF+2)
      NFR = IFIX((FF-FI)/FS+.0001)+1
      QA = FI-FS
      NK = NF+NFR-1
      IF (NFR.GT.30.OR.NK.GT.30) GO TO 570
      DO 660 KI = NF, NK
         QA = QA+FS
         FRST(KI) = QA
  660 CONTINUE
      NF = NK
      GO TO 690
  670 NF = NF+1
      GO TO 690
  680 NF = NF+2
  690 CONTINUE
C
C  **STORE THE NUMBER OF FREQUENCIES OF EACH BAND IN "IBAND"**
C
      IBAND(NBAND) = NF
      GO TO 600
  700 WRITE (IOT,1840)
      GO TO 1640
  710 WRITE (IOT,1850)
      GO TO 1640
  720 WRITE (IOT,1860) NF
      GO TO 1640
  730 IF (NBAND.LE.0) GO TO 700
      NF = NF-1
      IF ((NOISE.NE.0).AND.(NF.GT.10)) GO TO 720
      NFOPT = NF
      FF = FRST(NF)
      IF (NF.EQ.1.OR.IBANKF.EQ.0.OR.ISTRIP.GE.1) GO TO 750
```

```
C
C      IF(NF.EQ.1.OR.(IBANKF.EQ.0.AND.ISTRIP.EQ.0)) GOTO 1313
C
      DO 740 IX1 = 2, NF
         IF (FRST(IX1).GT.FRST(IX1-1)) GO TO 740
         WRITE (IOT,1870) (FRST(IX2),IX2=1,NF)
         GO TO 1640
  740 CONTINUE
  750 IF (ISTRIP.EQ.0.) GO TO 760
      IF (ISTRIP.EQ.1.) FREQ = FRST(1)
      WRITE (IOT,1880) FREQ
  760 CONTINUE
      IF (NOOPT.EQ.0) GO TO 840
C
C  ** READ NOISE DATA FOR "NOF" CODE **
C
      NOLINE = NF*NOOPT+1
      K = 1
      IF (NOOPT.LE.2) GO TO 770
      WRITE (IOT,1890)
      GO TO 1640
  770 CONTINUE
      DO 830 I = 1, NOLINE
         JJ = I
         IF (I.GT.NF) K = 2
         IF (I.GT.NF) JJ = I-NF
         CALL MODSIN (IFR,NOITEM)
         IF (NOITEM.GT.0) GO TO 790
  780    WRITE (IOT,1900)
         WRITE (IOT,1910) NF
         GO TO 1640
  790    IDCLOC = IDCLOC+1
         IF (ITRAP.EQ.1) DC0(IDCLOC) = ARRAY(ITRAP)
         IF (NOITEM.EQ.(5+ITRAP)) GO TO 810
         IF (ARRAY(ITRAP+1).EQ.END1.AND.I.EQ.NOLINE) GO TO 830
         IF (ARRAY(ITRAP+1).EQ.END1) GO TO 780
         IF (I.LT.NOLINE) GO TO 800
         WRITE (IOT,1920)
         GO TO 1640
  800    WRITE (IOT,1930)
         GO TO 1640
  810    DO 820 J = 1, 5
            FIG(K,JJ,J) = ARRAY(ITRAP+J)
            FIGG(K,JJ,J) = FIG(K,JJ,J)
  820    CONTINUE
  830 CONTINUE
C
C   **DETERMINE DESIRED PRINT OPTION AND ASSIGN VALUE TO NSTAR**
C
  840 NSTAR = 35-INC3(IPRI)
      IF (INC3(IPRI).EQ.43) NSTAR = 8
      IF (INC3(IPRI).GT.43.AND.INC3(IPRI).LT.47) NSTAR = 5
      IF (INC3(IPRI).EQ.56) NSTAR = 9
      IF (INC3(IPRI).EQ.57) NSTAR = 10
      IF (NSTAR.EQ.10) DC3(IPRI) = S3
```

```
      IF (NSTAR.EQ.10) INC3(IPRI) = 43
      IF (NSTAR.EQ.9) FPERT = DATX(IPRI,2)
      IF (NSTAR.NE.1) GO TO 850
      ITEST = DATX(IPRI,3)
      FPERT = DATX(IPRI,4)
      IF (ITEST.EQ.1.OR.ITEST.EQ.2.OR.ITEST.EQ.3) IVG = ITEST
      IF (IVG.NE.0) GO TO 850
      ITEST = DATX(IPRI,4)
      FPERT = DATX(IPRI,3)
      IF (ITEST.EQ.1.OR.ITEST.EQ.2.OR.ITEST.EQ.3) IVG = ITEST
      IF (IVG.EQ.0) FPERT = DATX(IPRI,3)
  850 CONTINUE
C
C     WRITE(IOT,)IVG,FPERT
C
      IF (NSTAR.NE.8) GO TO 890
      DATX(IPRI,1) = 50
      DO 880 I = 1, NF
         CALL MODSIN (IFR,NOITEM)
         IDCLOC = IDCLOC+1
         IF (ITRAP.EQ.1) DCO(IDCLOC) = ARRAY(ITRAP)
         IF (NOITEM.EQ.(4+ITRAP)) GO TO 860
         WRITE (IOT,1990) (IBUF(III),III=1,LDC)
         WRITE (IOT,1940)
         GO TO 1640
  860    DO 870 J = 1, 4
            SLIMP(I,J) = ARRAY(ITRAP+J)
  870    CONTINUE
  880 CONTINUE
      CALL MODSIN (IFR,NOITEM)
      IDCLOC = IDCLOC+1
      IF (ITRAP.EQ.1) DCO(IDCLOC) = ARRAY(ITRAP)
      IF (ARRAY(ITRAP+1).EQ.END1) GO TO 900
      WRITE (IOT,1950)
      GO TO 1640
  890 IF (NSTAR.NE.10) GO TO 900
      NSTAR = 8
      DATX(IPRI,1) = 50.
C
C   ** IF ITHR=0 DO NOT READ THREE PORT S-DATA **
C
  900 IF (ITHR.EQ.0) GO TO 1000
      NTHR = NF*ITHR
      N3THR = 3*NTHR
      N = 1
      K = 1
      INF = 1
      JNF = 1
C
C   ** READ POLAR 3-PORT PARAMETERS**
C
      NOLINE = N3THR+1
      DO 990 I = 1, NOLINE
         CALL MODSIN (IFR,NOITEM)
         IF (NOITEM.GT.0) GO TO 920
```

```
  910     WRITE (IOT,1960)
          WRITE (IOT,1970) N3THR,NF,ITHR
          GO TO 1640
  920     IDCLOC = IDCLOC+1
          IF (ITRAP.EQ.1) DC0(IDCLOC) = ARRAY(ITRAP)
          IF (NOITEM.EQ.(6+ITRAP)) GO TO 940
          IF (ARRAY(ITRAP+1).EQ.END1.AND.I.EQ.NOLINE) GO TO 990
          IF (ARRAY(ITRAP+1).EQ.END1) GO TO 910
          IF (I.LT.NOLINE) GO TO 930
          WRITE (IOT,1980)
          GO TO 1640
  930     WRITE (IOT,1990) (IBUF(II),II=1,LDC)
          WRITE (IOT,2010)
  940     DO 950 JL = 1, 6
             S3P(N,JL) = ARRAY(ITRAP+JL)
  950     CONTINUE
          IF (N.NE.3) GO TO 990
          DO 960 L = 1, 3
             DO 960 M = 1, 3
                PPRD = S3P(L,2*M)*RPD
                S3R(K,L,M) = CMPLX(S3P(L,2*M-1)*COS(PPRD),S3P(L,2*M-1)*
     *             SIN(PPRD))
  960     CONTINUE
          IF (INF.NE.NF) GO TO 970
          JNF = JNF+1
          K = JNF
          INF = 1
          GO TO 980
  970     K = K+ITHR
          INF = INF+1
  980     N = 0
  990     N = N+1
C
C******PROCESS "TWO" OR DATA BANK ELEMENTS
C
 1000 IF (NT.EQ.0) GO TO 1240
      NSD = NF*NT
      IF (NSD.GT.30) GO TO 1060
      DO 1220 I = 1, NT
         K = I
         IF (DC1(KBANK(I)).NE.TWO) GO TO 1200
         IF (KYM.EQ.1) GO TO 1150
         GO TO (1010,1020,1020), KSIN
 1010    NOLINE = NF
         GO TO 1030
 1020    NOLINE = KSIN-1
C
C******READ S-PARAMETERS FROM DATA FILE - PROCESS S1 OPTION
C
 1030    DO 1080 J = 1, NOLINE
            CALL MODSIN (IFR,NOITEM)
            IF (NOITEM.GT.0) GO TO 1040
            WRITE (IOT,2020)
            GO TO 1640
 1040       IDCLOC = IDCLOC+1
```

```
            IF (ITRAP.EQ.1) DCO(IDCLOC) = ARRAY(ITRAP)
            IF (NOITEM.EQ.(8+ITRAP)) GO TO 1070
            IF (ARRAY(ITRAP+1).EQ.END1) GO TO 1050
            WRITE (IOT,2030) (IBUF(II),II=1,LDC)
            GO TO 1640
 1050       WRITE (IOT,2040)
            GO TO 1640
 1060       WRITE (IOT,2050)
            GO TO 1640
 1070       PS(K,1,1) = ARRAY(ITRAP+1)
            PS(K,1,2) = ARRAY(ITRAP+2)
            PS(K,2,1) = ARRAY(ITRAP+3)
            PS(K,2,2) = ARRAY(ITRAP+4)
            PS(K,1,3) = ARRAY(ITRAP+5)
            PS(K,1,4) = ARRAY(ITRAP+6)
            PS(K,2,3) = ARRAY(ITRAP+7)
            PS(K,2,4) = ARRAY(ITRAP+8)
            K = K+NT
 1080    CONTINUE
         GO TO (1220,1090,1120), KSIN
C
C******PROCESS S-PARAMETERS WITH S2 OPTION
C
 1090    DO 1110 J = 2, NF
            DO 1100 L = 1, 2
               DO 1100 M = 1, 4
                  PS(K,L,M) = PS(I,L,M)
 1100       CONTINUE
            K = K+NT
 1110    CONTINUE
         GO TO 1220
C
C******PROCESS S-PARAMETERS WITH S3 OPTION
C
 1120    BW = FRST(NF)-FRST(1)
         FI = FRST(1)
         JEND = I+NT
         DO 1130 J = I, JEND, NT
            DO 1130 L = 1, 2
               DO 1130 M = 1, 2
                  PPRD = RPD*PS(J,L,2*M)
                  M1 = 2*M-1
                  BBB(J,L,M) = CMPLX(PS(J,L,M1)*COS(PPRD),PS(J,L,M1)*SIN
     *                  (PPRD))
 1130    CONTINUE
         DO 1140 J = 1, 2
            DO 1140 L = 1, 2
               SLR = (REAL(BBB(I+NT,J,L))-REAL(BBB(I,J,L)))/BW
               SLI = (AIMAG(BBB(I+NT,J,L))-AIMAG(BBB(I,J,L)))/BW
               DO 1140 M = 2, NF
                  XR = REAL(BBB(I,J,L))+SLR*(FRST(M)-FI)
                  XI = AIMAG(BBB(I,J,L))+SLI*(FRST(M)-FI)
                  N = (M-1)*NT+I
                  BBB(N,J,L) = CMPLX(XR,XI)
 1140    CONTINUE
```

```
         GO TO 1220
C
C******READ AND PROCESS Y-PARAMETERS
C
 1150    DO 1190 J = 1, NF
            CALL MODSIN (IFR,NOITEM)
            IF (NOITEM.GT.0) GO TO 1160
            WRITE (IOT,2060)
            GO TO 1640
 1160       IDCLOC = IDCLOC+1
            IF (ITRAP.EQ.1) DC0(IDCLOC) = ARRAY(ITRAP)
            IF (NOITEM.EQ.(8+ITRAP)) GO TO 1180
            IF (ARRAY(ITRAP+1).EQ.END1) GO TO 1170
            WRITE (IOT,2070) (IBUF(II),II=1,LDC)
            GO TO 1640
 1170       WRITE (IOT,2080)
            GO TO 1640
 1180       YMAT(K,1,1) = CMPLX(ARRAY(ITRAP+1),ARRAY(ITRAP+2))
            YMAT(K,2,1) = CMPLX(ARRAY(ITRAP+3),ARRAY(ITRAP+4))
            YMAT(K,1,2) = CMPLX(ARRAY(ITRAP+5),ARRAY(ITRAP+6))
            YMAT(K,2,2) = CMPLX(ARRAY(ITRAP+7),ARRAY(ITRAP+8))
            K = K+NT
 1190    CONTINUE
         GO TO 1220
C
C******READ AND PROCESS S-PARAMETERS FROM EXTERNAL DATA BANK
C
 1200    IF (NF.LE.20) GO TO 1210
         WRITE (IOT,2090)
         GO TO 1640
 1210    CALL MFGSUB (DC1(KBANK(I)),DC3(KBANK(I)),I,ZCH)
         IF (JERRX.EQ.1) GO TO 1640
         DATX(KBANK(I),1) = 50.0
         IF (DC1(KBANK(I)).EQ.OWN) DATX(KBANK(I),1) = ZCH
         IF (DATX(KBANK(I),1).EQ.0.0) DATX(KBANK(I),1) = 50.0
         DC1(KBANK(I)) = TWO
         DC3(KBANK(I)) = S1
         INC1(KBANK(I)) = 17
         INC3(KBANK(I)) = 29
 1220 CONTINUE
      IF (KSIN.EQ.2) KSIN = 1
      CALL MODSIN (IFR,NOITEM)
      IF (NOITEM.GT.0) GO TO 1230
      WRITE (IOT,2100)
      GO TO 1640
 1230 IDCLOC = IDCLOC+1
      IF (ITRAP.EQ.1) DC0(IDCLOC) = ARRAY(ITRAP)
      IF (ARRAY(ITRAP+1).EQ.END1) GO TO 1240
      WRITE (IOT,2110) (IBUF(II),II=1,LDC)
      GO TO 1640
C
C******PROCESS "IMP" AND "ONE" ELEMENTS
C
 1240 IF (KODE.NE.6.OR.IPRIN.GT.0) GO TO 1250
      WRITE (IOT,2120)
```

```
 1250 IF (NIMP.EQ.0) GO TO 1320
C
C   **READ DATX FOR TWO-PORT "IMP" OR ONE-PORT "ONE" **
C
      NFIMP = NIMP*NF
      INF = 1
      JNF = 1
      K = 1
      NOLINE = NFIMP+1
      DO 1310 I = 1, NOLINE
         CALL MODSIN (IFR,NOITEM)
         IF (NOITEM.GT.0) GO TO 1270
 1260    WRITE (IOT,2130)
         WRITE (IOT,2140) NFIMP,NF,NIMP
         GO TO 1640
 1270    IDCLOC = IDCLOC+1
         IF (ITRAP.EQ.1) DCO(IDCLOC) = ARRAY(ITRAP)
         IF (NOITEM.EQ.(2+ITRAP).AND.I.LT.NOLINE) GO TO 1290
         IF (ARRAY(ITRAP+1).EQ.END1.AND.I.EQ.NOLINE) GO TO 1310
         IF (ARRAY(ITRAP+1).EQ.END1) GO TO 1260
         IF (NOITEM.NE.(ITRAP+2)) GO TO 1280
         WRITE (IOT,2150)
         GO TO 1640
 1280    WRITE (IOT,1990) (IBUF(II),II=1,LDC)
         WRITE (IOT,2160)
         GO TO 1640
 1290    VIMP(K,1) = ARRAY(ITRAP+1)
         VIMP(K,2) = ARRAY(ITRAP+2)
         IF (INF.NE.NF) GO TO 1300
         JNF = JNF+1
         K = JNF
         INF = 1
         GO TO 1310
 1300    K = K+NIMP
         INF = INF+1
 1310 CONTINUE
C
C *****READ ERROR FUNCTION GOAL AND WEIGHTING FACTORS
C
C *****ERROR FUNCTION GOAL
C
 1320 IF (KODE.NE.1.AND.KODE.NE.5) GO TO 1510
      CALL MODSIN (IFR,NOITEM)
      IF (NOITEM) 1420 , 1420 , 1330
 1330 IDCLOC = IDCLOC+1
      IF (ITRAP.EQ.1) DCO(IDCLOC) = ARRAY(ITRAP)
      IF (NOITEM.NE.(1+ITRAP)) GO TO 1430
      CONV = ARRAY(1+ITRAP)
C
C *****WEIGHTING FACTORS - ONE PER FREQUENCY BAND
C
      KKI = NBAND+1
      DO 1400 KI = 1, KKI
         CALL MODSIN (IFR,NOITEM)
         IF (NOITEM) 1420 , 1420 , 1340
```

```
 1340     IDCLOC = IDCLOC+1
          IF (ITRAP.EQ.1) DCO(IDCLOC) = ARRAY(ITRAP)
          IF (ARRAY(1+ITRAP).EQ.END1) GO TO 1420
          IF (KI.EQ.KKI) GO TO 1410
          IF (NOITEM.LT.(ITRAP+4).OR.NOITEM.GT.(ITRAP+6)) GO TO 1440
C
C *****CHECK AND STORE FIRST FOUR FACTORS
C
          DO 1360 I = 1, 4
             IF (IARRAY(ITRAP+I).NE.0) GO TO 1450
             IF (I.EQ.4) GO TO 1350
             WI(KI,I) = ARRAY(ITRAP+I)
             GO TO 1360
 1350        S21(KI) = ARRAY(ITRAP+4)
 1360     CONTINUE
C
C ******IS FIFTH FACTOR "LT" OR "GT"
C
          IF (NOITEM.EQ.(ITRAP+4)) GO TO 1400
          IF (IARRAY(ITRAP+5).EQ.0) GO TO 1390
          IF (ARRAY(ITRAP+5).EQ.AALT) GO TO 1370
          IF (ARRAY(ITRAP+5).NE.AAGT) GO TO 1460
          ILTGT(KI) = 3
          GO TO 1380
 1370     ILTGT(KI) = 2
C
C *****FIFTH OR SIXTH ITEM MAY BE GROUP DELAY FACTOR
C
 1380     IF (NOITEM.EQ.(ITRAP+5)) GO TO 1400
          IF (IARRAY(ITRAP+6).NE.0) GO TO 1470
          WI(KI,6) = ARRAY(ITRAP+6)
          GO TO 1400
 1390     WI(KI,6) = ARRAY(ITRAP+5)
          IF (NOITEM.EQ.(ITRAP+6)) GO TO 1480
 1400 CONTINUE
C
C *****ERROR STATEMENTS
C
 1410 WRITE (IOT,2170) NBAND
      GO TO 1630
 1420 IF (KI.EQ.KKI) GO TO 1490
      WRITE (IOT,2180) NBAND
      GO TO 1630
 1430 WRITE (IOT,2000) (IBUF(II),II=1,LDC)
      WRITE (IOT,2190)
      GO TO 1630
 1440 WRITE (IOT,2000) (IBUF(II),II=1,LDC)
      WRITE (IOT,2200)
      GO TO 1630
 1450 WRITE (IOT,2000) (IBUF(II),II=1,LDC)
      WRITE (IOT,2210)
      GO TO 1630
 1460 WRITE (IOT,2000) (IBUF(II),II=1,LDC)
      WRITE (IOT,2220)
      GO TO 1630
```

```
 1470 WRITE (IOT,2000) (IBUF(II),II=1,LDC)
      WRITE (IOT,2230)
      GO TO 1630
 1480 WRITE (IOT,2000) (IBUF(II),II=1,LDC)
      WRITE (IOT,2240)
      GO TO 1630
 1490 CONTINUE
      NFOPT = IBAND(1)
      SUM2 = WI(2,1)+WI(2,2)+WI(2,3)
      SUM3 = WI(3,1)+WI(3,2)+WI(3,3)
      IF (SUM2.NE.0.0) NFOPT = IBAND(2)
      IF (SUM3.NE.0.0) NFOPT = IBAND(3)
 1500 IF (KODE.LT.3.AND.NOPT.EQ.0) GO TO 1530
      IF (KODE.EQ.5.AND.NOPT.EQ.0) GO TO 1530
      IF (JERRX.EQ.1) GO TO 1640
      IF (IPRIN.EQ.1) GO TO 1620
      IF (KODE.EQ.1.AND.MKODE.NE.1.AND.ISENS.EQ.0) GO TO 1620
      IF (KODE.EQ.1.AND.ISENS.EQ.0) WRITE (IOT,2250) NOPT
      IF (KODE.EQ.1.AND.ISENS.EQ.1.AND.MKODE.EQ.1) WRITE (IOT,2260) NOPT
 1510 IF (JERRX.EQ.1) GO TO 1640
      IF (IPRIN.GT.0) GO TO 1620
      IF (KODE.EQ.3.AND.MKODE.EQ.0) GO TO 1520
      IF (KODE.GT.1.AND.KODE.LT.6) GO TO 1620
      IF (IPNOPR.NE.0.AND.MIXPR.EQ.0) GO TO 1620
 1520 GO TO (1540,1550,1560,1570,1580,1580,1600,1610,1590), NSTAR
 1530 WRITE (IOT,2270)
      GO TO 1640
 1540 WRITE (IOT,2280) DATX(IPRI,1),DATX(IPRI,2)
      GO TO 1620
 1550 WRITE (IOT,2290) DATX(IPRI,1),DATX(IPRI,2)
      GO TO 1620
 1560 WRITE (IOT,2300) DATX(IPRI,1),DATX(IPRI,2)
      GO TO 1620
 1570 WRITE (IOT,2310)
      GO TO 1620
 1580 WRITE (IOT,2320) DATX(IPRI,1)
      GO TO 1620
 1590 WRITE (IOT,2330) DATX(IPRI,1)
      GO TO 1620
 1600 WRITE (IOT,2340)
      GO TO 1620
C
C  **PRINT HEADING FOR "S3" PRINT OPTION**
C
 1610 WRITE (IOT,2350)
      WRITE (IOT,2360) (FRST(J),(SLIMP(J,I),I=1,4),J=1,NF)
      WRITE (IOT,2370)
 1620 CONTINUE
      RETURN
 1630 CONTINUE
 1640 JERRX = 1
      RETURN
 1650    FORMAT (/" FOUND CHARACTERS WHERE NUMBERS EXPECTED"/136A1/)
 1660    FORMAT (/" NOT ENOUGH DATA IN PARENTHESES - 2 OR 3 REQ""D"/
     *      136A1/)
```

```
 1670    FORMAT (/" ONLY 15 VARIABLES ALLOWED"/)
 1680    FORMAT (/" LIMITS NOT ALLOWED ON NON-OPTIMIZED VARIABLES"/136A1
    *      /)
 1675    FORMAT (/" ONLY 6 TOLERANCED PARAMETERS ALLOWED"/)
 1690    FORMAT (/
    *      " 3RD PARAMETER IN PARENTHESES MUST BE ""@B"", ""B"",
    *      ", ""%T"" OR ""T""/
    *      136A1/)
 1700    FORMAT (" ALL TWO-PORT OPTIONS MUST BE THE SAME")
 1710    FORMAT (" NON-STANDARD CODE USED IN COLUMN 1")
 1720    FORMAT (" ** ""NEG"" IMPLEMENTATION HAS BEEN CHANGED,",
    *      " READ APPENDIX 9 FOR DETAILS **")
 1730    FORMAT (" NON-STANDARD CODE USED IN COLUMN 2")
 1740    FORMAT (" NON-STANDARD CODE USED IN COLUMN 3")
 1750 FORMAT (" THREE- AND FOUR-PORT DEFINE MUST BE PUT INTO ""B1"" OR "
    *"B2""")
 1760    FORMAT (" CANNOT USE WW,XX,YY OR ZZ WITH B2")
 1770 FORMAT (//" ERROR IN DATA FILE - NO PRINTOUT INSTRUCTION")
 1780 FORMAT (//," ERROR IN DATA FILE - #END# MISSING AFTER",
    *   " CIRCUIT DESCRIPTION ")
 1790 FORMAT (" MIXING Y AND S PARAMETERS IS ILLEGAL")
 1800 FORMAT (" NUMBER OF FREQUENCIES IS GREATER THAN 30")
 1810 FORMAT (" THE MAP OPTION IS NOT ALLOWED WHEN MODELS USED")
 1820 FORMAT (//," ""RJX"" ELEMENT MUST ONLY BE USED WITH ""MAP"" ",
    *   "OPTION"/)
 1830 FORMAT (" TOO MANY FREQUENCIES ON A LINE - 15 MAXIMUM")
 1840 FORMAT (//," MISSING FREQUENCY LINE IN DATA FILE")
 1850 FORMAT (" ""END"" IS MISSING IN DATA FILE AFTER FREQUENCY ","DATA"
    *   )
 1860 FORMAT (" ",I3," FREQUENCIES NOT ALLOWED FOR NOISE ANALYSIS",
    *   "--MAXIMUM IS 10")
 1870    FORMAT (" ALL FREQUENCIES MUST BE IN ASCENDING ORDER"/
    *      " CURRENT ORDER IS:"/9(1X,F8.0,1X,F8.0,1X,F8.0,1X,F8.0,1X,
    *      F8.0/))
 1880 FORMAT (/" TRANSMISSION LINE LOSS IN DATA FILE DEFINED"," FOR",
    *   F8.1," MHZ"/" LOSS INCREASES AS THE SQUARE ROOT",
    *   " OF FREQUENCY"/)
 1890 FORMAT (/," ONLY TWO ""NOF"" CODES ALLOWED   ")
 1900    FORMAT (/," INSUFFICIENT LINES OF NOISE FIGURE DATA")
 1910    FORMAT (" ",I3," LINES OF DATA REQUIRED BETWEEN SECOND ",
    *      " AND THIRD #END#")
 1920    FORMAT (" ""END"" IS MISSING AFTER NOISE FIGURE DATA")
 1930    FORMAT (/," WRONG AMOUNT OF DATA ON NOISE FIGURE LINE- 5,",
    *      "ITEMS REQUIRED")
 1940    FORMAT (" SOURCE AND LOAD SECTION MUST CONTAIN 4 ITEMS ",
    *      "PER LINE "/)
 1950 FORMAT (" IMPROPER DATA OF ""END"" IS MISSING IN SOURCE ,",
    *   "AND LOAD SECTION "/)
 1960    FORMAT (" "," INSUFFICIENT NUMBER OF 3-PORT DATA")
 1970    FORMAT (" NUMBER OF REQUIRED LINES = ",I2/
    *      " NUMBER OF FREQUENCIES = ",I2," NUMBER OF DEVICES = ",I2)
 1980    FORMAT (" ""END"" IS MISSING AFTER 3-PORT DATA")
 1990    FORMAT (136A1)
 2000    FORMAT (136A1)
 2010    FORMAT (" WRONG AMOUNT OF DATA ON 3-PORT LINE - 6 ITEMS ",
```

```
     *      "REQUIRED")
2020        FORMAT (" REACHED END-OF-FILE WHILE READING S-PARAMETERS")
2030        FORMAT (1X,136A1/" WRONG AMOUNT OF DATA ON S-PARAMETER ",
     *         "LINE - 8 ITEMS REQUIRED")
2040        FORMAT (" NOT ENOUGH LINES OF S-PARAMETERS IN FILE")
2050        FORMAT (" PRODUCT OF FREQUENCIES AND TWO-PORTS EXCEEDS ,",
     *         " 30, STOP EXECUTION ")
2060        FORMAT (" REACHED END-OF-FILE WHILE READING Y-PARAMETERS")
2070        FORMAT (1X,136A1/" WRONG AMOUNT OF DATA ON Y-PARAMETER ",
     *         "LINE, 8 ITEMS REQUIRED")
2080        FORMAT (" NOT ENOUGH LINES OF Y-PARAMETERS IN FILE")
2090     FORMAT (" TRANSISTOR INTERPOLATOR MAXIMUM IS 20 FREQUENCIES")
2100 FORMAT (" #END# IS MISSING AFTER S- OR Y-PARAMETERS")
2110 FORMAT (1X,136A1/" TOO MANY LINES OF S- OR Y-PARAMETERS"/
     *    " ABOVE LINE IS FIRST PAST THE REQUIREMENT")
2120 FORMAT (//9X,"DEVICE PARAMETERS WITH VARIOUS IMPEDANCES",
     *    " IN TWO-PORT ""RJX"" "/)
2130     FORMAT (/," INSUFFICIENT DATA FOR ""IMP"" ELEMENT ")
2140     FORMAT (" REQUIRED NUMBER OF LINES = ",I2/" NUMBER OF ",
     *       "FREQUENCIES = ",I2,"  NUMBER OF ""IMP"" ELEMENTS = ",I2)
2150     FORMAT (" ""END"" IS MISSING AFTER ONE-PORT DATA")
2160     FORMAT (" WRONG AMOUNT OF DATA ON ""IMP"" DATA LINE -",
     *       " 2 ITEMS REQUIRED")
2170 FORMAT (" TOO MANY LINES OF WEIGHTING FACTORS"/1X,I2,
     *    " LINES REQUIRED")
2180 FORMAT (" INSUFFICIENT DATA FOR WEIGHTING FACTORS AND"/
     *    " OPTIMIZATION CRITERIA, OR ""END"" IS MISSING"/1X,I2,
     *    " LINES REQUIRED")
2190 FORMAT (" ONLY ONE ITEM ALLOWED ON ERROR FUNCTION",
     *    " SPECIFICATION LINE")
2200 FORMAT (" THERE MUST BE BETWEEN FOUR AND SIX ITEMS",
     *    " ON EACH WIGHTING FACTOR LINE ")
2210 FORMAT (" THE FIRST FOUR WEIGHTING FACTORS MUST BE NUMERIC")
2220 FORMAT (" THE FIFTH FACTOR MUST BE ""LT"", ""GT"" OR THE GROUP",
     *    " DELAY WEIGHTING FACTOR")
2230 FORMAT (" THE SIXTH FACTOR MUST BE NUMERIC")
2240 FORMAT (" ONLY FIVE NUMERIC WEIGHTING FACTORS ARE ALLOWED")
2250 FORMAT (//18X,"CIRCUIT OPTIMIZATION WITH ",I2," VARIABLES"//2X,
     *    "INITIAL CIRCUIT ANALYSIS ")
2260 FORMAT (//18X,"SENSITIVITY ANALYSIS WITH ",I2," VARIABLES"//2X,
     *    "INITIAL CIRCUIT ANALYSIS ")
2270 FORMAT (/," PLEASE INDICATE IN YOUR DATA FILE THE ",
     *    " VARIABLE ELEMENTS")
2280 FORMAT (//,3X,"VOLTAGE GAIN FROM",F9.1," OHM SOURCE INTO",F9.1,
     *    " OHM LOAD"//5X,"FREQ.     GAIN(DB)",3X,"PHASE(DEGREES)  DELAY"
     *    )
2290 FORMAT (//" INPUT REFL. COEF. AND VSWR IN",F5.0," OHM SYSTEM WITH"
     *    ,F7.1," OHM LOAD"//3X,"F(MHZ)",6X,"RHO(MAGN.<ANGLE)",6X,"VSWR",
     *    4X,"RET L/G (DB)  Z(R+JX)","OHM"/)
2300 FORMAT (//," OUTPUT REFL. COEF. AND VSWR IN",F5.0,
     *    " OHM SYSTEM WITH ",F7.1," OHM SOURCE"//3X,"F(MHZ)",6X,
     *    "RHO(MAGN.<ANGLE)",6X,"VSWR",4X,"RET L/G (DB)  Z(R+JX), ","OHM"
     *    /)
2310 FORMAT (/,9X,"TWO-PORT Y-PARAMETER MATRIX IN MILLIMHOS",/,
     *    " FREQ. ",8X,"Y11",13X,"Y21",13X,"Y12",13X,"Y22"/)
```

```
2320 FORMAT (//18X,"POLAR S-PARAMETERS IN",F7.1," OHM SYSTEM"/3X,
    *   "FREQ.",5X,"S11",10X,"S21",10X,"S12",9X,"S22",7X,"S21",4X,"K"/
    *   4X,"    ",2X,"(MAGN<ANGL) ( MAGN<ANGL) ( MAGN<ANGL) ",
    *   "(MAGN<ANGL)",3X,"DB   FACT."/)
2330 FORMAT (//18X,"POLAR S-PARAMETERS IN",F7.1," OHM SYSTEM"/3X,
    *   "FREQ.",5X,"S11",10X,"S21",10X,"S12",9X,"S22",6X,"DELAY",3X,"K"
    *   /4X,"    ",2X,"(MAGN<ANGL) (  DB <ANGL) ( MAGN<ANGL) ",
    *   "(MAGN<ANGL)",2X,"      FACT."/)
2340 FORMAT (/,9X,"TWO-PORT Z-PARAMETER MATRIX IN OHMS     ",/,
    *   " FREQ. ",9X,"Z11",13X,"Z21",13X,"Z12",13X,"Z22",/)
2350 FORMAT (//14X,"POLAR S-PARAMETERS WITH COMPLEX LOAD AND"," SOURCE"
    *   //"    FREQ.   SOURCE IMP. (R,JX) OHMS    LOAD",
    *   " IMP. (R,JX) OHMS")
2360 FORMAT (1X,F10.3,2X,"(",F9.2,",",F10.2,")",5X,"(",F9.2,",",F10.2,
    *   ")")
2370 FORMAT (" ",/4X,"F",8X,"S11",10X,"S21",10X,"S12",9X,"S22",7X,"S21"
    *   ,5X,"K"/4X,"     (MAGN.<ANGL) (MAGN.<ANGL) (MAGN.<ANGL)",
    *   " (MAGN.<ANGL)   DB    FACTOR"/)
     END
```

```
      SUBROUTINE SEARCH
C
C        RE:     09/24/85
C
      COMMON KODE,IFLAG,XITIME,ITRAP,JTRAP,ISENS,IPRIN,MKODE,EX,
     +W,FR,NT,NF,INDFF,NCO,IXX,MCODE,FF,IMA,IMA1,NOISE,IBANKF,IDCLOC
      COMMON/TAKE/ ARRAY(28),IARRAY(28)
CBOB  COMMON/CO2/U(15),G(15),V,N,CONV,INEXP
      COMMON/CO2/U(15),G(15),V,N,CONV,INEXP,IGR(15)                    12-18-85
      COMMON/CSERCH/ALAMDA,GO(15),G1T(15),GA(15),H(15,15)
     +         ,KOUNT,LIMIT,MAT,NITER,VO,V1T,VO1,VO2,VO3
     +         ,XO(15),X1(15),X1T(15),XA(15),XST1(15),XST2(15)
     +         ,LDSERC,NLIST,ILIST
      COMMON/NSERCH/DMIN,IEXIT,VDIFF,TOT,GS,GSO
      COMMON/IOS/ IOT,INPET,INP,IFR,ITO,IED,IOPT,ISCR
C+
      COMMON/QUAMOD/IQUAD,IFTP,IAPPR,MAJOR
C+
C
C        KODE=1: FIRST CALL     KODE>1: SUBSEQUENT CALLS
C
      IF(KODE.NE.1) GOTO 70
C
C        INITIALIZATION FOR THE FIRST CALL ONLY
C
      ILIST=0
      KLIST=0
      LIMIT=0
      MAT=0
      NITER=0
      NLIST=0
      VO1=0.0
      VO2=0.0
      VO3=0.0
      DMIN=1.0
      DO 20 I=1,N
      DO 10 J=1,N
 10   H(J,I)=0.0
      H(I,I)=1.0
      IGR(I)=0
 20   X1T(I)=0.0
      IF(IPRIN.EQ.1) GOTO 60
C
C        PRINT HEADER
C
      IF(ISENS.EQ.1) GOTO 40
      WRITE(IOT,30)
 30   FORMAT(//46H OPTIMIZATION BEGINS WITH FOLLOWING VARIABLES ,
     +13HAND GRADIENTS//6X,9HVARIABLES,17X,9HGRADIENTS)
      GOTO 60
 40   WRITE(IOT,50)
 50   FORMAT(//47H SENSITIVITY ANALYSIS WITH FOLLOWING VARIABLES ,
     +13HAND GRADIENTS//6X,9HVARIABLES,17X,9HGRADIENTS)
 60   KODE=2
      LDSERC=1
```

```
      RETURN
C
C        SET SOME CONSTANTS
C
 70   ZERO1=0.01
      IBNDRY=0
      ALP=0.5
      ZERO=0.000001
      MUNG=0
      KPRINT=0
      IEXIT=0
      NP=N+1
      N1=N-1
      NN=N*NP/2
C
C        LDSERC=1: FIRST FUNCTION EVALUATION
C              =2: OPTIMIZATION IN PROGRESS
C
      IF(LDSERC.EQ.2) GOTO 280
      V0=V
      V1T=V                                                          11-08-85
      DO 80 I=1,N
      X0(I)=U(I)
      X1T(I)=U(I)                                                    11-07-85
      G1T(I)=G(I)
 80   G0(I)=G(I)
C
C        PRINT THE STARTING POINT
C
      DO 90 I=1,N
      ALT=EXP(X0(I))
      G0II=G0(I)/ALT
90    WRITE(IOT,105)I,ALT,I,G0(I)
C00   FORMAT(4H   (,I2,2H):,G12.5,10X,1H(,I2,2H):,G12.5)
105   FORMAT(4H   (,I2,2H):,G12.5,10X,1H(,I2,2H):,G12.5,2X)
110   FORMAT(2X,8HERR. F.=F14.7)
120   FORMAT(9X,12H----****----)
      WRITE(IOT,110)V0
C
C        IPRIN=1: PRINTOUTS AT EXPERT LEVEL
C             =0: PRINTOUTS AT NON-EXPERT LEVEL
C
      IF(IPRIN.NE.1) WRITE(IOT,120)
C
C     ISENS=1 INDICATES SENSITIVITY ANALYSIS, ONLY EXIT
C
      IF(ISENS.NE.1)GOTO 130
      ISENS=2
      RETURN
C
130   IF(IPRIN.NE.1) WRITE(IOT,140)
      IF(IPRIN.EQ.1) WRITE(IOT,150)
140   FORMAT(38H HOW MANY ITERATIONS BEFORE NEXT STOP
     +32H('0' RESULTS IN FINAL ANALYSIS.) ,/20H WANT INTERMEDIATE P,
     +41HRINTS (YES=1,NO=0). TYPE TWO NUMBERS; I,J    )
```

```
150   FORMAT(47H NO. OF ITERATIONS? PRINT INTERMEDIATE RESULTS?  ,
     + 10H(Y=1,N=0) )
      CALL TAKEIN(INPET,NOITEM)
      LIMIT=ARRAY(1)
      NLIST=ARRAY(2)
      ILIST=1-NLIST
      NITER=0
      IF(LIMIT.GT.0) GOTO 160
      IEXIT=1
      IFLAG=1
      GOTO 510
C
C        ITERATIVE LOOP STARTS HERE
C
160   NITER=NITER+1
C
C        SOLVE FOR GA= - H*GO, H STORES THE LDL FACTORS
C
C        FORWARD SUBSTITUTION, XA IS USED FOR TEMPORARY STORAGE
C
      XA(1)=-GO(1)
      IF(N.EQ.1) GOTO 190
      DO 180 I=2,N
      I1=I-1
      Z=-GO(I)
      DO 170 J=1,I1
170   Z=Z-H(I,J)*XA(J)
180   XA(I)=Z
C
C        BACKWARD SUBSTITUTION
C
190   GA(N)=XA(N)/H(N,N)
      IF(N.EQ.1) GOTO 220
      N1=N-1
      DO 210 I=1,N1
      I1=N-I
      I2=I1+1
      Z=0.0
      DO 200 J=I2,N
200   Z=Z+H(I1,J)*GA(J)
210   GA(I1)=XA(I1)/H(I1,I1)-Z
C
C        CHECK THE DIRECTIONAL GRADIENT
C        EXIT IF IT IS POSITIVE OR SUFFICIENTLY CLOSE TO ZERO
C
220   GS=0.0
      DO 230 I=1,N
230   GS=GS+GA(I)*GO(I)
      IF(GS.LT.-ZERO) GOTO 240
      IEXIT=2
      GOTO 460
C
240   GS0=GS
      ALAMDA=-2.0*V/GS
      IF(ALAMDA.GT.1.0) ALAMDA=1.0
```

```
      TOT=0.0
      VDIFF=V0
C+
      MAJOR=1
C+
250   IEXIT=3
C
C        COMPUTE THE INCREMENTS DX.  EXIT IF THEY ARE ALL VERY SMALL
C        OTHERWISE A NEW TRIAL POINT IS GIVEN BY X1T=X0+DX
C
      DO 260 I=1,N
      DX=ALAMDA*GA(I)
      IF(ABS(DX).GT.ZERO) IEXIT=0
      IF(IGR(I).EQ.0) GOTO 260                                          12-18-85
      TEMP=IGR(I)*DX                                                    12-18-85
      IF(TEMP.GT.0) DX=0.0                                              12-18-85
      IF(TEMP.LT.0) IGR(I)=0                                            12-18-85
260   X1T(I)=X0(I)+DX
      IF(IEXIT.EQ.3) GOTO 460
C
C        RETURN TO EVALUATE FUNCTION VALUE AND GRADIENTS AT X1T
C
      DO 270 I=1,N
C70   IF(G(I).NE.1.0E-6) U(I)=X1T(I)
270   U(I)=X1T(I)                                                       12-18-85
      MUNG=2
      LDSERC=2
      RETURN
C
280   V1T=V
C+
      MAJOR=0
C+
      IBNDRY=1
      DO 290 I=1,N
CBOB  IF(G(I).NE.1.0E-6) IBNDRY=0
      IF(IGR(I).EQ.0) IBNDRY=0                                           2-18-86
290   G1T(I)=G(I)
C
C        IBNDRY=1: ALLVARIABLES ARE FROZEN AT BOUNDARY, EXIT
C
      IF(IBNDRY.EQ.0) GOTO 300
      IEXIT=4
      GOTO 570
300   IF(V1T.GT.CONV) GOTO 310
C
C        THE FUNCTION VALUE FALLS BELOW CONVERGENCE THRESHOLD, EXIT
C
      IEXIT=5
      GOTO 460
C
C        COMPUTE THE DIRECTIONAL GRADIENT AT THE TRIAL POINT X1T
C
310   GYS=0.0
      DO 320 I=1,N
```

```
320   GYS=GYS+GA(I)*G1T(I)
C
C
C        TERMINATE THE LINE SEARCH IF THE DIRECTIONAL GRADIENT
C           HAS DECREASED BY MORE THAN TEN PERCENT
C        OTHERWISE ESTIMATE A NEW TRIAL POINT BY LINEAR EXTRAPOLATION
C           OR CUBIC INTERPOLATION AS APPROPRIATE
C
      IF(V1T.GE.V0) GOTO 350
CBOB  IF(ABS(GYS/GS0).LE.0.9) GOTO 360
      IF(ABS(GYS).LE.0.9*ABS(GS0))  GOTO 360                          12-18-85
      IF(GYS.GT.0.0) GOTO 350
C
C        LINEAR EXTRAPOLATION
C
      TOT=TOT+ALAMDA
      Z=10.0
      IF(GS.GE.GYS) GOTO 330
      Z1=10.0*ABS(GS-GYS)
      Z2=ABS(GYS)
      IF(Z2.GT.Z1) GOTO 330
      Z=10.0*Z2/Z1
330   ALAMDA=ALAMDA*Z
      V0=V1T
      GS=GYS
      DO 340 I=1,N
340   X0(I)=X1T(I)
      GOTO 250
C
C     CUBIC INTERPOLATION
C
350   Z=3.0*(V0-V1T)/ALAMDA+GYS+GS
      ZZ=SQRT(Z*Z-GS*GYS)
      GZ=GYS+ZZ
      Z=1.0-(GZ-Z)/(ZZ+GZ-GS)
      ALAMDA=ALAMDA*Z
      GOTO 250
C
C     LINE SEARCH COMPLETED.  THE MATRIX H SHOULD BE UPDATED.
C     NOTE:  ARRAYS XA, GA, AND X1 ARE USED AS WORKING SPACE.
C
360   ALAMDA=ALAMDA+TOT
      V0=V1T
      VDIFF=VDIFF-V0
      DGS=GYS-GS0
      LINK=1
C
C     CHOOSE FROM THE DFP AND THE COMPLEMENTARY DFP FORMULAS.
C     UPDATING H, BASED ON THE FOLLOWING TEST:
C
      IF(DGS+ALAMDA*GS0.GT.0.0) GOTO 380
C
      DO 370 I=1,N
370   GA(I)=G1T(I)-G0(I)
      SIG=1.0/(ALAMDA*DGS)
```

```
      GOTO 400
C
380   ZZ=ALAMDA/(DGS-ALAMDA*GSO)
      Z=DGS*ZZ-1.0
      DO 390 I=1,N
390   GA(I)=Z*GO(I)+G1T(I)
      SIG=1.0/(ZZ*DGS*DGS)
C
C     FORWARD SUBSTITUTION FOR XA=L*GA, WHERE H=LDL AND GA IS
C     COMPUTED ABOVE.
C
400   XA(1)=GA(1)
      DO 420 I=2,N
      I1=I-1
      Z=GA(I)
      DO 410 J=1,I1
410   Z=Z-H(I,J)*XA(J)
420   XA(I)=Z
C
C     UPDATE DIAGONAL ELEMENTS.  POSITIVE DEFINITENESS IN INSURED.
C
      DO 430 I=1,N
      Z=H(I,I)+SIG*XA(I)*XA(I)
      IF(Z.LE.0.0) Z=DMIN
      IF(Z.LT.DMIN) DMIN=Z
      H(I,I)=Z
      X1(I)=XA(I)*SIG/Z
430   SIG=SIG-X1(I)*X1(I)*Z
C
C     UPDATE THE L MATRIX
C
      N1=N-1
      DO 440 I=1,N1
      I1=I+1
      DO 440 J=I1,N
      GA(J)=GA(J)-H(I,J)*XA(I)
      H(I,J)=H(I,J)+X1(I)*GA(J)
440   H(J,I)=H(I,J)
C
      IF(LINK.EQ.2) GOTO 460
      LINK=2
      DO 450 I=1,N
450   GA(I)=GO(I)
      SIG=1.0/GSO
      IF(DGS+ALAMDA*GSO.GT.0.0) SIG=-ZZ
      GOTO 400
C
C     UPDATING OF H COMPLETED.
C
460   VO=V1T
      VO3=VO2
      VO2=VO1
      VO1=VO
      DO 470 I=1,N
      XO(I)=X1T(I)
```

```
470   GO(I)=G1T(I)
      IF(NITER.GE.LIMIT.OR.IEXIT.GT.0) ILIST=0
      IF(ILIST.NE.0) GOTO 490
C
C     ILIST=0: PRINT VARIABLES AND GRADIENTS
C
      DO 480 I=1,N
      ALT=EXP(X0(I))
      GOII=GO(I)/ALT
CBOB  IF(GO(I).EQ.1.0E-6) ALT=EXP(U(I))
      IF(IGR(I).NE.0) ALT=EXP(U(I))
480   WRITE(IOT,105)I,ALT,I,GO(I)
490   WRITE(IOT,110) VO
      IF(IPRIN.NE.1.AND.ILIST.EQ.0) WRITE(IOT,120)
C
      INEXP=1
C     IF INEXP=0, CHECK FOR STRICT DECREASE IN FUNCTION VALUE
C     AND ENABLE FRACTIONAL TERMINATION
C
      IF(INEXP.EQ.1) GOTO 500
      VOO=ABS(VO2-VO1)
      V1P=VO2*0.01
      IF(VOO.LT.V1P) IEXIT=6
      VOO=ABS(VO3-VO1)
      V4P=VO3*0.04
      IF(VOO.LT.V4P) IEXIT=6
C
500   IF(IEXIT.GT.0) GOTO 510
      IF(NITER-LIMIT) 160,130,130
C
C     EXIT.  SET LDSERC=-1, KODE=4, AND MUNG=1.  THE MAIN PROGRAM
C     WILL DETECT LDSERC<0.
C
510   IF(IEXIT.EQ.1) WRITE(IOT,520)
      IF(IEXIT.EQ.2) WRITE(IOT,530)
      IF(IEXIT.EQ.3) WRITE(IOT,540)
      IF(IEXIT.EQ.5) WRITE(IOT,550)
      IF(IEXIT.EQ.6) WRITE(IOT,560)
520   FORMAT(43H SEARCH INTERRUPTED, FINAL ANALYSIS FOLLOWS  /)
530   FORMAT(47H GRADIENT TERMINATION WITH ABOVE VALUES. FINAL  ,
     *16HANALYSIS FOLLOWS /)
540   FORMAT(44H LOCAL MIN REACHED.  GRADIENT SEARCH CANNOT   ,
     *7HIMPROVE)
550   FORMAT(47H FUNCTION TERMINATION WITH ABOVE VALUES. FINAL ,
     *16HANALYSIS FOLLOWS /)
560   FORMAT(49H FRACTIONAL TERMINATION WITH ABOVE VALUES. FINAL  ,
     *16HANALYSIS FOLLOWS /)
      GOTO 610
C
570   DO 580 I=1,N
      ALT=EXP(U(I))
580   WRITE(IOT,590) I, ALT
590   FORMAT(4H   (,I2,2H):,G12.5)
      WRITE(IOT,110) V
      IF(IPRIN.NE.1) WRITE(IOT,120)
```

```
      WRITE(IOT,600)
600   FORMAT(51H ALL VARIABLES AT BOUNDARY, FINAL ANALYSIS FOLLOWS /)
C
610   KODE=4
      MUNG=1
      LDSERC=-1
      RETURN
      END
```

```
      SUBROUTINE ANALYZ
C
      COMMON KODE,IFLAG,XITIME,ITRAP,JTRAP,ISENS,IPRIN,MKODE,EX,W,FR,NT,
     *   NF,INDFF,NCO,IXX,MCODE,FF,IMA,IMA1,NOISE,IBANKF,IDCLOC
C
      COMMON /COMB/ U11,U22,ZIN,ATTN,CAIND,IND,I,IJ,IJJ,IND3R,INDTR,J,
     *   KIMP,M,N,NBAND,NCB,NEWEX,NFOPT,PPRD
      COMMON /HVPLOT/ HORI(7),VERT(7),EPLOT(7,7),SENDE(15),BBB(30,2,2)
C
      COMMON /ANALQ/ ERRD,AOPT,INRJX,JSTK,JSOK,KLM,KYM,MODIM,MODEL,NOOPT
     *   ,IPRI,ITHR,IN1,IN2,IN3,NOIT,IDIV,ITER,IPRAM,ERSET,ERRAN,ERNO,
     *   SAVEC,ILTGT(3),S3P(3,6),VAX(15),IP1(15),IP2(15)
C
      COMMON /KSIO/ KSIN,KSOUT
      COMMON /DATAS/ PERT,PI,DPR,TWOPI,RPD
      COMMON /ANALZ/ IBAND(3),RIND(70),INC1(70),INC2(70),INC3(70),KX(15)
     *   ,KY(15)
      COMMON /TAKE/ ARRAY(28),IARRAY(28)
      COMMON /TAKE1/ IBUF(136),LDC
C
      COMMON /TAKE2/ PERUP(3,15)
C
CBOB  COMMON /CO2/ SEND(15),GR(15),ERR,NOPT,CONV,INEXP
      COMMON /CO2/ SEND(15),GR(15),ERR,NOPT,CONV,INEXP,IGR(15)          12-16-85
      COMMON /CO3/ CARAY(30,2,2),DATX(70,6)
C
C
      COMMON /CO4/ VIMP(30,2),FRST(45),DC0(100),DC1(70),DC2(70),DC3(70),
     *   FIG(3,10,5),FIN(3,10),OUT(2,4),WI(3,6),S21(3),FIGG(2,10,5)
C
      COMMON /CO5/ SLIMP(30,4)
C
      COMMON /CO6/ S3R(16,3,3)
C
      COMMON /CO7/ YMAT(15,2,2)
      COMMON /CO8/ PS(30,2,4)
      COMMON /CO9/ AAA(30,2,2),A(2,2),AB1(2,2),AB2(2,2),AB3(2,2),AB4(2,2
     *   )
      COMMON /PPLOT/ NPLOT,PARAY(30,2,5),NPOINT,IPNOPR,MIXPR,OLDT
C
      COMMON /DATANA/ ERRD1,ERRSD,KKX,KKY,TURN,YOS,YSS,ZOS,ZSS,AIM,ANGL,
     *   ANGLE,CHARI,DBL,ERRAI,GAINDB,P11,P12,P21,P22,QX,QY,U12,U21,UD11
     *   ,UN1,UN2,UP1,UP2,UDEN,UN3,UN4,VSWR,AB,AC,AD,AE,AF,CALEN,CSQ,
     *   DENO,DS,DS2,DX,EJ,QL,SSQ,WXX,XJ,YIN,YINI,YINR,YJ,ZCST,ZD,ZINI,
     *   ZINR,ZOE,ZOO,ZP,ALE,AMJ,GNO(2,10),POW,SQU,STRC(30),FACT(2,10),
     *   EXHOLD,NSTNS
      COMMON /CNAERR/ IFTLER,AMET,ALOSS
      COMMON /IOS/ IOT,INPET,INP,IFR,ITO,IED,IOPT,ISCR
C
C
      COMMON /BOMB/ JERRX
C
      COMMON /CNOISE/ DDX(64),REFR,GN,RN,IN22,INOIS,PAV(10),IA
      COMMON /CGRID/ WRST(15),IFUNCT,IFUX,ICOUNT,IIII,CAC,XPER
      COMMON /MCARLO/ SUMS11(30),SUMS22(30),SUMSCL(30),SQDS11(30),SQDS22
```

```
     *   (30),SQDSCL(30)
         COMPLEX SUMS11,SUMS22
C
         COMMON /IVG/ IVG
         COMMON /NFILE/ IPINT
C
         COMMON /PRTOPT/ NSTAR
         COMMON /CODLAY/ DELAY,FPERT
C
C
         COMMON /IMPONE/ NIMP
         COMMON /QUAMOD/IQUAD,IFTP,IAPPR,MAJOR
C
         DOUBLE PRECISION PERT,ERRSD,GGD,ERRD1,ERRD
C
         DIMENSION VARI(7),VARX(17),VARY(17)
         INTEGER POW
         COMPLEX CARAY,ANG
         COMPLEX BBB,GNO,STRC,YIN,ZIN,CUNIT
         COMPLEX AAA,A,AB1,AB2,AB3,AB4,S3R,YMAT,TANG
         COMPLEX U11,U12,U21,U22,UN1,UN2,UD11,UP1,UP2,P11,P12,P21,P22
C
         COMPLEX VG1,VG2,VG3
C
         DATA VARI / 10.0,2.5,1.25,1.0,0.8,0.4,0.1 /
         DATA VARX / 0.0001,0.2,0.5,1.0,2.0,5.0,100.0,10*0.0 /
         DATA VARY / 7*0.0,0.2,0.5,1.0,2.0,5.0,-0.2,-0.5,-1.0,-2.0,-5.0 /
C
         IF (FPERT.EQ.0.0) FPERT = 0.001
         IFPERT = 0
         IF (NSTAR.EQ.1.OR.NSTAR.EQ.9) IFPERT = -1
         ERINI = 1.E12
         ICST = 0
         IGRAD = 0
         ICOUNT = 0
         CUNIT = CMPLX(1.,0.0)
         IF (NT.EQ.0.AND.ITHR.NE.0) NT = 1
         ERINJ = 0.0
         IF (KODE.EQ.10.AND.MKODE.NE.3) MKODE = 2
         IF (KODE.EQ.10) KODE = 9
         IF (MKODE.EQ.3) ICOUNT = 0
         IF (KODE.EQ.8) GO TO 110
         IF (KODE.GT.4) GO TO 90
         ERRD = 0.
         DELAYP = 0.0
         ERRAN = 0.0
         ERRDL = 0.0
         IF (KODE.NE.1) GO TO 20
         DO 10 I = 1, NOPT
            SEND(I) = ALOG(DATX(KX(I),KY(I)))
   10 CONTINUE
         GO TO 50
   20 DO 40 I = 1, NOPT
C           IF (GR(I).NE.0.000001) GR(I) = 0.0
            GR(I)=0.0
```

```
          IF (PERUP(2,I).EQ.0.) GO TO 30
C         IF (SEND(I).LT.(ALOG(PERUP(3,I))).OR.SEND(I).GT.(ALOG(PERUP(2,I
C     *      )))) GR(I) = 0.000001
C         X = ALOG(PERUP(2,I))
C         IF (SEND(I).GT.X) SEND(I) = ALOG(PERUP(2,I))
C         X = ALOG(PERUP(3,I))
C         IF (SEND(I).LT.X) SEND(I) = ALOG(PERUP(3,I))
            X=ALOG(PERUP(2,I))
            IF(SEND(I).LT.X) GOTO 21
            SEND(I)=X
            IGR(I)=1
   21      X=ALOG(PERUP(3,I))
            IF(SEND(I).GT.X) GOTO 30
            SEND(I)=X
            IGR(I)=-1
   30     IF (SEND(I).LT.-100) SEND(I) = -100
          SENDE(I) = EXP(SEND(I))
          DATX(KX(I),KY(I)) = SENDE(I)
   40 CONTINUE
   50 CONTINUE
      IPRAM = 0
      GO TO 85
   60 IPRAM = IPRAM+1
      I = IPRAM
      MCODE = 1
      I88 = KX(IPRAM)
      I99 = KY(IPRAM)
      SAVEC = DATX(I88,I99)
      IF (PERUP(2,I).EQ.0) GO TO 70
      IF (GR(I).EQ.0.000001) GO TO 60
   70 CONTINUE
      DATX(I88,I99) = DATX(I88,I99)+PERT*DATX(I88,I99)
   85 ERSET = 0.0
      ERRAN = 0.0
      ERRDL = 0.0
      DELAYP = 0.0
      ERNO = 0.0
      KIMP = 0
      ERRD = 0.
      IF (NOISE.EQ.0) GO TO 90
      DO 80 INOI = 1, 2
         DO 80 II = 1, NF
            DO 80 KK = 1, 5
               FIG(INOI,II,KK) = FIGG(INOI,II,KK)
   80 CONTINUE
   90 CONTINUE
      IF (KODE.EQ.6) GO TO 240
      IF (KODE.NE.5) GO TO 260
      IF (IMA.EQ.0) GO TO 210
C
C  **  COARSE SEARCH**
C
  100 MKODE = 0
      IIII = 0
  110 CALL GRID
```

```
      IF (IFTLER.NE.0) GO TO 2460
      IIII = 0
      IF (IFUX.EQ.IFUNCT) GO TO 160
  120 IIII = IIII+1
      IF (IIII.GT.IFUX) GO TO 2190
      IF (KODE.EQ.10) GO TO 130
      IRAX = INT(RANF(0.)*IFUNCT)+1
      JXX = IFUNCT
  130 CONTINUE
      DO 150 J = 1, NOPT
         IF (KODE.EQ.10) GO TO 140
         ICC = INT(IRAX*IP1(J)/JXX+.001)
         DATX(KX(J),KY(J)) = VAX(J)+GR(J)*ICC
         IRAX = IRAX-INT(ICC*JXX/IP1(J)+.001)
         JXX = INT(JXX/IP1(J)+0.001)
         GO TO 150
  140    IF (IP1(J).EQ.0) GO TO 150
         DATX(KX(J),KY(J)) = VAX(J)+RANF(0.0)*GR(J)
  150 CONTINUE
      GO TO 260
  160 DO 170 J = 1, NOPT
         DATX(KX(J),KY(J)) = VAX(J)
  170 CONTINUE
      GO TO 260
  180 DO 200 IIII = 1, NOPT
         IF (IP2(IIII).GE.IP1(IIII)) GO TO 200
         IP2(IIII) = IP2(IIII)+1
         R = IP2(IIII)-1
         DATX(KX(IIII),KY(IIII)) = VAX(IIII)+GR(IIII)*R
         IF (IIII.EQ.1) GO TO 260
         IIIIM1 = IIII-1
C
         DO 190 J = 1, IIIIM1
            IP2(J) = 1
            IF (IP1(J).EQ.1) GO TO 190
            DATX(KX(J),KY(J)) = VAX(J)
  190    CONTINUE
C
         GO TO 260
  200 CONTINUE
      GO TO 2190
  210 COM1 = DATX(KX(1),KY(1))
      COM2 = DATX(KX(2),KY(2))
      IJII = 0
      IKII = 0
C
C******************  L  O  O  P  ********************************
C
C  **BEGIN LOOP FOR SWEEP OPTION.  COMPUTE ERRORS**    LOOP-A
C
  220 IJII = IJII+1
      DATX(KX(1),KY(1)) = VARI(IJII)*COM1
      VERT(IJII) = DATX(KX(1),KY(1))
C
  230 IKII = IKII+1
```

```
      DATX(KX(2),KY(2)) = VARI(IKII)*COM2
      IF (IJII.GT.1) GO TO 260
      HORI(IKII) = DATX(KX(2),KY(2))
      GO TO 260
  240 CONTINUE
C
C  **BEGIN LOOP FOR MAPPING WITH VARIOUS IMPEDANCES**  LOOP-B
C
  250 KLM = KLM+1
  260 CONTINUE
      ICOUNT = ICOUNT+1
  270 CONTINUE
      IF (NPLOT.EQ.0) NPOINT = 0
C
C  **BEGIN FREQUENCY DO-LOOP**                         LOOP-C
C
      INDFF = 1
      IF (KODE.NE.8) GO TO 280
      IF (ICOUNT.EQ.1) WRITE (IOT,2490)
      WRITE (IOT,2500) (DATX(KX(J),KY(J)),J=1,NOPT)
  280 FR = FRST(INDFF)
      IF (IFPERT.EQ.-1) FR = (1.0-FPERT)*FR
      W = TWOPI*FR
C
C  **BEGIN INTERCONNECTION DO-LOOP**                   LOOP-D
C
      IXX = 1
  290 IN1 = INC1(IXX)
      IN2 = INC2(IXX)
      IN3 = INC3(IXX)
      IF (IN1.LT.1.OR.IN1.GT.67) GO TO 2470
      GO TO (330,360,420,890,900,910,950,470,480,1330,550,560,630,640,
     *   760,760,1210,830,870,870,870,840,840,880,1360,1370,1380,1430,
     *   870,1450,1480,1490,870,870,870,870,870,1500,300,1530,1540,1550,
     *   1720,1600,1610,1660,1550,1660,1680,2060,1970,1720,1720,1700,
     *   1700,1600,1600,870,870,1750,1750,1750,1750,870,860,1510,1520),
     *   IN1
C
C*****CALL USER"S OWN OVERLAY - ELEMENTS "ELA", "ELB" AND "ELC"
C
  300 X = 1.
      IF (IN1.EQ.43) X = -X
      IF (DATX(IXX,3).EQ.0.) DATX(IXX,3) = 25.
      R = DATX(IXX,1)+X*DATX(IXX,2)*FR/DATX(IXX,3)
      IF (2-IN3) 2350 , 310 , 320
  310 CALL SIMP (A,R,0.)
      GO TO 1980
  320 R = 1./R
      CALL PIMP (A,R,0.)
      GO TO 1980
C
C*****COMPUTE ABCD MATRIX FOR RESISTOR
C
  330 IF (2-IN3) 2350 , 340 , 350
  340 CALL SIMP (A,DATX(IXX,1),0.)
```

```
      GO TO 1980
  350 G = 1./DATX(IXX,1)
      CALL PIMP (A,G,0.)
      GO TO 1980
C
C   **COMPUTE ABCD MATRIX FOR INDUCTOR**
C
  360 IF (2-IN3) 2350 , 370 , 390
  370 WL = W*DATX(IXX,1)*1.E-3
      CALL SIMP (A,0.,WL)
      IF (DATX(IXX,2).EQ.0.0.OR.INDFF.GT.1) GO TO 380
      RIND(IXX) = TWOPI*DATX(IXX,1)*1.E-3*DATX(IXX,3)/DATX(IXX,2)
  380 CALL SIMP (A,RIND(IXX),WL)
      GO TO 1980
  390 IF (DATX(IXX,2).EQ.0.0) GO TO 410
      IF (INDFF.GT.1) GO TO 400
      RIND(IXX) = TWOPI*DATX(IXX,1)*1.E-3*DATX(IXX,3)/DATX(IXX,2)
  400 B = W*DATX(IXX,1)*1.E-3
      D = 1./(RIND(IXX)*RIND(IXX)+B*B)
      Z = RIND(IXX)*D
      WL = -B*D
      CALL PIMP (A,Z,WL)
      GO TO 1980
  410 WL = -1.E3/(W*DATX(IXX,1))
      CALL PIMP (A,0.,WL)
      GO TO 1980
C
C   **COMPUTE ABCD MATRIX FOR CAPACITOR**
C
  420 C = 1.E-6*DATX(IXX,1)
      R = 0.0
      IF (2-IN3) 2350 , 430 , 450
  430 WC = -1.E6/(W*DATX(IXX,1))
      IF (DATX(IXX,2).EQ.0.) GO TO 440
      R = 1.E6/(TWOPI*DATX(IXX,1)*DATX(IXX,3))*DATX(IXX,2)
      GO TO 570
  440 CALL SIMP (A,R,WC)
      GO TO 1980
  450 IF (DATX(IXX,2).EQ.0.0) GO TO 460
      R = 1.E6/(TWOPI*DATX(IXX,1)*DATX(IXX,3))*DATX(IXX,2)
      GO TO 580
      B = W*DATX(IXX,1)*1.E-6
      B2 = B*B
      D = 1./(1.+B2*R*R)
      Z = B2*R*D
      WC = B*D
      CALL PIMP (A,Z,WC)
      GO TO 1980
  460 WC = W*DATX(IXX,1)*1.E-6
      CALL PIMP (A,0.,WC)
      GO TO 1980
C
C   **COMPUTE ABCD MATRIX FOR SERIES RL**
C
  470 R = DATX(IXX,1)
```

```
      X = DATX(IXX,2)*1.E-3
      IF (2-IN3) 2350 , 510 , 520
C
C   **COMPUTE ABCD MATRIX FOR PARALLEL RL**
C
  480 R = DATX(IXX,1)
      X = DATX(IXX,2)*1.E-3
      B = W*X
      IF (2-IN3) 2350 , 490 , 500
  490 B2 = B*B
      D = 1./(R*R+B2)
      Z = R*B2*D
      WL = R*R*B*D
      GO TO 540
  500 D = 1.0/(R*B)
      Z = B*D
      WL = -R*D
      GO TO 530
  510 WL = W*X
      Z = R
      GO TO 540
  520 B = W*X
      D = 1./(R*R+B*B)
      Z = R*D
      WL = -B*D
  530 CALL PIMP (A,Z,WL)
      GO TO 1980
  540 CALL SIMP (A,Z,WL)
      GO TO 1980
C
C   **COMPUTE ABCD MATRIX FOR SERIES RC**
C
  550 C = 1.E-6*DATX(IXX,2)
      R = DATX(IXX,1)
      IF (2-IN3) 2350 , 590 , 600
C
C   **COMPUTE ABCD MATRIX FOR PARALLEL RC**
C
  560 C = 1.E-6*DATX(IXX,2)
      R = DATX(IXX,1)
      IF (2-IN3) 2350 , 570 , 580
  570 B = W*C
      D = 1./(1.+R*R*B*B)
      Z = R*D
      WC = -R*R*B*D
      GO TO 620
  580 Z = 1./R
      WC = W*C
      GO TO 610
  590 Z = R
      WC = -1./(W*C)
      GO TO 620
  600 B = W*C
      B2 = B*B
      D = 1./(1.+B2*R*R)
```

```
      Z = B2*R*D
      WC = B*D
  610 CALL PIMP (A,Z,WC)
      GO TO 1980
  620 CALL SIMP (A,Z,WC)
      GO TO 1980
C
C   **COMPUTE ABCD MATRIX FOR SERIES LC**
C
  630 X = DATX(IXX,1)*1.E-3
      Y = DATX(IXX,2)
      Z = 0.0
      IF (2-IN3) 2350 , 700 , 720
C
C  ** COMPUTE ABCD MATRIX FOR PARALLEL LC **
C
  640 IF (DATX(IXX,3).EQ.0.0) GO TO 670
      IF (2-IN3) 2350 , 650 , 660
  650 X = DATX(IXX,1)*1.E-3
      Y = DATX(IXX,2)*1.E-6
      RIND(IXX) = TWOPI*X*DATX(IXX,4)/DATX(IXX,3)
      B = 1.-W*W*X*Y
      B1 = W*W*RIND(IXX)*RIND(IXX)*Y
      D = 1./(B*B+B1*Y)
      Z = (RIND(IXX)*B+B1/RIND(IXX)*X)*D
      WX = (W*X*B-B1/W)*D
      GO TO 750
  660 X = DATX(IXX,1)*1.E-3
      Y = DATX(IXX,2)*1.E-6
      RIND(IXX) = TWOPI*X*DATX(IXX,4)/DATX(IXX,3)
      D = 1./(RIND(IXX)*RIND(IXX)+X*X*W*W)
      Z = RIND(IXX)*D
      WX = W*Y-W*X*D
      GO TO 740
  670 X = DATX(IXX,1)*1.E-3
      Y = DATX(IXX,2)
      Z = 0.0
      IF (2-IN3) 2350 , 680 , 690
  680 WX = W*X/(1.-W*W*X*Y*1.E-6)
      GO TO 750
  690 WX = (W*W*X*Y*1.E-6-1.)/(W*X)
      GO TO 740
  700 WX = W*X-1.E6/(W*Y)
      IF (DATX(IXX,3).EQ.0.0) GO TO 750
      IF (INDFF.GT.1) GO TO 710
      RIND(IXX) = TWOPI*X*DATX(IXX,4)/DATX(IXX,3)
  710 Z = RIND(IXX)
      GO TO 750
  720 WX = (W*Y)/(1.E6-W*W*X*Y)
      IF (DATX(IXX,3).EQ.0.0) GO TO 740
      IF (INDFF.GT.1) GO TO 730
      RIND(IXX) = TWOPI*X*DATX(IXX,4)/DATX(IXX,3)
  730 Y = DATX(IXX,2)*1.E-6
      B = 1./W/Y-W*X
      D = 1./(RIND(IXX)*RIND(IXX)+B*B)
```

```
      Z = RIND(IXX)*D
      WX = B*D
  740 CALL PIMP (A,Z,WX)
      GO TO 1980
  750 CALL SIMP (A,Z,WX)
      GO TO 1980
C
C   **COMPUTE ABCD MATRIX FOR SERIES RLC (RX)**
C
  760 C = 1.E-6*DATX(IXX,3)
      R = DATX(IXX,1)
      X = DATX(IXX,2)*1.E-3
      IZJ = IN1-14
      GO TO (770,780), IZJ
  770 IF (2-IN3) 2350 , 810 , 820
C
C   **COMPUTE ABCD MATRIX FOR PARALLEL RLC (RX)**
C
  780 IF (2-IN3) 2350 , 790 , 800
  790 W2 = W*W
      DA = R*(W2*X*C-1.)
      D = 1./(W2*X*X+DA*DA)
      Z = R*W2*X*X*D
      WC = -(R*W*X*DA*D)
      GO TO 620
  800 Z = 1./R
      WC = W*C-1./(W*X)
      GO TO 610
  810 Z = R
      WC = W*X-1./(W*C)
      GO TO 620
  820 W2 = W*W
      D = 1./(W2*(C*(R*R*C+W2*X*X*C-2.*X))+1.)
      Z = R*W2*C*C*D
      WC = W*C*(1.-W2*X*C)*D
      GO TO 610
C
C   **COMPUTE ABCD MATRIX FOR IDEAL TRANSFORMER**
C
  830 IF (IN3.EQ.35) TURN = DATX(IXX,1)
      IF (IN3.EQ.38) TURN = SQRT(DATX(IXX,1)/DATX(IXX,2))
      IF (IN2.GT.25) TURN = -TURN
      A(1,1) = TURN
      A(2,2) = 1.0/TURN
      A(1,2) = 0.0
      A(2,1) = 0.0
      GO TO 1980
C
C   **COMPUTE ABCD MATRIX FOR CONTROLLED CURRENT SOURCE**
C
  840 IF (IN3.NE.36.AND.IN3.NE.37) GO TO 2360
      X = DATX(IXX,1)
      IF (IN3.EQ.37) GO TO 850
      Q = -1./DATX(IXX,3)
      Q1 = Q/DATX(IXX,2)
```

```
      A(1,1) = X*Q1
      A(2,1) = Q1
      A(1,2) = Q*X
      A(2,2) = Q
      GO TO 1980
C
C   **COMPUTE ABCD MATRIX FOR VOLTAGE CONTROLLED CURRENT SOURCE**
C
  850 Q = -1000./DATX(IXX,3)
      Q1 = Q/DATX(IXX,2)
      A(1,1) = Q1
      A(2,1) = Q1/X
      A(1,2) = Q
      A(2,2) = Q/X
      GO TO 1980
C
C    COMPUTE A GENERALIZED CURRENT/VOLTAGE SOURCE
C
  860 IF (DATX(IXX,4).EQ.0.) ANGE = 1E18
      IF (DATX(IXX,4).NE.0.) ANGE = DATX(IXX,4)
      IF (DATX(IXX,1).EQ.0..OR.DATX(IXX,2).EQ.0.) GO TO 2450
      ANG = DATX(IXX,3)/CMPLX(1.,FR/ANGE)*CEXP(CMPLX(0.,-W*DATX(IXX,5)))
      IF (IN3.EQ.5) AAA(IN2,2,1) = ANG/DATX(IXX,1)
      IF (IN3.EQ.24) AAA(IN2,2,1) = ANG/DATX(IXX,2)
      IF (IN3.EQ.52) AAA(IN2,2,1) = ANG/DATX(IXX,2)/DATX(IXX,1)
      IF (IN3.EQ.53) AAA(IN2,2,1) = ANG*.001
      IF (IN3.NE.5.AND.IN3.NE.24.AND.IN3.NE.34.AND.IN3.LT.52) GO TO 2450
      AAA(IN2,1,1) = CMPLX(1./DATX(IXX,1),0.)
      AAA(IN2,1,2) = CMPLX(1./DATX(IXX,2),0.)
      AAA(IN2,2,2) = CMPLX(988.,0.)
      IF (IN3.EQ.34) AAA(IN2,2,2) = CMPLX(988.5,0.)
      IF (IN3.EQ.34) AAA(IN2,2,1) = CMPLX(10.**(DATX(IXX,3)/20.),0.)*
     *   CEXP(CMPLX(0.,DATX(IXX,4)))
      IF (IN3.EQ.34.AND.DATX(IXX,4).GT.360) AAA(IN2,2,2) = CMPLX(988.7,
     *   0.)
      GO TO 2000
C
C   ** COMPUTE ABCD MATRIX FOR TRANSMISSION LINE(LOSSY/LOSSLESS)***
C
  870 ALOSS = FRST(1)
      CALL STRIP
C
C      IF (JERRX .NE. 0)  GOTO 99999
C      GOTO 989
C
      IF (JERRX.EQ.0) GO TO 2000
      IF (IFTLER.NE.23) GO TO 2480
      RETURN
C
C   **COMPUTE ABCD MATRIX FOR IDEAL OP. AMP.**
C
  880 A(1,1) = 1.0/DATX(IXX,1)
      A(2,2) = A(1,1)
      A(1,2) = 0.0
      A(2,1) = A(1,2)
```

```
      GO TO 1980
C
C   **CASCADE TWO ABCD MATRICES**
C
  890 A(1,1) = AAA(IN2,1,1)*AAA(IN3,1,1)+AAA(IN2,1,2)*AAA(IN3,2,1)
      A(2,1) = AAA(IN2,2,1)*AAA(IN3,1,1)+AAA(IN2,2,2)*AAA(IN3,2,1)
      A(1,2) = AAA(IN2,1,1)*AAA(IN3,1,2)+AAA(IN2,1,2)*AAA(IN3,2,2)
      A(2,2) = AAA(IN2,2,1)*AAA(IN3,1,2)+AAA(IN2,2,2)*AAA(IN3,2,2)
      GO TO 1980
C
C                **PARALLEL TWO ABCD MATRICES THRU Y-PARAMETERS*
C
  900 IF (CABS(AAA(IN2,1,2)).EQ.0.0.OR.CABS(AAA(IN3,1,2)).EQ.0.0) GO TO
     *   2370
      IE1 = 1
      IE2 = 3
      GO TO 920
C
C                **SERIES TWO ABCD MATRICES THRU Z-PARAMETERS**
C
  910 IF (CABS(AAA(IN2,2,1)).EQ.0.0.OR.CABS(AAA(IN3,2,1)).EQ.0.0) GO TO
     *   2370
      IE1 = 2
      IE2 = 2
C
  920 DO 930 I = 1, 2
         DO 930 J = 1, 2
            AB1(I,J) = AAA(IN2,I,J)
            AB2(I,J) = AAA(IN3,I,J)
  930 CONTINUE
C
      CALL ABCDM (AB1,AB3,DATX(IPRI,1),IE1)
      CALL ABCDM (AB2,AB4,DATX(IPRI,1),IE1)
C
      DO 940 I = 1, 2
         DO 940 J = 1, 2
            AB1(I,J) = AB3(I,J)+AB4(I,J)
  940 CONTINUE
C
      CALL ABCDM (AB1,A,DATX(IPRI,1),IE2)
      GO TO 1980
C
C   **PRINT RESULTS**
C
  950 IF (IN3.NE.31.AND.IN3.NE.28) GO TO 990
C
C  **PRINT OPTION "YM", PRINT Y-MATRIX**
C
      DO 960 I = 1, 2
         DO 960 J = 1, 2
            A(I,J) = AAA(IN2,I,J)
  960 CONTINUE
      IF (CABS(A(1,2)).EQ.0.0.AND.IN3.EQ.31) GO TO 2380
      IF (CABS(A(2,1)).EQ.0.0.AND.IN3.EQ.28) GO TO 2380
      IF (IN3.EQ.28) CALL ABCDM (A,AB1,DATX(IPRI,1),2)
```

```
      IF (IN3.EQ.28) GO TO 980
      CALL ABCDM (A,AB1,DATX(IPRI,1),1)
      DO 970 I = 1, 2
         DO 970 J = 1, 2
            AB1(I,J) = 1000.*AB1(I,J)
  970 CONTINUE
      WRITE (IOT,2510) FR,((AB1(I,J),I=1,2),J=1,2)
      GO TO 2000
  980 WRITE (IOT,2520) FR,((AB1(I,J),I=1,2),J=1,2)
      GO TO 2000
C
C   **TEST FOR NON-ZERO CHARACTERISTIC IMPEDANCE**
C
  990 IF (IN3.NE.34.AND.DATX(IXX,1).EQ.0.) GO TO 2390
      IF (NOOPT.GT.0) CALL SNOIS1
C
C  **PRINT OPTION "IR", "OR", "S1 TO S6", OR "VG"**
C
      IF (NSTAR.EQ.9) GO TO 1130
C
      IF (IN3.GT.42.AND.IN3.LT.47) GO TO 1130
      NSTNS = 35-IN3
      GO TO (1000,1050,1050,2070,1130,1130), NSTNS
C
C   **COMPUTE VOLTAGE GAIN**
C
 1000 CONTINUE
      IF (IVG.NE.0) GO TO 1020
 1010 WRITE (IOT,2530)
      CALL TAKEIN (INPET,NOITEM)
      IVG = ARRAY(1)
 1020 CONTINUE
      IF (IVG.NE.1.AND.IVG.NE.2.AND.IVG.NE.3) GO TO 1010
      VG1 = 0.0
      VG2 = 0.0
      VG3 = 1.0
      RSRCE = DATX(IXX,1)
      RLOAD = DATX(IXX,2)
      IF (RLOAD.EQ.0.0) GO TO 1030
      VG1 = 1.0/(AAA(IN2,1,1)+AAA(IN2,1,2)/RLOAD)
      ZIN = AAA(IN2,1,1)*RLOAD+AAA(IN2,1,2)
      ZIN = ZIN/(AAA(IN2,2,1)*RLOAD+AAA(IN2,2,2))
      VG2 = VG1*(ZIN/(ZIN+RSRCE))
      VG3 = VG2*(1.0+RSRCE/RLOAD)
 1030 CONTINUE
      IF (IVG.EQ.1) U11 = VG2
      IF (IVG.EQ.2) U11 = VG3
      IF (IVG.EQ.3) U11 = VG1
      CALL RPOLAR (U11,ANGLE,OUT(2,1))
      IF (IFPERT.NE.-1) GO TO 1040
      ANGLE1 = ANGLE
      IF (ANGLE1.LT.0.0) ANGLE1 = 360.+ANGLE1
      GO TO 2060
 1040 ANGLE0 = ANGLE
      IF (ANGLE0.LT.0.0) ANGLE0 = 360.+ANGLE0
```

```
      DELAY = (ANGLE1-ANGLE0)/(360.*FPERT*FR)
      DDELAY = 1000.*DELAY
      IF (CAIND.EQ.1) OUT(2,1) = ATTN*OUT(2,1)
      IF (KODE.EQ.1.AND.MKODE.NE.1) GO TO 2010
      IF (KODE.EQ.2.OR.KODE.EQ.5) GO TO 2010
      GAINDB = 20.*ALOG10(OUT(2,1))
      IF (KODE.EQ.10) CAC = GAINDB-PARAY(INDFF,1,1)
      IF (KODE.EQ.10) CALL GRID
      IF (KODE.EQ.10) GO TO 2070
C
C *** VG-----PRINT OPTION
C
      CARAY(INDFF,2,1) = CMPLX(GAINDB,0.)
      PARAY(INDFF,1,1) = GAINDB
      IF (IPNOPR.EQ.0.OR.IXX.NE.NPLOT) WRITE (IOT,2540) FR,GAINDB,ANGLE,
     *   DDELAY
      IF (IXX.NE.NPLOT) GO TO 2000
      IF (NPOINT.EQ.0) OLDT = -10000.
      IF (OLDT.GT.FR) NPOINT = 0
      IF (NPOINT.GE.30) GO TO 2000
      NPOINT = NPOINT+1
      OLDT = FR
      PARAY(NPOINT,1,1) = GAINDB
      PARAY(NPOINT,2,1) = DDELAY
      GO TO 2000
C
C  **COMPUTE Z(IN) OR Z(OUT)**
C
 1050 I = NSTNS-1
      J = 4-NSTNS
      U22 = AAA(IN2,I,I)*DATX(IXX,2)+AAA(IN2,1,2)
      U22 = U22/(AAA(IN2,2,1)*DATX(IXX,2)+AAA(IN2,J,J))
      U11 = (U22-DATX(IXX,1))/(U22+DATX(IXX,1))
      C = CABS(U11)
      IF (C.LT.0.001) C = 0.001
      CAC = CABS(U11-CARAY(INDFF,1,1))
      IF (KODE.EQ.10) CALL GRID
      IF (KODE.EQ.10) GO TO 2060
      CARAY(INDFF,1,1) = U11
      OUT(1,1) = C
      DBL = 20.0*ALOG10(C)
      IF (KODE.EQ.1.AND.MKODE.NE.1) GO TO 2010
      OUT(2,1) = DBL
      OUT(1,3) = 101.0
      OUT(2,3) = 1.0
C
C  **TEST FOR INFINITE VSWR**
C
      IF (C.GE.1.0) GO TO 1060
      VSWR = (1.0+C)/(1.0-C)
      OUT(2,3) = VSWR
 1060 IF (KODE.EQ.2.AND.AOPT.EQ.0.0) GO TO 2010
      IF (KODE.EQ.5.AND.AOPT.EQ.0.0) GO TO 2010
      CALL RPOLAR (U11,ANGLE,AM)
      IF (ANGLE) 1070 , 1080 , 1080
```

```
 1070 ANGL = 360.+ANGLE
      GO TO 1090
 1080 ANGL = ANGLE
 1090 IF (INDFF.EQ.1) GO TO 1100
      ERRAN = ERRAN+ABS(ANGL-ERRAI)
 1100 ERRAI = ANGL
      IF (KODE.EQ.2.OR.KODE.EQ.5) GO TO 2010
      IF (C.GE.0.9995) GO TO 1110
      IF (IPNOPR.EQ.0.OR.IXX.NE.NPLOT) WRITE (IOT,2550) FR,C,ANGLE,VSWR,
     *   DBL,U22
      GO TO 1120
 1110 IF (IPNOPR.EQ.0.OR.IXX.NE.NPLOT) WRITE (IOT,2560) FR,C,ANGLE,DBL,
     *   U22
 1120 IF (IXX.NE.NPLOT) GO TO 2060
      IF (NPOINT.EQ.0) OLDT = -10000.
      IF (OLDT.GT.FR) NPOINT = 0
      IF (NPOINT.GE.30) GO TO 2060
      NPOINT = NPOINT+1
      OLDT = FR
      CARAY(NPOINT,1,1) = U11
      CARAY(NPOINT,2,2) = U11
      PARAY(NPOINT,1,1) = C
      PARAY(NPOINT,2,1) = ANGLE
      PARAY(NPOINT,1,2) = VSWR
      PARAY(NPOINT,2,2) = DBL
      GO TO 2060
 1130 IF (KODE.NE.2.AND.KODE.NE.5.AND.IN3.NE.43) GO TO 1160
C
C  **PRINT OPTION "S1 TO S6"**
C
      IF (IN3.NE.43) GO TO 1150
C
C    **COMPUTE S-MATRIX FOR "S3" OPTION**
C
      DO 1140 I = 1, 2
         DO 1140 J = 1, 2
            AB1(I,J) = AAA(IN2,I,J)
 1140 CONTINUE
      CALL ABCDM (AB1,AB2,50.0,4)
      UP1 = CMPLX(SLIMP(INDFF,1),SLIMP(INDFF,2))
      UP1 = (UP1-50.)/(UP1+50.)
      UP2 = CMPLX(SLIMP(INDFF,3),SLIMP(INDFF,4))
      UP2 = (UP2-50.)/(UP2+50.)
      QX = CABS(UP1*UP1)
      QY = CABS(UP2*UP2)
      P11 = (1.-CONJG(UP1))/(CABS(1.-UP1))*SQRT(1.-QX)
      P12 = (1.-CONJG(UP2))/(CABS(1.-UP2))*SQRT(1.-QY)
      P21 = CONJG(P11)
      P22 = CONJG(P12)
      UD11 = 1./((1.-UP1*AB2(1,1))*(1.-UP2*AB2(2,2))-UP1*UP2*AB2(2,1))
      A(1,1) = ((UD11*P21/P11)*((1.-UP2*AB2(2,2))*(AB2(1,1)-CONJG(UP1))+
     *   UP2*AB2(1,2)*AB2(2,1)))
      A(1,2) = ((UD11*P22/P11)*(1.-QX)*AB2(1,2))
      A(2,1) = ((UD11*P21/P12)*(1.-QY)*AB2(2,1))
      A(2,2) = ((UD11*P22/P12)*((1.-UP1*AB2(1,1))*(AB2(2,2)-CONJG(UP2))+
```

```
     *   UP1*AB2(1,2)*AB2(2,1)))
         IF (KODE.NE.2.AND.KODE.NE.5) GO TO 1180
         OUT(1,1) = CABS(A(1,1))
         OUT(2,1) = CABS(A(2,1))
         OUT(2,3) = CABS(A(2,2))
         OUT(1,3) = 0.0
         IF (CAIND.EQ.1) OUT(2,1) = ATTN*OUT(2,1)
         GO TO 2000
C
C    ** COMPUTE MAGN. OF S11,S21 AND S22 FOR OPTIMIZATION,
C       (S1 AND S2 OPTION)**
C
 1150 CHARI = DATX(IXX,1)
         U11 = AAA(IN2,1,1)
         U12 = AAA(IN2,1,2)/CHARI
         U21 = AAA(IN2,2,1)*CHARI
         U22 = AAA(IN2,2,2)
         UDEN = CABS(1./(U11+U12+U21+U22))
         UN1 = U12-U21
         UN2 = U11-U22
         UN3 = CABS(UN1+UN2)
         UN4 = CABS(UN1-UN2)
         OUT(1,1) = UN3*UDEN
         OUT(2,1) = 2.*UDEN
         OUT(2,3) = UN4*UDEN
         OUT(1,3) = 0.0
         IF (CAIND.EQ.1) OUT(2,1) = ATTN*OUT(2,1)
         IF (NSTAR.NE.9) GO TO 2000
 1160 DO 1170 I = 1, 2
            DO 1170 J = 1, 2
               AB4(I,J) = AAA(IN2,I,J)
 1170 CONTINUE
         CALL ABCDM (AB4,A,DATX(IXX,1),4)
 1180 CONTINUE
         CAC = CABS(A(1,1)-CARAY(INDFF,1,1))
         IF (KODE.EQ.10) CALL GRID
C
C *****COMPUTE DELAY FOR "S7" OPTION
C
         IF (NSTAR.NE.9) GO TO 1200
         IF (IFPERT.NE.-1) GO TO 1190
         CALL RPOLAR (A(2,1),ANGLE1,AMAG)
         IF (ANGLE1.LT.0.0) ANGLE1 = 360.+ANGLE1
         GO TO 2060
 1190 CALL RPOLAR (A(2,1),ANGLE0,AMAG)
         IF (ANGLE0.LT.0.0) ANGLE0 = 360.+ANGLE0
         DELAY = (ANGLE1-ANGLE0)/(360.0*FPERT*FR)
 1200 IF (KODE.EQ.2.OR.KODE.EQ.5) GO TO 2000
         IF (KODE.NE.10) CALL SWRIT
         GO TO 2000
C
C    **CONVERT POLAR S- OR RECT. Y-PARAMETERS TO ABCD PARAMETERS**
C
 1210 INDTR = INDTR+1
         IF ((NOIT.GT.0.OR.ITER.GE.1).AND.KODE.NE.10) GO TO 1310
```

```
      IF (KYM.EQ.0) GO TO 1240
      DO 1220 I = 1, 2
         DO 1220 J = 1, 2
            AB1(I,J) = YMAT(INDTR,I,J)
 1220 CONTINUE
      CALL ABCDM (AB1,A,DATX(IPRI,1),3)
      DO 1230 I = 1, 2
         DO 1230 J = 1, 2
            BBB(INDTR,I,J) = A(I,J)
 1230 CONTINUE
      GO TO 1980
 1240 IF (DATX(IXX,1).EQ.0.) GO TO 2400
      IF (KSIN.EQ.3) GO TO 1280
      DO 1250 I = 1, 2
         DO 1250 J = 1, 2
            PPRD = RPD*PS(INDTR,I,2*J)
            K = I+2*J-2
            M = 2*J-1
            X = PS(INDTR,I,M)
            XCOORD = X*COS(PPRD)
            YCOORD = X*SIN(PPRD)
            IF (KODE.NE.10) GO TO 1250
            DELTA = PERUP(1,K)*X
            XCOORD = XCOORD+DELTA*(1.0-2.0*RANF(0.0))
            YCOORD = YCOORD+DELTA*(1.0-2.0*RANF(0.0))
 1250       AB1(I,J) = CMPLX(XCOORD,YCOORD)
      CALL ABCDM (AB1,A,DATX(IXX,1),5)
      GO TO 1260
 1260 DO 1270 I = 1, 2
         DO 1270 J = 1, 2
            BBB(INDTR,I,J) = A(I,J)
 1270 CONTINUE
      GO TO 1980
C
C1025  IF(IN2.EQ.JSTK) GO TO 104
C
 1280 CONTINUE
      DO 1290 I = 1, 2
         DO 1290 J = 1, 2
            AB1(I,J) = BBB(INDTR,I,J)
 1290 CONTINUE
      CALL ABCDM (AB1,A,DATX(IXX,1),5)
      DO 1300 I = 1, 2
         DO 1300 J = 1, 2
            BBB(INDTR,I,J) = A(I,J)
 1300 CONTINUE
 1310 DO 1320 I = 1, 2
         DO 1320 J = 1, 2
            AAA(IN2,I,J) = BBB(INDTR,I,J)
 1320 CONTINUE
      GO TO 2000
C
C   **CASCADE IN3-IN2+1 ABCD MATRICES**
C
 1330 I = IN2
```

```
      J = IN2+1
      IF (CABS(AAA(I,1,1)).LT.1.0E-12.OR.CABS(AAA(J,1,1)).LT.1.0E-12) GO
     *    TO 2440
      AB1(1,1) = AAA(I,1,1)*AAA(J,1,1)+AAA(I,1,2)*AAA(J,2,1)
      AB1(2,1) = AAA(I,2,1)*AAA(J,1,1)+AAA(I,2,2)*AAA(J,2,1)
      AB1(1,2) = AAA(I,1,1)*AAA(J,1,2)+AAA(I,1,2)*AAA(J,2,2)
      AB1(2,2) = AAA(I,2,1)*AAA(J,1,2)+AAA(I,2,2)*AAA(J,2,2)
      NCB = IN3-IN2-1
      IF (NCB.EQ.0) GO TO 1730
      J = J+1
      DO 1350 IJJ = 1, NCB
         IJ = IJJ
         IF (CABS(AAA(J,1,1)).LT.1E-12) GO TO 2440
         A(1,1) = AB1(1,1)*AAA(J,1,1)+AB1(1,2)*AAA(J,2,1)
         A(1,2) = AB1(1,1)*AAA(J,1,2)+AB1(1,2)*AAA(J,2,2)
         A(2,1) = AB1(2,1)*AAA(J,1,1)+AB1(2,2)*AAA(J,2,1)
         A(2,2) = AB1(2,1)*AAA(J,1,2)+AB1(2,2)*AAA(J,2,2)
         IF (IJ.EQ.NCB) GO TO 1980
         DO 1340 M = 1, 2
            DO 1340 N = 1, 2
               AB1(N,M) = A(N,M)
1340     CONTINUE
         J = J+1
1350  CONTINUE
      GO TO 1980
1360  ZIN = (AAA(IN3,1,1)*DATX(IXX,1)+AAA(IN3,1,2))/(AAA(IN3,2,1)*
     *   DATX(IXX,1)+AAA(IN3,2,2))
      ZINR = REAL(ZIN)
      ZINI = AIMAG(ZIN)
      CALL SIMP (A,ZINR,ZINI)
      GO TO 1980
1370  YIN = (AAA(IN3,2,1)*DATX(IXX,1)+AAA(IN3,2,2))/(AAA(IN3,1,1)*
     *   DATX(IXX,1)+AAA(IN3,1,2))
      YINR = REAL(YIN)
      YINI = AIMAG(YIN)
      CALL PIMP (A,YINR,YINI)
      GO TO 1980
1380  R = VARX(KLM)*DATX(INRJX,1)
      WX = VARY(KLM)*DATX(INRJX,1)
      IF (IN3.EQ.1) GO TO 1390
      CALL SIMP (A,R,WX)
      IF (WX) 1410 , 1400 , 1420
1390  DENO = R*R+WX*WX
      G = R/DENO
      B = -WX/DENO
      CALL PIMP (A,G,B)
      IF (WX) 1410 , 1400 , 1420
1400  WRITE (IOT,2570) R,WX
      GO TO 1980
1410  C = 1.E6/(W*ABS(WX))
      WXX = ABS(WX)
      WRITE (IOT,2580) R,WXX,C
      GO TO 1980
1420  QL = 1000.*WX/W
      WRITE (IOT,2590) R,WX,QL
```

```
      GO TO 1980
C
C   **COMPUTE ATTENUATION AND ABCD MATRIX OF LOSSY COAX. CABLE**
C
 1430 IF (CAIND.EQ.1) GO TO 1440
      CALEN = DATX(IXX,1)/(DATX(IXX,2)*SQRT(FF)+DATX(IXX,3)*FF)
      CAIND = 1
 1440 ATTN = -.11513*CALEN*(DATX(IXX,2)*SQRT(FR)+DATX(IXX,3)*FR)
      ATTN = EXP(ATTN)
      AAA(IN2,1,1) = 1.0
      AAA(IN2,1,2) = 0.0
      AAA(IN2,2,1) = 0.0
      AAA(IN2,2,2) = 1.0
      GO TO 2000
C
C   **DATA GIVEN IN DATA FILE**
C
 1450 KIMP = KIMP+1
      IF (2-IN3) 2350 , 1460 , 1470
 1460 CALL SIMP (A,VIMP(KIMP,1),VIMP(KIMP,2))
      GO TO 1980
 1470 AB = VIMP(KIMP,1)
      AC = VIMP(KIMP,2)
      AD = 1./(AB*AB+AC*AC)
      AE = AB*AD
      AF = -AC*AD
      CALL PIMP (A,AE,AF)
      GO TO 1980
C
C  ** COMPUTE ABCD MATRIX IN THE "IN2" LOCATION THAT IS THE
C     INVERSE OF THE SERIES IMPEDANCE IN LOCATION "IN3"***
C
 1480 AAA(IN2,1,1) = 1.0
      AAA(IN2,1,2) = 0.0
      AAA(IN2,2,1) = AAA(IN3,1,2)/DATX(IXX,1)/DATX(IXX,1)
      AAA(IN2,2,2) = 1.0
      GO TO 2000
C
C   **COMPUTE ABCD MATRIX FOR CIRCULAR OPEN STUB**
C
 1490 CALL CSTUB
      IF (JERRX.EQ.1) GO TO 2480
      IF (INDFF.EQ.1.AND.ITER.EQ.0.AND.ICST.EQ.0) WRITE (IOT,2600)
      IF (ICST.EQ.0) ICST = 1
      IF (IFTLER.NE.0) GO TO 2480
C
C
      AAA(IN2,1,1) = 1.0
      AAA(IN2,1,2) = 0.0
      AAA(IN2,2,1) = CMPLX(0.,ZCST)
      AAA(IN2,2,2) = 1.0
      GO TO 2000
C
C  **CODE "EQU" -- CREATE NEW TWO-PORT**
C
```

```
 1500 AAA(IN2,1,1) = AAA(IN3,1,1)
      AAA(IN2,1,2) = AAA(IN3,1,2)
      AAA(IN2,2,1) = AAA(IN3,2,1)
      AAA(IN2,2,2) = AAA(IN3,2,2)
      GO TO 2000
C
C  **  VARIABLE GAIN (VGB , 1521) AND PHASE (VPB, 1522) BLOCKS  **
C
 1510 X = DATX(IXX,1)
      AAA(IN2,2,1) = CMPLX(0.,0.)
      AAA(IN2,1,2) = CMPLX(0.,0.)
      IF (IN3.EQ.42) X = -X
      IF (DATX(IXX,4).LT.FR.AND.DATX(IXX,4).NE.0.) X = X+(ALOG(FR)-ALOG(
     *   DATX(IXX,4)))*DATX(IXX,2)*1.4427
      IF (DATX(IXX,5).LT.FR.AND.DATX(IXX,5).NE.0.) X = X-(ALOG(FR)-ALOG(
     *   DATX(IXX,5)))*DATX(IXX,3)*1.4427
      AAA(IN2,1,1) = CMPLX(EXP(-.115*X),0.)
      AAA(IN2,2,2) = AAA(IN2,1,1)
      GO TO 2000
 1520 X = DATX(IXX,1)*.01745
      AAA(IN2,2,1) = CMPLX(0.,0.)
      AAA(IN2,1,2) = CMPLX(0.,0.)
      IF (IN3.EQ.42) X = -X
      IF (DATX(IXX,4).LT.FR.AND.DATX(IXX,4).NE.0.) X = X+(ALOG(FR)-ALOG(
     *   DATX(IXX,4)))*DATX(IXX,2)*.02518
      IF (DATX(IXX,5).LT.FR.AND.DATX(IXX,5).NE.0.) X = X-(ALOG(FR)-ALOG(
     *   DATX(IXX,5)))*DATX(IXX,3)*.02518
      AAA(IN2,1,1) = 1./CMPLX(COS(X),SIN(X))
      AAA(IN2,2,2) = AAA(IN2,1,1)
      GO TO 2000
C
C **CODE "CHA":INTERCHANGE INPUT AND OUTPUT, CREATE NEW TWO-PORT*
C
 1530 U11 = 1./(AAA(IN3,1,1)*AAA(IN3,2,2)-AAA(IN3,1,2)*AAA(IN3,2,1))
      A(1,1) = AAA(IN3,2,2)*U11
      A(2,2) = AAA(IN3,1,1)*U11
      A(1,2) = AAA(IN3,1,2)*U11
      A(2,1) = AAA(IN3,2,1)*U11
      GO TO 1980
C
C  **COMPUTE ABCD MATRIX OF MUTUAL INDUCTOR**
C
 1540 IF (IN3.EQ.41) AMU = -DATX(IXX,3)
      IF (IN3.EQ.42) AMU = DATX(IXX,3)
      AAA(IN2,1,1) = (-DATX(IXX,1)/AMU)
      AAA(IN2,1,2) = CMPLX(0.,-((DATX(IXX,1)*DATX(IXX,2)-DATX(IXX,3)*
     *   DATX(IXX,3))/(AMU*1E3))*W)
      AAA(IN2,2,1) = CMPLX(0.,1.E3/(AMU*W))
      AAA(IN2,2,2) = (-DATX(IXX,2)/AMU)
      GO TO 2000
C
C  **COMPARE TWO ABCD MATRICES AND DETERMINE ERROR DIFFERENCE**
C  **AAA(IN3) IS THE MODEL AND AAA(IN2) IS THE MEASURED DATX**
C
 1550 IF (KODE.EQ.9) GO TO 2060
```

```
      DO 1560 I = 1, 2
         DO 1560 J = 1, 2
C
C       AB4(I,J)=AAA(IN3,I,J)
C
              AB2(I,J) = AAA(IN3,I,J)
C
C       IF (CABS(AB4(I,J)).LT.0.01) AB4(I,J)=CMPLX(0.01,0.0)
C
              AB1(I,J) = AAA(IN2,I,J)
 1560 CONTINUE
      CALL ABCDM (AB1,AB3,DATX(IPRI,1),4)
      CALL ABCDM (AB2,AB4,DATX(IPRI,1),4)
      DO 1570 I = 1, 2
         DO 1570 J = 1, 2
            IF (CABS(AB4(I,J)).LT..01) AB4(I,J) = CMPLX(.01,0.)
 1570 CONTINUE
      IF (KODE.EQ.1.OR.ERR.EQ.0.0) GO TO 1580
      IF (ERR.GT.0.002.OR.INEXP.EQ.1) GO TO 1580
      WRITE (IOT,2610)
      CALL TAKEIN (INPET,NOITEM)
      NEWEX = ARRAY(1)
      INEXP = 1
      IDIV = 2*NEWEX-4
C
C     DIFFERENT WEIGHTS IN DIFFERENT BANDS
C
 1580 CONTINUE
      IROW = 1
      IF (INDFF.GT.IBAND(1).AND.INDFF.LE.IBAND(2)) IROW = 2
      IF (INDFF.GT.IBAND(2).AND.INDFF.LE.IBAND(3)) IROW = 3
      IF (IN1.EQ.47) GO TO 1590
      IF (WI(IROW,1).NE.0.0) ERSET = WI(IROW,1)*(CABS(1.0-AB3(1,1)/AB4(1
     *   ,1)))**NEWEX+ERSET
      IF (WI(IROW,2).NE.0.0) ERSET = WI(IROW,2)*(CABS(1.0-AB3(2,2)/AB4(2
     *   ,2)))**NEWEX+ERSET
      IF (WI(IROW,3).NE.0.0) ERSET = WI(IROW,3)*(CABS(1.0-AB3(2,1)/AB4(2
     *   ,1)))**NEWEX+ERSET
      IF (S21(IROW).NE.0.0) ERSET = S21(IROW)*(CABS(1.0-AB3(1,2)/AB4(1,2
     *   )))**NEWEX+ERSET
      GO TO 2000
 1590 IF (WI(IROW,1).NE.0.0) ERSET = WI(IROW,1)*(CABS(AB3(1,1))-(AB4(1,1
     *   )))**NEWEX+ERSET
      IF (WI(IROW,2).NE.0.0) ERSET = WI(IROW,2)*(CABS(AB3(2,2))-(AB4(2,2
     *   )))**NEWEX+ERSET
      IF (WI(IROW,3).NE.0.0) ERSET = WI(IROW,3)*(CABS(AB3(2,1))-(AB4(2,1
     *   )))**NEWEX+ERSET
      IF (S21(IROW).NE.0.0) ERSET = S21(IROW)*(CABS(AB3(1,2))-(AB4(1,2))
     *   )**NEWEX+ERSET
      GO TO 2000
C
C      ** CALL SUBROUTINE NOISE TO COMPUTE NOISE FIGURES  ***
C
 1600 CALL SNOISE
      IF (IFTLER.EQ.8.OR.IFTLER.EQ.9) GO TO 2480
```

```
      GO TO 2000
C
C  **STORE IMPEDANCE FOR "ONE" AND "IMP" IN "VIMP" IN PROPER FORM
C
 1610 KIMP = KIMP+1
      IF (ITER.GE.1) GO TO 1650
      IF (IN3.EQ.38) GO TO 1640
      IF (IN3.EQ.33) GO TO 1620
      GO TO 2410
 1620 IF (DATX(IXX,1).EQ.0.0) GO TO 2420
      R = RPD*VIMP(KIMP,2)
      U11 = CMPLX(VIMP(KIMP,1)*COS(R),VIMP(KIMP,1)*SIN(R))
C
C **IF THE REFL. COEFF. IS NOT FOR MODEL, THEN CONVERT TO IMPEDAN
C
      IF (IN2.NE.JSOK) GO TO 1630
      IF (MODIM.EQ.1) U11 = DATX(IXX,1)*(1.+U11)/(1.-U11)
      STRC(INDFF) = U11
      GO TO 2000
 1630 U11 = DATX(IXX,1)*(1.+U11)/(1.-U11)
      VIMP(KIMP,1) = REAL(U11)
      VIMP(KIMP,2) = AIMAG(U11)
 1640 IF (IN2.NE.JSOK) GO TO 1650
      U11 = CMPLX(VIMP(KIMP,1),VIMP(KIMP,2))
      IF (MODIM.EQ.2) U11 = (U11-DATX(IPRI,1))/(U11+DATX(IPRI,1))
      STRC(INDFF) = U11
      GO TO 2000
 1650 AAA(IN2,1,1) = 1.0
      AAA(IN2,1,2) = 1.0E12
      SQU = VIMP(KIMP,1)*VIMP(KIMP,1)+VIMP(KIMP,2)*VIMP(KIMP,2)
      AAA(IN2,2,1) = CMPLX(VIMP(KIMP,1)/SQU,-VIMP(KIMP,2)/SQU)
      AAA(IN2,2,2) = 1.E12*AAA(IN2,2,1)
      GO TO 2000
C
C  **"SEO" OR "SMO"- COMPUTE ERROR BETWEEN INPUT OF 2 1-PORTS**
C
 1660 IF (KODE.EQ.9) GO TO 2060
      IF (ITER.GE.1) GO TO 1670
      POW = DATX(IPRI,3)+2.
 1670 U22 = AAA(IN2,1,1)*DATX(IPRI,2)+AAA(IN2,1,2)
      U22 = U22/(AAA(IN2,2,1)*DATX(IPRI,2)+AAA(IN2,2,2))
      U11 = STRC(INDFF)
C
C -- WEIGHTING FACTORS FOR "SEO" AND "SMO"
C
      IROW = 1
      IF (INDFF.GT.IBAND(1).AND.INDFF.LE.IBAND(2)) IROW = 2
      IF (INDFF.GT.IBAND(2).AND.INDFF.LE.IBAND(3)) IROW = 3
      WT = WI(IROW,1)
C
C  **COMPUTE ERROR FOR "SEO"**
C
C
      IF (ABS(REAL(U11)).LT.0.1) U11 = CMPLX(.1,AIMAG(U11))
      IF (ABS(AIMAG(U11)).LT.0.1) U11 = CMPLX(REAL(U11),.1)
```

```
      IF (IN1.EQ.46) ERSET = WT*(ABS((REAL(U22)-REAL(U11))/REAL(U11))+
     *   ABS((AIMAG(U22)-AIMAG(U11))/AIMAG(U11)))**POW+ERSET
C
C
C  **COMPUTE ERROR FOR "SMO"**
C
      IF (IN1.NE.48) GO TO 2000
      U22 = (U22-DATX(IPRI,1))/(U22+DATX(IPRI,1))
      IF (CABS(U11).LT.0.05) U11 = CMPLX(0.05,0.0)
      EXHOLD = (1-CABS(U22)/CABS(U11))
      ERSET = ERSET+WT*EXHOLD**POW
      GO TO 2000
 1680 IND3R = IND3R+1
      IF (DATX(IXX,1).EQ.0.0) GO TO 2400
      IF (IN3.EQ.48) GO TO 1690
      ZIN = AAA(IN3,1,1)/AAA(IN3,2,1)
      U11 = (ZIN-DATX(IXX,1))/(ZIN+DATX(IXX,1))
      U11 = U11/(1.-U11*S3R(IND3R,2,2))
      AB1(1,1) = S3R(IND3R,1,1)+S3R(IND3R,1,2)*S3R(IND3R,2,1)*U11
      AB1(1,2) = S3R(IND3R,1,3)+S3R(IND3R,1,2)*S3R(IND3R,2,3)*U11
      AB1(2,1) = S3R(IND3R,3,1)+S3R(IND3R,3,2)*S3R(IND3R,2,1)*U11
      AB1(2,2) = S3R(IND3R,3,3)+S3R(IND3R,3,2)*S3R(IND3R,2,3)*U11
      CALL ABCDM (AB1,A,DATX(IXX,1),5)
      GO TO 1980
 1690 AAA(IN2,1,1) = CMPLX(IND3R+.5,0.)
      AAA(IN2,1,2) = CMPLX(DATX(IXX,1),0.)
      AAA(IN2,2,2) = CMPLX(987.,987.)
      GO TO 2060
C
C    **EXCHANGE PORT 1 OR PORT 2 AND GROUND**
C
C    **X1G     IN1=54?   X2G   IN1=55**
C
 1700 I = IN1-53
      J = 56-IN1
      P11 = AAA(IN3,I,I)-CUNIT
      IF (CABS(P11).EQ.0.0) GO TO 1710
      P11 = CUNIT/P11
C
C  TO ALLOW SAME CODE FOR ORIGINAL AND INTERCHANGED TWO-PORTS; FOR
C  EXAMPLE, "X1G AA AA", CHANGED LEFT HAND SIDE FROM "AAA(IN2,.,.)"
C  TO "A(.,.)" AND THEN JUMPED TO 980 RATHER THAN 989
C
      A(I,I) = AAA(IN3,I,I)*P11
      A(1,2) = AAA(IN3,1,2)*P11
      A(2,1) = AAA(IN3,2,1)*P11
      A(J,J) = AAA(IN3,2,1)*AAA(IN3,1,2)*P11-AAA(IN3,J,J)+CUNIT
      GO TO 1980
 1710 IFTLER = 39
      GO TO 2480
C
C  **NODAL ANALYSIS, "CON", "DEF" AND "MUC"**
C
 1720 CALL NODAL
      IF (JERRX.NE.0) GO TO 2480
```

```
      GO TO 2000
C
 1730 DO 1740 M = 1, 2
         DO 1740 N = 1, 2
            A(M,N) = AB1(M,N)
 1740 CONTINUE
      GO TO 1980
C
C  **SPECIAL CODES "SUM", "PRD", "DIF", AND "RAT"**
C
 1750 CONTINUE
C
C  **HOW MANY ENTRIES\**
C
      DO 1760 ITEMS1 = 1, 6
         ITEMS = 7-ITEMS1
         IF (DATX(IXX,ITEMS).NE.0.0) GO TO 1770
 1760 CONTINUE
 1770 CONTINUE
C
C  **READ DATA**
C
      IPOS1 = 0
      IPOS2 = 0
      IPOS3 = 0
      IPOS4 = 0
      SPDR1 = 0.0
      SPDR2 = 0.0
      IPOS1 = DATX(IXX,1)-2.5
      GO TO (1960,1780,1790,1800,1810,1820) ITEMS
 1780 IPOS2 = IPOS1
      SPDR1 = DATX(IXX,2)
      GO TO 1830
 1790 IPOS2 = DATX(IXX,2)-2.5
      SPDR1 = DATX(IXX,3)
      GO TO 1830
 1800 IPOS2 = IPOS1
      SPDR1 = DATX(IXX,2)
      IPOS3 = DATX(IXX,3)-2.5
      IPOS4 = IPOS3
      SPDR2 = DATX(IXX,4)
      GO TO 1830
 1810 IPOS2 = IPOS1
      SPDR1 = DATX(IXX,2)
      IPOS3 = DATX(IXX,3)-2.5
      IPOS4 = DATX(IXX,4)-2.5
      SPDR2 = DATX(IXX,5)
      GO TO 1830
 1820 IPOS2 = DATX(IXX,2)-2.5
      SPDR1 = DATX(IXX,3)
      IPOS3 = DATX(IXX,4)-2.5
      IPOS4 = DATX(IXX,5)-2.5
      SPDR2 = DATX(IXX,6)
 1830 CONTINUE
C
```

```
C  **FIND POINTER "J" TO INDEPENDENT VARIABLE**
C
      DO 1840 K = 1, 70
         J = IXX-K
         IF (J.EQ.0) GO TO 1900
         IF (INC2(J).EQ.IN3) GO TO 1850
 1840 CONTINUE
C
C  **FIND POINTER "L" TO DEPENDENT VARIABLE**
C
 1850 CONTINUE
      L = IXX
      DO 1860 K = 1, 70
         L = L+1
         IF (L.EQ.69) GO TO 1900
         IF (INC2(L).NE.IN2) GO TO 1860
         IF (INC1(L).LT.60.OR.INC1(L).GT.63) GO TO 1870
 1860 CONTINUE
 1870 CONTINUE
      IF (SPDR1.EQ.1.E-20) SPDR1 = 0.0
      IF (SPDR2.EQ.1.E-20) SPDR2 = 0.0
      IF (IPOS1.LT.1.OR.IPOS1.GT.6.OR.IPOS2.LT.1.OR.IPOS2.GT.6)
     *    GO TO 1960
      IF ((ITEMS.GT.3).AND.(IPOS3.LT.1.OR.IPOS3.GT.6.OR.IPOS4.LT.1.OR.
     *   IPOS4.GT.6)) GO TO 1960
      I = IN1-59
      IF (INDFF.NE.1.OR.ITER.NE.0) GO TO 1890
      DO 1880 JJ = 1, NOPT
         IF (KX(JJ).NE.IXX) GO TO 1880
         IF (I.EQ.1) PERUP(2,JJ) = SPDR1
         IF (I.EQ.1.AND.ITEMS.GT.3) PERUP(2,JJ+1) = SPDR2
         IF (I.EQ.3) PERUP(3,JJ) = SPDR1
         IF (I.EQ.3.AND.ITEMS.GT.3) PERUP(3,JJ+1) = SPDR2
 1880 CONTINUE
 1890 CONTINUE
      GO TO (1910,1920,1930,1940), I
 1900 IFTLER = 19
      GO TO 2480
C
C  **SUM**
C
 1910 TEMP1 = SPDR1-DATX(J,IPOS2)
      IF (ITEMS.GT.3) TEMP2 = SPDR2-DATX(J,IPOS4)
      GO TO 1950
C
C  **PRD**
C
 1920 TEMP1 = DATX(J,IPOS2)
      IF (TEMP1.LE.0.0) GO TO 1960
      TEMP1 = SPDR1/TEMP1
      IF (ITEMS.LE.3) GO TO 1950
      TEMP2 = DATX(J,IPOS4)
      IF (TEMP2.LE.0.0) GO TO 1960
      TEMP2 = SPDR2/TEMP2
      GO TO 1950
```

```
C
C  **DIF**
C
 1930 TEMP1 = DATX(J,IPOS2)-SPDR1
      IF (ITEMS.GT.3) TEMP2 = DATX(J,IPOS4)-SPDR2
      GO TO 1950
C
C  **RAT**
C
 1940 IF (SPDR1.LE.0.0) GO TO 1960
      TEMP1 = DATX(J,IPOS2)/SPDR1
      IF (ITEMS.LE.3) GO TO 1950
      IF (SPDR2.LE.0.0) GO TO 1960
      TEMP2 = DATX(J,IPOS4)/SPDR2
      GO TO 1950
 1950 IF (TEMP1.LE.0.0) GO TO 1960
      DATX(L,IPOS1) = TEMP1
      IF (ITEMS.LE.3) GO TO 2000
      IF (TEMP2.LE.0.0) GO TO 1960
      DATX(L,IPOS3) = TEMP2
      GO TO 2000
 1960 IFTLER = 37
      GO TO 2480
C
C
C  **NEG**
C
 1970 CONTINUE
      A(1,1) = AAA(IN3,1,1)
      A(2,2) = AAA(IN3,2,2)
      A(1,2) = -AAA(IN3,1,2)
      A(2,1) = -AAA(IN3,2,1)
      GO TO 1980
 1980 DO 1990 I = 1, 2
         DO 1990 J = 1, 2
            AAA(IN2,I,J) = A(I,J)
 1990 CONTINUE
 2000 CONTINUE
 2010 IF (KODE.NE.2.AND.KODE.NE.5) GO TO 2060
      IF (IFPERT.EQ.-1) GO TO 2060
      IF (MODEL.EQ.1) GO TO 2060
      IF (IN1.NE.7) GO TO 2060
      L = 1
      IF (INDFF.GT.IBAND(1)) L = 2
      IF (INDFF.GT.IBAND(2).AND.IBAND(2).NE.0) L = 3
      AA = OUT(1,1)
      BB = OUT(2,3)
      IF (OUT(1,3).GT.100.0) DD = OUT(2,1)
      IF (OUT(1,3).GT.100.0) GO TO 2020
      IF (OUT(2,1).EQ.0.0.AND.WI(L,3).NE.0.0) GO TO 2430
      IF (OUT(2,1).EQ.0.0) GO TO 2020
      DD = 20.0*ALOG10(OUT(2,1))
 2020 CC = DD-S21(L)
      IF (WI(L,1).LT.0.0) AA = 1.0/AA
      IF (WI(L,2).LT.0.0) BB = 1.0/BB
```

```
      IF (WI(L,3).LT.0.0) CC = 1.0/CC
      ILT = ILTGT(L)
C
      GO TO (2050,2040,2030), ILT
 2030 IF (DD.GT.S21(L)) CC = 0.0
      GO TO 2050
 2040 IF (DD.LT.S21(L)) CC = 0.0
 2050 ERRORN = ABS(WI(L,1))*AA*AA+ABS(WI(L,2))*(BB*BB)+ABS(WI(L,3))*CC*
     *   CC
      ERRD1 = ERRD+ERRORN
      ERRD = ERRD1
C
C  **STATEMENT 990 IS THE END OF THE INTERCONNECTION DO-LOOP
C
 2060 IXX = IXX+1
      IF (IXX.LE.NCO) GO TO 290
C
C  **STATEMENT 999 IS THE END OF THE FREQUENCY DO-LOOP
C
 2070 INOIS = 0
C
C *****COMPUTE ERROR OF GROUP DELAY FOR EACH FREQUENCY
C
      IF (NSTAR.NE.1.AND.NSTAR.NE.9) GO TO 2110
      IF (IFPERT.EQ.0) GO TO 2090
      IFPERT = 0
      IF (NT.GT.0) INDTR = INDTR-NT
C
C     IF(KIMP .GT. 0) KIMP=KIMP-1
C
      IF (KIMP.GT.0) KIMP = KIMP-NIMP
      IF (ITHR.GT.0) IND3R = IND3R-ITHR
C
C
      IF (NOISE.EQ.0) GO TO 280
      DO 2080 INOI = 1, 2
         DO 2080 KK = 1, 5
            FIG(INOI,INDFF,KK) = FIGG(INOI,INDFF,KK)
 2080 CONTINUE
      GO TO 280
 2090 IF (DELAYP.EQ.0.0) GO TO 2100
      DELAYI = (DELAY-DELAYP)*1000.
      DELAYI = DELAYI*DELAYI
      ERRDL = ERRDL+ABS(WI(L,6))*DELAYI
 2100 DELAYP = DELAY
      IFPERT = -1
      IF (INDFF.EQ.IBAND(1).OR.INDFF.EQ.IBAND(2).OR.INDFF.EQ.IBAND(3))
     *    DELAYP = 0.0
 2110 INDFF = INDFF+1
      IF (INDFF.LE.NF) GO TO 280
      ITER = 1
      IF (MODEL.EQ.1) GO TO 2120
      IF (NOOPT.NE.0.AND.NFOPT.EQ.0) NFOPT = NF
      ERRAN = ERRAN*ERRAN*AOPT
      IF (KODE.EQ.2.OR.KODE.EQ.5) ERRD = (ERRD1+ERNO+ERRAN+ERRDL)/NFOPT
```

```
      GO TO 2130
2120  ERRD = ERSET/NF*10.**IDIV
      ERSET = 0.0
2130  KIMP = 0
      ERNO = 0.
      INDTR = 0
      IND3R = 0
      ERRAN = 0.0
      DELAYP = 0.0
      ERRDL = 0.0
      IF (KODE.EQ.8.AND.ICOUNT.LT.IFUNCT.AND.NOOPT.NE.0) GO TO 2290
      IF (MKODE.EQ.3.AND.KODE.EQ.9) KODE = 10
      IF (MKODE.EQ.3) GO TO 270
      IF (KODE.EQ.8.AND.ICOUNT.LT.IFUNCT) GO TO 180
      IF (MKODE.EQ.2) KODE = 10
      IF (MKODE.EQ.2) GO TO 100
      IF (KODE.EQ.10) GO TO 120
      IF (KODE.GT.6.AND.NOOPT.EQ.0) GO TO 2340
      IF (NOOPT.NE.0.AND.KLM.EQ.1) GO TO 2270
      IF (NOOPT.NE.0.AND.KLM.GT.1) GO TO 2280
2140  CONTINUE
      IF (KLM.LT.17.AND.KODE.EQ.6) GO TO 250
      KLM = 0
      IF (KODE.NE.5) GO TO 2240
      IF (IMA.EQ.0) GO TO 2220
      IF (ERRD.LT.ERINJ) GO TO 2160
      DO 2150 K = 1, NOPT
         WRST(K) = DATX(KX(K),KY(K))
2150  CONTINUE
      ERINJ = ERRD
2160  IF (ERRD.GE.ERINI) GO TO 2180
      DO 2170 K = 1, NOPT
         SENDE(K) = DATX(KX(K),KY(K))
2170  CONTINUE
      ERINI = ERRD
      WRITE (IOT,2620) ERINI,ICOUNT,(SENDE(K),K=1,NOPT)
2180  ERRD = 0.0
      IF (XPER.EQ.100) GO TO 180
      GO TO 120
2190  IF (KODE.EQ.10) GO TO 2340
      ICOUNT = ICOUNT-1
      WRITE (IOT,2630) ICOUNT
      DO 2200 K = 1, NOPT
         WRITE (IOT,2640) K,SENDE(K)
2200  CONTINUE
      WRITE (IOT,2650)
      DO 2210 K = 1, NOPT
         WRITE (IOT,2640) K,WRST(K)
2210  CONTINUE
      IF (IMA.EQ.1) GO TO 2320
2220  EPLOT(IJII,IKII) = ERRD
      ERRD = 0.
      IF (IKII.LT.7) GO TO 230
      IKII = 0
      DATX(KX(2),KY(2)) = COM2
```

```
      NOIT = NOIT+1
      IF (IJII.LT.7) GO TO 220
      IJII = 0
      DATX(KX(1),KY(1)) = COM1
      WRITE (IOT,2660) (HORI(I),I=1,7)
      DO 2230 I = 1, 7
         WRITE (IOT,2670) VERT(I),(EPLOT(I,J),J=1,7)
 2230 CONTINUE
 2240 CONTINUE
      IF (KODE.NE.2) GO TO 2260

      IF (IQUAD.NE.1) GOTO 2245
      IF (IPRAM.EQ.0) ERR=ERRD
      IF (IPRAM.EQ.0) GOTO 2260
2245  CONTINUE

      IF (IPRAM.NE.0) GO TO 2250
      ERRSD = ERRD
      GO TO 60
C
C   **COMPUTE GRADIENTS (DOUBLE PRECISION) AND STORE IN GGD**
C
 2250 DATX(I88,I99) = SAVEC
C
C
      GGD = (ERRD-ERRSD)/PERT
      IF (GR(IPRAM).NE.0.000001) GR(IPRAM) = GGD
      IF (IPRAM.LT.NOPT) GO TO 60
C
C
      ERR = ERRSD
 2260 INDTR = 0
      IND3R = 0
      KIMP = 0
C
C **TRANSFER GRADIENTS INTO GR (SINGLE PRECISION)**
C
      IF (NOOPT.EQ.0.OR.KODE.EQ.6) GO TO 2340
      IF (KODE.EQ.2.OR.KODE.EQ.5) GO TO 2300
      IF (KODE.EQ.1.AND.MKODE.NE.1) GO TO 2300
 2270 IF (IPRIN.NE.1.AND.NOOPT.EQ.2) WRITE (IOT,2690)
      IF (IPRIN.NE.1) WRITE (IOT,2700)
      WRITE (IOT,2680)
 2280 IF (KODE.EQ.1.AND.MKODE.NE.1) GO TO 2300
 2290 IF (KODE.EQ.8.AND.ICOUNT.EQ.1.AND.IPRIN.NE.1.AND.NOOPT.EQ.2) WRITE
     *    (IOT,2690)
      IF (KODE.EQ.8.AND.ICOUNT.EQ.1.AND.IPRIN.NE.1) WRITE (IOT,2700)
C
C    **PRINT NOISE FIGURE DATA**
C
      IA = -1
      CALL SNOIS2
      WRITE (IOT,2710)
C
C
```

```
 2300 IF (KODE.EQ.3.OR.KODE.EQ.4.OR.(KODE.EQ.9.AND.MKODE.NE.2))
     *    GO TO 2340
      IF (MKODE.EQ.2) KODE = 10
      IF (MKODE.EQ.2) GO TO 100
      IF (KODE.EQ.10) GO TO 120
C
C
      DO 2310 J = 1, 2
         DO 2310 I = 1, NF
            DO 2310 K = 1, 5
               FIG(J,I,K) = FIGG(J,I,K)
 2310 CONTINUE
      IF (KLM.GT.0.AND.KODE.EQ.6) GO TO 2140
      IF (KODE.EQ.8.AND.ICOUNT.LT.IFUNCT) GO TO 180
      GO TO 2340
 2320 WRITE (IOT,2720)
      WRITE (IOT,2730)
      CALL TAKEIN (INPET,NOITEM)
      KODE = ARRAY(1)
      IF (KODE.EQ.0) GO TO 2340
      IF (KODE.NE.1.AND.KODE.NE.3.AND.KODE.NE.13) GO TO 2320
C
      DO 2330 I = 1, NOPT
         DATX(KX(I),KY(I)) = SENDE(I)
         VAX(I) = SENDE(I)
 2330 CONTINUE
 2340 RETURN
C
C***** DIAGNOSTIC PRINT STATEMENTS
C
 2350 IFTLER = 1
      GO TO 2480
 2360 IFTLER = 2
      GO TO 2480
 2370 IFTLER = 4
      GO TO 2480
 2380 IFTLER = 5
      GO TO 2480
 2390 IFTLER = 6
      GO TO 2480
 2400 IFTLER = 7
      GO TO 2480
 2410 IFTLER = 10
      GO TO 2480
 2420 IFTLER = 11
      GO TO 2480
 2430 IFTLER = 12
      GO TO 2480
 2440 IFTLER = 18
      GO TO 2480
 2450 IFTLER = 24
      GO TO 2480
 2460 IFTLER = 28
      GO TO 2480
 2470 IFTLER = 31
```

```
      GO TO 2480
 2480 CONTINUE
      JERRX = 1
      GO TO 2340
C
 2490 FORMAT (2X," SWEPT VARIABLE VALUES:")
 2500 FORMAT (15F10.2)
 2510 FORMAT (" ",F7.1,4(F9.1,F7.1))
 2520 FORMAT (" ",F7.1,8F8.1)
 2530 FORMAT (" OPTIONS: V(OUT)/V(S) [1], INSERTION GAIN [2]",
     *   " AND V(OUT)/V(1) [3]"/" ENTER 1, 2 OR 3")
 2540 FORMAT (" ",2X,F9.3,2F12.2,4X,F11.5)
 2550 FORMAT (" ",F9.3,3X,F6.3,5X,F6.1,F10.2,":1",F10.2,2F10.2)
 2560 FORMAT (" ",F9.3,3X,F6.3,5X,F6.1,12X,F10.2,2F10.2)
 2570 FORMAT (4X,"IMP.=(",F6.1,",",F6.1,") OHMS")
 2580 FORMAT (4X,"IMP.=(",F6.1,",-J",F6.1,") OHMS",5X,"C=",F10.2," PF")
 2590 FORMAT (4X,"IMP.=(",F6.1,", J",F6.1,") OHMS",5X,"L=",F10.2," NH")
 2600 FORMAT (" THE CST CODE IS AN APPROXIMATION.")
 2610 FORMAT (" MESSAGE TO USER-"/" PARTIAL GRADIENTS ARE ",
     *   "GETTING TOO SMALL FOR EFFECTIVE SEARCH."/" DO YOU WISH",
     *   " TO INCREASE THE VALUE OF THE EXPONENT OF THE ERROR/",
     *   " FUNCTION FROM 2 TO SOME HIGHER INTEGER NUMBER(= OR =)"/
     *   " IF ""NO"" TYPE 2 AND HIT RETURN, IF ""YES"" TYPE THE NEW ",
     *   "EXPONENT "/" (MUST BE EVEN NUMBER) AND HIT RETURN ")
 2620 FORMAT (" ",F14.4,I10/,15F10.2)
 2630 FORMAT (" OPTIMUM VARIABLES AFTER",I7," FUNCTION ","EVALUATIONS")
 2640 FORMAT (" VAR(",I2,")=",G14.8)
 2650 FORMAT (/" WORST CASE VARIABLE VALUES")
 2660 FORMAT (//12X,"ERROR FUNCTION VALUES WITH TWO COMPONENTS",
     *   " VARYING "//"  VAR.1",33X,"VAR.2",/" ","*-----*    ",
     *   "*-------------------------------------------------",
     *   "--------*"/" ",8X,7(2X,F7.2)//)
 2670 FORMAT (" ",F7.2,1X,7(2X,F7.2),/)
 2680 FORMAT (" NOISE FIGURE DATA FILE REQUIREMENTS CHANGED,"/
     *   " PLEASE SEE NEW EXAMPLE 12 IN THE MANUAL")
 2690 FORMAT (/" ",26X,"OVERALL AMPLIFIER")
 2700 FORMAT (" ",26X,"NOISE FIGURE DATA",/"   FREQ",5X,
     *   "MIN. NOISE FIG.   OPT. NOISE SOURCE   ACTUAL NF  NORM  ",/
     *   "    MHZ",11X,"DB",13X,"MAGN   ANGLE",8X,"DB",7X,"RN")
 2710 FORMAT (/)
 2720 FORMAT ("HOW TO USE CURRENT RESULTS\")
 2730 FORMAT (" ANAL(1), OPT(3), 1+3 (13) OR QUIT (0)  ")
C
      END
```

```
      SUBROUTINE FTP
C
      DIMENSION X(21),CX(21),V(420),AF(1),TAF(1),H(21)
      COMMON KODE,IFLAG,XITIME,ITRAP,JTRAP,ISENS,IPRIN,MKODE,EX,W,FR,NT,
     +NF,INDFF,NCO,IXX,MCODE,FF,IMA,IMA1,NOISE,IBANKF,IDCLOC
      COMMON/CO2/SEND(15),GR(15),ERR,NOPT,CONV,INEXP,IGR(15)
      COMMON/CO3/CARAY(30,2,2),DATX(70,6)
      COMMON/ANALZ/IBAND(3),RIND(70),INC1(70),INC2(70),INC3(70),KX(15),
     +KY(15)
      COMMON/TOLER/NVERT,TOL(15),NRANK,NCOUNT,NCALL,TRES,WIT(65)
      COMMON/QUAMOD/IQUAD,IFTP,IAPPR,MAJOR
      COMMON /MIM/B(253,1),IV(6,64)
      COMMON /BLKMRE/RMU(6,64),NRAND
      COMMON /BLKMRF/INTOL,KXTOL(6),KYTOL(6),IRATOL(6),PERTOL(2,6)
      COMMON /BLKMRG/NSPACE,KXMOD(21),KYMOD(21)
      COMPLEX CARAY
C
C     THIS SUBROUTINE IS DESIGNED TO SOLVE SEVERAL VERSIONS OF THE
C     FIXED TOLERANCE PROBLEM (FTP). CURRENTLY AVAILABLE ARE VERSIONS
C     A (FTP WITH ALL VERTICES OF THE TOLERANCE REGION),
C     B (FTP WITH A SPECIFIED NUMBER OF WORST VERTICES),
C     C (FTP WITH A THRESHOLD VALUE ON THE OBJECTIVE FUNCTION),
C     D (FTP WITH WEIGHTED VERTICES PLUS WEIGHTED NOMINAL),
C     E (FTP WITH MONTE-CARLO DESIGN)
C
      IF (KODE.NE.1) GO TO 900
10    PRINT(6,1000)
1000  FORMAT(" DO YOU WANT TO USE QUADRATIC APPROXIMATION FOR THE"/,
     +" FIXED TOLERANCE PROBLEM (1/0 FOR YES/NO) ? ",)
      READ(5,*) IQUAD
      IF(IQUAD.NE.1.AND.IQUAD.NE.0) GOTO 10
      IF(IQUAD.NE.1) GOTO 800
C
C     THE QUADRATIC APPROXIMATION OPTION HAS BEEN SELECTED
C     THE QUADRATIC MODEL IS TO BE ESTABLISHED
C
      NSPACE=INTOL
      DO 100 I=1,INTOL
      KXMOD(I)=KXTOL(I)
      KYMOD(I)=KYTOL(I)
      X(I)=DATX(KXTOL(I),KYTOL(I))
      H(I)=(PERTOL(1,I)+PERTOL(2,I))/2.0
      IF(IRATOL(I).EQ.0) H(I)=H(I)/100.0*X(I)
100   CONTINUE
      DO 120 I=1,NOPT
      DO 110 J=1,INTOL
110   IF(KX(I).EQ.KXTOL(J).AND.KY(I).EQ.KYTOL(J)) GOTO 120
      NSPACE=NSPACE+1
      KXMOD(NSPACE)=KX(I)
      KYMOD(NSPACE)=KY(I)
      X(NSPACE)=DATX(KX(I),KY(I))
      H(NSPACE)=0.10*X(NSPACE)
120   CONTINUE
60    CONTINUE
      N=NSPACE
```

```
      NOBP=(N+1)*(N+2)/2
      NREM=N*(N-1)
      IF(NREM.LT.1) GOTO 140
      CALL RANSET(N*2+1)
      DO 130 MR=1,NREM
      V(MR)=2.0*RANF(0)-1.0
      IF(V(MR).EQ.0.0) V(MR)=0.1
130   CONTINUE
140   NR=1
      MP=1
      IJ=1
      CALL ANALYZ
      KODE=2
      CALL MODEL4(N,NR,NOBP,X,B,V,CX,AF,TAF,H,MP,NREM,IJ)
      KODE=1
      DO 150 I=1,NSPACE
150   DATX(KXMOD(I),KYMOD(I))=CX(I)
C
800   PRINT(6,1200)
1200  FORMAT(" SELECT A VERSION OF THE FTP :"/,
     +5X,"1 - FTP WITH ALL VERTICES OF THE TOLERANCE REGION"/,
     +5X,"2 - FTP WITH SPECIFIED NUMBER OF WORST VERTICES"/,
     +5X,"3 - FTP WITH A THRESHOLD VALUE ON THE OBJECTIVE FUNCTION"/,
     +5X,"4 - FTP WITH WEIGHTED VERTICES PLUS WEIGHTED NOMINAL"/,
     +5X,"5 - FTP WITH MONTE-CARLO DESIGN ")
      READ(5,*)IFTP
900   IF(IFTP.EQ.1)GO TO 910
      IF(IFTP.EQ.2)GO TO 920
      IF(IFTP.EQ.3)GO TO 930
      IF(IFTP.EQ.4)GO TO 940
      IF(IFTP.EQ.5)GO TO 950
      GO TO 800
910   CALL FTPAQ
      GO TO 990
920   CALL FTPBQ
      GO TO 990
930   CALL FTPCQ
      GO TO 990
940   CALL FTPDQ
      GO TO 990
950   CALL FTPEQ
990   IF(MAJOR.EQ.1) NCALL=NCALL+1
      RETURN
      END
```

```
      SUBROUTINE FTPAQ
C
      DIMENSION XMOD(21),ER(1),GE(42,1),H(21,21),Y(21)
      DIMENSION ZT(64,6)
      DIMENSION XNOM(15),XT(15),XX(15),DXDXO(15),X(30),GFTP(15)
      COMMON KODE,IFLAG,XITIME,ITRAP,JTRAP,ISENS,IPRIN,MKODE,EX,W,FR,NT,
     +NF,INDFF,NCO,IXX,MCODE,FF,IMA,IMA1,NOISE,IBANKF,IDCLOC
      COMMON /CO2/SEND(15),GR(15),ERR,NOPT,CONV,INEXP,IGR(15)
      COMMON /CO3/CARAY(30,2,2),DATX(70,6)
      COMMON /ANALZ/IBAND(3),RIND(70),INC1(70),INC2(70),INC3(70),KX(15),
     +KY(15)
      COMMON /TOLER/NVERT,TOL(15),NRANK,NCOUNT,NCALL,TRES,WIT(65)
      COMMON /QUAMOD/IQUAD,IFTP,IAPPR,MAJOR
      COMMON /MIM/B(253,1),IV(6,64)
      COMMON/BLKMRF/INTOL,KXTOL(6),KYTOL(6),IRATOL(6),PERTOL(2,6)
      COMMON /BLKMRG/NSPACE,KXMOD(21),KYMOD(21)
      COMPLEX CARAY
      DATA ZP/1H+/,ZM/1H-/
C
      IF(KODE.NE.1)GO TO 100
      NVERT=2**INTOL
      CALL VERT(IV,INTOL,NVERT)
      DO 10 I=1,INTOL
10    XNOM(I)=DATX(KXTOL(I),KYTOL(I))
      CALL ANALYZ
      IF(IQUAD.NE.1)GO TO 20
      PRINT(6,1000)
1000  FORMAT(" THE OPTION YOU HAVE SELECTED REQUIRES SOME"
     +/" COMMENTS. SINCE THE QUADRATIC MODEL DEVELOPED EARLIER IN"
     +/" THIS RUN IS VALID ONLY FOR A FEW ITERATIONS IT IS RECOMMENDED"
     +/" THAT YOU RUN OPTIMIZATION ONLY FOR 5 ITERATIONS. IN ORDER"
     +/" TO CONTINUE OPTIMIZATION USING QUADRATIC APPROXIMATION QUIT"
     +/" AFTER 5 ITERATIONS, SAVE RESULTS AND RESTART THE PROCESS")
20    GOTO 920
C
C
100   DO 110 I=1,NOPT
110   DATX(KX(I),KY(I))=EXP(SEND(I))
      DO 120 I=1,INTOL
120   XNOM(I)=DATX(KXTOL(I),KYTOL(I))
      EFTP=0.0
      DO 130 I=1,NOPT
130   GFTP(I)=0.0
      IF(IQUAD.EQ.1)GO TO 500
C
      DO 300 J=1,NVERT
      DO 140 K=1,INTOL
      IPM=IV(K,J)
      TOLL=PERTOL(1,K)
      IF(IPM.EQ.-1) TOLL=-PERTOL(2,K)
      IF(IRATOL(K).EQ.0) TOLL=TOLL/100.0*XNOM(K)
      XX(K)=XNOM(K)+TOLL
140   CONTINUE
      DO 150 I=1,INTOL
```

```
150   DATX(KXTOL(I),KYTOL(I))=XX(I)
      DO 160 I=1,NOPT
160   SEND(I)=ALOG(DATX(KX(I),KY(I)))
      CALL ANALYZ
      EFTP=EFTP+ERR
      DO 190 JJ=1,NOPT
C   CHECK IF A VARIABLE HAS ABSOLUTE TOLERANCES AND IF SO, ADJUST THE
C   GRADIENT WHICH IS RETURNED FROM ANALYZ
      GGR=GR(JJ)
      DO 170 KK=1,INTOL
      IFF=0
      IF(KX(JJ).EQ.KXTOL(KK).AND.KY(JJ).EQ.KYTOL(KK).
     +AND.IRATOL(KK).EQ.1) IFF=1
      IF(IFF.EQ.1) GGR=GGR*XNOM(KK)/XX(KK)
      IF(IFF.EQ.1) GOTO 180
170   CONTINUE
180   GFTP(JJ)=GFTP(JJ)+GGR
190   CONTINUE
300   CONTINUE
      GO TO 900
C
C
C   HERE WE USE QUADRATIC APPROXIMATION AND PERFORM QUADRATIC
C   POLYNOMIAL EVALUATION
C
500   DO 510 I=1,NSPACE
510   XMOD(I)=DATX(KXMOD(I),KYMOD(I))
      NOBP=(NSPACE+1)*(NSPACE+2)/2
      N=NSPACE*2
      NN=NSPACE
      NAC=1
      NV=1
      LL=0
      DO 700 J=1,NVERT
      DO 520 I=1,INTOL
      DXDXO(I)=1.0
      IPM=IV(I,J)
      TOLL=PERTOL(1,I)
      IF(IPM.LT.0) TOLL=-PERTOL(2,I)
      IF(IRATOL(I).EQ.0) DXDXO(I)=1.0+TOLL/100.0
      IF(IRATOL(I).EQ.0) TOLL=TOLL/100.0*XNOM(I)
      XMOD(I)=XNOM(I)+TOLL
520   CONTINUE
      CALL QPE(N,NN,NAC,XMOD,ER,GE,B,IV,H,Y,NOBP,NV,LL)
      EFTP=EFTP+ER(1)
      LSPACE=INTOL+1
      DO 550 K=1,NOPT
      DO 540 L=1,INTOL
      IF(KX(K).EQ.KXTOL(L).AND.KY(K).EQ.KYTOL(L)) GOTO 530
      GOTO 540
530   GFTP(K)=GFTP(K)+GE(L,1)*XNOM(L)*DXDXO(L)
      GOTO 550
540   CONTINUE
      GFTP(K)=GFTP(K)+GE(LSPACE,1)*XMOD(LSPACE)
      LSPACE=LSPACE+1
```

```
550   CONTINUE
700   CONTINUE
C
C
900   ERR=EFTP
      DO 910 I=1,NOPT
910   GR(I)=GFTP(I)
C
920   DO 930 I=1,INTOL
930   DATX(KXTOL(I),KYTOL(I))=XNOM(I)
      DO 940 I=1,NOPT
940   SEND(I)=ALOG(DATX(KX(I),KY(I)))
C
      RETURN
      END
```

```
      SUBROUTINE FTPBQ
C
      DIMENSION XMOD(21),ER(1),GE(42,1),H(21,21),Y(21)
      DIMENSION XT(15),XNOM(15),XX(15),GFTP(15),FERR(64),NRF(64)
      DIMENSION ZT(64,6),DXDXO(15)
      COMMON KODE,IFLAG,XITIME,ITRAP,JTRAP,ISENS,IPRIN,MKODE,EX,
     *W,FR,NT,NF,INDFF,NCO,IXX,MCODE,FF,IMA,IMA1,NOISE,IBANKF,IDCLOC
      COMMON/CO2/SEND(15),GR(15),ERR,NOPT,CONV,INEXP,IGR(15)
      COMMON /CO3/ CARAY(30,2,2),DATX(70,6)
      COMMON/ANALZ/IBAND(3),RIND(70),INC1(70),INC2(70),INC3(70),
     *KX(15),KY(15)
      COMMON/TOLER/NVERT,TOL(15),NRANK,NCOUNT,NCALL,TRES,WIT(65)
      COMMON /MIM/B(253,1),IV(6,64)
      COMMON /QUAMOD/IQUAD,IFTP,IAPPR,MAJOR
      COMMON/BLKMRF/INTOL,KXTOL(6),KYTOL(6),IRATOL(6),PERTOL(2,6)
      COMMON /BLKMRG/NSPACE,KXMOD(21),KYMOD(21)
      COMPLEX CARAY
      DATA ZP/1H+/,ZM/1H-/
C
10    NCOUNT=NCOUNT+1
      IF(KODE.NE.1) GOTO 100
      NVERT=2**INTOL
      CALL VERT(IV,INTOL,NVERT)
      DO 20 I=1,INTOL
20    XNOM(I)=DATX(KXTOL(I),KYTOL(I))
      CALL ANALYZ
      PRINT(6,1000)
1000  FORMAT(" WE RECOMMEND THAT THE NUMBER OF WORST VERTICES "
     +/" YOU ARE ABOUT TO ENTER SHOULD BE AT LEAST 2/3 OF THE NUMBER"
     +/" OF ALL POSSIBLE VERTICES OF THE TOLERANCE REGION TO OBTAIN"
     +/" REASONABLE RESULTS."/)
30    PRINT(6,1100) NVERT
1100  FORMAT(" ENTER THE NUMBER OF WORST VERTICES FOR THE FTP ",
     +"(MAX",I3,"):")
      READ(5,*) NRANK
      IF(NRANK.LT.1.OR.NRANK.GT.NVERT) GOTO 30
      IF(IQUAD.EQ.0) GOTO 40
      PRINT(6,1200)
1200  FORMAT(" THE OPTION YOU HAVE SELECTED REQUIRES SOME"
     +/" COMMENTS. SINCE THE QUADRATIC MODEL DEVELOPED EARLIER IN"
     +/" THIS RUN IS VALID ONLY FOR A FEW ITERATIONS IT IS RECOMMENDED"
     +/" THAT YOU RUN OPTIMIZATION ONLY FOR 5 ITERATIONS. IN ORDER"
     +/" TO CONTINUE OPTIMIZATION USING QUADRATIC APPROXIMATION QUIT"
     +/" AFTER 5 ITERATIONS, SAVE RESULTS AND RESTART THE PROCESS")
40    GOTO 920
C
C
100   DO 110 I=1,NOPT
110   DATX(KX(I),KY(I))=EXP(SEND(I))
      DO 120 I=1,INTOL
120   XNOM(I)=DATX(KXTOL(I),KYTOL(I))
      EFTP=0.0
      DO 130 I=1,NOPT
130   GFTP(I)=0.0
```

```
      IF(IQUAD.EQ.1) GOTO 500
C
      DO 300 J=1,NVERT
      IF(NCOUNT.LT.3)GO TO 150
      DO 140 I=1,NRANK
      IF(J.EQ.NRF(I))GO TO 150
140   CONTINUE
      GO TO 300
150   DO 160 K=1,INTOL
      IPM=IV(K,J)
      Z=ZP
      IF(IPM.EQ.-1) Z=ZM
      TOLL=PERTOL(1,K)
      IF(IPM.EQ.-1) TOLL=-PERTOL(2,K)
      IF(IRATOL(K).EQ.0) TOLL=TOLL/100.0*XNOM(K)
      XX(K)=XNOM(K)+TOLL
      ZT(J,K)=Z
160   CONTINUE
      DO 170 I=1,INTOL
170   DATX(KXTOL(I),KYTOL(I))=XX(I)
      DO 180 I=1,NOPT
180   SEND(I)=ALOG(DATX(KX(I),KY(I)))
      CALL ANALYZ
      FERR(J)=ERR
      EFTP=EFTP+ERR
      DO 210 JJ=1,NOPT
C   CHECK IF A VARIABLE HAS ABSOLUTE TOLERANCES AND IF SO, ADJUST THE
C   GRADIENT WHICH IS RETURNED FROM ANALYZ
      GGR=GR(JJ)
      DO 190 KK=1,INTOL
      IFF=0
      IF(KX(JJ).EQ.KXTOL(KK).AND.KY(JJ).EQ.KYTOL(KK).
     +AND.IRATOL(KK).EQ.1) IFF=1
      IF(IFF.EQ.1) GGR=GGR*XNOM(KK)/XX(KK)
      IF(IFF.EQ.1) GOTO 200
190   CONTINUE
200   GFTP(JJ)=GFTP(JJ)+GGR
210   CONTINUE
300   CONTINUE
      GOTO 800
C
C   HERE WE USE QUADRATIC APPROXIMATION AND PERFORM QUADRATIC
C   POLYNOMIAL EVALUATION
C
500   DO 510 I=1,NSPACE
510   XMOD(I)=DATX(KXMOD(I),KYMOD(I))
      NOBP=(NSPACE+1)*(NSPACE+2)/2
      N=NSPACE*2
      NN=NSPACE
      NAC=1
      NV=1
      LL=0
      DO 700 J=1,NVERT
      IF(NCOUNT.LT.3) GOTO 530
      DO 520 I=1,NRANK
```

```
      IF(J.EQ.NRF(I)) GOTO 530
520   CONTINUE
      GO TO 700
530   DO 540 I=1,INTOL
      DXDXO(I)=1.0
      IPM=IV(I,J)
      Z=ZP
      IF(IPM.EQ.-1) Z=ZM
      ZT(J,I)=Z
      TOLL=PERTOL(1,I)
      IF(IPM.LT.0) TOLL=-PERTOL(2,I)
      IF(IRATOL(I).EQ.0) DXDXO(I)=1.0+TOLL/100.0
      IF(IRATOL(I).EQ.0) TOLL=TOLL/100.0*XNOM(I)
      XMOD(I)=XNOM(I)+TOLL
540   CONTINUE
      CALL QPE(N,NN,NAC,XMOD,ER,GE,B,IV,H,Y,NOBP,NV,LL)
      FERR(J)=ER(1)
      EFTP=EFTP+ER(1)
      LSPACE=INTOL+1
      DO 570 K=1,NOPT
      DO 560 L=1,INTOL
      IF(KX(K).EQ.KXTOL(L).AND.KY(K).EQ.KYTOL(L)) GOTO 550
      GOTO 560
550   GFTP(K)=GFTP(K)+GE(L,1)*XNOM(L)*DXDXO(L)
      GOTO 570
560   CONTINUE
      GFTP(K)=GFTP(K)+GE(LSPACE,1)*XMOD(LSPACE)
      LSPACE=LSPACE+1
570   CONTINUE
700   CONTINUE
C
800   IF(NCOUNT.GT.2)GO TO 900
C SORT ERROR FUNCTIONS IN FROM LARGEST TO SMALLEST
      DO 810 I=1,NVERT
810   NRF(I)=I
      LEND=NVERT-1
      DO 840 I1=1,LEND
      LBEG=I1+1
      DO 830 I2=LBEG,NVERT
      IF(FERR(I1).GE.FERR(I2)) GOTO 820
      TMP1=FERR(I1)
      TMP2=NRF(I1)
      FERR(I1)=FERR(I2)
      NRF(I1)=NRF(I2)
      FERR(I2)=TMP1
      NRF(I2)=TMP2
820   CONTINUE
830   CONTINUE
840   CONTINUE
      DO 850 I=1,NVERT
850   WRITE(6,1300)I,NRF(I),FERR(I),(ZT(NRF(I),J),J=1,INTOL)
1300  FORMAT(2X," I=",I3,2X," VERT.NO.=",I3,2X," ERR=",F12.7,2X,6A1)
C
C
900   ERR=EFTP
```

```
      DO 910 I=1,NOPT
910   GR(I)=GFTP(I)
C
920   DO 930 I=1,INTOL
930   DATX(KXTOL(I),KYTOL(I))=XNOM(I)
      DO 940 I=1,NOPT
940   SEND(I)=ALOG(DATX(KX(I),KY(I)))
      IF(NCOUNT.EQ.2)GO TO 10
C
      RETURN
      END
```

```
      SUBROUTINE FTPCQ
C
      DIMENSION XMOD(21),ER(1),GE(42,1),H(21,21),Y(21),FERR(64),NRF(64)
      DIMENSION ZT(64,6)
      DIMENSION XNOM(15),XT(15),XX(15),DXDXO(15),X(30),GFTP(15)
      COMMON KODE,IFLAG,XITIME,ITRAP,JTRAP,ISENS,IPRIN,MKODE,EX,W,FR,NT,
     +NF,INDFF,NCO,IXX,MCODE,FF,IMA,IMA1,NOISE,IBANKF,IDCLOC
      COMMON /CO2/SEND(15),GR(15),ERR,NOPT,CONV,INEXP,IGR(15)
      COMMON /CO3/CARAY(30,2,2),DATX(70,6)
      COMMON /ANALZ/IBAND(3),RIND(70),INC1(70),INC2(70),INC3(70),KX(15),
     +KY(15)
      COMMON /TOLER/NVERT,TOL(15),NRANK,NCOUNT,NCALL,TRES,WIT(65)
      COMMON /QUAMOD/IQUAD,IFTP,IAPPR,MAJOR
      COMMON /MIM/B(253,1),IV(6,64)
      COMMON/LAST/XNOMLST(15),NRFLST(64),FERRLST(64),ZTLST(64,6)
      COMMON/BLKMRF/INTOL,KXTOL(6),KYTOL(6),IRATOL(6),PERTOL(2,6)
      COMMON /BLKMRG/NSPACE,KXMOD(21),KYMOD(21)
      COMPLEX CARAY
      DATA ZP/1H+/,ZM/1H-/
C
C
      NVERT=2**INTOL
      IF(MAJOR.NE.1)GO TO 20
      DO 10 I=1,NVERT
10    WRITE(6,1000)I,NRFLST(I),FERRLST(I),(ZTLST(NRFLST(I),J),J=1,INTOL)
1000  FORMAT(2X," I=",I3,2X," VERT.NO.=",I3,2X," ERR=",F12.7,2X,6A1)
20    IF(KODE.NE.1)GO TO 100
      CALL VERT(IV,INTOL,NVERT)
      DO 30 I=1,INTOL
30    XNOM(I)=DATX(KXTOL(I),KYTOL(I))
      CALL ANALYZ
      PRINT(6,1100)
1100  FORMAT("  ENTER THRESHOLD VALUE FOR THE FUNCTION CHARACTERIZING",
     +/" A GIVEN VERTEX.  FROM OUR COMPUTATIONAL EXPERIENCE WE RECOM",
     +"MEND",/" THAT THE THRESHOLD VALUE BE  5*OBJECTIVE FUNCTION",
     +" VALUE AT THE",/" NOMINAL DESIGN PROBLEM")
      READ(5,*)TRES
      IF(IQUAD.EQ.0) GOTO 40
      PRINT(6,1200)
1200  FORMAT(" THE OPTION YOU HAVE SELECTED REQUIRES SOME"
     +/" COMMENTS. SINCE THE QUADRATIC MODEL DEVELOPED EARLIER IN"
     +/" THIS RUN IS VALID ONLY FOR A FEW ITERATIONS IT IS RECOMMENDED"
     +/" THAT YOU RUN OPTIMIZATION ONLY FOR 5 ITERATIONS. IN ORDER"
     +/" TO CONTINUE OPTIMIZATION USING QUADRATIC APPROXIMATION QUIT"
     +/" AFTER 5 ITERATIONS, SAVE RESULTS AND RESTART THE PROCESS")
40    GOTO 920
C
C
100   DO 110 I=1,NOPT
110   DATX(KX(I),KY(I))=EXP(SEND(I))
      DO 120 I=1,INTOL
120   XNOM(I)=DATX(KXTOL(I),KYTOL(I))
      EFTP=0.0
      DO 130 I=1,NOPT
```

```
130   GFTP(I)=0.0
      IF(IQUAD.EQ.1) GOTO 500
C
      DO 300 J=1,NVERT
      DO 140 K=1,INTOL
      IPM=IV(K,J)
      Z=ZP
      IF(IPM.EQ.-1) Z=ZM
      TOLL=PERTOL(1,K)
      IF(IPM.EQ.-1) TOLL=-PERTOL(2,K)
      IF(IRATOL(K).EQ.0) TOLL=TOLL/100.0*XNOM(K)
      XX(K)=XNOM(K)+TOLL
      ZT(J,K)=Z
140   CONTINUE
      DO 150 I=1,INTOL
150   DATX(KXTOL(I),KYTOL(I))=XX(I)
      DO 160 I=1,NOPT
160   SEND(I)=ALOG(DATX(KX(I),KY(I)))
      CALL ANALYZ
      FERR(J)=ERR
      IF(ERR.LE.TRES) GOTO 300
170   EFTP=EFTP+(ERR-TRES)
      DO 200 JJ=1,NOPT
C   CHECK IF A VARIABLE HAS ABSOLUTE TOLERANCES AND IF SO, ADJUST THE
C   GRADIENT WHICH IS RETURNED FROM ANALYZ
      GGR=GR(JJ)
      DO 180 KK=1,INTOL
      IFF=0
      IF(KX(JJ).EQ.KXTOL(KK).AND.KY(JJ).EQ.KYTOL(KK).
     +AND.IRATOL(KK).EQ.1) IFF=1
      IF(IFF.EQ.1) GGR=GGR*XNOM(KK)/XX(KK)
      IF(IFF.EQ.1) GOTO 190
180   CONTINUE
190   GFTP(JJ)=GFTP(JJ)+GGR
200   CONTINUE
300   CONTINUE
      GOTO 800
C
C   HERE WE USE QUADRATIC APPROXIMATION AND PERFORM QUADRATIC
C   POLYNOMIAL EVALUATION
C
500   DO 510 I=1,NSPACE
510   XMOD(I)=DATX(KXMOD(I),KYMOD(I))
      NOBP=(NSPACE+1)*(NSPACE+2)/2
      N=NSPACE*2
      NN=NSPACE
      NAC=1
      NV=1
      LL=0
      DO 700 J=1,NVERT
      DO 520 I=1,INTOL
      DXDXO(I)=1.0
      IPM=IV(I,J)
      Z=ZP
      IF(IPM.EQ.-1) Z=ZM
```

```
      ZT(J,I)=Z
      TOLL=PERTOL(1,I)
      IF(IPM.LT.0) TOLL=-PERTOL(2,I)
      IF(IRATOL(I).EQ.0) DXDXO(I)=1.0+TOLL/100.0
      IF(IRATOL(I).EQ.0) TOLL=TOLL/100.0*XNOM(I)
      XMOD(I)=XNOM(I)+TOLL
520   CONTINUE
      CALL QPE(N,NN,NAC,XMOD,ER,GE,B,IV,H,Y,NOBP,NV,LL)
      FERR(J)=ER(1)
      IF(ER(1).LE.TRES) GOTO 700
      EFTP=EFTP+(ER(1)-TRES)
      LSPACE=INTOL+1
      DO 550 K=1,NOPT
      DO 540 L=1,INTOL
      IF(KX(K).EQ.KXTOL(L).AND.KY(K).EQ.KYTOL(L)) GOTO 530
      GOTO 540
530   GFTP(K)=GFTP(K)+GE(L,1)*XNOM(L)*DXDXO(L)
      GOTO 550
540   CONTINUE
      GFTP(K)=GFTP(K)+GE(LSPACE,1)*XMOD(LSPACE)
      LSPACE=LSPACE+1
550   CONTINUE
700   CONTINUE
C
C SORT ERROR FUNCTIONS IN FROM LARGEST TO SMALLEST
800   LEND=NVERT-1
      DO 810 I=1,NVERT
810   NRF(I)=I
      DO 840 I1=1,LEND
      LBEG=I1+1
      DO 830 I2=LBEG,NVERT
      IF(FERR(I1).GE.FERR(I2)) GOTO 820
      TMP1=FERR(I1)
      TMP2=NRF(I1)
      FERR(I1)=FERR(I2)
      NRF(I1)=NRF(I2)
      FERR(I2)=TMP1
      NRF(I2)=TMP2
820   CONTINUE
830   CONTINUE
840   CONTINUE
      DO 850 I=1,INTOL
850   XNOMLST(I)=XNOM(I)
      DO 870 J=1,NVERT
      DO 860 I=1,INTOL
860   ZTLST(J,I)=ZT(J,I)
      NRFLST(J)=NRF(J)
      FERRLST(J)=FERR(J)
870   CONTINUE
C
C
900   ERR=EFTP
      DO 910 I=1,NOPT
910   GR(I)=GFTP(I)
C
```

```
920   DO 930 I=1,INTOL
930   DATX(KXTOL(I),KYTOL(I))=XNOM(I)
      DO 940 I=1,NOPT
940   SEND(I)=ALOG(DATX(KX(I),KY(I)))
C
      RETURN
      END
```

```
      SUBROUTINE FTPDQ
C
      DIMENSION XMOD(21),ER(1),GE(42,1),H(21,21),Y(21),ZT(64,6)
      DIMENSION XNOM(15),XT(15),XX(15),DXDXO(15),X(30),GFTP(15)
      COMMON KODE,IFLAG,XITIME,ITRAP,JTRAP,ISENS,IPRIN,MKODE,EX,W,FR,NT,
     +NF,INDFF,NCO,IXX,MCODE,FF,IMA,IMA1,NOISE,IBANKF,IDCLOC
      COMMON /CO2/SEND(15),GR(15),ERR,NOPT,CONV,INEXP,IGR(15)
      COMMON /CO3/CARAY(30,2,2),DATX(70,6)
      COMMON /ANALZ/IBAND(3),RIND(70),INC1(70),INC2(70),INC3(70),KX(15),
     +KY(15)
      COMMON /TOLER/NVERT,TOL(15),NRANK,NCOUNT,NCALL,TRES,WIT(65)
      COMMON /QUAMOD/IQUAD,IFTP,IAPPR,MAJOR
      COMMON /MIM/B(253,1),IV(6,64)
      COMMON/BLKMRF/INTOL,KXTOL(6),KYTOL(6),IRATOL(6),PERTOL(2,6)
      COMMON /BLKMRG/NSPACE,KXMOD(21),KYMOD(21)
      COMPLEX CARAY
C
      IF(KODE.NE.1)GO TO 100
      NVERT=2**INTOL
      CALL VERT(IV,INTOL,NVERT)
      CALL ANALYZ
      DO 10 I=1,NVERT
10    WIT(I)=1.0
      WIT(NVERT+1)=FLOAT(NVERT)
      DO 20 I=1,INTOL
20    XNOM(I)=DATX(KXTOL(I),KYTOL(I))
      IF(IQUAD.NE.1)GO TO 30
      PRINT(6,1000)
1000  FORMAT(" THE OPTION YOU HAVE SELECTED REQUIRES SOME"
     +/" COMMENTS. SINCE THE QUADRATIC MODEL DEVELOPED EARLIER IN"
     +/" THIS RUN IS VALID ONLY FOR A FEW ITERATIONS IT IS RECOMMENDED"
     +/" THAT YOU RUN OPTIMIZATION ONLY FOR 5 ITERATIONS. IN ORDER"
     +/" TO CONTINUE OPTIMIZATION USING QUADRATIC APPROXIMATION QUIT"
     +/" AFTER 5 ITERATIONS, SAVE RESULTS AND RESTART THE PROCESS")
30    GOTO 920
C
C
100   DO 110 I=1,NOPT
110   DATX(KX(I),KY(I))=EXP(SEND(I))
      DO 120 I=1,INTOL
120   XNOM(I)=DATX(KXTOL(I),KYTOL(I))
      EFTP=0.0
      DO 130 I=1,NOPT
130   GFTP(I)=0.0
      IF(IQUAD.EQ.1) GOTO 500
C
      JLAST=NVERT+1
      DO 300 J=1,JLAST
      DO 140 K=1,INTOL
      XX(K)=XNOM(K)
      IF(J.EQ.JLAST) GOTO 140
      IPM=IV(K,J)
      TOLL=PERTOL(1,K)
      IF(IPM.EQ.-1) TOLL=-PERTOL(2,K)
```

```
      IF(IRATOL(K).EQ.0) TOLL=TOLL/100.0*XNOM(K)
      XX(K)=XNOM(K)+TOLL
140   CONTINUE
      DO 150 I=1,INTOL
150   DATX(KXTOL(I),KYTOL(I))=XX(I)
      DO 160 I=1,NOPT
160   SEND(I)=ALOG(DATX(KX(I),KY(I)))
      CALL ANALYZ
      EFTP=EFTP+WIT(J)*ERR
      DO 190 JJ=1,NOPT
C     CHECK IF A VARIABLE HAS ABSOLUTE TOLERANCES AND IF SO, ADJUST THE
C     GRADIENT WHICH IS RETURNED FROM ANALYZ
      GGR=GR(JJ)*WIT(J)
      IF(J.EQ.JLAST) GOTO 180
      DO 170 KK=1,INTOL
      IFF=0
      IF(KX(JJ).EQ.KXTOL(KK).AND.KY(JJ).EQ.KYTOL(KK).
     +AND.IRATOL(KK).EQ.1) IFF=1
      IF(IFF.EQ.1) GGR=GGR*XNOM(KK)/XX(KK)
      IF(IFF.EQ.1) GOTO 180
170   CONTINUE
180   GFTP(JJ)=GFTP(JJ)+GGR
190   CONTINUE
300   CONTINUE
      GOTO 900
C
C
C     HERE WE USE QUADRATIC APPROXIMATION AND PERFORM THE QUADRATIC
C     POLYNOMIAL EVALUATION
C
500   DO 510 I=1,NSPACE
510   XMOD(I)=DATX(KXMOD(I),KYMOD(I))
      NOBP=(NSPACE+1)*(NSPACE+2)/2
      N=NSPACE*2
      NN=NSPACE
      NAC=1
      NV=1
      LL=0
      JLAST=NVERT+1
      DO 700 J=1,JLAST
      DO 520 I=1,INTOL
      DXDXO(I)=1.0
      XMOD(I)=XNOM(I)
      IF(J.EQ.JLAST) GOTO 520
      IPM=IV(I,J)
      TOLL=PERTOL(1,I)
      IF(IPM.LT.0) TOLL=-PERTOL(2,I)
      IF(IRATOL(I).EQ.0) DXDXO(I)=1.0+TOLL/100.0
      IF(IRATOL(I).EQ.0) TOLL=TOLL/100.0*XNOM(I)
      XMOD(I)=XNOM(I)+TOLL
520   CONTINUE
      CALL QPE(N,NN,NAC,XMOD,ER,GE,B,IV,H,Y,NOBP,NV,LL)
      EFTP=EFTP+WIT(J)*ER(1)
      LSPACE=INTOL+1
      DO 550 K=1,NOPT
```

```
      DO 540 L=1,INTOL
      IF(KX(K).EQ.KXTOL(L).AND.KY(K).EQ.KYTOL(L)) GOTO 530
      GOTO 540
530   GFTP(K)=GFTP(K)+WIT(J)*GE(L,1)*XNOM(L)*DXDXO(L)
      GOTO 550
540   CONTINUE
      GFTP(K)=GFTP(K)+WIT(J)*GE(LSPACE,1)*XMOD(LSPACE)
      LSPACE=LSPACE+1
550   CONTINUE
700   CONTINUE
C
C
900   ERR=EFTP
      DO 910 I=1,NOPT
910   GR(I)=GFTP(I)
C
920   DO 930 I=1,INTOL
930   DATX(KXTOL(I),KYTOL(I))=XNOM(I)
      DO 940 I=1,NOPT
940   SEND(I)=ALOG(DATX(KX(I),KY(I)))
C
      RETURN
      END
```

```
      SUBROUTINE FTPEQ
C
      DIMENSION XMOD(21),ER(1),GE(42,1),H(21,21),Y(21),ZT(64,6)
      DIMENSION XNOM(15),XT(15),XX(15),DXDXO(15),X(30),GFTP(15)
      COMMON KODE,IFLAG,XITIME,ITRAP,JTRAP,ISENS,IPRIN,MKODE,EX,W,FR,NT,
     +NF,INDFF,NCO,IXX,MCODE,FF,IMA,IMA1,NOISE,IBANKF,IDCLOC
      COMMON /CO2/SEND(15),GR(15),ERR,NOPT,CONV,INEXP,IGR(15)
      COMMON /CO3/CARAY(30,2,2),DATX(70,6)
      COMMON /ANALZ/IBAND(3),RIND(70),INC1(70),INC2(70),INC3(70),KX(15),
     +KY(15)
      COMMON /TOLER/NVERT,TOL(15),NRANK,NCOUNT,NCALL,TRES,WIT(65)
      COMMON /QUAMOD/IQUAD,IFTP,IAPPR,MAJOR
      COMMON /MIM/B(253,1),IV(6,64)
      COMMON /BLKMRE/RMU(6,64),NRAND
      COMMON/BLKMRF/INTOL,KXTOL(6),KYTOL(6),IRATOL(6),PERTOL(2,6)
      COMMON /BLKMRG/NSPACE,KXMOD(21),KYMOD(21)
      COMPLEX CARAY
C
      IF(KODE.NE.1) GOTO 100
      NVERT=2**INTOL
      DO 10 I=1,INTOL
10    XNOM(I)=DATX(KXTOL(I),KYTOL(I))
      CALL ANALYZ
20    PRINT*," ENTER THE NUMBER OF RANDOM POINTS DESIRED (64 MAX) ",
      READ(5,*) NRAND
      IF(NRAND.LT.1.OR.NRAND.GT.64) GOTO 20
C OBTAIN A RMU(J,I) WITH VALUES BETWEEN +1.0 AND -1.0
      DO 30 J=1,INTOL
C SEED THE RANDOM NUMBER GENERATOR
      CALL RANSET(J*2+1)
      DO 30 I=1,NRAND
C EVERY CALL TO FUNCTION RANF() OBTAINS A RANDOM NUMBER BETWEEN
C 0.0 AND 1.0
      RMU(J,I)=2.0*RANF(0)-1.0
30    CONTINUE
      IF(IQUAD.NE.1)GO TO 40
      PRINT(6,1000)
1000  FORMAT(" THE OPTION YOU HAVE SELECTED REQUIRES SOME"
     +/" COMMENTS. SINCE THE QUADRATIC MODEL DEVELOPED EARLIER IN"
     +/" THIS RUN IS VALID ONLY FOR A FEW ITERATIONS IT IS RECOMMENDED"
     +/" THAT YOU RUN OPTIMIZATION ONLY FOR 5 ITERATIONS. IN ORDER"
     +/" TO CONTINUE OPTIMIZATION USING QUADRATIC APPROXIMATION QUIT"
     +/" AFTER 5 ITERATIONS, SAVE RESULTS AND RESTART THE PROCESS")
40    GOTO 920
C
C
100   DO 110 I=1,NOPT
110   DATX(KX(I),KY(I))=EXP(SEND(I))
      DO 120 I=1,INTOL
120   XNOM(I)=DATX(KXTOL(I),KYTOL(I))
      EFTP=0.0
      DO 130 I=1,NOPT
130   GFTP(I)=0.0
      IF(IQUAD.EQ.1) GOTO 500
```

```
C
      DO 300 J=1,NRAND
      DO 140 K=1,INTOL
      XMU=RMU(K,J)
      TOLL=PERTOL(1,K)*XMU
      IF(XMU.LT.0.0) TOLL=PERTOL(2,K)*XMU
      IF(IRATOL(K).EQ.0) TOLL=TOLL/100.0*XNOM(K)
      XX(K)=XNOM(K)+TOLL
140   CONTINUE
      DO 150 I=1,INTOL
150   DATX(KXTOL(I),KYTOL(I))=XX(I)
      DO 160 I=1,NOPT
160   SEND(I)=ALOG(DATX(KX(I),KY(I)))
      CALL ANALYZ
      EFTP=EFTP+ERR
      DO 190 JJ=1,NOPT
C   CHECK IF A VARIABLE HAS ABSOLUTE TOLERANCES AND IF SO, ADJUST THE
C   GRADIENT WHICH IS RETURNED FROM ANALYZ
      GGR=GR(JJ)
      DO 170 KK=1,INTOL
      IFF=0
      IF(KX(JJ).EQ.KXTOL(KK).AND.KY(JJ).EQ.KYTOL(KK).
     +AND.IRATOL(KK).EQ.1) IFF=1
      IF(IFF.EQ.1) GGR=GGR*XNOM(KK)/XX(KK)
      IF(IFF.EQ.1) GOTO 180
170   CONTINUE
180   GFTP(JJ)=GFTP(JJ)+GGR
190   CONTINUE
300   CONTINUE
      GOTO 900
C
C
C   HERE WE USE QUADRATIC APPROXIMATION AND PERFORM THE QUADRATIC
C   POLYNOMIAL EVALUATION
C
500   DO 510 I=1,NSPACE
510   XMOD(I)=DATX(KXMOD(I),KYMOD(I))
      NOBP=(NSPACE+1)*(NSPACE+2)/2
      N=NSPACE*2
      NN=NSPACE
      NAC=1
      NV=1
      LL=0
      DO 700 J=1,NRAND
      DO 520 K=1,INTOL
      DXDXO(K)=1.0
      XMU=RMU(K,J)
      TOLL=PERTOL(1,K)*XMU
      IF(XMU.LT.0.0) TOLL=PERTOL(2,K)*XMU
      IF(IRATOL(K).EQ.0) DXDXO(K)=1.0+TOLL/100.0
      IF(IRATOL(K).EQ.0) TOLL=TOLL/100.0*XNOM(K)
      XMOD(K)=XNOM(K)+TOLL
520   CONTINUE
      CALL QPE(N,NN,NAC,XMOD,ER,GE,B,IV,H,Y,NOBP,NV,LL)
      EFTP=EFTP+ER(1)
```

```
      LSPACE=INTOL+1
      DO 550 K=1,NOPT
      DO 540 L=1,INTOL
      IF(KX(K).EQ.KXTOL(L).AND.KY(K).EQ.KYTOL(L)) GOTO 530
      GOTO 540
530   GFTP(K)=GFTP(K)+GE(L,1)*XNOM(L)*DXDXO(L)
      GOTO 550
540   CONTINUE
      GFTP(K)=GFTP(K)+GE(LSPACE,1)*XMOD(LSPACE)
      LSPACE=LSPACE+1
550   CONTINUE
700   CONTINUE
C
C
900   ERR=EFTP
      DO 910 I=1,NOPT
910   GR(I)=GFTP(I)
C
920   DO 930 I=1,INTOL
930   DATX(KXTOL(I),KYTOL(I))=XNOM(I)
      DO 940 I=1,NOPT
940   SEND(I)=ALOG(DATX(KX(I),KY(I)))
C
      RETURN
      END
```

```
      SUBROUTINE MODEL4(N,NR,NOBP,X,B,V,CX,AF,TAF,H,MP,NREM,IJ)
C
C      THIS SUBROUTINE CALCULATES COEFFICIENTS OF QUADRATIC MODELS
C      OF SURFACES IN OPTIMAL DESIGN
C
C      N - NUMBER OF TOLERANCED PARAMETERS
C
C      NR - NUMBER OF FUNCTIONS TO BE INTERPOLATED
C
C      NOBP - NUMBER OF BASE POINTS=(N+1)*(N+2)/2
C
C      X - ARRAY OF DIMENSION N IN WHICH CENTER OF INTERPOLATION
C          IS GIVEN
C
C      B - ARRAY OF DIMENSIONS NOPB BY NR IN WHICH COEFFICIENTS
C          OF THE INTERPOLATING POLYNOMIALS ARE RETURNED, E.G.,
C          B(5,2) IS THE 5TH COEFFICIENT OF THE 2ND POLYNOMIAL
C
C      NREM = N*(N-1)
C
C      V - ARRAY OF DIMENSIONS NREM IN WHICH THE USER SUPPLIES
C          NUMBERS SUCH THAT ABS(V) IS LESS THAN ONE AND NOT EQUAL
C          TO ZERO. THESE NUMBERS ARE USED TO CHANGE TWO PARAMETERS
C          AT A TIME
C
C      CX - WORKING ARRAY OF DIMENSION N TO STORE THE CENTER OF
C           INTERPOLATION
C
C      AF - ARRAY OF DIMENSION NR USED TO STORE ACTUAL FUNCTION
C           VALUES AT A CERTAIN BASE POINT
C
C      TAF - ARRAY OF DIMENSION NR USED TO STORE 2*AF
C
C      H - ARRAY OF DIMENSION N IN WHICH THE STEPS DEFINING THE
C          SIZE OF THE INTERPOLATION REGION ARE SUPPLIED BY THE USER
C
C      MP - 1 IF QUADRATIC MODEL IS TO BE PRINTED, OTHERWISE SET TO
C           ZERO
C
C      IJ - 1 FOR THE FIRST CALL TO MODEL4, OTHERWISE SET TO ZERO
C
C      SUBROUTINE ACFUNS(X,AF) IS CALLED TO SUPPLY VALUES OF AF(NR)
C      AT POINT X(N)
C
C
      REAL X(N),B(NOBP,NR),V(NREM),CX(N),AF(NR),TAF(NR),H(N)
      COMMON/ANALZ/IBAND(3),RIND(70),INC1(70),INC2(70),INC3(70),KX(15),
     +KY(15)
      COMMON/CO3/CARAY(30,2,2),DATX(70,6)
      COMPLEX CARAY
      COMMON/BLKMRF/INTOL,KXTOL(6),KYTOL(6),IRATOL(6),PERTOL(2,6)
      COMMON /BLKMRG/NSPACE,KXMOD(21),KYMOD(21)
C
      IF(N.GT.1) GOTO 2
      PRINT*," STOP --- MODELLING IN ONE DIMENSION --- BUG "
```

```
      STOP
2     IF(IJ.EQ.0) GO TO 10
      MM=((N+2)*(N+1))/2
      MM1=MM-1
      MMN=MM-N-1
      LK=N-1
10    DO 20 J=1,N
      CX(J)=X(J)
20    CONTINUE
      CALL ACFUNS(X,AF)
      DO 25 J=1,NR
      B(MM,J)=AF(J)
      TAF(J)=AF(J)+AF(J)
25    CONTINUE
      DO 40 I=1,N
      X(I)=CX(I)+H(I)
      CALL ACFUNS(X,AF)
      DO 30 J=1,NR
      B(I,J)=AF(J)
30    CONTINUE
      X(I)=CX(I)-H(I)
      CALL ACFUNS(X,AF)
      TH=H(I)+H(I)
      HH=H(I)*H(I)
      THH=HH+HH
      DO 35 J=1,NR
      B(MMN+I,J)=(B(I,J)-AF(J))/TH
      B(I,J)=(B(I,J)+AF(J)-TAF(J))/THH
35    CONTINUE
      X(I)=CX(I)
40    CONTINUE
      IR=N
      IRR=1
      DO 50 I=1,LK
      I1=I+1
      MI=MMN+I
      DO 50 K=I1,N
      VHI=V(IRR)*H(I)
      VHK=V(IRR+1)*H(K)
      MK=MMN+K
      IR=IR+1
      IRR=IRR+2
      X(I)=CX(I)+VHI
      X(K)=CX(K)+VHK
      DV=VHI*VHK
      CALL ACFUNS(X,AF)
      DO 45 J=1,NR
      B(IR,J)=(AF(J)-B(MM,J)-VHI*(VHI*B(I,J)+B(MI,J))
     +  -VHK*(VHK*B(K,J)+B(MK,J)))/DV
45    CONTINUE
      X(I)=CX(I)
      X(K)=CX(K)
50    CONTINUE
      DO 65 J=1,NR
      DO 55 I=1,N
```

```
      MI=MMN+I
      A=B(I,J)*CX(I)
      B(MM,J)=B(MM,J)+CX(I)*(A-B(MI,J))
      B(MI,J)=B(MI,J)-A-A
55    CONTINUE
      IR=N
      DO 60 I=1,LK
      MI=MMN+I
      I1=I+1
      DO 60 K=I1,N
      MK=MMN+K
      IR=IR+1
      A=CX(I)*B(IR,J)
      B(MM,J)=B(MM,J)+A*CX(K)
      B(MI,J)=B(MI,J)-CX(K)*B(IR,J)
      B(MK,J)=B(MK,J)-A
60    CONTINUE
65    CONTINUE
C
      IF(MP.NE.1) RETURN
      WRITE(6,200)
      DO 70 I=1,N
      WRITE(6,250) H(I)
70    CONTINUE
      DO 150 MA=1,NR
      WRITE(6,270)MA
      WRITE(6,280) (B(I,MA),I=1,MM)
150   CONTINUE
200   FORMAT(//4X,"STEPS AROUND CENTER OF INTERPOLATION")
250   FORMAT(4X,E16.8)
270   FORMAT(/,4X,"ER(",I2,")",/,4X,6("-"))
C 280 FORMAT(4(3X,E16.8),/,4(3X,E16.8),/,2(3X,E16.8)/)
280   FORMAT(4(3X,E16.8))
      RETURN
      END
```

```
      SUBROUTINE QPE(N,NN,NAC,X,ER,GE,B,IV,H,Y,NOBP,NV,LL)
C
C      N - NUMBER OF PARAMETERS + NUMBER OF TOLERANCES
C
C      NN - NUMBER OF NOMINAL VALUES OF PARAMETERS (OR NUMBER
C           OF VARIABLES IN CASE OF A SINGLE POINT)
C
C      NAC - NUMBER OF ACTUAL CONSTRAINTS, E.G., IF THERE ARE FOUR
C            VERTICES FOR TWO CONSTRAINTS NAC=2, 8 ERROR FUNCTIONS
C            RESULT
C
C      X - INPUT VECTOR OF DIMENSION N
C
C      ER - ARRAY OF DIMENSION NAC IF LL=0, OTHERWISE (LL=1)OF
C           DIMENSION NV*NAC
C
C      GE - ARRAY OF DIMENSION (N,M) STORING GRADIENTS OF ERROR
C           FUNCTIONS; M IS THE NUMBER OF ERROR FUNCTIONS
C
C      B - ARRAY OF POLYNOMIAL COEFFICIENTS, DIMENSIONED (NOPB,NAC),
C          WHERE NOBP=(NN+1)*(NN+2)/2
C
C      IV - ARRAY OF DIMENSIONS (NN,NV) DEFINING VERTICES; IT IS SUPPLIE
C           BY THE USER; IF EVALUATION AT A SINGLE POINT IS REQUIRED
C           (LL=0) THEN NV=1 AND IV IS NOT SUPPLIED
C
C      H - WORKING ARRAY OF DIMENSIONS (NN,NN)
C
C      Y - WORKING ARRAY OF DIMENSION NN
C
C      NOBP = (NN+1)*(NN+2)/2 USER SUPPLIED
C           NN
C      NV=2  NUMBER OF VERTICES (EQUALS 1 FOR A SINGLE POINT)
C
C      LL - INTEGER SUPPLIED BY THE USER, IF LL=1 EVALUATION AT VERTICES
C           IS REQUIRED, IF LL=0 EVALUATION AT A SINGLE POINT IS REQUIRE
C
      REAL    X(1),ER(1),GE(12,64),B(NOBP,NAC),H(NN,NN),Y(NN)
      INTEGER IV(NN,NV)
C
      MM=NOBP
      MM1=MM-1
      LK=NN-1
      IF(LL.EQ.1)GO TO 10
      DO 5 I=1,NN
      Y(I)=X(I)
5     CONTINUE
      GO TO 25
C
C      STARTING AT THE VERTEX Y(I)=X(I)-X(I+NN)  X(I+NN) IS EPS(I)
C
10    DO 20 I=1,NN
      Y(I)=X(I)-X(I+NN)
20    CONTINUE
25    DO 100 K1=1,NAC
```

```
      K=1+NV*(K1-1)
      IF(LL.NE.1)K=K1
      S=B(MM,K1)
      DO 30 I=1,NN
      T=B(I,K1)*Y(I)
      S=S+T*Y(I)
      GE(I,K)=T+T
      H(I,I)=B(I,K1)+B(I,K1)
30    CONTINUE
      KM=NN
      DO 40 I=1,LK
      LH=I+1
      DO 40 J=LH,NN
      KM=KM+1
      T=B(KM,K1)*Y(I)
      S=S+T*Y(J)
      GE(I,K)=GE(I,K)+B(KM,K1)*Y(J)
      GE(J,K)=GE(J,K)+T
      H(I,J)=B(KM,K1)
      H(J,I)=H(I,J)
40    CONTINUE
      DO 50 I=1,NN
      KM=KM+1
      S=S+B(KM,K1)*Y(I)
      GE(I,K)=GE(I,K)+B(KM,K1)
50    CONTINUE
      ER(K)=S
      IF(LL.NE.1)GO TO 100
C
C      OTHER VERTICES
C
      JU=1
      L=K
      LG=1
      MI=K-1
      DO 90 I=1,NN
      M=MI
      INN=I+NN
      EP=X(INN)
      TE=EP+EP
      DO 70 IR=1,NN
      H(IR,I)=TE*H(IR,I)
70    CONTINUE
      EPH=EP*H(I,I)
      DO 80 J=1,JU
      M=M+1
      L=L+1
      LG=LG+1
      ER(L)=ER(M)+TE*GE(I,M)+EPH
      DO 80 IR=1,NN
      GE(IR,L)=GE(IR,M)+H(IR,I)
80    CONTINUE
      JU=JU+JU
90    CONTINUE
      L=K
```

```
      DO 99 II=1,NV
      DO 95 I=1,NN
      J=I+NN
      GE(J,L)=FLOAT(IV(I,II))*GE(I,L)
95    CONTINUE
      L=L+1
99    CONTINUE
100   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE VERT(IV,NN,NV)
C
C     THIS SUBROUTINE GENERATES THE MU'S ACCORDING TO
C     THE COMMON NUMBERING SCHEME FOR THE VERTICES
C
C     NN - NUMBER OF PARAMETERS (INPUT)
C     NV - NUMBER OF VERTICES (INPUT)
C     IV - ARRAY OF MU'S(OUTPUT)
C
      DIMENSION IV(6,64)
      II=1
      MI=NV
      DO 30 I=1,NN
      IR=0
      IS=-1
      DO 20 J=1,MI
      DO 10 K=1,II
      IR=IR+1
      IV(I,IR)=IS
10    CONTINUE
      IS=-IS
20    CONTINUE
      II=II+II
      MI=MI/2
30    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ACFUNS(X,AF)
C
      DIMENSION X(21),AF(1)
      COMMON KODE,IFLAG,XITIME,ITRAP,JTRAP,ISENS,IPRIN,MKODE,EX,W,FR,NT,
     +NF,INDFF,NCO,IXX,MCODE,FF,IMA,IMA1,NOISE,IBANKF,IDCLOC
      COMMON/CO2/SEND(15),GR(15),ERR,NOPT,CONV,INEXP,IGR(15)
      COMMON/ANALZ/IBAND(3),RIND(70),INC1(70),INC2(70),INC3(70),KX(15),
     +KY(15)
      COMMON/CO3/CARAY(30,2,2),DATX(70,6)
      COMPLEX CARAY
      COMMON /BLKMRG/NSPACE,KXMOD(21),KYMOD(21)
C
      DO 5 I=1,NSPACE
      DATX(KXMOD(I),KYMOD(I))=X(I)
5     CONTINUE
C
      DO 10 I=1,NOPT
      SEND(I)=ALOG(DATX(KX(I),KY(I)))
10    CONTINUE
C
      CALL ANALYZ
C     WRITE(6,2000)KODE,(X(I),I=1,NOPT),ERR
2000  FORMAT(" KODE=",I3,/,2X,6(E13.7,2X)/" ERR=",F16.8)
      AF(1)=ERR
      RETURN
      END
```