

OSA

FET MODELLING USING THE FETSC AND FETSC1 SYSTEMS, PART II: LISTING

OSA-87-FC-3-L

February 23, 1987

Optimization Systems Associates

163 Watson's Lane, Dundas, Ontario, Canada

© 1983 J. W. Bandler, President

FET MODELLING USING THE FETSC AND FETSC1 SYSTEMS, PART II: LISTING

OSA-87-FC-3-L

February 23, 1987

SUBROUTINES OF THE FETSC AND FETSC1 SYSTEMS

All subroutines of the FETSC and FETSC1 systems are listed in full in this report.

Routines for the FETSC system

File name	Routine	Index
FETSC	BRIEF	LISTING-4
	CMPARE	LISTING-5
	FETMDL	LISTING-6
	FETSIM	LISTING-9
	GRAD	LISTING-10
	HEAD	LISTING-11
	HELP	LISTING-12
	OPTIMI	LISTING-13
	PARSER	LISTING-15
	PASSIVE_INTERRUPT	LISTING-17
	PRESENT	LISTING-18
	SAVCKT	LISTING-19
	TAKECHAR	LISTING-20
	WDA	LISTING-21
	WPA	LISTING-22
FETFDF	CSOLU	LISTING-23
	FDF	LISTING-25
	SIMFET	LISTING-27
SCFILE	BLOCK DATA A	LISTING-31
	BLKBLK	LISTING-32
	CHKTYP	LISTING-34
	DATABLK	LISTING-35
	ERMSG	LISTING-36
	FERMSG	LISTING-37
	FREQBLK	LISTING-38
	GETCHAR	LISTING-40
	GETFET	LISTING-41
	GETFRQ	LISTING-44
	GETINT	LISTING-45
	GETREAL	LISTING-46
	GETSYMBL	LISTING-47
	GETUNIT	LISTING-48
	GETVAR	LISTING-49
	GETWRD	LISTING-51
	NODUMMY	LISTING-52
	NOSYMBL	LISTING-53
	OPTBLK	LISTING-54
	OUTBLK	LISTING-57
	PAD	LISTING-58
	RDCKT	LISTING-59
	RDSCF	LISTING-61

Routines for the FETSC1 system

File name	Routine	Index
FETSC1	BRIEF	LISTING-63
	CMPARE	LISTING-64
	FETMDL	LISTING-65
	FETSIM	LISTING-68
	GRAD	LISTING-69
	HEAD	LISTING-70
	HELP	LISTING-71
	OPTIMI	LISTING-72
	PARSER	LISTING-74
	PASSIVE_INTERRUPT	LISTING-76
	PRESENT	LISTING-77
	SAVCKT	LISTING-78
	TAKECHAR	LISTING-79
	WDA	LISTING-80
	WPA	LISTING-81
FETFDF1	CSOLU	LISTING-82
	FDF	LISTING-84
	SIMFET	LISTING-86
SCFILE1	BLOCK DATA A	LISTING-90
	BLKBLK	LISTING-91
	CHKTYP	LISTING-93
	DATABLK	LISTING-94
	ERMSG	LISTING-95
	FERMSG	LISTING-96
	FREQBLK	LISTING-97
	GETCHAR	LISTING-99
	GETFET	LISTING-100
	GETFRQ	LISTING-103
	GETINT	LISTING-104
	GETREAL	LISTING-105
	GETSYMBL	LISTING-106
	GETUNIT	LISTING-107
	GETVAR	LISTING-108
	GETWRD	LISTING-110
	NODUMMY	LISTING-111
	NOSYMBL	LISTING-112
	OPTBLK	LISTING-113
	OUTBLK	LISTING-116
	PAD	LISTING-117
	RDCKT	LISTING-118
	RDSCF	LISTING-120

Routines common to both the FETSC and FETSC1 systems

File name	Routine	Index
L1ABC	ADDCOL	LISTING-122
	BFGS	LISTING-124
	DELCOL	LISTING-125
	HACCUM	LISTING-126
	L1A	LISTING-127
	L1B	LISTING-128
	L1C	LISTING-129
	L1LP	LISTING-133
	L1STA2	LISTING-139
	LINSYS	LISTING-144
	RSOLV	LISTING-146
	UTRNS	LISTING-147
	UTTRNS	LISTING-148
QUASI2	PENALTY	LISTING-149
	QUASI2	LISTING-150

```
c*************************************************************************

c       file name: FETSC.FOR

c       for program FETSC (VAX version).
c       this file contains the main program itself: FETMDL and other
c       related subroutines

c*************************************************************************



        SUBROUTINE BRIEF(FILNAM,CKTFIL,NCKT)

c       display a summary of the input infomation

        CHARACTER*12 FILNAM,CKTFIL(6)
        IF(NCKT.EQ.0)CALL ERMSG(32,I,J,*20)
        KK=NCKT+3
        WRITE(*,10)FILNAM,NCKT,(CKTFIL(K),K=4,KK)
10      FORMAT(5X,'Working file: ',A12/5X,'Number of circuits: ',I2/5X,
     *  'Measurement files: ',3(A12,3X))
20      RETURN
        END
```

```
      SUBROUTINE CMPARE(JOPT,NCKT,PA,PA0,PDIM,CKTNAME)
      PARAMETER (MP=11,MX=40,MF=20)

c     compare the optimal solution with the starting point
c     (command 'COMPARE')    starting point in PA0, solution in PA

      DIMENSION PA(MP,3),PDIM(MP),PP(6),PA0(MP,3)
      CHARACTER CKTNAME(6)*8,PN(MP)*8,FMT1(3)*15,FMT3(3)*22
      CHARACTER C1*8,C2*8,FMT2(3)*19
      SAVE PN,FMT1,FMT2,FMT3
      DATA PN/'RG  (OH)','RD  (OH)','GDS(/OH)','RI  (OH)','RS  (OH)',
     *'LS  (nH)','CGS (pF)','CDG (pF)','CDS (pF)','G  (/OH)','T   (ps)'/
      DATA FMT1/'(13X,A8)       ','(11X,A8,22X,A8)','(7X,3(A8,14X)) '/
      DATA FMT2,FMT3/'(7X,A8,7X,A8)      ','(2(5X,A8,5X,A8,4X))',
     * '(3(2X,A8,2X,A8,2X))','(2(1X,F14.5),2X,A8)   ',
     *'(2(2(3X,F10.4),4X),A8)','(3(2(2X,F8.4),2X),A8) '/
      IF(NCKT.LE.0)CALL ERMSG(32,I,J,*30)
      IF(JOPT.EQ.0)CALL ERMSG(35,I,J,*30)
      C1='starting'
      C2='solution'
      WRITE(*,*)'          FET MODEL PARAMETERS'
      WRITE(*,FMT1(NCKT))(CKTNAME(K),K=1,NCKT)
      WRITE(*,FMT2(NCKT))(C1,C2,K=1,NCKT)
      KK=2*NCKT
      DO 20 I=1,MP
      DO 10 K=1,NCKT
      K2=2*K
      K1=K2-1
      PP(K1)=PA0(I,K)
10    PP(K2)=PA(I,K)
      WRITE(*,FMT3(NCKT))(PP(K),K=1,KK),PN(I)
20    CONTINUE
      WRITE(*,*)
30    RETURN
      END
```

```
      PROGRAM FETMDL

c     main program of FETSC
c     VAX version
c     A FET modelling program developed for Raytheon Company
c     equivalent circuit has 11 parameters

      PARAMETER (MP=11,MX=40,MF=20)
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     DIMENSIONAL LIMITS:     MP  maximum number of circuit parameters
c                             MX  maximum number of variables
c                             MF  maximum number of frequencies
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,A*80,KYWRD*2,STRG*17
      CHARACTER FILNAM*12,OLDFIL*12,FILEC*27
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),INXMDA(3),X(MX)
      DIMENSION PA(MP,3),INPX(MP,3),BND(2,MX),INDX(MX),WGT(8,MF,3)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8),FREQ(MF),INP0(MP)
      DIMENSION PDIM(MP),DAT(8,MF,3),FM(8,MF,3),F1(8,MF,3),INPP(MP)
      LOGICAL THERE
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      COMMON/BLK1/PDIM,DAT,INPP,PA0(MP,3)
      DATA INP0/1,3,4,2,8,9,6,5,7,10,11/

c     make sure the message file is present

      INQUIRE(FILE='FETSC.MSG',EXIST=THERE)
      IF(.NOT.THERE)CALL FERMSG(4,'FETSC.MSG   ',I,*2)

      CALL HEAD
      CALL HELP
      OPEN(1,FILE='FETSC.MSG',STATUS='OLD')
      DO 200 I=1,40
200   READ(1,'(A)')MSG(I)
      CLOSE(1)
      DO 210 I=1,MP
210   INPP(I)=INP0(I)

c     JOPT: have we done optimization?  0/1 = no/yes
c     JERR: have we done simulation?  0/1 = no/yes
c     NCKT: have we read circuit files?  0/(1,2,3)  = no/yes

      JOPT=0
      JERR=0
      NCKT=0

c     loop. read, interpret and act upon a command

100   WRITE(*,'(A$)')' COMD>'
      READ(*,'(A)')A
      CALL PARSER(A,KYCD,FILNAM)
      GOTO(1,1,2,3,4,5,6,7,7,8,9,14,10,11,12,13,100),KYCD
```

```
c       commands 'L1' and 'L2'

1       CALL OPTIMI(JOPT,KYCD,OLDFIL,CKTFIL)
        IF(JOPT.EQ.0)GOTO 100
        JERR=0
        GOTO 6

2       STOP

c       command 'DIR'

3       CALL LIB$SPAWN('DIR')
        WRITE(*,*)
        GOTO 100

4       CALL HELP
        GOTO 100

c       command 'PARAMETER'

5       CALL WPA(NCKT,PA,PDIM,OLDFIL,CKTNAME)
        GOTO 100

c       command 'COMPARE'

6       CALL CMPARE(JOPT,NCKT,PA,PAO,PDIM,CKTNAME)
        GOTO 100

c       commands 'SWEEP' and '%ER'

7       CALL FETSIM(FM,F1,JERR)
        CALL WDA(FM,F1,KYCD)
        GOTO 100

c       command 'SENSITIVITY'

8       CALL GRAD(PDIM,CKTNAME)
        GOTO 100

c       command 'PRESENT'

9       CALL PRESENT
        GOTO 100

c       command 'SAVE'

10      CALL SAVCKT(FILNAM,OLDFIL,NX,X)
        GOTO 100

c       command 'READ'

11      CALL RDCKT(FILNAM,OLDFIL,IERROR)
        JOPT=0
        JERR=0
```

```
        GOTO 100

c       command 'EDIT'

12      STRG(1:3)='ED '
        STRG(4:15)=FILNAM
        STRG(16:17)='  '
        WRITE(*,'(10X,''V A X    E D T    E D I T O R''/)')
        CALL LIB$SPAWN(STRG)
        GOTO 100

c       command 'TYPE'

13      STRG(1:5)='TYPE '
        STRG(6:17)=FILNAM
        CALL LIB$SPAWN(STRG)
        GOTO 100

c       command 'BRIEFING'

14      CALL BRIEF(OLDFIL,CKTFIL,NCKT)
        GOTO 100

        END
```

```
      SUBROUTINE FETSIM(FM,F1,JERR)
      PARAMETER (MP=11,MX=40,MF=20)

c     organize model simulation for all frequency points
c     call SIMFET for each frequency

      DIMENSION INXMDA(3),PA(MP,3),INPX(MP,3),BND(2,MX),WGT(8,MF,3)
      DIMENSION INDX(MX),FREQ(MF),F(8),DF(8,MP),INPP(MP),X(MX)
      DIMENSION PDIM(MP),DAT(8,MF,3),FM(8,MF,3),F1(8,MF,3)
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      COMMON/BLK1/PDIM,DAT,INPP,PA0(MP,3)
      IF(NCKT.LE.0)CALL ERMSG(32,I,J,*100)
      IF(JERR.GT.0)RETURN
      C1=90.0/ASIN(1.0)
      DO 90 I=1,NCKT
      DO 50 J=1,NFR
      CALL SIMFET(FREQ(J),PA(1,I),F,DF,JERR,I)
      DO 40 K=1,4
      K1=2*K-1
      K2=K1+1
      FM(K1,J,I)=SQRT(F(K1)*F(K1)+F(K2)*F(K2))

c     avoid discontinuity in ATAN2

            TEMPA=C1*ATAN2(F(K2),F(K1))
            TEMPB=DAT(K2,J,I)
            IF(ABS(TEMPA-TEMPB).GT.180.0)THEN
                  IF(TEMPA.GT.TEMPB)THEN
                        TEMPA=TEMPA-360.0
                  ELSE
                        TEMPA=TEMPA+360.0
                  ENDIF
            ENDIF
40    FM(K2,J,I)=TEMPA

      DO 50 K=1,8
      T=DAT(K,J,I)
      IF(ABS(T).GT.0.001)THEN
            T=100.0*ABS((FM(K,J,I)/T)-1.0)
            F1(K,J,I)=AMIN1(999.0,T)
      ELSE
            F1(K,J,I)=0.0
      ENDIF
50    CONTINUE
90    CONTINUE
      JERR=1
100   RETURN
      END
```

```
      SUBROUTINE GRAD(PDIM,CKTNAME)
      PARAMETER (MP=11,MX=40,MF=20)

c     organize model sensitivity evaluation and least-squares objective
c     call SIMFET

      CHARACTER PN(MP)*8,FMT1(3)*11,FMT2(3)*19,C*8,CKTNAME(6)*8
      DIMENSION PA(MP,3),X(MX),INPX(MP,3),BND(2,MX),INDX(MX),G(MP,3)
      DIMENSION INXMDA(3),FREQ(MF),WGT(8,MF,3),PDIM(MP),F(8),DF(8,MP)
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      COMMON/BLK2/SNR(8,3),KFR(MF,3),DTM(8,MF,3),KOUNT,KODE
      SAVE PN,FMT1,FMT2
      DATA PN/'RG  (OH)','RD  (OH)','GDS(/OH)','RI  (OH)','RS  (OH)',
     *'LS  (nH)','CGS (pF)','CDG (pF)','CDS (pF)','G  (/OH)','T   (ps)'/
      DATA FMT1/'(12X,A8)   ','(2(12X,A8))','(3(12X,A8))'/
      DATA FMT2/'(5X,F15.6,5X,A8)   ','(2(5X,F15.6),5X,A8)',
     *'(3(5X,F15.6),5X,A8)'/
      IF(NCKT.EQ.0)CALL ERMSG(32,I,J,*100)
      M=0
      JSEN=1
      FOBJ=0.0
      DO 10 I=1,NCKT
      DO 10 IP=1,MP
10    G(IP,I)=0.0
      DO 20 I=1,NCKT
      DO 20 J=1,NFR
      CALL SIMFET(FREQ(J),PA(1,I),F,DF,JSEN,I)
      M=M+KFR(J,I)
      DO 20 K=1,8
      WF=SNR(K,I)*WGT(K,J,I)*(F(K)-DTM(K,J,I))
      FOBJ=FOBJ+WF*WF
      WF=2.0*WF
      DO 20 IP=1,MP
20    G(IP,I)=G(IP,I)+WF*DF(K,IP)
      RM=M
      FOBJ=FOBJ*8.0/RM
      C='gradient'
      DO 30 I=1,3
      DO 30 IP=1,MP
30    G(IP,I)=G(IP,I)*PA(IP,I)*8.0/RM
      WRITE(*,90)FOBJ
      WRITE(*,FMT1(NCKT))(CKTNAME(I),I=1,NCKT)
      WRITE(*,FMT1(NCKT))(C,I=1,NCKT)
      DO 40 IP=1,MP
40    WRITE(*,FMT2(NCKT))(G(IP,I),I=1,NCKT),PN(IP)
      WRITE(*,*)
90    FORMAT(10X,'SENSITIVITY OF THE CURRENT MODEL'/10X,
     *  'least-squares error:',F15.8)
100   RETURN
      END
```

```
      SUBROUTINE HEAD

c     escape sequence for clearing the screen

      PRINT*,CHAR(27)//'[2J'
      PRINT*,CHAR(27)//'[H'

      WRITE(*,1)
1     FORMAT(33X,'FET MODELLING'//25X,'Interactive Command System'/)
      RETURN
      END
```

```
      SUBROUTINE HELP
      WRITE(*,10)
10    FORMAT(10X,'Commands Available')
      WRITE(*,20)
20    FORMAT(5X,'L1',12X,'start modelling using L1 optimization'/5X,
     *'L2',12X,'start modelling using least-squares optimization'/5X,
     *'STOP          stop this program and return to VMS'/5X,
     *'DIR           list the file directory'/5X,
     *'HELP          display the available commands'/5X,
     *'PARAMETER     display the current model parameters'/5X,
     *'COMPARE       compare starting point with optimization solution'/
     *5X,'%ERROR',8X,'display percentage errors in S-parameter matching'
     */5X,'SWEEP         generate S-parameters of the current model'/
     *5X,'SENSITIVITY   sensitivity analysis for the current model'/
     *5X,'SAVE fname    save current data in a CKT file'/5X,
     *'READ fname    read desired data from a file'/5X,
     *'EDIT fn.ext   edit a disk file using full sreen editor'/5X,
     *'TYPE fn.ext   display the content of a disk file'/5X,
     *'PRESENTATION  call the user-defined routine PRESENT'/5X,
     *'BRIEFING      display the current status'/)
      RETURN
      END
```

```
      SUBROUTINE OPTIMI(JOPT,KYCD,FILNAM,CKTFIL)
      PARAMETER (MP=11,MX=40,MF=20)

c     prepare for optimization. call L1C or QUASI2

      CHARACTER*12 FILNAM,CKTFIL(6)
      DIMENSION PA(MP,3),X(MX),INPX(MP,3),BND(2,MX),INDX(MX),PAO(MP,3)
      DIMENSION INXMDA(3),FREQ(MF),WGT(8,MF,3),XX(MX),INPP(MP)
      DIMENSION PDIM(MP),DAT(8,MF,3),FM(8,MF,3),F1(8,MF,3)
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      COMMON/BLK1/PDIM,DAT,INPP,PAO
      COMMON/BLK2/SNR(8,3),KFR(MF,3),DTM(8,MF,3),KOUNT,KODE
      COMMON/L11/MARK,FF1(500),DFF1(500,MX),FF(500),DFF(500,MX)
      COMMON/L12/C(MX),DC(MX,MX),CLOC(MX)

      IF(NCKT.LE.0)CALL ERMSG(32,I,J,*100)

      N=NX
      IF(NX.LE.0)CALL ERMSG(33,I,J,*100)
      JOPT=0
      L=0

c     exponential transformation on the variables
c     lower bounds on the original variables are conveniently handled

      DO 20 I=1,N
      TEMP=X(I)-BND(1,I)
      IF(TEMP.GT.4.5E-5)THEN
              XX(I)=ALOG(TEMP)
      ELSE
              XX(I)=-10.0
      ENDIF
      IF(INDX(I).EQ.2)THEN

c     transform the upper bounds on the original variables

              L=L+1
              DO 10 J=1,N
10            DC(L,J)=0.0
              DC(L,I)=-1.0
              C(L)=ALOG(BND(2,I)-BND(1,I))
      ENDIF
20    CONTINUE

      M=0
      DO 60 I=1,NCKT
      DO 60 J=1,NFR
60    M=M+KFR(J,I)

c     save the starting point

      DO 70 I=1,NCKT
      DO 70 J=1,MP
70    PAO(J,I)=PA(J,I)
```

```
      IF(M.EQ.0)CALL ERMSG(34,I,J,*100)
      JOPT=1
      KOUNT=0
      KODE=KYCD
      DX=0.2
      MAXF=1000
      EPS=0.01
      IF(ETERM.GE.1.E-8)EPS=ETERM
      LEQ=0
      CALL BRIEF(FILNAM,CKTFIL,NCKT)

c     start optimization

      WRITE(*,*)
      WRITE(*,*)'            Start modelling ...'
      WRITE(*,*)'     You can press any key to interrupt optimization'
      WRITE(*,*)
      CALL LIB$INIT_TIMER
      IF(KODE.EQ.1)CALL L1C(N,M,L,LEQ,XX,DX,EPS,MAXF)
      IF(KODE.EQ.2)CALL QUASI2(N,M,L,LEQ,XX,DX,EPS,MAXF)
      CALL LIB$STAT_TIMER(2,ICPU)
      WRITE(*,*)
      IF(MARK.GT.0)THEN
             WRITE(*,*)'      Optimization Completed'
      ELSE
             WRITE(*,*)'      Optimization Interrupted'
      ENDIF
      WRITE(*,'(6X,''after '',I4,'' iterations'',F10.2,'' CPU seconds''
     *          )')MAXF,REAL(ICPU)*0.01

c     reverse the exponential transformation on the variables

      DO 80 I=1,NX
80    X(I)=EXP(XX(I))+BND(1,I)
      DO 90 K=1,NCKT
      DO 90 I=1,MP
      IF(INPX(I,K).GT.0)PA(I,K)=X(INPX(I,K))
90    CONTINUE

100   RETURN
      END
```

```
      SUBROUTINE PARSER(A,KYCD,FILNAM)

c     interpret a input command contained in the ASCII string A
c     return a integer code KYCD

      CHARACTER A*80,WRD*80,COMD(16)*2,KYWRD*2,FILNAM*12
      LOGICAL THERE
      SAVE COMD
      DATA COMD/'L1','L2','ST','DI','HE','PA','CO','%E','SW',
     *          'SE','PR','BR','SA','RE','ED','TY'/
      KYCD=17
      J0=1
      CALL GETWRD(A,J0,J1,WRD,LENG)
      IF(LENG.EQ.0)RETURN
      IF(LENG.EQ.1)GOTO 20
      KYWRD=WRD(1:2)
      DO 10 I=1,16
      IF(KYWRD.EQ.COMD(I))GOTO 30
10    CONTINUE
20    CALL ERMSG(31,I,J,*100)

c     is it a file related command?

30    IF(I.GE.13)THEN

c     yes, search the input string for a file name

40          CALL GETWRD(A,J0,J1,WRD,LENG)
            IF(LENG.EQ.0)THEN

c           file name missing, ask for it

                WRITE(*,'(A$)')'     File name: '
                READ(*,'(A)')A
                J0=1
                GOTO 40
            ENDIF
            FILNAM='            '
            FILNAM=WRD(1:LENG)
            IF(I.GE.15)GOTO 60

c           command 'save' or 'read'. put in the omitted extension '.DAT'

            DO 50 K=1,LENG
            IF(WRD(K:K).EQ.'.')GOTO 60
50          CONTINUE
            L1=LENG+1
            IF(L1.GT.9)L1=9
            L2=L1+3
            FILNAM(L1:L2)='.DAT'

60          IF(I.EQ.16.OR.I.EQ.14)THEN

c           command 'read' or 'type'. make sure the file exits
```

```
                    INQUIRE(FILE=FILNAM,EXIST=THERE)
                    IF(.NOT.THERE)CALL FERMSG(1,FILNAM,I,*100)
                ENDIF
            ENDIF

            KYCD=I
100         RETURN
            END
```

```
      subroutine passive_interrupt(cont)

c     check if any key has been pressed

      implicit integer*4 (a-z)
      integer*2 iosb(4)
      integer*2 chan/0/
      integer*4 quadword(2),time
      logical cont
      character*50 line
      common/io_channel/chan
      l = 1
      if (chan .eq. 0) call sys$assign('TT',chan,,)
      status = sys$qiow(,%val(chan),%val(55+128+64),iosb,,,
     1       %val(%loc(line)),%val(1),%val(time),quadword,,)
      l = iosb(2)
      if (l.eq.0) then
            cont = .true.
      else
            cont = .false.
      endif
      end
```

```
      SUBROUTINE PRESENT
      PARAMETER (MP=11,MX=40,MF=20)

c     gateway for user implemented operations. See manual
c     basic information:
c             NCKT:   number of circuits
c             PA:     current values of circuit parameters

      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),INXMDA(3),X(MX)
      DIMENSION PA(MP,3),INPX(MP,3),BND(2,MX),INDX(MX),WGT(8,MF,3)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8),FREQ(MF),PDIM(MP)
      DIMENSION DAT(8,MF,3),INPP(MP),PA0(MP,3)
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      COMMON/BLK1/PDIM,DAT,INPP,PA0

      RETURN
      END
```

```
      SUBROUTINE SAVCKT(FILNAM,OLDFIL,NX,X)
      PARAMETER (MP=11,MX=40,MF=20)

c     save the current infomation to a Super-Compact file

      CHARACTER TT(100)*80,VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,FILNAM*12,OLDFIL*12,AX*8,AT*80
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),X(MX)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8)
      INTEGER*4 STR$TRIM,ICODE
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/SC1/TD(6),IPT(18),XDIM(MX),VDIM(MX),IVTYP(MX)

      IF(NX.LE.0)GOTO 20
      DO 10 I=1,NX
      TEMP=X(I)/XDIM(I)
      IF(TEMP.GT.99.9999.OR.TEMP.LT.0.001)THEN
            WRITE(AX,'(E8.3)')TEMP
      ELSE
            WRITE(AX,'(F8.5)')TEMP
      ENDIF
      J1=LOCX(2,I)
      J2=J1+7
      TT(LOCX(1,I))(J1:J2)=AX
10    CONTINUE
20    OPEN(2,FILE=FILNAM,STATUS='NEW',CARRIAGECONTROL='LIST')
      OLDFIL=FILNAM
      DO 30 I=1,LINET
      AT=TT(I)(1:80)
      ICODE=STR$TRIM(AT,AT,LENG)
      WRITE(2,'(A)')AT(1:LENG)
30    CONTINUE
      CLOSE(2)
      RETURN
      END
```

```
      SUBROUTINE TAKECHAR(CHR)

C     read a single character from keyboard. no echo. no <CR>

      IMPLICIT INTEGER*4 (A-Z)
      INTEGER*2 IOSB(4)
      INTEGER*2 CHAN/0/
      CHARACTER CHR
      COMMON/IO_CHANNEL/CHAN
      L=1
      IF(CHAN.EQ.0)CALL SYS$ASSIGN('TT',CHAN)
      STATUS=SYS$QIOW(,%VAL(CHAN),%VAL(55+64),IOSB,,,
     * %VAL(%LOC(CHR)),%VAL(L),,,,)
      RETURN
      END
```

```
      SUBROUTINE WDA(FM,F1,KYCD)

c     display simulation results (commands 'SWEEP' and '%ER')

      PARAMETER (MP=11,MX=40,MF=20)
      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,TN(8)*4,CHR
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),INXMDA(3),X(MX)
      DIMENSION PA(MP,3),WGT(8,MF,3),INPX(MP,3),BND(2,MX),INDX(MX)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8),FREQ(MF)
      DIMENSION PDIM(MP),DAT(8,MF,3),FM(8,MF,3),F1(8,MF,3),INPP(MP)
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1      TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      COMMON/BLK1/PDIM,DAT,INPP,PA0(MP,3)
      SAVE TN
      DATA TN/'MS11','AS11','MS21','AS21','MS12','AS12','MS22','AS22'/
      IF(NCKT.LE.0)RETURN
      I=1
10    IF(KYCD.EQ.8)THEN
            WRITE(*,110)CKTNAME(I),CKTFIL(INXMDA(I))
            WRITE(*,'(6X,''(GHZ)'',8(4X,A4))')(TN(K),K=1,8)
            DO 20 J=1,NFR
20          WRITE(*,120)FREQ(J),(F1(K,J,I),K=1,8)
      ELSE
            JO=6
30          WRITE(JO,130)CKTNAME(I)
            WRITE(JO,'(1X,''*    (GHZ)'',8(4X,A4))')(TN(K),K=1,8)
            DO 40 J=1,NFR
40          WRITE(JO,140)FREQ(J),(FM(K,J,I),K=1,8)
            IF(CKTFIL(I)(1:1).NE.' ')THEN
                  IF(JO.EQ.2)THEN
                        CLOSE(2)
                  ELSE
                        OPEN(2,FILE=CKTFIL(I),STATUS='NEW')
                        JO=2
                        GOTO 30
                  ENDIF
            ENDIF
      ENDIF
      IF(I.GE.NCKT)GOTO 100
      WRITE(*,'(/5X,''Press <RETURN> to continue ...'')')
      READ(*,'(A)')CHR
      I=I+1
      WRITE(*,*)
      GOTO 10
100   RETURN
110   FORMAT(/19X,'PERCENTAGE ERRORS IN S-PARAMETER MATCHING'/20X,
     *'ABS[ (Model-Measurement)/Measurement ] %'/3X,'Frequency',7X,
     *'model: ',A8,6X,'measurement: ',A12)
120   FORMAT(1X,F9.1,2X,8(F7.2,'%'))
130   FORMAT(1X,'*',15X,'S-PARAMETERS OF MODEL ',A8/1X,'*  Frequency')
140   FORMAT(1X,F9.1,2X,4(F8.4,F8.2))
      END
```

```
      SUBROUTINE WPA(NCKT,PA,PDIM,FILNAM,CKTNAME)
      PARAMETER (MP=11,MX=40,MF=20)

c     display the current values of the parameters (command 'PA')

      DIMENSION PA(MP,3),PDIM(MP),PP(3)
      CHARACTER CKTNAME(6)*8,PN(MP)*8,FMT1(3)*15,FMT2(3)*19
      CHARACTER FILNAM*12
      SAVE PN,FMT1,FMT2
      DATA PN/'RG  (OH)','RD  (OH)','GDS(/OH)','RI  (OH)','RS  (OH)',
     *'LS  (nH)','CGS (pF)','CDG (pF)','CDS (pF)','G  (/OH)','T   (ps)'/
      DATA FMT1/'(16X,A8)       ','(12X,A8,10X,A8)','(5X,3(A8,10X)) '/
      DATA FMT2/'(5X,F18.5,5X,A8)   ','(1X,2F18.5,5X,A8)  ',
     *'(1X,3(F12.5,6X),A8)'/

      IF(NCKT.LE.0)CALL ERMSG(32,I,J,*30)
      WRITE(*,'(6X,''FET MODEL PARAMETERS    file: '',A12)')FILNAM
      WRITE(*,FMT1(NCKT))(CKTNAME(K),K=1,NCKT)
      DO 20 I=1,MP
      DO 10 K=1,NCKT
10    PP(K)=PA(I,K)*PDIM(I)
      WRITE(*,FMT2(NCKT))(PP(K),K=1,NCKT),PN(I)
20    CONTINUE
      WRITE(*,*)
30    RETURN
      END
```

```
c*************************************************************************

c       file name: FETFDF.FOR

c       for program FETSC (VAX version).
c       this group of files are responsible for model simulation

c*************************************************************************



        SUBROUTINE CSOLU(Y,A,B,C,D)

c       solve a complex system of linear equations
c       using LU factorization

        COMPLEX*8 Y(6,6),A(6),B(6),C(6),D(6),Z,T1,T2,T3,T4
        DO 20 I=1,5
        Z=Y(I,I)
        IF(Z.EQ.(0.0,0.0))GO TO 110
        M=I+1
        DO 20 J=M,6
        Y(I,J)=Y(I,J)/Z
        DO 10 K=M,6
10      Y(K,J)=Y(K,J)-Y(K,I)*Y(I,J)
20      CONTINUE

        A(1)=1.0/Y(1,1)
        C(1)=1.0
        DO 40 I=2,6
        T1=0.0
        T2=0.0
        DO 30 K=1,I-1
        T1=T1-Y(I,K)*A(K)
30      T2=T2-Y(K,I)*C(K)
        A(I)=T1/Y(I,I)
40      C(I)=T2

        B(6)=1.0/Y(6,6)
        C(6)=C(6)/Y(6,6)
        D(6)=1.0/Y(6,6)
        DO 60 L=2,6
        I=7-L
        T1=A(I)
        T2=0.0
        T3=C(I)
        T4=0.0
        DO 50 K=I+1,6
        T1=T1-Y(I,K)*A(K)
        T2=T2-Y(I,K)*B(K)
        T3=T3-Y(K,I)*C(K)
50      T4=T4-Y(K,I)*D(K)
        A(I)=T1
        B(I)=T2
        C(I)=T3/Y(I,I)
```

```
60      D(I)=T4/Y(I,I)

        RETURN

110     WRITE(*,*)'FATAL ERROR IN CSOLU: ZERO ON THE DIAGONAL'
        STOP
        END
```

```
      SUBROUTINE FDF(N,M,XX)
      PARAMETER (MP=11,MX=40,MF=20)

c     organize calculation of error functions and their derivatives
c     called by the optimizer

      CHARACTER CHR
      LOGICAL CONT
      DIMENSION XX(N),FS(8),DFS(8,MP),INDX(MX)
      DIMENSION INXMDA(3),PA(MP,3),X(MX),INPX(MP,3),BND(2,MX)
      DIMENSION FREQ(MF),XA(MX),WGT(8,MF,3)
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      COMMON/BLK2/SNR(8,3),KFR(MF,3),DTM(8,MF,3),KOUNT,KODE
      COMMON/L11/MARK,F(500),DF(500,MX),FF(500),DFF(500,MX)

      KOUNT=KOUNT+1
      M1=0
      JSEN=1
      FOBJ=0.0
      CC1=90.0/ASIN(1.0)

c     reserve the exponential transformation on the variables

      DO 10 I=1,N
      XA(I)=EXP(XX(I))
      X(I)=XA(I)+BND(1,I)
      DO 10 J=1,M
10    DF(J,I)=0.0

c     loop for NCKT circuits

      DO 60 K=1,NCKT

c           update the variable parameters

            DO 20 I=1,MP
            IF(INPX(I,K).GT.0)PA(I,K)=X(INPX(I,K))
20          CONTINUE

c     sweep for all frequencies

      DO 60 J=1,NFR
            IF(KFR(J,K).EQ.0.AND.K.NE.JOUT)GOTO 60
            CALL SIMFET(FREQ(J),PA(1,K),FS,DFS,JSEN,K)

c           8 responses (real & imag. parts of 4 s-parameters)

            DO 40 L=1,8
                  IF(WGT(L,J,K).LE.1.E-5)GOTO 40
                  M1=M1+1
                  WT=WGT(L,J,K)*SNR(L,K)
                  F(M1)=WT*(FS(L)-DTM(L,J,K))

c                 compute the objective function
```

```
                        IF(KODE.EQ.1)THEN
                                FOBJ=FOBJ+ABS(F(M1))
                        ELSE
                                FOBJ=FOBJ+F(M1)*F(M1)
                        ENDIF

c                       assemble the gradients

                        DO 30 I=1,MP
                        I1=INPX(I,K)
                        IF(I1.GT.0)DF(M1,I1)=DF(M1,I1)+WT*DFS(L,I)*XA(I1)
30                      CONTINUE
40              CONTINUE
60      CONTINUE

c       check for user interrupt

        CALL PASSIVE_INTERRUPT(CONT)
        IF(.NOT.CONT)THEN

c               interrupt handler

70              WRITE(*,80)
80              FORMAT(///15X,'USER INTERRUPT'/6X,'Press  T  to terminate',
     1           ' optimization'/6X,'Or press <CR> to continue')
                CALL TAKECHAR(CHR)
                IF(CHR.EQ.'T'.OR.CHR.EQ.'t')THEN

c                       'T' has been pressed: terminate optimization

                        MARK=0

                ELSEIF(CHR.NE.CHAR(13))THEN

c                       neither 'T' nor <CR>. wait for another key

                        GOTO 70
                ENDIF
        ENDIF

c       display the count of iteration and the value of objective function

        RM=M
        FOBJ=FOBJ*8.0/RM
        IF (KODE.EQ.1)THEN
           WRITE(*,'(6X,''Iteration'',I3,5X,''L1 error'',F12.7)')KOUNT,FOBJ
        ELSE
           WRITE(*,'(6X,''Iteration'',I3,5X,''L2 error'',F12.7)')KOUNT,FOBJ
        ENDIF

        RETURN
        END
```

```
      SUBROUTINE SIMFET(FREQ,PA,F,DF,JSEN,ICKT)
      PARAMETER (MP=11,MX=40,MF=20)

c     organize model simulation for a single frequency (FREQ)
c     expect 11 parameters in the following order:
c     RG(OH),  RD(OH),  GDS(/OH),  RI(OH),  RS(OH),  LS(nH),  CGS(pF)
c     CDG(pF), CDS(pF), G(/OH),    T(ps)

      DIMENSION PA(MP),PAP(MP),PP(MP),F(8),DF(8,MP),DFS(8,MP)
      COMMON/BLK1/PDIM(MP),DAT(8,MF,3),INPP(MP),PA0(MP,3)
      COMMON/FETCONST/F3DB(3),GGS(3),FDIM
      COMPLEX*8 SMX(4),DSMX(4,MP),PH(6),QH(6),P(6),Q(6)
      COMPLEX*8 Y(6,6),S,Z(9),Z2P(4),ZDELTA,DZ2P(4,MP)
      COMPLEX*8 SMXMU(4),DSMXT(4),C1,C2,C3,C4,C5,C6
      INTEGER INDEX1(7),INDEX2(7)
      SAVE INDEX1,INDEX2
      DATA INDEX1,INDEX2/1,3,5,4,2,2,4,2,4,6,5,5,3,5/

      DO 1 I=1,MP
1     PAP(I)=PA(I)
      DO 2 I=1,4
      IF(I.EQ.3)GOTO 2
      IF(PAP(I).GT.1.E-6)THEN
            PAP(I)=1.0/PAP(I)
      ELSE
            PAP(I)=1.E6
      ENDIF
2     CONTINUE
      DO 3 I=1,MP
      II=INPP(I)
3     PP(II)=PAP(I)

c     assemble the Y matrix

            DO 10 I=1,6
            DO 10 J=1,6
10          Y(J,I)=(0.0,0.0)

            TWOPIF=6.283185*FREQ
            C1=CMPLX(0.0,TWOPIF)
            C2=0.001*C1
            C3=CMPLX(1.0,FREQ/F3DB(ICKT))
            B=TWOPIF*0.001*PP(11)
            C4=CMPLX(COS(B),-SIN(B))/C3

            DO 20 I=1,4
20          Z(I)=PP(I)
            DO 30 I=5,7
30          Z(I)=C2*PP(I)
            Z(8)=1.0/(PP(8)+C1*PP(9))
            Z(9)=PP(10)*C4

            Y(1,1)=Z(1)
            Y(1,2)=-Z(1)
            Y(2,2)=Z(1)+Z(6)+Z(5)+GGS(ICKT)
```

```
                Y(2,3)=-Z(6)-GGS(ICKT)
                Y(2,5)=-Z(5)
                Y(3,3)=Z(2)+Z(6)+GGS(ICKT)
                Y(3,4)=-Z(2)
                Y(4,4)=Z(2)+Z(4)+Z(7)+Z(8)
                Y(4,5)=-Z(4)-Z(7)
                Y(5,5)=Z(5)+Z(4)+Z(7)+Z(3)
                Y(5,6)=-Z(3)
                Y(6,6)=Z(3)

                DO 40 I=1,5
                DO 40 J=I+1,6
40              Y(J,I)=Y(I,J)

                Y(5,2)=Y(5,2)+Z(9)
                Y(5,3)=Y(5,3)-Z(9)
                Y(4,2)=Y(4,2)-Z(9)
                Y(4,3)=Y(4,3)+Z(9)

c       solve the nodal equation for 4 solutions

        CALL CSOLU(Y,P,Q,PH,QH)

c       construct the two-port Z matrix

                Z2P(1)=P(1)/50.0
                Z2P(2)=P(6)/50.0
                Z2P(3)=Q(1)/50.0
                Z2P(4)=Q(6)/50.0
                ZDELTA=(Z2P(1)+1.0)*(Z2P(4)+1.0)-Z2P(2)*Z2P(3)

c       compute the two-port S matrix

                SMX(1)=1.0-2.0*(Z2P(4)+1.0)/ZDELTA
                SMX(2)=2.0*Z2P(2)/ZDELTA
                SMX(3)=2.0*Z2P(3)/ZDELTA
                SMX(4)=1.0-2.0*(Z2P(1)+1.0)/ZDELTA

        IF(JSEN.EQ.0)GOTO 150

c       sensitivity analysis

                DO 60 I=1,7
                J1=INDEX1(I)
                J2=INDEX2(I)
                DZ2P(1,I)=(PH(J1)-PH(J2))*(P(J2)-P(J1))
                DZ2P(2,I)=(QH(J1)-QH(J2))*(P(J2)-P(J1))
                DZ2P(3,I)=(PH(J1)-PH(J2))*(Q(J2)-Q(J1))
60              DZ2P(4,I)=(QH(J1)-QH(J2))*(Q(J2)-Q(J1))

                DO 70 I=5,7
                DO 70 J=1,4
70              DZ2P(J,I)=C2*DZ2P(J,I)

                C5=Z(8)*Z(8)
```

```
                DZ2P(1,8)=C5*PH(4)*P(4)
                DZ2P(2,8)=C5*QH(4)*P(4)
                DZ2P(3,8)=C5*PH(4)*Q(4)
                DZ2P(4,8)=C5*QH(4)*Q(4)

                DO 80 J=1,4
80              DZ2P(J,9)=C1*DZ2P(J,8)

                DZ2P(1,10)=C4*(PH(5)-PH(4))*(P(3)-P(2))
                DZ2P(2,10)=C4*(QH(5)-QH(4))*(P(3)-P(2))
                DZ2P(3,10)=C4*(PH(5)-PH(4))*(Q(3)-Q(2))
                DZ2P(4,10)=C4*(QH(5)-QH(4))*(Q(3)-Q(2))

                C6=-PP(10)*C2
                DO 90 J=1,4
90              DZ2P(J,11)=C6*DZ2P(J,10)

                SMXMU(1)=1.0-SMX(1)
                SMXMU(2)=-SMX(2)
                SMXMU(3)=-SMX(3)
                SMXMU(4)=1.0-SMX(4)

                DO 100 I=1,MP
                DSMXT(1)=SMXMU(1)*DZ2P(1,I)+SMXMU(3)*DZ2P(2,I)
                DSMXT(2)=SMXMU(2)*DZ2P(1,I)+SMXMU(4)*DZ2P(2,I)
                DSMXT(3)=SMXMU(1)*DZ2P(3,I)+SMXMU(3)*DZ2P(4,I)
                DSMXT(4)=SMXMU(2)*DZ2P(3,I)+SMXMU(4)*DZ2P(4,I)

                DSMX(1,I)=(DSMXT(1)*SMXMU(1)+DSMXT(3)*SMXMU(2))/100.0
                DSMX(2,I)=(DSMXT(2)*SMXMU(1)+DSMXT(4)*SMXMU(2))/100.0
                DSMX(3,I)=(DSMXT(1)*SMXMU(3)+DSMXT(3)*SMXMU(4))/100.0
100             DSMX(4,I)=(DSMXT(2)*SMXMU(3)+DSMXT(4)*SMXMU(4))/100.0

                DO 110 K=1,4
                K1=2*K-1
                DO 110 I=1,MP
                DFS(K1,I)=REAL(DSMX(K,I))
110             DFS(K1+1,I)=AIMAG(DSMX(K,I))

                DO 120 I=1,MP
                I1=INPP(I)
                DO 120 J=1,8
120             DF(J,I)=DFS(J,I1)
                DO 140 I=1,4
                IF(I.EQ.3)GOTO 140
                PR=-PAP(I)*PAP(I)
                DO 130 J=1,8
130             DF(J,I)=DF(J,I)*PR
140             CONTINUE

c       separate the real and imaginary parts of the S-parameters

150             DO 160 K=1,4
                K1=2*K-1
                F(K1)=REAL(SMX(K))
```

```
160             F(Kl+1)=AIMAG(SMX(K))

        RETURN
        END
```

```
c*************************************************************************

c       file name: SCFILE.FOR

c       for program FETSC (VAX version).
c       this group of subroutines constitutes a file handler to process
c       a circuit file and the associated measurement data files

c*************************************************************************



        BLOCK DATA A
        DIMENSION TYPDIM(6),IPTYP(18),IVTYP(40),XDIM(40),VDIM(40)
        COMMON/SC1/TYPDIM,IPTYP,XDIM,VDIM,IVTYP
        DATA TYPDIM/1.E-9,1.0,1.0,1.E9,1.E12,1.E12/
        DATA IPTYP/3,5,1,6,3,5,5,5,3,2,2,2,2,5,5,4,4,4/
        END
```

```
      SUBROUTINE BLKBLK(IERROR)
      PARAMETER (MP=11,MX=40,MF=20)

c     decode the 'BLK' block

      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,WRD*80,KYWRD*3,CHARR
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),INXMDA(3),X(MX)
      DIMENSION PA(MP,3),WGT(8,MF,3),INPX(MP,3),BND(2,MX),INDX(MX)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8),FREQ(MF)
      DIMENSION TYPDIM(6),IPTYP(18),IVTYP(MX),XDIM(MX),VDIM(MX)
      COMMON/SC1/TYPDIM,IPTYP,XDIM,VDIM,IVTYP
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      COMMON/FETCONST/F3DB(3),GGS(3),FDIM
      LTOP=LOCBLK(1,1)
      LBOTTOM=LOCBLK(2,1)
      IERROR=1
      IF(LTOP.EQ.0.OR.LBOTTOM.LE.LTOP)CALL ERMSG(1,I,J1,*80)
      DO 10 K=1,6
10    CKTNAME(K)='        '
      NX=0
      NVAR=0
      I=LTOP
      NCKT=0
      NFET=0
20    I=I+1
      IF(I.GT.LBOTTOM)GOTO 50
      J0=1
      CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
      IF(LENG.EQ.0)GOTO 20
      CALL GETSYMBL(WRD,LENG,IP,IFLAG)
      IF(NFET.EQ.1)GOTO 30
      IF(IFLAG.EQ.3)THEN
              LENG=IP-1
              J0=J1+IP
              IF(LENG.EQ.0.OR.LENG.GT.8)CALL ERMSG(6,I,J1,*80)
              NVAR=NVAR+1
              VNAME(NVAR)='        '
              VNAME(NVAR)(1:LENG)=WRD(1:LENG)
              CALL GETVAR(I,J0,NVAR,*80)
              IF(LOCV(3,NVAR).NE.0)THEN
                      NX=NX+1
                      INVAR(NVAR)=NX
                      X(NX)=VAL(1,NVAR)
                      XDIM(NX)=VDIM(NVAR)
                      LOCX(1,NX)=LOCV(1,NVAR)
                      LOCX(2,NX)=LOCV(2,NVAR)
                      INDX(NX)=LOCV(3,NVAR)
                      IF(INDX(NX).EQ.2)THEN
                              BND(1,NX)=VAL(2,NVAR)
                              BND(2,NX)=VAL(3,NVAR)
                      ENDIF
              ENDIF
```

```
              GOTO 20
            ENDIF
            IF(LENG.NE.3.OR.WRD(1:3).NE.'FET')CALL ERMSG(14,I,1,*80)
            CALL GETFET(I,J0,*80)
            NFET=1
            GOTO 20
30          IF(IFLAG.NE.3.OR.IP.NE.LENG.OR.LENG.LT.2)CALL ERMSG(16,I,1,*80)
            LENG=LENG-1
            NND=NCKT*3-2
            CKTNAME(NCKT)=WRD(1:LENG)
            CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
            IF(LENG.NE.4.OR.WRD(1:4).NE.'2POR')CALL ERMSG(18,I,J1,*80)
            CALL GETINT(TT(1,I),J0,J1,ND1,*90)
            IF(ND1.NE.NODE(NND))CALL ERMSG(15,I,J1,*80)
            CALL GETINT(TT(1,I),J0,J1,ND2,*90)
            IF(ND2.NE.NODE(NND+1))CALL ERMSG(15,I,J1,*80)
            NFET=0
            GOTO 20
50          IF(NFET.NE.0)CALL ERMSG(16,I,1,*80)
            IERROR=0
80          RETURN
90          CALL ERMSG(7,I,J1,*80)
            END
```

```
      SUBROUTINE CHKTYP(ITY,I,*)

c     make sure that a parameter and the symbolic variable assigned
c     to it have consistent types (capacitor, inductor, etc.)

      DIMENSION ITY(3)
      IF(ITY(2).EQ.7)GOTO 90
      IF(ITY(1).EQ.0.OR.ITY(1).EQ.7)GOTO 20
      IF(ITY(2).NE.0.AND.ITY(1).NE.ITY(2))GOTO 90
      IF(ITY(3).EQ.0.OR.ITY(3).EQ.7)GOTO 80
      IF(ITY(1).NE.ITY(3))GOTO 90
20    IF(ITY(3).EQ.0.OR.ITY(3).EQ.7)GOTO 80
      IF(ITY(2).EQ.0)GOTO 80
      IF(ITY(2).NE.ITY(3))GOTO 90
80    IF((ITY(1).NE.0).AND.(ITY(1).NE.7))ITY(2)=ITY(1)
      IF((ITY(3).NE.0).AND.(ITY(3).NE.7))ITY(2)=ITY(3)
      RETURN
90    CALL ERMSG(29,I,1,*100)
100   RETURN 1
      END
```

```
      SUBROUTINE DATABLK(IERROR)
      PARAMETER (MP=11,MX=40,MF=20)

c     decode the 'DATA' block

      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,WRD*80
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),INXMDA(3),X(MX)
      DIMENSION PA(MP,3),WGT(8,MF,3),INPX(MP,3),BND(2,MX),INDX(MX)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8),FREQ(MF)
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      DO 10 I=1,6
10    CKTFIL(I)='            '
      IERROR=1
      LTOP=LOCBLK(1,4)
      LBOTTOM=LOCBLK(2,4)
      IF(LTOP.EQ.0.OR.LTOP.GE.LBOTTOM)CALL ERMSG(35,I,J1,*100)
      LTOP=LTOP+1
      NMDA=3
      DO 90 I=LTOP,LBOTTOM
      J0=1
      CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
      IF(LENG.EQ.0)GOTO 90
      CALL GETSYMBL(WRD,LENG,IP,IFLAG)
      IF(IP.EQ.1.OR.IFLAG.NE.3)CALL ERMSG(30,I,J1,*100)
      NMDA=NMDA+1
      IF(NMDA.GT.6)CALL ERMSG(5,I,J1,*100)
      LENG=IP-1
      CKTNAME(NMDA)(1:LENG)=WRD(1:LENG)
      J0=IP+1
      CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
      IF(LENG.EQ.0)CALL ERMSG(17,I,J1,*100)
      CKTFIL(NMDA)(1:LENG)=WRD(1:LENG)
90    CONTINUE
      IERROR=0
100   RETURN
      END
```

```
      SUBROUTINE ERMSG(IER,ILINE,ICOL,*)
      PARAMETER (MP=11,MX=40,MF=20)

c     analyze an error and display the corresponding message

      CHARACTER TT(100)*80,VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,T*79
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8)
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      IF(IER.LE.3)THEN
              WRITE(*,10)MSG(IER)
      ELSEIF(IER.LE.30)THEN
              T=TT(ILINE)(1:79)
              WRITE(*,10)MSG(IER)
              WRITE(*,20)ILINE,ICOL
              WRITE(*,'(1X,A79)')T
      ELSE
              WRITE(*,30)MSG(IER)
      ENDIF
      RETURN 1
10    FORMAT(5X,'Error in circuit file: ',A30)
20    FORMAT(5X,'Error in circuit file at line ',I3,' after column',I3)
30    FORMAT(5X,'Error: ',A30)
      END
```

```
      SUBROUTINE FERMSG(IER,FILNAM,ILINE,*)

c     display file-related error messages

      CHARACTER FILNAM*12
      IF(IER.EQ.1)WRITE(*,10)FILNAM
      IF(IER.EQ.2)WRITE(*,20)FILNAM,ILINE
      IF(IER.EQ.3)WRITE(*,30)FILNAM
      IF(IER.EQ.4)WRITE(*,40)FILNAM
      RETURN 1
10    FORMAT(5X,'File not found: ',A12)
20    FORMAT(5X,'Bad data in the file ',A12,' at line',I3)
30    FORMAT(5X,'Not enough data in file ',A12)
40    FORMAT(5X,'File missing: ',A12)
      END
```

```
      SUBROUTINE FREQBLK(IERROR)
      PARAMETER (MP=11,MX=40,MF=20)

c     decode the 'FREQ' block

      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,WRD*80
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),INXMDA(3),X(MX)
      DIMENSION PA(MP,3),WGT(8,MF,3),INPX(MP,3),BND(2,MX),INDX(MX)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8),FREQ(MF)
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      IERROR=1
      LTOP=LOCBLK(1,2)
      LBOTTOM=LOCBLK(2,2)
      IF(LTOP.EQ.0.OR.LTOP.GE.LBOTTOM)CALL ERMSG(2,I,J1,*100)
      LTOP=LTOP+1
      NFR=0
      EPS=1.E-12
      DO 30 I=LTOP,LBOTTOM
      J0=1
10    J00=J0
      CALL GETWRD(TT(1,I),J00,J1,WRD,LENG)
      IF(LENG.EQ.0)GOTO 30
      II=ICHAR(WRD(1:1))
      IF(II.GE.65.AND.II.LE.122)THEN
              CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
              IF(LENG.EQ.4.AND.WRD(1:4).EQ.'STEP')THEN
                      CALL GETFRQ(I,J0,J1,FLOW,*100)
                      CALL GETFRQ(I,J0,J1,FHIGH,*100)
                      CALL GETFRQ(I,J0,J1,DFR,*100)
                      IF(DFR.LT.EPS)GOTO 120
                      IF(FLOW.LT.EPS)GOTO 120
                      IF(FHIGH.LT.EPS)GOTO 120
                      IF(FHIGH.LT.FLOW)THEN
                              TEMP=FHIGH
                              FHIGH=FLOW
                              FLOW=TEMP
                      ENDIF
                      N1=INT((FHIGH-FLOW)/DFR)+1
                      N2=NFR+N1
                      IF(N2.GT.MF)GOTO 110
                      DO 20 J=1,N1
                      FREQ(NFR+J)=FLOW
20                    FLOW=FLOW+DFR
                      NFR=N2
              ELSE
                      CALL ERMSG(8,I,J1,*100)
              ENDIF
      ELSE
              IF(NFR.GE.MF)GOTO 110
              NFR=NFR+1
              CALL GETFRQ(I,J0,J1,FREQ(NFR),*100)
      ENDIF
```

```
      GOTO 10
30    CONTINUE
      IF(NFR.EQ.0)CALL ERMSG(2,I,J1,*100)
      IERROR=0
100   RETURN
110   CALL ERMSG(19,I,J1,*100)
120   CALL ERMSG(20,I,J1,*100)
      END
```

```
      SUBROUTINE GETCHAR(STRG,ITOP,ICHR,CHARR,*)

C     get a single character from a line starting from the ITOPth position

      CHARACTER STRG(80),CHARR
      ICHR=0
      IF(ITOP.GT.80)RETURN 1
      DO 10 I=ITOP,80
      IF(STRG(I).NE.' ')GOTO 20
10    CONTINUE
      RETURN 1
20    CHARR=STRG(I)
      ICHR=I
      ITOP=I+1
      RETURN
      END
```

```
      SUBROUTINE GETFET(I,J0,*)
      PARAMETER (MP=11,MX=40,MF=20)

c     decode the FET element

      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,CHARR,PARNAM(18)*3,WRD*80
      CHARACTER KYWRD*3,VLABL*8
      DIMENSION IDP(18),ICO(18),VAL(3,MX),LOCV(3,MX),INXMDA(3),X(MX)
      DIMENSION PA(MP,3),WGT(8,MF,3),INPX(MP,3),BND(2,MX),INDX(MX)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8),FREQ(MF)
      DIMENSION TYPDIM(6),IPTYP(18),IVTYP(MX),XDIM(MX),VDIM(MX)
      COMMON/SC1/TYPDIM,IPTYP,XDIM,VDIM,IVTYP
      COMMON/TS1/LOCBLK(2,10),LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      COMMON/FETCONST/F3DB(3),GGS(3),FDIM
      SAVE PARNAM,IDP,ICO,IL
      DATA PARNAM/'G  ','CGS','F  ','T  ','GGS','CDG','CDC','CDS',
     1'GDS','RI ','RG ','RD ','RS ','CGE','CDE','LG ','LD ','LS '/
      DATA IDP/10,7,6,11,6,8,6,9,3,4,1,2,5,6,6,6,6,6/
      DATA ICO/2,2,1,2,-1,2,0,2,2,2,2,2,2,0,0,0,0,2/
      IG=NCKT*3+1
      NCKT=NCKT+1
      IF(NCKT.GT.3)CALL ERMSG(5,I,1,*150)
      CALL GETINT(TT(1,I),J0,J1,NODE(IG),*140)
      CALL GETINT(TT(1,I),J0,J1,NODE(IG+1),*140)
      CALL GETCHAR(TT(1,I),J0,J1,CHARR,*10)
      IF(CHARR.EQ.'0')GOTO 10
      IF(CHARR.EQ.'G')THEN
              J0=J1
              GOTO 10
      ELSE
              CALL ERMSG(21,I,J1,*150)
      ENDIF
10    DO 20 K=1,MP
      PA(K,NCKT)=0.0
20    INPX(K,NCKT)=0
      F3DB(NCKT)=0.0
      GGS(NCKT)=0.0
      K=1
30    CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
      IF(LENG.EQ.0)THEN
              I=I+1
              J0=1
              CALL GETCHAR(TT(1,I),J0,J1,CHARR,*40)
              IF(CHARR.EQ.'+')GOTO 30
40            I=I-1
              GOTO 100
      ENDIF
      CALL NOSYMBL(WRD,LENG,J0,J1)
      IF(LENG.EQ.0.OR.LENG.GT.3)GOTO 60
      KYWRD='   '
      KYWRD(1:LENG)=WRD(1:LENG)
      DO 50 KK=K,18
```

```
      IF(KYWRD.EQ.PARNAM(KK))GOTO 70
50    CONTINUE
60    CALL ERMSG(8,I,J1,*150)
70    K=KK+1
      IPAR=IDP(KK)
      JTYP=IPTYP(KK)
      J00=J0
      CALL GETCHAR(TT(1,I),J00,J1,CHARR,*160)
      IF(CHARR.EQ.'=')J0=J00
      J00=J0
      CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
      IF(LENG.EQ.0)GOTO 160
      IA=ICHAR(WRD(1:1))
      IF(IA.GE.65.AND.IA.LE.122)THEN
              IF(NVAR.EQ.0)GOTO 170
              IF(LENG.GT.8)CALL ERMSG(6,I,J1,*150)
              VLABL='        '
              VLABL(1:LENG)=WRD(1:LENG)
              DO 80 L=1,NVAR
              IF(VLABL.EQ.VNAME(L))GOTO 90
80            CONTINUE
              GOTO 170
      ELSE
              J0=J00
              CALL GETVAR(I,J0,MX,*150)
              IF(LOCV(3,MX).NE.0)THEN
                      NX=NX+1
                      INVAR(MX)=NX
                      X(NX)=VAL(1,MX)
                      XDIM(NX)=VDIM(MX)
                      LOCX(1,NX)=LOCV(1,MX)
                      LOCX(2,NX)=LOCV(2,MX)
                      INDX(NX)=LOCV(3,MX)
                      IF(INDX(NX).EQ.2)THEN
                              BND(1,NX)=VAL(2,MX)
                              BND(2,NX)=VAL(3,MX)
                      ENDIF
              ENDIF
              L=MX
      ENDIF
90    IF(IVTYP(L).EQ.0)THEN
              IVTYP(L)=JTYP
              VAL(1,L)=VAL(1,L)*TYPDIM(JTYP)
              IF(LOCV(3,L).NE.0)THEN
                      JX=INVAR(L)
                      X(JX)=X(JX)*TYPDIM(JTYP)
                      XDIM(JX)=XDIM(JX)*TYPDIM(JTYP)
                      IF(INDX(JX).EQ.2)THEN
                              BND(1,JX)=BND(1,JX)*TYPDIM(JTYP)
                              BND(2,JX)=BND(2,JX)*TYPDIM(JTYP)
                      ENDIF
              ENDIF
      ELSEIF(IVTYP(L).NE.JTYP)THEN
              CALL ERMSG(29,I,J1,*150)
      ENDIF
```

```
      PA(IPAR,NCKT)=VAL(1,L)
      IF(LOCV(3,L).EQ.0)THEN
              INPX(IPAR,NCKT)=0
      ELSE
              INPX(IPAR,NCKT)=INVAR(L)
      ENDIF
      IJ=INPX(6,NCKT)
      ZZ=PA(6,NCKT)
      IF(ICO(KK).EQ.1)THEN
              F3DB(NCKT)=ZZ
              IF(IJ.NE.0)CALL ERMSG(11,I,J1,*150)
      ELSEIF(ICO(KK).EQ.-1)THEN
              GGS(NCKT)=ZZ
              IF(IJ.NE.0)CALL ERMSG(12,I,J1,*150)
      ELSEIF(ICO(KK).EQ.0)THEN
              IF(IJ.NE.0.OR.ZZ.NE.0.0)CALL ERMSG(13,I,J1,*150)
      ENDIF
      IF(K.LE.18)GOTO 30
100   RETURN
140   CALL ERMSG(7,I,J1,*150)
150   RETURN1
160   CALL ERMSG(4,I,J1,*150)
170   CALL ERMSG(10,I,J1,*150)
      END
```

```
      SUBROUTINE GETFRQ(I,J0,J1,TEMP,*)
      PARAMETER (MP=11,MX=40,MF=20)

c     decode a single frequency

      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX), NODE(9),LOCX(2,MX)
      DIMENSION TYPDIM(6),IPTYP(18),IVTYP(MX),XDIM(MX),VDIM(MX)
      COMMON/SC1/TYPDIM,IPTYP,XDIM,VDIM,IVTYP
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR(MX),LOCX,VAL,DIM(8),
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      CALL GETVAR(I,J0,MX,*100)
      IF(IVTYP(MX).GT.1)CALL ERMSG(29,I,J1,*100)
      IF(LOCV(3,MX).NE.0)CALL ERMSG(4,I,J1,*100)
      TEMP=VAL(1,MX)
      RETURN
100   RETURN 1
      END
```

```
      SUBROUTINE GETINT(STRG,ITOP,IP,IVAL,*)

c     try to convert an ASCII string into an integer

      CHARACTER STRG(80),WRD*80,WRD1*20
      CALL GETWRD(STRG,ITOP,IP,WRD,LENG)
      IF(LENG.EQ.0)RETURN 1
      WRD1='                    '
      WRD1(1:LENG)=WRD(1:LENG)
      READ(WRD1,*,ERR=10,END=10)IVAL
      IF(IVAL.GE.0)RETURN
10    RETURN 1
      END
```

```
      SUBROUTINE GETREAL(WRD,LENG,VALUE,*)

C     try to convert an ASCII string into a real value

      CHARACTER WRD*80,WRD1*20
      IF(LENG.EQ.0)GOTO 10
      WRD1='                    '
      WRD1(1:LENG)=WRD(1:LENG)
      READ(WRD1,*,ERR=10,END=10)VALUE
      IF(VALUE.GE.0.0)RETURN
10    RETURN 1
      END
```

```
      SUBROUTINE GETSYMBL(WRD,LENG,IP,IFLAG)

c     detect special symbols in a word

      CHARACTER SYMBOLS(10),WRD*80,A
      SAVE SYMBOLS
      DATA SYMBOLS/'=','?',':','+','-','%','*','/','>','<'/
      IFLAG=0
      DO 10 I=1,LENG
      A=WRD(I:I)
      DO 10 J=1,4
      IF(A.EQ.SYMBOLS(J))GOTO 20
10    CONTINUE
      RETURN
20    IP=I
      IFLAG=J
      RETURN
      END
```

```
      SUBROUTINE GETUNIT(WRD,LENG,IP,ITYPE,VAL)

C     decode the unit attached to a numeral

      CHARACTER WRD*80,KYWRD*4,UNAM(6,7)*4
      DIMENSION DDIM(6,7)
      SAVE UNAM,DDIM
      DATA UNAM/'HZ  ','KHZ ','MHZ ','GHZ ','    ','    ','OH  ','KOH ',
     1'MOH ','    ','    ','    ','/OH ','/KOH','/MOH','    ','    ',
     2'    ','L   ','MH  ','UH  ','NH  ','PH  ','    ','F   ','MF  ',
     3'UF  ','NF  ','PF  ','FF  ','SEC ','MS  ','US  ','NS  ','    ',
     4'    ','%   ','    ','    ','    ','    ','    '/
      DATA DDIM/1.,1.E3,1.E6,1.E9,0.,0.,1.,1.E3,1.E6,0.,0.,0.,1.,1.E-3,
     1  1.E-6,0.,0.,0.,1.,1.E-3,1.E-6,1.E-9,1.E-12,0.,1.,1.E-3,1.E-6,
     2  1.E-9,1.E-12,1.E-15,1.0,1.E-3,1.E-6,1.E-9,8*0.0/
      IP=0
      ITYPE=0
      VAL=1.0
      DO 10 I=LENG,1,-1
      IA=ICHAR(WRD(I:I))
      IF(IA.GE.48.AND.IA.LE.57)GOTO 20
       IF(IA.EQ.4)GOTO 20
10     CONTINUE
       RETURN
20     IF(I.EQ.LENG)RETURN
       IP=I+1
       IG=LENG-IP+1
       IF(IG.GT.4)GOTO 40
       KYWRD='    '
       KYWRD(1:IG)=WRD(IP:LENG)
       DO 30 I=1,7
       DO 30 J=1,6
       IF(KYWRD.EQ.UNAM(J,I))GOTO 50
30     CONTINUE
40     IP=-1
       RETURN
50     ITYPE=I
       VAL=DDIM(J,I)
       RETURN
       END
```

```
      SUBROUTINE GETVAR(I,J0,NVR,*)
      PARAMETER (MP=11,MX=40,MF=20)

c     decode infomation relating to a variable including
c     its value, type (bounded or not) and the bounds on it

      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,WRD*80,A
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),INXMDA(3),ITY(3)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8),VU(3),VL(3)
      DIMENSION TYPDIM(6),IPTYP(18),IVTYP(MX),XDIM(MX),VDIM(MX)
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/SC1/TYPDIM,IPTYP,XDIM,VDIM,IVTYP
      J00=J0
      INDXV=0
      CALL GETCHAR(TT(1,I),J00,J1,A,*70)
      IF(A.EQ.'?')THEN
              INDXV=1
              J0=J00
              JX=J00
      ENDIF
      KOUNT=1
10    CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
      CALL NOSYMBL(WRD,LENG,J0,J1)
      IF(LENG.EQ.0)GOTO 70
      CALL GETUNIT(WRD,LENG,IP,ITY(KOUNT),VU(KOUNT))
      IF(IP.LT.0)CALL ERMSG(25,I,J1,*70)
      IF(IP.GT.0)LENG=IP-1
      CALL GETREAL(WRD,LENG,VL(KOUNT),*70)
      IF(INDXV.EQ.0)GOTO 25
      IF(KOUNT.EQ.1)THEN
              J00=J0
              CALL GETCHAR(TT(1,I),J00,J11,A,*80)
              IF(A.EQ.'?')GOTO 20
              JX=J0
              KOUNT=2
              LOCV(3,NVR)=2
      ELSEIF(KOUNT.EQ.2)THEN
              KOUNT=3
              JX1=J1+LENG
              CALL PAD(TT(1,I),JX,JX1,ISHIFT)
              J0=J0+ISHIFT
      ELSE
              CALL CHKTYP(ITY,I,*90)
              CALL GETCHAR(TT(1,I),J0,J1,A,*80)
              IF(A.NE.'?')GOTO 80
              GOTO 30
      ENDIF
      GOTO 10
20    JX1=J1+LENG
      CALL PAD(TT(1,I),JX,JX1,ISHIFT)
      J0=J00+ISHIFT
25    LOCV(3,NVR)=INDXV
      LOCV(2,NVR)=JX
```

```
      VDIM(NVR)=VU(1)
      IF(ITY(1).EQ.7)THEN
              CALL ERMSG(29,I,1,*90)
      ELSEIF(ITY(1).NE.0)THEN
              VDIM(NVR)=VDIM(NVR)*TYPDIM(ITY(1))
      ENDIF
      IVTYP(NVR)=ITY(1)
      VAL(1,NVR)=VL(1)*VDIM(NVR)
      GOTO 40
30    LOCV(2,NVR)=JX
      VDIM(NVR)=VU(2)
      IF(ITY(2).NE.0)VDIM(NVR)=VDIM(NVR)*TYPDIM(ITY(2))
      VAL(1,NVR)=VL(2)*VDIM(NVR)
      IF(ITY(1).EQ.7)THEN
              VAL(2,NVR)=VAL(1,NVR)*VL(1)*0.01
      ELSE
              IF(ITY(2).NE.0)VU(1)=VU(1)*TYPDIM(ITY(2))
              VAL(2,NVR)=VL(1)*VU(1)
      ENDIF
      IF(ITY(3).EQ.7)THEN
              VAL(3,NVR)=VAL(1,NVR)*VL(3)*0.01
      ELSE
              IF(ITY(2).NE.0)VU(3)=VU(3)*TYPDIM(ITY(2))
              VAL(3,NVR)=VL(3)*VU(3)
      ENDIF
      IF(VAL(3,NVR).LE.VAL(2,NVR))CALL ERMSG(26,I,1,*90)
      IVTYP(NVR)=ITY(2)
40    LOCV(1,NVR)=I
      RETURN
70    CALL ERMSG(4,I,J1,*90)
80    CALL ERMSG(22,I,J1,*90)
90    RETURN1
      END
```

```
      SUBROUTINE GETWRD(STRG,ITOP,IWRD,WRD,LENG)

c     get a word from a line (STRG) starting from the ITOPth position.
c     words are separated by blanks. return the word in WRD
c     its length in LENG

      CHARACTER STRG(80),WRD*80,A
      LENG=0
      IF(ITOP.GT.80)RETURN
      DO 10 I=ITOP,80
      IF(STRG(I).NE.' ')GOTO 20
10    CONTINUE
      RETURN
20    IF(STRG(I).EQ.'*')RETURN
      IWRD=I
      IW=1
      DO 30 I=IWRD,80
      A=STRG(I)
      IF(A.EQ.' ')GOTO 40
      IA=ICHAR(A)
      IF(IA.GE.97.AND.IA.LE.122)A=CHAR(IA-32)
      WRD(IW:IW)=A
      IW=IW+1
30    CONTINUE
40    ITOP=I
      LENG=ITOP-IWRD
      RETURN
      END
```

```
      SUBROUTINE NODUMMY(ILINE)

c     skip over comment lines in the measurement files

      CHARACTER CLINE*80,DUMMY
1     READ(1,'(A)',END=4)CLINE
      J0=1
      ILINE=ILINE+1
2     CALL GETCHAR(CLINE,J0,J1,DUMMY,*1)
      IF(DUMMY.EQ.CHAR(9))GOTO 2
      IF(DUMMY.EQ.'*')GOTO 1
      IDUMMY=ICHAR(DUMMY)-48
      IF(IDUMMY.GE.0.AND.IDUMMY.LE.9)GOTO 3
      IF(DUMMY.NE.'+'.AND.DUMMY.NE.'-'.AND.DUMMY.NE.'.')GOTO 1
3     BACKSPACE 1
      ILINE=ILINE-1
4     RETURN
      END
```

```
      SUBROUTINE NOSYMBL(WRD,LENG,J0,J1)

c     detect special symbols within a word

      CHARACTER WRD*80
      IF(LENG.EQ.0)RETURN
      CALL GETSYMBL(WRD,LENG,IP,IFLAG)
      IF(IFLAG.GT.0)THEN
              J0=J1+IP-1
              LENG=IP-1
      ENDIF
      RETURN
      END
```

```
      SUBROUTINE OPTBLK(IERROR)
      PARAMETER (MP=11,MX=40,MF=20)

c     decode the 'OPT' block

      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8
      CHARACTER WRD*80,PNAME*8,CNAME*7,CMDA(8)*7,CHARR
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),INXMDA(3),X(MX)
      DIMENSION PA(MP,3),WGT(8,MF,3),INPX(MP,3),BND(2,MX),INDX(MX)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8),FREQ(MF),WW(4)
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      DO 10 I=1,3
      DO 10 J=1,8
      DO 10 K=1,NFR
10    WGT(K,J,I)=0.0
      ETERM=0.0
      IERROR=1
      LTOP=LOCBLK(1,3)
      LBOTTOM=LOCBLK(2,3)
      IF(LTOP.EQ.0.OR.LTOP.GE.LBOTTOM)GOTO 210
      I=LTOP
      JCKT=0
20    I=I+1
      IF(I.GT.LBOTTOM)GOTO 150
      J0=1
30    CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
      IF(LENG.EQ.0)GOTO 20
      CHARR=WRD(1:1)
      IF(CHARR.EQ.'F'.AND.JCKT.GT.0)THEN
              IF(LENG.GT.1.AND.WRD(2:2).NE.'=')GOTO 100
              J0=J1+1
              CALL GETCHAR(TT(1,I),J0,J1,CHARR,*240)
              IF(CHARR.NE.'=')GOTO 220
              CALL GETFRQ(I,J0,J1,FLOW,*200)
              FHIGH=FLOW
              J00=J0
              CALL GETCHAR(TT(1,I),J00,J1,CHARR,*20)
              II=ICHAR(CHARR)
              IF(II.GE.43.AND.II.LE.57)CALL GETFRQ(I,J0,J1,FHIGH,*200)
              GOTO 30
      ELSEIF(CHARR.EQ.'S'.AND.JCKT.GT.0)THEN
              IF(LENG.GT.1.AND.WRD(2:2).NE.'=')GOTO 100
              J0=J1+1
              CALL GETCHAR(TT(1,I),J0,J1,CHARR,*240)
              IF(CHARR.NE.'=')GOTO 220
              CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
              IF(LENG.EQ.0.OR.LENG.GT.8)GOTO 240
              PNAME='        '
              PNAME(1:LENG)=WRD(1:LENG)
              DO 40 K1=4,6
              IF(PNAME.EQ.CKTNAME(K1))GOTO 50
40            CONTINUE
```

```
                GOTO 220
50              INXMDA(JCKT)=K1
                J00=J0
                CALL GETCHAR(TT(1,I),J00,J1,CHARR,*80)
                IF(CHARR.EQ.'W')THEN
                        J0=J00
                        CALL GETCHAR(TT(1,I),J00,J1,CHARR,*60)
                        IF(CHARR.EQ.'=')J0=J00
60                      DO 70 J=1,4
                        CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
                        CALL GETREAL(WRD,LENG,WW(J),*230)
70                      CONTINUE
                        TEMP=WW(2)
                        WW(2)=WW(3)
                        WW(3)=TEMP
                ENDIF
80              DO 95 J=1,NFR
                IF(FREQ(J).LT.FLOW.OR.FREQ(J).GT.FHIGH)GOTO 95
                DO 90 L=1,4
                L2=2*L
                L1=L2-1
                WGT(L1,J,JCKT)=WW(L)
90              WGT(L2,J,JCKT)=WW(L)
95              CONTINUE
                GOTO 30
        ELSEIF(LENG.EQ.4.AND.WRD(1:4).EQ.'TERM')THEN
                J0=J1+4
                CALL GETCHAR(TT(1,I),J0,J1,CHARR,*250)
                IF(CHARR.NE.'=')GOTO 220
                CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
                CALL GETREAL(WRD,LENG,ETERM,*230)
                GOTO 30
        ENDIF
100     IF(LENG.GT.8)GOTO 250
        PNAME='        '
        PNAME(1:LENG)=WRD(1:LENG)
        DO 110 K0=1,3
        IF(PNAME.EQ.CKTNAME(K0))GOTO 120
110     CONTINUE
        GOTO 250
120     JCKT=K0
        FHIGH=1.E20
        FLOW=-100.0
        DO 130 L=1,4
130     WW(L)=1.0
        GOTO 30
150     DO 160 I=1,NCKT
        IF(INXMDA(I).GT.0)GOTO 160
                WRITE(*,260)CKTNAME(I)
                GOTO 200
160     CONTINUE
        IERROR=0
200     RETURN
210     CALL ERMSG(3,I,J1,*200)
220     CALL ERMSG(8,I,J1,*200)
```

```
230     CALL ERMSG(4,I,J1,*200)
240     CALL ERMSG(24,I,J1,*200)
250     IF(JCKT.EQ.0)CALL ERMSG(16,I,J1,*200)
        GOTO 220
260     FORMAT(5X,'Error in CKT file:  ',A8,' not used for modelling')
        END
```

```
      SUBROUTINE OUTBLK(IERROR,NCKT)
      PARAMETER (MP=11,MX=40,MF=20)

c     decode the 'OUT' block

      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,WRD*80,WRD1*8,WRD2*3
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),INXMDA(3)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8)
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/BLKPLT/IX(10),IY(10,8,3),YMIN(8,3),YR(8,3),JOUT
      JOUT=1
      IF(NCKT.LE.1)RETURN
      LTOP=LOCBLK(1,5)
      LBOTTOM=LOCBLK(2,5)
      IF(LTOP.EQ.0.OR.LTOP.GE.LBOTTOM)RETURN
      LTOP=LTOP+1
      DO 20 I=LTOP,LBOTTOM
      J0=1
      CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
      IF(LENG.EQ.0)GOTO 20
      IF(LENG.NE.3)GOTO 40
      WRD2=WRD(1:3)
      IF(WRD2.EQ.'PRI')THEN
            CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
            IF(LENG.NE.3)GOTO 20
            WRD2=WRD(1:3)
            IF(WRD2.NE.'PLO')GOTO 20
      ENDIF
      IF(WRD2.EQ.'PLO')THEN
            CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
            IF(LENG.EQ.0.OR.LENG.GT.8)GOTO 50
            WRD1='        '
            WRD1(1:LENG)=WRD(1:LENG)
            DO 10 K=1,NCKT
            IF(WRD1.EQ.CKTNAME(K))GOTO 30
10          CONTINUE
            GOTO 50
      ELSE
            GOTO 40
      ENDIF
20    CONTINUE
      RETURN
30    JOUT=K
      RETURN
40    CALL ERMSG(8,I,J1,*60)
50    CALL ERMSG(16,I,J1,*60)
60    IERROR=1
      RETURN
      END
```

```
      SUBROUTINE PAD(STRG,JX,JX1,ISHIFT)

c     insert blanks between words so that 8 digits
c     can be accommodated

      CHARACTER STRG(80)
      ISHIFT=8-JX1+JX
      IF(ISHIFT.LE.0)RETURN
      I00=80-ISHIFT
      IF(I00.LE.JX1)RETURN
      DO 10 I=I00,JX1,-1
10    STRG(I+ISHIFT)=STRG(I)
      J3=JX+7
      DO 20 I=JX1,J3
20    STRG(I)='0'
      RETURN
      END
```

```
      SUBROUTINE RDCKT(FILNAM,OLDFIL,IERROR)
      PARAMETER (MP=11,MX=40,MF=20)

c     RDCKT is called by the main program.  It is the link between
c     this group of files and the rest of the program.
c     Organizes the process of reading circuit and measurement files.

      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,A*80,KYWRD*2,STRG*17
      CHARACTER FILNAM*12,OLDFIL*12
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),INXMDA(3)
      DIMENSION PA(MP,3),X(MX),INPX(MP,3),BND(2,MX),INDX(MX)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8),FREQ(MF)
      DIMENSION PDIM(MP),DAT(8,MF,3),INPP(MP),WGT(8,MF,3)
      LOGICAL THERE
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      COMMON/BLK1/PDIM,DAT,INPP,PA0(MP,3)
      COMMON/BLK2/SNR(8,3),KFR(MF,3),DTM(8,MF,3),KOUNT,KODE
      COMMON/FETCONST/F3DB(3),GGS(3),FDIM

      OPEN(1,FILE=FILNAM,STATUS='OLD')
      WRITE(*,'(5X,''Reading file: '',A12)')FILNAM

c     read circuit file

      CALL RDSCF(IERROR)
      WRITE(*,*)
      IF(IERROR.NE.1)CALL BLKBLK(IERROR)
      IF(IERROR.NE.1)CALL FREQBLK(IERROR)
      IF(IERROR.NE.1)CALL DATABLK(IERROR)
      IF(IERROR.NE.1)CALL OPTBLK(IERROR)
      IF(IERROR.NE.1)CALL OUTBLK(IERROR,NCKT)
      CLOSE(1)
      IF(IERROR.EQ.1)GOTO 230
      OLDFIL=FILNAM

c     read measurement files

      DO 40 K=1,NCKT
      FILNAM=CKTFIL(INXMDA(K))
      INQUIRE(FILE=FILNAM,EXIST=THERE)
      IF(.NOT.THERE)CALL FERMSG(1,FILNAM,I,*220)
      OPEN(1,FILE=FILNAM,STATUS='OLD')
      EPS=0.001*FREQ(1)
      ILINE=1
      DO 30 J=1,NFR
20    CALL NODUMMY(ILINE)
      READ(1,*,ERR=190,END=200)FRE,(DAT(M,J,K),M=1,8)
      ILINE=ILINE+1
      IF(ABS(FRE-FREQ(J)).GE.EPS)GOTO 20
30    CONTINUE
      CLOSE(1)
40    CONTINUE
```

```
      DO 50 J=1,6
50    DIM(J)=1.0
      DO 60 J=1,MP
60    PDIM(J)=1.0
      FDIM=1.0

c     transform the measurement data to rectangular form

      C1=ASIN(1.0)/90.0
      DO 70 I=1,NCKT
      DO 70 K=1,4
      K1=2*K-1
      K2=K1+1
      DO 70 J=1,NFR
      T1=DAT(K1,J,I)
      T2=DAT(K2,J,I)*C1
      DTM(K1,J,I)=T1*COS(T2)
      DTM(K2,J,I)=T1*SIN(T2)
70    CONTINUE
      DO 100 I=1,NCKT
      DO 100 K=1,8
      SNR(K,I)=1.0
      IF(NFR.EQ.1)GOTO 100
      SMAX=-2000.0
      SMIN=2000.0
      DO 80 J=1,NFR
      IF(DTM(K,J,I).GT.SMAX)SMAX=DTM(K,J,I)
      IF(DTM(K,J,I).LT.SMIN)SMIN=DTM(K,J,I)
80    CONTINUE
      SNR(K,I)=1.0/(SMAX-SMIN)
100   CONTINUE

c     count the error functions

      MM=0
      DO 120 I=1,NCKT
      DO 120 J=1,NFR
      M1=0
      DO 110 K=1,8
      IF(WGT(K,J,I).GT.1.E-5)M1=M1+1
110   CONTINUE
      MM=MM+M1
120   KFR(J,I)=M1

      CALL BRIEF(OLDFIL,CKTFIL,NCKT)
180   RETURN
190   CALL FERMSG(2,FILNAM,ILINE,*210)
200   CALL FERMSG(3,FILNAM,ILINE,*210)
210   CLOSE(1)
220   IERROR=1
230   NCKT=0
      GOTO 180
      END
```

```
      SUBROUTINE RDSCF(IERROR)
      PARAMETER (MP=11,MX=40,MF=20)

c     put the contents of a the circuit file in char array TT(80,100)
c     locate various blocks in the file
c     block i  begins at the LOCBLK(1,i)th line and ends at
c     the LOCBLK(2,i)th line

      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,T*80,A,B,BLKNAME(6)*5,WRD*80
      CHARACTER KYWRD*5
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),INXMDA(3)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8)
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      SAVE BLKNAME
      DATA BLKNAME/'BLK  ','FREQ ','OPT  ','DATA ','OUT  ','     '/
      B=CHAR(9)
      DO 25 I=1,100
      READ(1,'(A)',ERR=30,END=30)T
      DO 10 J=1,80
10    TT(J,I)=' '
      K=1
      DO 20 J=1,80
      IF(T(J:J).EQ.B)THEN

c     a tab

            K=(K/9+1)*9
      ELSE
            IF(K.GT.80)GOTO 25
            TT(K,I)=T(J:J)
            K=K+1
      ENDIF
20    CONTINUE
25    CONTINUE
30    LINET=I-1
      DO 40 I=1,10
40    LOCBLK(1,I)=0
      IPRECED=0
      DO 80 I=1,LINET
      J0=1
      KYWRD='     '
      CALL GETWRD(TT(1,I),J0,J2,WRD,LENG)
      IF(LENG.EQ.0)GOTO 80
      IF(IPRECED.EQ.0)THEN
            IF(LENG.GT.4)GOTO 60
            KYWRD(1:LENG)=WRD(1:LENG)
            DO 50 J=1,5
            IF(KYWRD.EQ.BLKNAME(J))GOTO 70
50          CONTINUE
60          CALL ERMSG(27,I,1,*100)
70          IPRECED=J
            LOCBLK(1,IPRECED)=I
      ELSEIF(LENG.EQ.3.AND.WRD(1:3).EQ.'END')THEN
```

```
                LOCBLK(2,IPRECED)=I-1
                IPRECED=0
        ENDIF
80      CONTINUE
        IF(IPRECED.NE.0)CALL ERMSG(28,LINET,1,*100)
        IERROR=0
        RETURN
100     IERROR=1
        RETURN
        END
```

```
c*************************************************************************

c       file name: FETSC1.FOR

c       for program FETSC1 (VAX version).
c       this file contains the main program itself: FETMDL and other
c       related subroutines

c*************************************************************************



        SUBROUTINE BRIEF(FILNAM,CKTFIL,NCKT)

c       display a summary of the input infomation

        CHARACTER*12 FILNAM,CKTFIL(6)
        IF(NCKT.EQ.0)CALL ERMSG(32,I,J,*20)
        KK=NCKT+3
        WRITE(*,10)FILNAM,NCKT,(CKTFIL(K),K=4,KK)
10      FORMAT(5X,'Working file: ',A12/5X,'Number of circuits: ',I2/5X,
     *  'Measurement files: ',3(A12,3X))
20      RETURN
        END
```

```
      SUBROUTINE CMPARE(JOPT,NCKT,PA,PA0,PDIM,CKTNAME)
      PARAMETER (MP=13,MX=40,MF=20)

c     compare the optimal solution with the starting point
c     (command 'COMPARE')    starting point in PA0, solution in PA

      DIMENSION PA(MP,3),PDIM(MP),PP(6),PA0(MP,3)
      CHARACTER CKTNAME(6)*8,PN(MP)*8,FMT1(3)*15,FMT3(3)*22
      CHARACTER C1*8,C2*8,FMT2(3)*19
      SAVE PN,FMT1,FMT2,FMT3

ccc   DATA PN/'RG  (OH)','RD  (OH)','GDS(/OH)','RI  (OH)','RS  (OH)',
ccc  *'LS  (nH)','CGS (pF)','CDG (pF)','CDS (pF)','G  (/OH)','T   (ps)'/
      DATA PN/'RG  (OH)','RD  (OH)','GDS(/OH)','RI  (OH)','RS  (OH)',
     *'LS  (nH)','CGS (pF)','CDG (pF)','CDS (pF)','G  (/OH)','T   (ps)',
     *'LG  (nH)','LD  (nH)'/

      DATA FMT1/'(13X,A8)       ','(11X,A8,22X,A8)','(7X,3(A8,14X)) '/
      DATA FMT2,FMT3/'(7X,A8,7X,A8)      ','(2(5X,A8,5X,A8,4X))',
     * '(3(2X,A8,2X,A8,2X))','(2(1X,F14.5),2X,A8)   ',
     *'(2(2(3X,F10.4),4X),A8)','(3(2(2X,F8.4),2X),A8) '/
      IF(NCKT.LE.0)CALL ERMSG(32,I,J,*30)
      IF(JOPT.EQ.0)CALL ERMSG(35,I,J,*30)
      C1='starting'
      C2='solution'
      WRITE(*,*)'          FET MODEL PARAMETERS'
      WRITE(*,FMT1(NCKT))(CKTNAME(K),K=1,NCKT)
      WRITE(*,FMT2(NCKT))(C1,C2,K=1,NCKT)
      KK=2*NCKT
      DO 20 I=1,MP
      DO 10 K=1,NCKT
      K2=2*K
      K1=K2-1
      PP(K1)=PA0(I,K)
10    PP(K2)=PA(I,K)
      WRITE(*,FMT3(NCKT))(PP(K),K=1,KK),PN(I)
20    CONTINUE
      WRITE(*,*)
30    RETURN
      END
```

```
      PROGRAM FETMDL

c     main program of FETSC1
c     VAX version
c     A FET modelling program developed for Raytheon Company
c     equivalent circuit has 13 parameters including LS and LD
c     symbol 'ccc' in columns 1-3 indicates lines that have been
c     changed from FETSC.   Also, in each subroutine the statement
c             PARAMETER (MP=11,MX=40,MF=20)
c     has been changed to
c             PARAMETER (MP=13,MX=40,MF=20)

      PARAMETER (MP=13,MX=40,MF=20)
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     DIMENSIONAL LIMITS:     MP  maximum number of circuit parameters
c                             MX  maximum number of variables
c                             MF  maximum number of frequencies
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,A*80,KYWRD*2,STRG*17
      CHARACTER FILNAM*12,OLDFIL*12,FILEC*27
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),INXMDA(3),X(MX)
      DIMENSION PA(MP,3),INPX(MP,3),BND(2,MX),INDX(MX),WGT(8,MF,3)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8),FREQ(MF),INPO(MP)
      DIMENSION PDIM(MP),DAT(8,MF,3),FM(8,MF,3),F1(8,MF,3),INPP(MP)
      LOGICAL THERE
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1      TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      COMMON/BLK1/PDIM,DAT,INPP,PA0(MP,3)
ccc   DATA INPO/1,3,4,2,8,9,6,5,7,10,11/
      DATA INPO/1,3,4,2,8,9,6,5,7,10,11,12,13/

c     make sure the message file is present

      INQUIRE(FILE='FETSC.MSG',EXIST=THERE)
      IF(.NOT.THERE)CALL FERMSG(4,'FETSC.MSG   ',I,*2)

      CALL HEAD
      CALL HELP
      OPEN(1,FILE='FETSC.MSG',STATUS='OLD')
      DO 200 I=1,40
200   READ(1,'(A)')MSG(I)
      CLOSE(1)
      DO 210 I=1,MP
210   INPP(I)=INPO(I)

c     JOPT: have we done optimization?  0/1 = no/yes
c     JERR: have we done simulation?  0/1 = no/yes
c     NCKT: have we read circuit files?  0/(1,2,3)  = no/yes

      JOPT=0
      JERR=0
      NCKT=0
```

```
c       loop. read, interpret and act upon a command

100     WRITE(*,'(A$)')' COMD>'
        READ(*,'(A)')A
        CALL PARSER(A,KYCD,FILNAM)
        GOTO(1,1,2,3,4,5,6,7,7,8,9,14,10,11,12,13,100),KYCD

c       commands 'L1' and 'L2'

1       CALL OPTIMI(JOPT,KYCD,OLDFIL,CKTFIL)
        IF(JOPT.EQ.0)GOTO 100
        JERR=0
        GOTO 6

2       STOP

c       command 'DIR'

3       CALL LIB$SPAWN('DIR')
        WRITE(*,*)
        GOTO 100

4       CALL HELP
        GOTO 100

c       command 'PARAMETER'

5       CALL WPA(NCKT,PA,PDIM,OLDFIL,CKTNAME)
        GOTO 100

c       command 'COMPARE'

6       CALL CMPARE(JOPT,NCKT,PA,PAO,PDIM,CKTNAME)
        GOTO 100

c       commands 'SWEEP' and '%ER'

7       CALL FETSIM(FM,F1,JERR)
        CALL WDA(FM,F1,KYCD)
        GOTO 100

c       command 'SENSITIVITY'

8       CALL GRAD(PDIM,CKTNAME)
        GOTO 100

c       command 'PRESENT'

9       CALL PRESENT
        GOTO 100

c       command 'SAVE'

10      CALL SAVCKT(FILNAM,OLDFIL,NX,X)
        GOTO 100
```

```
c       command 'READ'

11      CALL RDCKT(FILNAM,OLDFIL,IERROR)
        JOPT=0
        JERR=0
        GOTO 100

c       command 'EDIT'

12      STRG(1:3)='ED '
        STRG(4:15)=FILNAM
        STRG(16:17)='  '
        WRITE(*,'(10X,''V A X    E D T    E D I T O R''/)')
        CALL LIB$SPAWN(STRG)
        GOTO 100

c       command 'TYPE'

13      STRG(1:5)='TYPE '
        STRG(6:17)=FILNAM
        CALL LIB$SPAWN(STRG)
        GOTO 100

c       command 'BRIEFING'

14      CALL BRIEF(OLDFIL,CKTFIL,NCKT)
        GOTO 100

        END
```

```
      SUBROUTINE FETSIM(FM,F1,JERR)
      PARAMETER (MP=13,MX=40,MF=20)

c     organize model simulation for all frequency points
c     call SIMFET for each frequency

      DIMENSION INXMDA(3),PA(MP,3),INPX(MP,3),BND(2,MX),WGT(8,MF,3)
      DIMENSION INDX(MX),FREQ(MF),F(8),DF(8,MP),INPP(MP),X(MX)
      DIMENSION PDIM(MP),DAT(8,MF,3),FM(8,MF,3),F1(8,MF,3)
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      COMMON/BLK1/PDIM,DAT,INPP,PA0(MP,3)
      IF(NCKT.LE.0)CALL ERMSG(32,I,J,*100)
      IF(JERR.GT.0)RETURN
      C1=90.0/ASIN(1.0)
      DO 90 I=1,NCKT
      DO 50 J=1,NFR
      CALL SIMFET(FREQ(J),PA(1,I),F,DF,JERR,I)
      DO 40 K=1,4
      K1=2*K-1
      K2=K1+1
      FM(K1,J,I)=SQRT(F(K1)*F(K1)+F(K2)*F(K2))

c     avoid discontinuity in ATAN2

            TEMPA=C1*ATAN2(F(K2),F(K1))
            TEMPB=DAT(K2,J,I)
            IF(ABS(TEMPA-TEMPB).GT.180.0)THEN
                  IF(TEMPA.GT.TEMPB)THEN
                        TEMPA=TEMPA-360.0
                  ELSE
                        TEMPA=TEMPA+360.0
                  ENDIF
            ENDIF
40    FM(K2,J,I)=TEMPA

      DO 50 K=1,8
      T=DAT(K,J,I)
      IF(ABS(T).GT.0.001)THEN
            T=100.0*ABS((FM(K,J,I)/T)-1.0)
            F1(K,J,I)=AMIN1(999.0,T)
      ELSE
            F1(K,J,I)=0.0
      ENDIF
50    CONTINUE
90    CONTINUE
      JERR=1
100   RETURN
      END
```

```
      SUBROUTINE GRAD(PDIM,CKTNAME)
      PARAMETER (MP=13,MX=40,MF=20)

c     organize model sensitivity evaluation and least-squares objective
c     call SIMFET

      CHARACTER PN(MP)*8,FMT1(3)*11,FMT2(3)*19,C*8,CKTNAME(6)*8
      DIMENSION PA(MP,3),X(MX),INPX(MP,3),BND(2,MX),INDX(MX),G(MP,3)
      DIMENSION INXMDA(3),FREQ(MF),WGT(8,MF,3),PDIM(MP),F(8),DF(8,MP)
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      COMMON/BLK2/SNR(8,3),KFR(MF,3),DTM(8,MF,3),KOUNT,KODE
      SAVE PN,FMT1,FMT2

ccc   DATA PN/'RG  (OH)','RD  (OH)','GDS(/OH)','RI  (OH)','RS  (OH)',
ccc  *'LS  (nH)','CGS (pF)','CDG (pF)','CDS (pF)','G  (/OH)','T   (ps)'/
      DATA PN/'RG  (OH)','RD  (OH)','GDS(/OH)','RI  (OH)','RS  (OH)',
     *'LS  (nH)','CGS (pF)','CDG (pF)','CDS (pF)','G  (/OH)','T   (ps)',
     *'LG  (nH)','LD  (nH)'/

      DATA FMT1/'(12X,A8)   ','(2(12X,A8))','(3(12X,A8))'/
      DATA FMT2/'(5X,F15.6,5X,A8)   ','(2(5X,F15.6),5X,A8)',
     *'(3(5X,F15.6),5X,A8)'/
      IF(NCKT.EQ.0)CALL ERMSG(32,I,J,*100)
      M=0
      JSEN=1
      FOBJ=0.0
      DO 10 I=1,NCKT
      DO 10 IP=1,MP
10    G(IP,I)=0.0
      DO 20 I=1,NCKT
      DO 20 J=1,NFR
      CALL SIMFET(FREQ(J),PA(1,I),F,DF,JSEN,I)
      M=M+KFR(J,I)
      DO 20 K=1,8
      WF=SNR(K,I)*WGT(K,J,I)*(F(K)-DTM(K,J,I))
      FOBJ=FOBJ+WF*WF
      WF=2.0*WF
      DO 20 IP=1,MP
20    G(IP,I)=G(IP,I)+WF*DF(K,IP)
      RM=M
      FOBJ=FOBJ*8.0/RM
      C='gradient'
      DO 30 I=1,3
      DO 30 IP=1,MP
30    G(IP,I)=G(IP,I)*PA(IP,I)*8.0/RM
      WRITE(*,90)FOBJ
      WRITE(*,FMT1(NCKT))(CKTNAME(I),I=1,NCKT)
      WRITE(*,FMT1(NCKT))(C,I=1,NCKT)
      DO 40 IP=1,MP
40    WRITE(*,FMT2(NCKT))(G(IP,I),I=1,NCKT),PN(IP)
      WRITE(*,*)
90    FORMAT(10X,'SENSITIVITY OF THE CURRENT MODEL'/10X,
     *  'least-squares error:',F15.8)
100   RETURN
      END
```

```
      SUBROUTINE HEAD

c     escape sequence for clearing the screen

      PRINT*,CHAR(27)//'[2J'
      PRINT*,CHAR(27)//'[H'

      WRITE(*,1)
1     FORMAT(33X,'FET MODELLING'//25X,'Interactive Command System'/)
      RETURN
      END
```

```
      SUBROUTINE HELP
      WRITE(*,10)
10    FORMAT(10X,'Commands Available')
      WRITE(*,20)
20    FORMAT(5X,'L1',12X,'start modelling using L1 optimization'/5X,
     *'L2',12X,'start modelling using least-squares optimization'/5X,
     *'STOP          stop this program and return to VMS'/5X,
     *'DIR           list the file directory'/5X,
     *'HELP          display the available commands'/5X,
     *'PARAMETER     display the current model parameters'/5X,
     *'COMPARE       compare starting point with optimization solution'/
     *5X,'%ERROR',8X,'display percentage errors in S-parameter matching'
     */5X,'SWEEP         generate S-parameters of the current model'/
     *5X,'SENSITIVITY   sensitivity analysis for the current model'/
     *5X,'SAVE fname    save current data in a CKT file'/5X,
     *'READ fname    read desired data from a file'/5X,
     *'EDIT fn.ext   edit a disk file using full sreen editor'/5X,
     *'TYPE fn.ext   display the content of a disk file'/5X,
     *'PRESENTATION  call the user-defined routine PRESENT'/5X,
     *'BRIEFING      display the current status'/)
      RETURN
      END
```

```
      SUBROUTINE OPTIMI(JOPT,KYCD,FILNAM,CKTFIL)
      PARAMETER (MP=13,MX=40,MF=20)

c     prepare for optimization. call L1C or QUASI2

      CHARACTER*12 FILNAM,CKTFIL(6)
      DIMENSION PA(MP,3),X(MX),INPX(MP,3),BND(2,MX),INDX(MX),PAO(MP,3)
      DIMENSION INXMDA(3),FREQ(MF),WGT(8,MF,3),XX(MX),INPP(MP)
      DIMENSION PDIM(MP),DAT(8,MF,3),FM(8,MF,3),F1(8,MF,3)
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      COMMON/BLK1/PDIM,DAT,INPP,PAO
      COMMON/BLK2/SNR(8,3),KFR(MF,3),DTM(8,MF,3),KOUNT,KODE
      COMMON/L11/MARK,FF1(500),DFF1(500,MX),FF(500),DFF(500,MX)
      COMMON/L12/C(MX),DC(MX,MX),CLOC(MX)

      IF(NCKT.LE.0)CALL ERMSG(32,I,J,*100)

      N=NX
      IF(NX.LE.0)CALL ERMSG(33,I,J,*100)
      JOPT=0
      L=0

c     exponential transformation on the variables
c     lower bounds on the original variables are conveniently handled

      DO 20 I=1,N
      TEMP=X(I)-BND(1,I)
      IF(TEMP.GT.4.5E-5)THEN
              XX(I)=ALOG(TEMP)
      ELSE
              XX(I)=-10.0
      ENDIF
      IF(INDX(I).EQ.2)THEN

c     transform the upper bounds on the original variables

              L=L+1
              DO 10 J=1,N
10            DC(L,J)=0.0
              DC(L,I)=-1.0
              C(L)=ALOG(BND(2,I)-BND(1,I))
      ENDIF
20    CONTINUE

      M=0
      DO 60 I=1,NCKT
      DO 60 J=1,NFR
60    M=M+KFR(J,I)

c     save the starting point

      DO 70 I=1,NCKT
      DO 70 J=1,MP
70    PAO(J,I)=PA(J,I)
```

```
      IF(M.EQ.0)CALL ERMSG(34,I,J,*100)
      JOPT=1
      KOUNT=0
      KODE=KYCD
      DX=0.2
      MAXF=1000
      EPS=0.01
      IF(ETERM.GE.1.E-8)EPS=ETERM
      LEQ=0
      CALL BRIEF(FILNAM,CKTFIL,NCKT)

c     start optimization

      WRITE(*,*)
      WRITE(*,*)'           Start modelling ...'
      WRITE(*,*)'     You can press any key to interrupt optimization'
      WRITE(*,*)
      CALL LIB$INIT_TIMER
      IF(KODE.EQ.1)CALL L1C(N,M,L,LEQ,XX,DX,EPS,MAXF)
      IF(KODE.EQ.2)CALL QUASI2(N,M,L,LEQ,XX,DX,EPS,MAXF)
      CALL LIB$STAT_TIMER(2,ICPU)
      WRITE(*,*)
      IF(MARK.GT.0)THEN
              WRITE(*,*)'      Optimization Completed'
      ELSE
              WRITE(*,*)'      Optimization Interrupted'
      ENDIF
      WRITE(*,'(6X,''after '',I4,'' iterations'',F10.2,'' CPU seconds''
     *          )')MAXF,REAL(ICPU)*0.01

c     reverse the exponential transformation on the variables

      DO 80 I=1,NX
80    X(I)=EXP(XX(I))+BND(1,I)
      DO 90 K=1,NCKT
      DO 90 I=1,MP
      IF(INPX(I,K).GT.0)PA(I,K)=X(INPX(I,K))
90    CONTINUE

100   RETURN
      END
```

```
      SUBROUTINE PARSER(A,KYCD,FILNAM)

c     interpret a input command contained in the ASCII string A
c     return a integer code KYCD

      CHARACTER A*80,WRD*80,COMD(16)*2,KYWRD*2,FILNAM*12
      LOGICAL THERE
      SAVE COMD
      DATA COMD/'L1','L2','ST','DI','HE','PA','CO','%E','SW',
     *          'SE','PR','BR','SA','RE','ED','TY'/
      KYCD=17
      J0=1
      CALL GETWRD(A,J0,J1,WRD,LENG)
      IF(LENG.EQ.0)RETURN
      IF(LENG.EQ.1)GOTO 20
      KYWRD=WRD(1:2)
      DO 10 I=1,16
      IF(KYWRD.EQ.COMD(I))GOTO 30
10    CONTINUE
20    CALL ERMSG(31,I,J,*100)

c     is it a file related command?

30    IF(I.GE.13)THEN

c     yes, search the input string for a file name

40          CALL GETWRD(A,J0,J1,WRD,LENG)
            IF(LENG.EQ.0)THEN

c           file name missing, ask for it

                WRITE(*,'(A$)')'     File name: '
                READ(*,'(A)')A
                J0=1
                GOTO 40
            ENDIF
            FILNAM='            '
            FILNAM=WRD(1:LENG)
            IF(I.GE.15)GOTO 60

c           command 'save' or 'read'. put in the omitted extension '.DAT'

            DO 50 K=1,LENG
            IF(WRD(K:K).EQ.'.')GOTO 60
50          CONTINUE
            L1=LENG+1
            IF(L1.GT.9)L1=9
            L2=L1+3
            FILNAM(L1:L2)='.DAT'

60          IF(I.EQ.16.OR.I.EQ.14)THEN

c           command 'read' or 'type'. make sure the file exits
```

```
                    INQUIRE(FILE=FILNAM,EXIST=THERE)
                    IF(.NOT.THERE)CALL FERMSG(1,FILNAM,I,*100)
                ENDIF
        ENDIF

        KYCD=I
100     RETURN
        END
```

```
      subroutine passive_interrupt(cont)

C     check if any key has been pressed

      implicit integer*4 (a-z)
      integer*2 iosb(4)
      integer*2 chan/0/
      integer*4 quadword(2),time
      logical cont
      character*50 line
      common/io_channel/chan
      l = 1
      if (chan .eq. 0) call sys$assign('TT',chan,,)
      status = sys$qiow(,%val(chan),%val(55+128+64),iosb,,,
     1       %val(%loc(line)),%val(l),%val(time),quadword,,)
      l = iosb(2)
      if (l.eq.0) then
              cont = .true.
      else
              cont = .false.
      endif
      end
```

```
      SUBROUTINE PRESENT
      PARAMETER (MP=13,MX=40,MF=20)

c     gateway for user implemented operations. See manual
c     basic information:
c             NCKT:   number of circuits
c             PA:     current values of circuit parameters

      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),INXMDA(3),X(MX)
      DIMENSION PA(MP,3),INPX(MP,3),BND(2,MX),INDX(MX),WGT(8,MF,3)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8),FREQ(MF),PDIM(MP)
      DIMENSION DAT(8,MF,3),INPP(MP),PA0(MP,3)
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      COMMON/BLK1/PDIM,DAT,INPP,PA0

      RETURN
      END
```

```
      SUBROUTINE SAVCKT(FILNAM,OLDFIL,NX,X)
      PARAMETER (MP=13,MX=40,MF=20)

c     save the current infomation to a Super-Compact file

      CHARACTER TT(100)*80,VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,FILNAM*12,OLDFIL*12,AX*8,AT*80
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),X(MX)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8)
      INTEGER*4 STR$TRIM,ICODE
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/SC1/TD(6),IPT(18),XDIM(MX),VDIM(MX),IVTYP(MX)

      IF(NX.LE.0)GOTO 20
      DO 10 I=1,NX
      TEMP=X(I)/XDIM(I)
      IF(TEMP.GT.99.9999.OR.TEMP.LT.0.001)THEN
            WRITE(AX,'(E8.3)')TEMP
      ELSE
            WRITE(AX,'(F8.5)')TEMP
      ENDIF
      J1=LOCX(2,I)
      J2=J1+7
      TT(LOCX(1,I))(J1:J2)=AX
10    CONTINUE
20    OPEN(2,FILE=FILNAM,STATUS='NEW',CARRIAGECONTROL='LIST')
      OLDFIL=FILNAM
      DO 30 I=1,LINET
      AT=TT(I)(1:80)
      ICODE=STR$TRIM(AT,AT,LENG)
      WRITE(2,'(A)')AT(1:LENG)
30    CONTINUE
      CLOSE(2)
      RETURN
      END
```

```
      SUBROUTINE TAKECHAR(CHR)

c     read a single character from keyboard. no echo. no <CR>

      IMPLICIT INTEGER*4 (A-Z)
      INTEGER*2 IOSB(4)
      INTEGER*2 CHAN/0/
      CHARACTER CHR
      COMMON/IO_CHANNEL/CHAN
      L=1
      IF(CHAN.EQ.0)CALL SYS$ASSIGN('TT',CHAN)
      STATUS=SYS$QIOW(,%VAL(CHAN),%VAL(55+64),IOSB,,,
     * %VAL(%LOC(CHR)),%VAL(L),,,,)
      RETURN
      END
```

```
      SUBROUTINE WDA(FM,F1,KYCD)

C     display simulation results (commands 'SWEEP' and '%ER')

      PARAMETER (MP=13,MX=40,MF=20)
      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,TN(8)*4,CHR
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),INXMDA(3),X(MX)
      DIMENSION PA(MP,3),WGT(8,MF,3),INPX(MP,3),BND(2,MX),INDX(MX)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8),FREQ(MF)
      DIMENSION PDIM(MP),DAT(8,MF,3),FM(8,MF,3),F1(8,MF,3),INPP(MP)
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      COMMON/BLK1/PDIM,DAT,INPP,PA0(MP,3)
      SAVE TN
      DATA TN/'MS11','AS11','MS21','AS21','MS12','AS12','MS22','AS22'/
      IF(NCKT.LE.0)RETURN
      I=1
10    IF(KYCD.EQ.8)THEN
            WRITE(*,110)CKTNAME(I),CKTFIL(INXMDA(I))
            WRITE(*,'(6X,''(GHZ)'',8(4X,A4))')(TN(K),K=1,8)
            DO 20 J=1,NFR
20          WRITE(*,120)FREQ(J),(F1(K,J,I),K=1,8)
      ELSE
            JO=6
30          WRITE(JO,130)CKTNAME(I)
            WRITE(JO,'(1X,''*    (GHZ)'',8(4X,A4))')(TN(K),K=1,8)
            DO 40 J=1,NFR
40          WRITE(JO,140)FREQ(J),(FM(K,J,I),K=1,8)
            IF(CKTFIL(I)(1:1).NE.' ')THEN
                  IF(JO.EQ.2)THEN
                        CLOSE(2)
                  ELSE
                        OPEN(2,FILE=CKTFIL(I),STATUS='NEW')
                        JO=2
                        GOTO 30
                  ENDIF
            ENDIF
      ENDIF
      IF(I.GE.NCKT)GOTO 100
      WRITE(*,'(/5X,''Press <RETURN> to continue ...'')')
      READ(*,'(A)')CHR
      I=I+1
      WRITE(*,*)
      GOTO 10
100   RETURN
110   FORMAT(/19X,'PERCENTAGE ERRORS IN S-PARAMETER MATCHING'/20X,
     *'ABS[ (Model-Measurement)/Measurement ] %'/3X,'Frequency',7X,
     *'model: ',A8,6X,'measurement: ',A12)
120   FORMAT(1X,F9.1,2X,8(F7.2,'%'))
130   FORMAT(1X,'*',15X,'S-PARAMETERS OF MODEL ',A8/1X,'*  Frequency')
140   FORMAT(1X,F9.1,2X,4(F8.4,F8.2))
      END
```

```
      SUBROUTINE WPA(NCKT,PA,PDIM,FILNAM,CKTNAME)
      PARAMETER (MP=13,MX=40,MF=20)

c     display the current values of the parameters (command 'PA')

      DIMENSION PA(MP,3),PDIM(MP),PP(3)
      CHARACTER CKTNAME(6)*8,PN(MP)*8,FMT1(3)*15,FMT2(3)*19
      CHARACTER FILNAM*12
      SAVE PN,FMT1,FMT2

ccc   DATA PN/'RG  (OH)','RD  (OH)','GDS(/OH)','RI  (OH)','RS  (OH)',
ccc  *'LS  (nH)','CGS (pF)','CDG (pF)','CDS (pF)','G  (/OH)','T   (ps)'/
      DATA PN/'RG  (OH)','RD  (OH)','GDS(/OH)','RI  (OH)','RS  (OH)',
     *'LS  (nH)','CGS (pF)','CDG (pF)','CDS (pF)','G  (/OH)','T   (ps)',
     *'LG  (nH)','LD  (nH)'/

      DATA FMT1/'(16X,A8)       ','(12X,A8,10X,A8)','(5X,3(A8,10X)) '/
      DATA FMT2/'(5X,F18.5,5X,A8)   ','(1X,2F18.5,5X,A8)  ',
     *'(1X,3(F12.5,6X),A8)'/

      IF(NCKT.LE.0)CALL ERMSG(32,I,J,*30)
      WRITE(*,'(6X,''FET MODEL PARAMETERS    file: '',A12)')FILNAM
      WRITE(*,FMT1(NCKT))(CKTNAME(K),K=1,NCKT)
      DO 20 I=1,MP
      DO 10 K=1,NCKT
10    PP(K)=PA(I,K)*PDIM(I)
      WRITE(*,FMT2(NCKT))(PP(K),K=1,NCKT),PN(I)
20    CONTINUE
      WRITE(*,*)
30    RETURN
      END
```

```
c*************************************************************************

c       file name: FETFDF1.FOR

c       for program FETSC1 (VAX version).
c       this group of files are responsible for model simulation

c*************************************************************************



        SUBROUTINE CSOLU(Y,A,B,C,D)

c       solve a complex system of linear equations
c       using LU factorization

        COMPLEX*8 Y(6,6),A(6),B(6),C(6),D(6),Z,T1,T2,T3,T4
        DO 20 I=1,5
        Z=Y(I,I)
        IF(Z.EQ.(0.0,0.0))GO TO 110
        M=I+1
        DO 20 J=M,6
        Y(I,J)=Y(I,J)/Z
        DO 10 K=M,6
10      Y(K,J)=Y(K,J)-Y(K,I)*Y(I,J)
20      CONTINUE

        A(1)=1.0/Y(1,1)
        C(1)=1.0
        DO 40 I=2,6
        T1=0.0
        T2=0.0
        DO 30 K=1,I-1
        T1=T1-Y(I,K)*A(K)
30      T2=T2-Y(K,I)*C(K)
        A(I)=T1/Y(I,I)
40      C(I)=T2

        B(6)=1.0/Y(6,6)
        C(6)=C(6)/Y(6,6)
        D(6)=1.0/Y(6,6)
        DO 60 L=2,6
        I=7-L
        T1=A(I)
        T2=0.0
        T3=C(I)
        T4=0.0
        DO 50 K=I+1,6
        T1=T1-Y(I,K)*A(K)
        T2=T2-Y(I,K)*B(K)
        T3=T3-Y(K,I)*C(K)
50      T4=T4-Y(K,I)*D(K)
        A(I)=T1
        B(I)=T2
        C(I)=T3/Y(I,I)
```

```
60      D(I)=T4/Y(I,I)

        RETURN

110     WRITE(*,*)'FATAL ERROR IN CSOLU: ZERO ON THE DIAGONAL'
        STOP
        END
```

```
      SUBROUTINE FDF(N,M,XX)
      PARAMETER (MP=13,MX=40,MF=20)

c     organize calculation of error functions and their derivatives
c     called by the optimizer

      CHARACTER CHR
      LOGICAL CONT
      DIMENSION XX(N),FS(8),DFS(8,MP),INDX(MX)
      DIMENSION INXMDA(3),PA(MP,3),X(MX),INPX(MP,3),BND(2,MX)
      DIMENSION FREQ(MF),XA(MX),WGT(8,MF,3)
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      COMMON/BLK2/SNR(8,3),KFR(MF,3),DTM(8,MF,3),KOUNT,KODE
      COMMON/L11/MARK,F(500),DF(500,MX),FF(500),DFF(500,MX)

      KOUNT=KOUNT+1
      M1=0
      JSEN=1
      FOBJ=0.0
      CC1=90.0/ASIN(1.0)

c     reserve the exponential transformation on the variables

      DO 10 I=1,N
      XA(I)=EXP(XX(I))
      X(I)=XA(I)+BND(1,I)
      DO 10 J=1,M
10    DF(J,I)=0.0

c     loop for NCKT circuits

      DO 60 K=1,NCKT

c           update the variable parameters

            DO 20 I=1,MP
            IF(INPX(I,K).GT.0)PA(I,K)=X(INPX(I,K))
20          CONTINUE

c     sweep for all frequencies

      DO 60 J=1,NFR
            IF(KFR(J,K).EQ.0.AND.K.NE.JOUT)GOTO 60
            CALL SIMFET(FREQ(J),PA(1,K),FS,DFS,JSEN,K)

c           8 responses (real & imag. parts of 4 s-parameters)

            DO 40 L=1,8
                  IF(WGT(L,J,K).LE.1.E-5)GOTO 40
                  M1=M1+1
                  WT=WGT(L,J,K)*SNR(L,K)
                  F(M1)=WT*(FS(L)-DTM(L,J,K))

c                 compute the objective function
```

```
                        IF(KODE.EQ.1)THEN
                                FOBJ=FOBJ+ABS(F(M1))
                        ELSE
                                FOBJ=FOBJ+F(M1)*F(M1)
                        ENDIF

c                       assemble the gradients

                        DO 30 I=1,MP
                        I1=INPX(I,K)
                        IF(I1.GT.0)DF(M1,I1)=DF(M1,I1)+WT*DFS(L,I)*XA(I1)
30                      CONTINUE
40              CONTINUE
60      CONTINUE

c       check for user interrupt

        CALL PASSIVE_INTERRUPT(CONT)
        IF(.NOT.CONT)THEN

c               interrupt handler

70              WRITE(*,80)
80              FORMAT(///15X,'USER INTERRUPT'/6X,'Press  T  to terminate',
     1           ' optimization'/6X,'Or press <CR> to continue')
                CALL TAKECHAR(CHR)
                IF(CHR.EQ.'T'.OR.CHR.EQ.'t')THEN

c                       'T' has been pressed: terminate optimization

                        MARK=0

                ELSEIF(CHR.NE.CHAR(13))THEN

c                       neither 'T' nor <CR>. wait for another key

                        GOTO 70
                ENDIF
        ENDIF

c       display the count of iteration and the value of objective function

        RM=M
        FOBJ=FOBJ*8.0/RM
        IF (KODE.EQ.1)THEN
           WRITE(*,'(6X,''Iteration'',I3,5X,''L1 error'',F12.7)')KOUNT,FOBJ
        ELSE
           WRITE(*,'(6X,''Iteration'',I3,5X,''L2 error'',F12.7)')KOUNT,FOBJ
        ENDIF

        RETURN
        END
```

```
      SUBROUTINE SIMFET(FREQ,PA,F,DF,JSEN,ICKT)
      PARAMETER (MP=13,MX=40,MF=20)

c     organize model simulation for a single frequency (FREQ)
c     expect 13 parameters in the following order:
c     RG(OH),  RD(OH),  GDS(/OH),  RI(OH),  RS(OH),  LS(nH),  CGS(pF)
c     CDG(pF), CDS(pF), G(/OH),    T(ps),   LG(nH),  LD(nH)

      DIMENSION PA(MP),PAP(MP),PP(MP),F(8),DF(8,MP),DFS(8,MP)
      COMMON/BLK1/PDIM(MP),DAT(8,MF,3),INPP(MP),PA0(MP,3)
      COMMON/FETCONST/F3DB(3),GGS(3),FDIM
      COMPLEX*8 SMX(4),DSMX(4,MP),PH(6),QH(6),P(6),Q(6)
      COMPLEX*8 Y(6,6),S,Z(9),Z2P(4),ZDELTA,DZ2P(4,MP)
      COMPLEX*8 SMXMU(4),DSMXT(4),C1,C2,C3,C4,C5,C6
      INTEGER INDEX1(7),INDEX2(7)
      SAVE INDEX1,INDEX2
      DATA INDEX1,INDEX2/1,3,5,4,2,2,4,2,4,6,5,5,3,5/

      DO 1 I=1,MP
1     PAP(I)=PA(I)

ccc   because of the inclusion of LG and LD, the inversion of RG -> GG
ccc   and RD -> GD are no longer appropriate, only RI (PA(4)) needs
ccc   to be inverted

ccc   DO 2 I=1,4
ccc   IF(I.EQ.3)GOTO 2
      I=4
      IF(PAP(I).GT.1.E-6)THEN
            PAP(I)=1.0/PAP(I)
      ELSE
            PAP(I)=1.E6
      ENDIF
ccc2  CONTINUE

      DO 3 I=1,MP
      II=INPP(I)
3     PP(II)=PAP(I)

c     assemble the Y matrix

            DO 10 I=1,6
            DO 10 J=1,6
10          Y(J,I)=(0.0,0.0)

            TWOPIF=6.283185*FREQ
            C1=CMPLX(0.0,TWOPIF)
            C2=0.001*C1
            C3=CMPLX(1.0,FREQ/F3DB(ICKT))
            B=TWOPIF*0.001*PP(11)
            C4=CMPLX(COS(B),-SIN(B))/C3

ccc         DO 20 I=1,4
ccc20       Z(I)=PP(I)
```

```
c               the following four lines replace the above two lines

                        Z(1)=1.0/(PP(1)+C1*PP(12))
                        Z(2)=PP(2)
                        Z(3)=1.0/(PP(3)+C1*PP(13))
                        Z(4)=PP(4)

                DO 30 I=5,7
30              Z(I)=C2*PP(I)
                Z(8)=1.0/(PP(8)+C1*PP(9))
                Z(9)=PP(10)*C4

                Y(1,1)=Z(1)
                Y(1,2)=-Z(1)
                Y(2,2)=Z(1)+Z(6)+Z(5)+GGS(ICKT)
                Y(2,3)=-Z(6)-GGS(ICKT)
                Y(2,5)=-Z(5)
                Y(3,3)=Z(2)+Z(6)+GGS(ICKT)
                Y(3,4)=-Z(2)
                Y(4,4)=Z(2)+Z(4)+Z(7)+Z(8)
                Y(4,5)=-Z(4)-Z(7)
                Y(5,5)=Z(5)+Z(4)+Z(7)+Z(3)
                Y(5,6)=-Z(3)
                Y(6,6)=Z(3)

                DO 40 I=1,5
                DO 40 J=I+1,6
40              Y(J,I)=Y(I,J)

                Y(5,2)=Y(5,2)+Z(9)
                Y(5,3)=Y(5,3)-Z(9)
                Y(4,2)=Y(4,2)-Z(9)
                Y(4,3)=Y(4,3)+Z(9)

c       solve the nodal equation for 4 solutions

        CALL CSOLU(Y,P,Q,PH,QH)

c       construct the two-port Z matrix

                Z2P(1)=P(1)/50.0
                Z2P(2)=P(6)/50.0
                Z2P(3)=Q(1)/50.0
                Z2P(4)=Q(6)/50.0
                ZDELTA=(Z2P(1)+1.0)*(Z2P(4)+1.0)-Z2P(2)*Z2P(3)

c       compute the two-port S matrix

                SMX(1)=1.0-2.0*(Z2P(4)+1.0)/ZDELTA
                SMX(2)=2.0*Z2P(2)/ZDELTA
                SMX(3)=2.0*Z2P(3)/ZDELTA
                SMX(4)=1.0-2.0*(Z2P(1)+1.0)/ZDELTA

        IF(JSEN.EQ.0)GOTO 150
```

```
c       sensitivity analysis

                DO 60 I=1,7
                J1=INDEX1(I)
                J2=INDEX2(I)
                DZ2P(1,I)=(PH(J1)-PH(J2))*(P(J2)-P(J1))
                DZ2P(2,I)=(QH(J1)-QH(J2))*(P(J2)-P(J1))
                DZ2P(3,I)=(PH(J1)-PH(J2))*(Q(J2)-Q(J1))
60              DZ2P(4,I)=(QH(J1)-QH(J2))*(Q(J2)-Q(J1))

ccc     the following 7 lines are newly added to account for LG and LD

                        C5=-Z(1)*Z(1)
                        C6=-Z(3)*Z(3)
                        DO 65 J=1,4
                        DZ2P(J,1)=DZ2P(J,1)*C5
                        DZ2P(J,12)=DZ2P(J,1)*C1
                        DZ2P(J,3)=DZ2P(J,3)*C6
65                      DZ2P(J,13)=DZ2P(J,3)*C1
ccc     end of changes

                DO 70 I=5,7
                DO 70 J=1,4
70              DZ2P(J,I)=C2*DZ2P(J,I)

                C5=Z(8)*Z(8)
                DZ2P(1,8)=C5*PH(4)*P(4)
                DZ2P(2,8)=C5*QH(4)*P(4)
                DZ2P(3,8)=C5*PH(4)*Q(4)
                DZ2P(4,8)=C5*QH(4)*Q(4)

                DO 80 J=1,4
80              DZ2P(J,9)=C1*DZ2P(J,8)

                DZ2P(1,10)=C4*(PH(5)-PH(4))*(P(3)-P(2))
                DZ2P(2,10)=C4*(QH(5)-QH(4))*(P(3)-P(2))
                DZ2P(3,10)=C4*(PH(5)-PH(4))*(Q(3)-Q(2))
                DZ2P(4,10)=C4*(QH(5)-QH(4))*(Q(3)-Q(2))

                C6=-PP(10)*C2
                DO 90 J=1,4
90              DZ2P(J,11)=C6*DZ2P(J,10)

                SMXMU(1)=1.0-SMX(1)
                SMXMU(2)=-SMX(2)
                SMXMU(3)=-SMX(3)
                SMXMU(4)=1.0-SMX(4)

                DO 100 I=1,MP
                DSMXT(1)=SMXMU(1)*DZ2P(1,I)+SMXMU(3)*DZ2P(2,I)
                DSMXT(2)=SMXMU(2)*DZ2P(1,I)+SMXMU(4)*DZ2P(2,I)
                DSMXT(3)=SMXMU(1)*DZ2P(3,I)+SMXMU(3)*DZ2P(4,I)
                DSMXT(4)=SMXMU(2)*DZ2P(3,I)+SMXMU(4)*DZ2P(4,I)

                DSMX(1,I)=(DSMXT(1)*SMXMU(1)+DSMXT(3)*SMXMU(2))/100.0
```

```
                DSMX(2,I)=(DSMXT(2)*SMXMU(1)+DSMXT(4)*SMXMU(2))/100.0
                DSMX(3,I)=(DSMXT(1)*SMXMU(3)+DSMXT(3)*SMXMU(4))/100.0
100             DSMX(4,I)=(DSMXT(2)*SMXMU(3)+DSMXT(4)*SMXMU(4))/100.0

                DO 110 K=1,4
                K1=2*K-1
                DO 110 I=1,MP
                DFS(K1,I)=REAL(DSMX(K,I))
110             DFS(K1+1,I)=AIMAG(DSMX(K,I))

                DO 120 I=1,MP
                I1=INPP(I)
                DO 120 J=1,8
120             DF(J,I)=DFS(J,I1)

ccc             DO 140 I=1,4
ccc             IF(I.EQ.3)GOTO 140
                I=4
                PR=-PAP(I)*PAP(I)
                DO 130 J=1,8
130             DF(J,I)=DF(J,I)*PR
ccc140          CONTINUE

c       separate the real and imaginary parts of the S-parameters

150             DO 160 K=1,4
                K1=2*K-1
                F(K1)=REAL(SMX(K))
160             F(K1+1)=AIMAG(SMX(K))

        RETURN
        END
```

```
c*********************************************************************

c       file name: SCFILE1.FOR

c       for program FETSC1 (VAX version).
c       this group of subroutines constitutes a file handler to process
c       a circuit file and the associated measurement data files

c*********************************************************************



        BLOCK DATA A
        DIMENSION TYPDIM(6),IPTYP(18),IVTYP(40),XDIM(40),VDIM(40)
        COMMON/SC1/TYPDIM,IPTYP,XDIM,VDIM,IVTYP
        DATA TYPDIM/1.E-9,1.0,1.0,1.E9,1.E12,1.E12/
        DATA IPTYP/3,5,1,6,3,5,5,5,3,2,2,2,2,5,5,4,4,4/
        END
```

```
      SUBROUTINE BLKBLK(IERROR)
      PARAMETER (MP=13,MX=40,MF=20)

c     decode the 'BLK' block

      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,WRD*80,KYWRD*3,CHARR
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),INXMDA(3),X(MX)
      DIMENSION PA(MP,3),WGT(8,MF,3),INPX(MP,3),BND(2,MX),INDX(MX)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8),FREQ(MF)
      DIMENSION TYPDIM(6),IPTYP(18),IVTYP(MX),XDIM(MX),VDIM(MX)
      COMMON/SC1/TYPDIM,IPTYP,XDIM,VDIM,IVTYP
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1      TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      COMMON/FETCONST/F3DB(3),GGS(3),FDIM
      LTOP=LOCBLK(1,1)
      LBOTTOM=LOCBLK(2,1)
      IERROR=1
      IF(LTOP.EQ.0.OR.LBOTTOM.LE.LTOP)CALL ERMSG(1,I,J1,*80)
      DO 10 K=1,6
10    CKTNAME(K)='        '
      NX=0
      NVAR=0
      I=LTOP
      NCKT=0
      NFET=0
20    I=I+1
      IF(I.GT.LBOTTOM)GOTO 50
      J0=1
      CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
      IF(LENG.EQ.0)GOTO 20
      CALL GETSYMBL(WRD,LENG,IP,IFLAG)
      IF(NFET.EQ.1)GOTO 30
      IF(IFLAG.EQ.3)THEN
              LENG=IP-1
              J0=J1+IP
              IF(LENG.EQ.0.OR.LENG.GT.8)CALL ERMSG(6,I,J1,*80)
              NVAR=NVAR+1
              VNAME(NVAR)='        '
              VNAME(NVAR)(1:LENG)=WRD(1:LENG)
              CALL GETVAR(I,J0,NVAR,*80)
              IF(LOCV(3,NVAR).NE.0)THEN
                      NX=NX+1
                      INVAR(NVAR)=NX
                      X(NX)=VAL(1,NVAR)
                      XDIM(NX)=VDIM(NVAR)
                      LOCX(1,NX)=LOCV(1,NVAR)
                      LOCX(2,NX)=LOCV(2,NVAR)
                      INDX(NX)=LOCV(3,NVAR)
                      IF(INDX(NX).EQ.2)THEN
                              BND(1,NX)=VAL(2,NVAR)
                              BND(2,NX)=VAL(3,NVAR)
                      ENDIF
              ENDIF
```

```
              GOTO 20
            ENDIF
            IF(LENG.NE.3.OR.WRD(1:3).NE.'FET')CALL ERMSG(14,I,1,*80)
            CALL GETFET(I,J0,*80)
            NFET=1
            GOTO 20
30          IF(IFLAG.NE.3.OR.IP.NE.LENG.OR.LENG.LT.2)CALL ERMSG(16,I,1,*80)
            LENG=LENG-1
            NND=NCKT*3-2
            CKTNAME(NCKT)=WRD(1:LENG)
            CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
            IF(LENG.NE.4.OR.WRD(1:4).NE.'2POR')CALL ERMSG(18,I,J1,*80)
            CALL GETINT(TT(1,I),J0,J1,ND1,*90)
            IF(ND1.NE.NODE(NND))CALL ERMSG(15,I,J1,*80)
            CALL GETINT(TT(1,I),J0,J1,ND2,*90)
            IF(ND2.NE.NODE(NND+1))CALL ERMSG(15,I,J1,*80)
            NFET=0
            GOTO 20
50          IF(NFET.NE.0)CALL ERMSG(16,I,1,*80)
            IERROR=0
80          RETURN
90          CALL ERMSG(7,I,J1,*80)
            END
```

```
      SUBROUTINE CHKTYP(ITY,I,*)

c     make sure that a parameter and the symbolic variable assigned
c     to it have consistent types (capacitor, inductor, etc.)

      DIMENSION ITY(3)
      IF(ITY(2).EQ.7)GOTO 90
      IF(ITY(1).EQ.0.OR.ITY(1).EQ.7)GOTO 20
      IF(ITY(2).NE.0.AND.ITY(1).NE.ITY(2))GOTO 90
      IF(ITY(3).EQ.0.OR.ITY(3).EQ.7)GOTO 80
      IF(ITY(1).NE.ITY(3))GOTO 90
20    IF(ITY(3).EQ.0.OR.ITY(3).EQ.7)GOTO 80
      IF(ITY(2).EQ.0)GOTO 80
      IF(ITY(2).NE.ITY(3))GOTO 90
80    IF((ITY(1).NE.0).AND.(ITY(1).NE.7))ITY(2)=ITY(1)
      IF((ITY(3).NE.0).AND.(ITY(3).NE.7))ITY(2)=ITY(3)
      RETURN
90    CALL ERMSG(29,I,1,*100)
100   RETURN 1
      END
```

```
      SUBROUTINE DATABLK(IERROR)
      PARAMETER (MP=13,MX=40,MF=20)

c     decode the 'DATA' block

      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,WRD*80
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),INXMDA(3),X(MX)
      DIMENSION PA(MP,3),WGT(8,MF,3),INPX(MP,3),BND(2,MX),INDX(MX)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8),FREQ(MF)
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      DO 10 I=1,6
10    CKTFIL(I)='            '
      IERROR=1
      LTOP=LOCBLK(1,4)
      LBOTTOM=LOCBLK(2,4)
      IF(LTOP.EQ.0.OR.LTOP.GE.LBOTTOM)CALL ERMSG(35,I,J1,*100)
      LTOP=LTOP+1
      NMDA=3
      DO 90 I=LTOP,LBOTTOM
      J0=1
      CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
      IF(LENG.EQ.0)GOTO 90
      CALL GETSYMBL(WRD,LENG,IP,IFLAG)
      IF(IP.EQ.1.OR.IFLAG.NE.3)CALL ERMSG(30,I,J1,*100)
      NMDA=NMDA+1
      IF(NMDA.GT.6)CALL ERMSG(5,I,J1,*100)
      LENG=IP-1
      CKTNAME(NMDA)(1:LENG)=WRD(1:LENG)
      J0=IP+1
      CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
      IF(LENG.EQ.0)CALL ERMSG(17,I,J1,*100)
      CKTFIL(NMDA)(1:LENG)=WRD(1:LENG)
90    CONTINUE
      IERROR=0
100   RETURN
      END
```

```
      SUBROUTINE ERMSG(IER,ILINE,ICOL,*)
      PARAMETER (MP=13,MX=40,MF=20)

c     analyze an error and display the corresponding message

      CHARACTER TT(100)*80,VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,T*79
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8)
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      IF(IER.LE.3)THEN
              WRITE(*,10)MSG(IER)
      ELSEIF(IER.LE.30)THEN
              T=TT(ILINE)(1:79)
              WRITE(*,10)MSG(IER)
              WRITE(*,20)ILINE,ICOL
              WRITE(*,'(1X,A79)')T
      ELSE
              WRITE(*,30)MSG(IER)
      ENDIF
      RETURN 1
10    FORMAT(5X,'Error in circuit file: ',A30)
20    FORMAT(5X,'Error in circuit file at line ',I3,' after column',I3)
30    FORMAT(5X,'Error: ',A30)
      END
```

```
      SUBROUTINE FERMSG(IER,FILNAM,ILINE,*)

C     display file-related error messages

      CHARACTER FILNAM*12
      IF(IER.EQ.1)WRITE(*,10)FILNAM
      IF(IER.EQ.2)WRITE(*,20)FILNAM,ILINE
      IF(IER.EQ.3)WRITE(*,30)FILNAM
      IF(IER.EQ.4)WRITE(*,40)FILNAM
      RETURN 1
10    FORMAT(5X,'File not found: ',A12)
20    FORMAT(5X,'Bad data in the file ',A12,' at line',I3)
30    FORMAT(5X,'Not enough data in file ',A12)
40    FORMAT(5X,'File missing: ',A12)
      END
```

```
      SUBROUTINE FREQBLK(IERROR)
      PARAMETER (MP=13,MX=40,MF=20)

c     decode the 'FREQ' block

      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,WRD*80
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),INXMDA(3),X(MX)
      DIMENSION PA(MP,3),WGT(8,MF,3),INPX(MP,3),BND(2,MX),INDX(MX)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8),FREQ(MF)
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      IERROR=1
      LTOP=LOCBLK(1,2)
      LBOTTOM=LOCBLK(2,2)
      IF(LTOP.EQ.0.OR.LTOP.GE.LBOTTOM)CALL ERMSG(2,I,J1,*100)
      LTOP=LTOP+1
      NFR=0
      EPS=1.E-12
      DO 30 I=LTOP,LBOTTOM
      J0=1
10    J00=J0
      CALL GETWRD(TT(1,I),J00,J1,WRD,LENG)
      IF(LENG.EQ.0)GOTO 30
      II=ICHAR(WRD(1:1))
      IF(II.GE.65.AND.II.LE.122)THEN
              CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
              IF(LENG.EQ.4.AND.WRD(1:4).EQ.'STEP')THEN
                      CALL GETFRQ(I,J0,J1,FLOW,*100)
                      CALL GETFRQ(I,J0,J1,FHIGH,*100)
                      CALL GETFRQ(I,J0,J1,DFR,*100)
                      IF(DFR.LT.EPS)GOTO 120
                      IF(FLOW.LT.EPS)GOTO 120
                      IF(FHIGH.LT.EPS)GOTO 120
                      IF(FHIGH.LT.FLOW)THEN
                              TEMP=FHIGH
                              FHIGH=FLOW
                              FLOW=TEMP
                      ENDIF
                      N1=INT((FHIGH-FLOW)/DFR)+1
                      N2=NFR+N1
                      IF(N2.GT.MF)GOTO 110
                      DO 20 J=1,N1
                      FREQ(NFR+J)=FLOW
20                    FLOW=FLOW+DFR
                      NFR=N2
              ELSE
                      CALL ERMSG(8,I,J1,*100)
              ENDIF
      ELSE
              IF(NFR.GE.MF)GOTO 110
              NFR=NFR+1
              CALL GETFRQ(I,J0,J1,FREQ(NFR),*100)
      ENDIF
```

```
      GOTO 10
30    CONTINUE
      IF(NFR.EQ.0)CALL ERMSG(2,I,J1,*100)
      IERROR=0
100   RETURN
110   CALL ERMSG(19,I,J1,*100)
120   CALL ERMSG(20,I,J1,*100)
      END
```

```
      SUBROUTINE GETCHAR(STRG,ITOP,ICHR,CHARR,*)

C     get a single character from a line starting from the ITOPth position

      CHARACTER STRG(80),CHARR
      ICHR=0
      IF(ITOP.GT.80)RETURN 1
      DO 10 I=ITOP,80
      IF(STRG(I).NE.' ')GOTO 20
10    CONTINUE
      RETURN 1
20    CHARR=STRG(I)
      ICHR=I
      ITOP=I+1
      RETURN
      END
```

```
      SUBROUTINE GETFET(I,J0,*)
      PARAMETER (MP=13,MX=40,MF=20)

c     decode the FET element

      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,CHARR,PARNAM(18)*3,WRD*80
      CHARACTER KYWRD*3,VLABL*8
      DIMENSION IDP(18),ICO(18),VAL(3,MX),LOCV(3,MX),INXMDA(3),X(MX)
      DIMENSION PA(MP,3),WGT(8,MF,3),INPX(MP,3),BND(2,MX),INDX(MX)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8),FREQ(MF)
      DIMENSION TYPDIM(6),IPTYP(18),IVTYP(MX),XDIM(MX),VDIM(MX)
      COMMON/SC1/TYPDIM,IPTYP,XDIM,VDIM,IVTYP
      COMMON/TS1/LOCBLK(2,10),LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      COMMON/FETCONST/F3DB(3),GGS(3),FDIM
      SAVE PARNAM,IDP,ICO,IL
      DATA PARNAM/'G  ','CGS','F  ','T  ','GGS','CDG','CDC','CDS',
     1'GDS','RI ','RG ','RD ','RS ','CGE','CDE','LG ','LD ','LS '/
ccc   DATA IDP/10,7,6,11,6,8,6,9,3,4,1,2,5,6,6,6,6,6/
      DATA IDP/10,7,6,11,6,8,6,9,3,4,1,2,5,6,6,12,13,6/
ccc   DATA ICO/2,2,1,2,-1,2,0,2,2,2,2,2,2,0,0,0,0,2/
      DATA ICO/2,2,1,2,-1,2,0,2,2,2,2,2,2,0,0,2,2,2/

      IG=NCKT*3+1
      NCKT=NCKT+1
      IF(NCKT.GT.3)CALL ERMSG(5,I,1,*150)
      CALL GETINT(TT(1,I),J0,J1,NODE(IG),*140)
      CALL GETINT(TT(1,I),J0,J1,NODE(IG+1),*140)
      CALL GETCHAR(TT(1,I),J0,J1,CHARR,*10)
      IF(CHARR.EQ.'0')GOTO 10
      IF(CHARR.EQ.'G')THEN
              J0=J1
              GOTO 10
      ELSE
              CALL ERMSG(21,I,J1,*150)
      ENDIF
10    DO 20 K=1,MP
      PA(K,NCKT)=0.0
20    INPX(K,NCKT)=0
      F3DB(NCKT)=0.0
      GGS(NCKT)=0.0
      K=1
30    CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
      IF(LENG.EQ.0)THEN
              I=I+1
              J0=1
              CALL GETCHAR(TT(1,I),J0,J1,CHARR,*40)
              IF(CHARR.EQ.'+')GOTO 30
40            I=I-1
              GOTO 100
      ENDIF
      CALL NOSYMBL(WRD,LENG,J0,J1)
      IF(LENG.EQ.0.OR.LENG.GT.3)GOTO 60
```

```
        KYWRD='    '
        KYWRD(1:LENG)=WRD(1:LENG)
        DO 50 KK=K,18
        IF(KYWRD.EQ.PARNAM(KK))GOTO 70
50      CONTINUE
60      CALL ERMSG(8,I,J1,*150)
70      K=KK+1
        IPAR=IDP(KK)
        JTYP=IPTYP(KK)
        JOO=JO
        CALL GETCHAR(TT(1,I),JOO,J1,CHARR,*160)
        IF(CHARR.EQ.'=')JO=JOO
        JOO=JO
        CALL GETWRD(TT(1,I),JO,J1,WRD,LENG)
        IF(LENG.EQ.0)GOTO 160
        IA=ICHAR(WRD(1:1))
        IF(IA.GE.65.AND.IA.LE.122)THEN
                IF(NVAR.EQ.0)GOTO 170
                IF(LENG.GT.8)CALL ERMSG(6,I,J1,*150)
                VLABL='        '
                VLABL(1:LENG)=WRD(1:LENG)
                DO 80 L=1,NVAR
                IF(VLABL.EQ.VNAME(L))GOTO 90
80              CONTINUE
                GOTO 170
        ELSE
                JO=JOO
                CALL GETVAR(I,JO,MX,*150)
                IF(LOCV(3,MX).NE.0)THEN
                        NX=NX+1
                        INVAR(MX)=NX
                        X(NX)=VAL(1,MX)
                        XDIM(NX)=VDIM(MX)
                        LOCX(1,NX)=LOCV(1,MX)
                        LOCX(2,NX)=LOCV(2,MX)
                        INDX(NX)=LOCV(3,MX)
                        IF(INDX(NX).EQ.2)THEN
                                BND(1,NX)=VAL(2,MX)
                                BND(2,NX)=VAL(3,MX)
                        ENDIF
                ENDIF
                L=MX
        ENDIF
90      IF(IVTYP(L).EQ.0)THEN
                IVTYP(L)=JTYP
                VAL(1,L)=VAL(1,L)*TYPDIM(JTYP)
                IF(LOCV(3,L).NE.0)THEN
                        JX=INVAR(L)
                        X(JX)=X(JX)*TYPDIM(JTYP)
                        XDIM(JX)=XDIM(JX)*TYPDIM(JTYP)
                        IF(INDX(JX).EQ.2)THEN
                                BND(1,JX)=BND(1,JX)*TYPDIM(JTYP)
                                BND(2,JX)=BND(2,JX)*TYPDIM(JTYP)
                        ENDIF
                ENDIF
```

```
      ELSEIF(IVTYP(L).NE.JTYP)THEN
              CALL ERMSG(29,I,J1,*150)
      ENDIF
      PA(IPAR,NCKT)=VAL(1,L)
      IF(LOCV(3,L).EQ.0)THEN
              INPX(IPAR,NCKT)=0
      ELSE
              INPX(IPAR,NCKT)=INVAR(L)
      ENDIF
      IJ=INPX(6,NCKT)
      ZZ=PA(6,NCKT)
      IF(ICO(KK).EQ.1)THEN
              F3DB(NCKT)=ZZ
              IF(IJ.NE.0)CALL ERMSG(11,I,J1,*150)
      ELSEIF(ICO(KK).EQ.-1)THEN
              GGS(NCKT)=ZZ
              IF(IJ.NE.0)CALL ERMSG(12,I,J1,*150)
      ELSEIF(ICO(KK).EQ.0)THEN
              IF(IJ.NE.0.OR.ZZ.NE.0.0)CALL ERMSG(13,I,J1,*150)
      ENDIF
      IF(K.LE.18)GOTO 30
100   RETURN
140   CALL ERMSG(7,I,J1,*150)
150   RETURN1
160   CALL ERMSG(4,I,J1,*150)
170   CALL ERMSG(10,I,J1,*150)
      END
```

```
      SUBROUTINE GETFRQ(I,J0,J1,TEMP,*)
      PARAMETER (MP=13,MX=40,MF=20)

c     decode a single frequency

      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX), NODE(9),LOCX(2,MX)
      DIMENSION TYPDIM(6),IPTYP(18),IVTYP(MX),XDIM(MX),VDIM(MX)
      COMMON/SC1/TYPDIM,IPTYP,XDIM,VDIM,IVTYP
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR(MX),LOCX,VAL,DIM(8),
     1      TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      CALL GETVAR(I,J0,MX,*100)
      IF(IVTYP(MX).GT.1)CALL ERMSG(29,I,J1,*100)
      IF(LOCV(3,MX).NE.0)CALL ERMSG(4,I,J1,*100)
      TEMP=VAL(1,MX)
      RETURN
100   RETURN 1
      END
```

```
      SUBROUTINE GETINT(STRG,ITOP,IP,IVAL,*)

C     try to convert an ASCII string into an integer

      CHARACTER STRG(80),WRD*80,WRD1*20
      CALL GETWRD(STRG,ITOP,IP,WRD,LENG)
      IF(LENG.EQ.0)RETURN 1
      WRD1='                    '
      WRD1(1:LENG)=WRD(1:LENG)
      READ(WRD1,*,ERR=10,END=10)IVAL
      IF(IVAL.GE.0)RETURN
10    RETURN 1
      END
```

```
      SUBROUTINE GETREAL(WRD,LENG,VALUE,*)

c     try to convert an ASCII string into a real value

      CHARACTER WRD*80,WRD1*20
      IF(LENG.EQ.0)GOTO 10
      WRD1='                    '
      WRD1(1:LENG)=WRD(1:LENG)
      READ(WRD1,*,ERR=10,END=10)VALUE
      IF(VALUE.GE.0.0)RETURN
10    RETURN 1
      END
```

```
      SUBROUTINE GETSYMBL(WRD,LENG,IP,IFLAG)

C     detect special symbols in a word

      CHARACTER SYMBOLS(10),WRD*80,A
      SAVE SYMBOLS
      DATA SYMBOLS/'=','?',':','+','-','%','*','/','>','<'/
      IFLAG=0
      DO 10 I=1,LENG
      A=WRD(I:I)
      DO 10 J=1,4
      IF(A.EQ.SYMBOLS(J))GOTO 20
10    CONTINUE
      RETURN
20    IP=I
      IFLAG=J
      RETURN
      END
```

```
      SUBROUTINE GETUNIT(WRD,LENG,IP,ITYPE,VAL)

c     decode the unit attached to a numeral

      CHARACTER WRD*80,KYWRD*4,UNAM(6,7)*4
      DIMENSION DDIM(6,7)
      SAVE UNAM,DDIM
      DATA UNAM/'HZ  ','KHZ ','MHZ ','GHZ ','    ','    ','OH  ','KOH ',
     1'MOH ','    ','    ','    ','/OH ','/KOH','/MOH','    ','    ',
     2'    ','L   ','MH  ','UH  ','NH  ','PH  ','    ','F   ','MF  ',
     3'UF  ','NF  ','PF  ','FF  ','SEC ','MS  ','US  ','NS  ','    ',
     4'    ','%   ','    ','    ','    ','    ','    ','    '/
      DATA DDIM/1.,1.E3,1.E6,1.E9,0.,0.,1.,1.E3,1.E6,0.,0.,0.,1.,1.E-3,
     1  1.E-6,0.,0.,0.,1.,1.E-3,1.E-6,1.E-9,1.E-12,0.,1.,1.E-3,1.E-6,
     2  1.E-9,1.E-12,1.E-15,1.0,1.E-3,1.E-6,1.E-9,8*0.0/
      IP=0
      ITYPE=0
      VAL=1.0
      DO 10 I=LENG,1,-1
      IA=ICHAR(WRD(I:I))
      IF(IA.GE.48.AND.IA.LE.57)GOTO 20
       IF(IA.EQ.4)GOTO 20
10     CONTINUE
       RETURN
20     IF(I.EQ.LENG)RETURN
       IP=I+1
       IG=LENG-IP+1
       IF(IG.GT.4)GOTO 40
       KYWRD='    '
       KYWRD(1:IG)=WRD(IP:LENG)
       DO 30 I=1,7
       DO 30 J=1,6
       IF(KYWRD.EQ.UNAM(J,I))GOTO 50
30     CONTINUE
40     IP=-1
       RETURN
50     ITYPE=I
       VAL=DDIM(J,I)
       RETURN
       END
```

```
      SUBROUTINE GETVAR(I,J0,NVR,*)
      PARAMETER (MP=13,MX=40,MF=20)

c     decode infomation relating to a variable including
c     its value, type (bounded or not) and the bounds on it

      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,WRD*80,A
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),INXMDA(3),ITY(3)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8),VU(3),VL(3)
      DIMENSION TYPDIM(6),IPTYP(18),IVTYP(MX),XDIM(MX),VDIM(MX)
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/SC1/TYPDIM,IPTYP,XDIM,VDIM,IVTYP
      J00=J0
      INDXV=0
      CALL GETCHAR(TT(1,I),J00,J1,A,*70)
      IF(A.EQ.'?')THEN
              INDXV=1
              J0=J00
              JX=J00
      ENDIF
      KOUNT=1
10    CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
      CALL NOSYMBL(WRD,LENG,J0,J1)
      IF(LENG.EQ.0)GOTO 70
      CALL GETUNIT(WRD,LENG,IP,ITY(KOUNT),VU(KOUNT))
      IF(IP.LT.0)CALL ERMSG(25,I,J1,*70)
      IF(IP.GT.0)LENG=IP-1
      CALL GETREAL(WRD,LENG,VL(KOUNT),*70)
      IF(INDXV.EQ.0)GOTO 25
      IF(KOUNT.EQ.1)THEN
              J00=J0
              CALL GETCHAR(TT(1,I),J00,J11,A,*80)
              IF(A.EQ.'?')GOTO 20
              JX=J0
              KOUNT=2
              LOCV(3,NVR)=2
      ELSEIF(KOUNT.EQ.2)THEN
              KOUNT=3
              JX1=J1+LENG
              CALL PAD(TT(1,I),JX,JX1,ISHIFT)
              J0=J0+ISHIFT
      ELSE
              CALL CHKTYP(ITY,I,*90)
              CALL GETCHAR(TT(1,I),J0,J1,A,*80)
              IF(A.NE.'?')GOTO 80
              GOTO 30
      ENDIF
      GOTO 10
20    JX1=J1+LENG
      CALL PAD(TT(1,I),JX,JX1,ISHIFT)
      J0=J00+ISHIFT
25    LOCV(3,NVR)=INDXV
      LOCV(2,NVR)=JX
```

```
      VDIM(NVR)=VU(1)
      IF(ITY(1).EQ.7)THEN
              CALL ERMSG(29,I,1,*90)
      ELSEIF(ITY(1).NE.0)THEN
              VDIM(NVR)=VDIM(NVR)*TYPDIM(ITY(1))
      ENDIF
      IVTYP(NVR)=ITY(1)
      VAL(1,NVR)=VL(1)*VDIM(NVR)
      GOTO 40
30    LOCV(2,NVR)=JX
      VDIM(NVR)=VU(2)
      IF(ITY(2).NE.0)VDIM(NVR)=VDIM(NVR)*TYPDIM(ITY(2))
      VAL(1,NVR)=VL(2)*VDIM(NVR)
      IF(ITY(1).EQ.7)THEN
              VAL(2,NVR)=VAL(1,NVR)*VL(1)*0.01
      ELSE
              IF(ITY(2).NE.0)VU(1)=VU(1)*TYPDIM(ITY(2))
              VAL(2,NVR)=VL(1)*VU(1)
      ENDIF
      IF(ITY(3).EQ.7)THEN
              VAL(3,NVR)=VAL(1,NVR)*VL(3)*0.01
      ELSE
              IF(ITY(2).NE.0)VU(3)=VU(3)*TYPDIM(ITY(2))
              VAL(3,NVR)=VL(3)*VU(3)
      ENDIF
      IF(VAL(3,NVR).LE.VAL(2,NVR))CALL ERMSG(26,I,1,*90)
      IVTYP(NVR)=ITY(2)
40    LOCV(1,NVR)=I
      RETURN
70    CALL ERMSG(4,I,J1,*90)
80    CALL ERMSG(22,I,J1,*90)
90    RETURN1
      END
```

```
      SUBROUTINE GETWRD(STRG,ITOP,IWRD,WRD,LENG)

c     get a word from a line (STRG) starting from the ITOPth position.
c     words are separated by blanks. return the word in WRD
c     its length in LENG

      CHARACTER STRG(80),WRD*80,A
      LENG=0
      IF(ITOP.GT.80)RETURN
      DO 10 I=ITOP,80
      IF(STRG(I).NE.' ')GOTO 20
10    CONTINUE
      RETURN
20    IF(STRG(I).EQ.'*')RETURN
      IWRD=I
      IW=1
      DO 30 I=IWRD,80
      A=STRG(I)
      IF(A.EQ.' ')GOTO 40
      IA=ICHAR(A)
      IF(IA.GE.97.AND.IA.LE.122)A=CHAR(IA-32)
      WRD(IW:IW)=A
      IW=IW+1
30    CONTINUE
40    ITOP=I
      LENG=ITOP-IWRD
      RETURN
      END
```

```
      SUBROUTINE NODUMMY(ILINE)

c     skip over comment lines in the measurement files

      CHARACTER CLINE*80,DUMMY
1     READ(1,'(A)',END=4)CLINE
      J0=1
      ILINE=ILINE+1
2     CALL GETCHAR(CLINE,J0,J1,DUMMY,*1)
      IF(DUMMY.EQ.CHAR(9))GOTO 2
      IF(DUMMY.EQ.'*')GOTO 1
      IDUMMY=ICHAR(DUMMY)-48
      IF(IDUMMY.GE.0.AND.IDUMMY.LE.9)GOTO 3
      IF(DUMMY.NE.'+'.AND.DUMMY.NE.'-'.AND.DUMMY.NE.'.')GOTO 1
3     BACKSPACE 1
      ILINE=ILINE-1
4     RETURN
      END
```

```
      SUBROUTINE NOSYMBL(WRD,LENG,J0,J1)

C     detect special symbols within a word

      CHARACTER WRD*80
      IF(LENG.EQ.0)RETURN
      CALL GETSYMBL(WRD,LENG,IP,IFLAG)
      IF(IFLAG.GT.0)THEN
              J0=J1+IP-1
              LENG=IP-1
      ENDIF
      RETURN
      END
```

```
      SUBROUTINE OPTBLK(IERROR)
      PARAMETER (MP=13,MX=40,MF=20)

c     decode the 'OPT' block

      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8
      CHARACTER WRD*80,PNAME*8,CNAME*7,CMDA(8)*7,CHARR
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),INXMDA(3),X(MX)
      DIMENSION PA(MP,3),WGT(8,MF,3),INPX(MP,3),BND(2,MX),INDX(MX)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8),FREQ(MF),WW(4)
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      DO 10 I=1,3
      DO 10 J=1,8
      DO 10 K=1,NFR
10    WGT(K,J,I)=0.0
      ETERM=0.0
      IERROR=1
      LTOP=LOCBLK(1,3)
      LBOTTOM=LOCBLK(2,3)
      IF(LTOP.EQ.0.OR.LTOP.GE.LBOTTOM)GOTO 210
      I=LTOP
      JCKT=0
20    I=I+1
      IF(I.GT.LBOTTOM)GOTO 150
      J0=1
30    CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
      IF(LENG.EQ.0)GOTO 20
      CHARR=WRD(1:1)
      IF(CHARR.EQ.'F'.AND.JCKT.GT.0)THEN
              IF(LENG.GT.1.AND.WRD(2:2).NE.'=')GOTO 100
              J0=J1+1
              CALL GETCHAR(TT(1,I),J0,J1,CHARR,*240)
              IF(CHARR.NE.'=')GOTO 220
              CALL GETFRQ(I,J0,J1,FLOW,*200)
              FHIGH=FLOW
              J00=J0
              CALL GETCHAR(TT(1,I),J00,J1,CHARR,*20)
              II=ICHAR(CHARR)
              IF(II.GE.43.AND.II.LE.57)CALL GETFRQ(I,J0,J1,FHIGH,*200)
              GOTO 30
      ELSEIF(CHARR.EQ.'S'.AND.JCKT.GT.0)THEN
              IF(LENG.GT.1.AND.WRD(2:2).NE.'=')GOTO 100
              J0=J1+1
              CALL GETCHAR(TT(1,I),J0,J1,CHARR,*240)
              IF(CHARR.NE.'=')GOTO 220
              CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
              IF(LENG.EQ.0.OR.LENG.GT.8)GOTO 240
              PNAME='        '
              PNAME(1:LENG)=WRD(1:LENG)
              DO 40 K1=4,6
              IF(PNAME.EQ.CKTNAME(K1))GOTO 50
40            CONTINUE
```

```
                GOTO 220
50              INXMDA(JCKT)=K1
                J00=J0
                CALL GETCHAR(TT(1,I),J00,J1,CHARR,*80)
                IF(CHARR.EQ.'W')THEN
                        J0=J00
                        CALL GETCHAR(TT(1,I),J00,J1,CHARR,*60)
                        IF(CHARR.EQ.'=')J0=J00
60                      DO 70 J=1,4
                        CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
                        CALL GETREAL(WRD,LENG,WW(J),*230)
70                      CONTINUE
                        TEMP=WW(2)
                        WW(2)=WW(3)
                        WW(3)=TEMP
                ENDIF
80              DO 95 J=1,NFR
                IF(FREQ(J).LT.FLOW.OR.FREQ(J).GT.FHIGH)GOTO 95
                DO 90 L=1,4
                L2=2*L
                L1=L2-1
                WGT(L1,J,JCKT)=WW(L)
90              WGT(L2,J,JCKT)=WW(L)
95              CONTINUE
                GOTO 30
        ELSEIF(LENG.EQ.4.AND.WRD(1:4).EQ.'TERM')THEN
                J0=J1+4
                CALL GETCHAR(TT(1,I),J0,J1,CHARR,*250)
                IF(CHARR.NE.'=')GOTO 220
                CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
                CALL GETREAL(WRD,LENG,ETERM,*230)
                GOTO 30
        ENDIF
100     IF(LENG.GT.8)GOTO 250
        PNAME='         '
        PNAME(1:LENG)=WRD(1:LENG)
        DO 110 K0=1,3
        IF(PNAME.EQ.CKTNAME(K0))GOTO 120
110     CONTINUE
        GOTO 250
120     JCKT=K0
        FHIGH=1.E20
        FLOW=-100.0
        DO 130 L=1,4
130     WW(L)=1.0
        GOTO 30
150     DO 160 I=1,NCKT
        IF(INXMDA(I).GT.0)GOTO 160
                WRITE(*,260)CKTNAME(I)
                GOTO 200
160     CONTINUE
        IERROR=0
200     RETURN
210     CALL ERMSG(3,I,J1,*200)
220     CALL ERMSG(8,I,J1,*200)
```

```
230     CALL ERMSG(4,I,J1,*200)
240     CALL ERMSG(24,I,J1,*200)
250     IF(JCKT.EQ.0)CALL ERMSG(16,I,J1,*200)
        GOTO 220
260     FORMAT(5X,'Error in CKT file:  ',A8,' not used for modelling')
        END
```

```
      SUBROUTINE OUTBLK(IERROR,NCKT)
      PARAMETER (MP=13,MX=40,MF=20)

c     decode the 'OUT' block

      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,WRD*80,WRD1*8,WRD2*3
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),INXMDA(3)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8)
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/BLKPLT/IX(10),IY(10,8,3),YMIN(8,3),YR(8,3),JOUT
      JOUT=1
      IF(NCKT.LE.1)RETURN
      LTOP=LOCBLK(1,5)
      LBOTTOM=LOCBLK(2,5)
      IF(LTOP.EQ.0.OR.LTOP.GE.LBOTTOM)RETURN
      LTOP=LTOP+1
      DO 20 I=LTOP,LBOTTOM
      J0=1
      CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
      IF(LENG.EQ.0)GOTO 20
      IF(LENG.NE.3)GOTO 40
      WRD2=WRD(1:3)
      IF(WRD2.EQ.'PRI')THEN
              CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
              IF(LENG.NE.3)GOTO 20
              WRD2=WRD(1:3)
              IF(WRD2.NE.'PLO')GOTO 20
      ENDIF
      IF(WRD2.EQ.'PLO')THEN
              CALL GETWRD(TT(1,I),J0,J1,WRD,LENG)
              IF(LENG.EQ.0.OR.LENG.GT.8)GOTO 50
              WRD1='        '
              WRD1(1:LENG)=WRD(1:LENG)
              DO 10 K=1,NCKT
              IF(WRD1.EQ.CKTNAME(K))GOTO 30
10            CONTINUE
              GOTO 50
      ELSE
              GOTO 40
      ENDIF
20    CONTINUE
      RETURN
30    JOUT=K
      RETURN
40    CALL ERMSG(8,I,J1,*60)
50    CALL ERMSG(16,I,J1,*60)
60    IERROR=1
      RETURN
      END
```

```
      SUBROUTINE PAD(STRG,JX,JX1,ISHIFT)

c     insert blanks between words so that 8 digits can be accomodated

      CHARACTER STRG(80)
      ISHIFT=8-JX1+JX
      IF(ISHIFT.LE.0)RETURN
      I00=80-ISHIFT
      IF(I00.LE.JX1)RETURN
      DO 10 I=I00,JX1,-1
10    STRG(I+ISHIFT)=STRG(I)
      J3=JX+7
      DO 20 I=JX1,J3
20    STRG(I)='0'
      RETURN
      END
```

```
      SUBROUTINE RDCKT(FILNAM,OLDFIL,IERROR)
      PARAMETER (MP=13,MX=40,MF=20)

c     RDCKT is called by the main program.  It is the link between
c     this group of files and the rest of the program.
c     Organizes the process of reading circuit and measurement files.

      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,A*80,KYWRD*2,STRG*17
      CHARACTER FILNAM*12,OLDFIL*12
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),INXMDA(3)
      DIMENSION PA(MP,3),X(MX),INPX(MP,3),BND(2,MX),INDX(MX)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8),FREQ(MF)
      DIMENSION PDIM(MP),DAT(8,MF,3),INPP(MP),WGT(8,MF,3)
      LOGICAL THERE
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      COMMON/TS2/NX,NCKT,NFR,INXMDA,INPX,INDX,PA,X,BND,FREQ,WGT,ETERM
      COMMON/BLK1/PDIM,DAT,INPP,PA0(MP,3)
      COMMON/BLK2/SNR(8,3),KFR(MF,3),DTM(8,MF,3),KOUNT,KODE
      COMMON/FETCONST/F3DB(3),GGS(3),FDIM

      OPEN(1,FILE=FILNAM,STATUS='OLD')
      WRITE(*,'(5X,''Reading file: '',A12)')FILNAM

c     read circuit file

      CALL RDSCF(IERROR)
      WRITE(*,*)
      IF(IERROR.NE.1)CALL BLKBLK(IERROR)
      IF(IERROR.NE.1)CALL FREQBLK(IERROR)
      IF(IERROR.NE.1)CALL DATABLK(IERROR)
      IF(IERROR.NE.1)CALL OPTBLK(IERROR)
      IF(IERROR.NE.1)CALL OUTBLK(IERROR,NCKT)
      CLOSE(1)
      IF(IERROR.EQ.1)GOTO 230
      OLDFIL=FILNAM

c     read measurement files

      DO 40 K=1,NCKT
      FILNAM=CKTFIL(INXMDA(K))
      INQUIRE(FILE=FILNAM,EXIST=THERE)
      IF(.NOT.THERE)CALL FERMSG(1,FILNAM,I,*220)
      OPEN(1,FILE=FILNAM,STATUS='OLD')
      EPS=0.001*FREQ(1)
      ILINE=1
      DO 30 J=1,NFR
20    CALL NODUMMY(ILINE)
      READ(1,*,ERR=190,END=200)FRE,(DAT(M,J,K),M=1,8)
      ILINE=ILINE+1
      IF(ABS(FRE-FREQ(J)).GE.EPS)GOTO 20
30    CONTINUE
      CLOSE(1)
40    CONTINUE
```

```
      DO 50 J=1,6
50    DIM(J)=1.0
      DO 60 J=1,MP
60    PDIM(J)=1.0
      FDIM=1.0

c     transform the measurement data to rectangular form

      C1=ASIN(1.0)/90.0
      DO 70 I=1,NCKT
      DO 70 K=1,4
      K1=2*K-1
      K2=K1+1
      DO 70 J=1,NFR
      T1=DAT(K1,J,I)
      T2=DAT(K2,J,I)*C1
      DTM(K1,J,I)=T1*COS(T2)
      DTM(K2,J,I)=T1*SIN(T2)
70    CONTINUE
      DO 100 I=1,NCKT
      DO 100 K=1,8
      SNR(K,I)=1.0
      IF(NFR.EQ.1)GOTO 100
      SMAX=-2000.0
      SMIN=2000.0
      DO 80 J=1,NFR
      IF(DTM(K,J,I).GT.SMAX)SMAX=DTM(K,J,I)
      IF(DTM(K,J,I).LT.SMIN)SMIN=DTM(K,J,I)
80    CONTINUE
      SNR(K,I)=1.0/(SMAX-SMIN)
100   CONTINUE

c     count the error functions

      MM=0
      DO 120 I=1,NCKT
      DO 120 J=1,NFR
      M1=0
      DO 110 K=1,8
      IF(WGT(K,J,I).GT.1.E-5)M1=M1+1
110   CONTINUE
      MM=MM+M1
120   KFR(J,I)=M1

      CALL BRIEF(OLDFIL,CKTFIL,NCKT)
180   RETURN
190   CALL FERMSG(2,FILNAM,ILINE,*210)
200   CALL FERMSG(3,FILNAM,ILINE,*210)
210   CLOSE(1)
220   IERROR=1
230   NCKT=0
      GOTO 180
      END
```

```
      SUBROUTINE RDSCF(IERROR)
      PARAMETER (MP=13,MX=40,MF=20)

c     put the contents of a the circuit file in char array TT(80,100)
c     locate various blocks in the file
c     block i  begins at the LOCBLK(1,i)th line and ends at
c     the LOCBLK(2,i)th line

      CHARACTER TT(80,100),VNAME(MX)*8,CKTNAME(6)*8,CKTFIL(6)*12
      CHARACTER MSG(40)*30,PROC(3)*8,T*80,A,B,BLKNAME(6)*5,WRD*80
      CHARACTER KYWRD*5
      DIMENSION LOCBLK(2,10),VAL(3,MX),LOCV(3,MX),INXMDA(3)
      DIMENSION NODE(9),LOCX(2,MX),INVAR(MX),DIM(8)
      COMMON/TS1/LOCBLK,LINET,NVAR,LOCV,NODE,INVAR,LOCX,VAL,DIM,
     1     TT,VNAME,CKTNAME,CKTFIL,PROC,MSG
      SAVE BLKNAME
      DATA BLKNAME/'BLK  ','FREQ ','OPT  ','DATA ','OUT  ','     '/
      B=CHAR(9)
      DO 25 I=1,100
      READ(1,'(A)',ERR=30,END=30)T
      DO 10 J=1,80
10    TT(J,I)=' '
      K=1
      DO 20 J=1,80
      IF(T(J:J).EQ.B)THEN

c     a tab

            K=(K/9+1)*9
      ELSE
            IF(K.GT.80)GOTO 25
            TT(K,I)=T(J:J)
            K=K+1
      ENDIF
20    CONTINUE
25    CONTINUE
30    LINET=I-1
      DO 40 I=1,10
40    LOCBLK(1,I)=0
      IPRECED=0
      DO 80 I=1,LINET
      J0=1
      KYWRD='     '
      CALL GETWRD(TT(1,I),J0,J2,WRD,LENG)
      IF(LENG.EQ.0)GOTO 80
      IF(IPRECED.EQ.0)THEN
            IF(LENG.GT.4)GOTO 60
            KYWRD(1:LENG)=WRD(1:LENG)
            DO 50 J=1,5
            IF(KYWRD.EQ.BLKNAME(J))GOTO 70
50          CONTINUE
60          CALL ERMSG(27,I,1,*100)
70          IPRECED=J
            LOCBLK(1,IPRECED)=I
      ELSEIF(LENG.EQ.3.AND.WRD(1:3).EQ.'END')THEN
```

```
                LOCBLK(2,IPRECED)=I-1
                IPRECED=0
        ENDIF
80      CONTINUE
        IF(IPRECED.NE.0)CALL ERMSG(28,LINET,1,*100)
        IERROR=0
        RETURN
100     IERROR=1
        RETURN
        END
```

```
c*************************************************************************

c       file name: L1ABC.FOR

c       for programs FETSC and FETSC1 (VAX version).
c       this group of subroutines constitutes the L1 optimization
c       package

c*************************************************************************


      SUBROUTINE ADDCOL(N,KCOL,KNEW,RIGHT,W,ACCUM,LRIGHT,EPS)
      PARAMETER (NX=40,MX=500,LX=40,NX2=NX+NX)
      COMMON/L13/XX(NX2),X1(NX),R(NX,NX),B(NX,NX),W0(NX),W1(NX),W2(NX),
     1    ASET(NX),U(NX,NX),KSET(NX),KSET0(NX),KSTATC(LX),KSTATF(MX)
      DIMENSION RIGHT(N),W(N)
      LOGICAL ACCUM,LRIGHT
      K1=KCOL+1
      K2=KCOL+KNEW
      DO 200 K=K1,K2
      S=0.0
      T=0.0
      IF(K.EQ.1) GOTO 20
      KK=K-1
      DO 10 I=1,KK
   10 T=T+R(I,K)**2
   20 DO 30 I=K,N
   30 S=S+R(I,K)**2
      T=T+S
      T=SQRT(T)
      S=SQRT(S)
      IF(T.EQ.0.0) RETURN
      IF(S.LT.T*EPS) RETURN
      TT=R(K,K)
      T=ABS(TT)
      ALFA=SQRT(S*(S+T))
      BETA=-SIGN(S,TT)
      R(K,K)=BETA
      W(K)=(TT-BETA)/ALFA
      IF(K.EQ.N) GOTO 80
      KK=K+1
      DO 40 I=KK,N
   40 W(I)=R(I,K)/ALFA
      IF(K.EQ.K2) GOTO 80
      DO 70 J=KK,K2
      T=0.0
      DO 50 I=K,N
   50 T=T+W(I)*R(I,J)
      DO 60 I=K,N
   60 R(I,J)=R(I,J)-T*W(I)
   70 CONTINUE
   80 IF(.NOT.LRIGHT) GOTO 110
      T=0.0
      DO 90 I=K,N
```

```
 90 T=T+W(I)*RIGHT(I)
    DO 100 I=K,N
100 RIGHT(I)=RIGHT(I)-T*W(I)
110 IF(ACCUM) GOTO 130
    DO 120 I=K,N
120 U(I,K)=W(I)
    GOTO 200
130 DO 160 I=1,N
    T=0.0
    DO 140 J=K,N
140 T=T+U(I,J)*W(J)
    DO 150 J=K,N
150 U(I,J)=U(I,J)-T*W(J)
160 CONTINUE
200 KCOL=KCOL+1
    RETURN
    END
```

```
      SUBROUTINE BFGS(N,SEPS)
      PARAMETER (NX=40,MX=500,LX=40,NX2=NX+NX)
      COMMON/L13/XX(NX2),X1(NX),R(NX,NX),B(NX,NX),W(NX),W1(NX),Y(NX),
     1    ASET(NX),U(NX,NX),KSET(NX),KSET0(NX),KSTATC(LX),KSTATF(MX)
      EPS=(N+10)*SEPS
      DO 20 I=1,N
      T=0.0
      DO 10 J=1,N
   10 T=T+B(I,J)*XX(J)
   20 W(I)=T
      YXX=0.0
      WXX=0.0
      DO 30 I=1,N
      YXX=YXX+Y(I)*XX(I)
   30 WXX=WXX+W(I)*XX(I)
      IF(WXX.LT.1.E-12.OR.YXX.LT.EPS*WXX) RETURN
      DO 40 I=1,N
   40 B(I,I)=B(I,I)+Y(I)**2/YXX-W(I)**2/WXX
      IF(N.EQ.1) RETURN
      DO 50 I=2,N
      I1=I-1
      DO 50 J=1,I1
      B(I,J)=B(I,J)+Y(I)*Y(J)/YXX-W(I)*W(J)/WXX
   50 B(J,I)=B(I,J)
      RETURN
      END
```

```
      SUBROUTINE DELCOL(K,N,KCOL)
      PARAMETER (NX=40,MX=500,LX=40,NX2=NX+NX)
      COMMON/L13/XOO(NX2),X1(NX),R(NX,NX),B(NX,NX),W(NX),H(NX),W2(NX),
     1    ASET(NX),U(NX,NX),KSET(NX),KSETO(NX),KSTATC(LX),KSTATF(MX)
      COMMON/L14/RIGHT(NX),FUP(MX),DLDF(MX),CUP(LX),DLDC(LX),
     1       STEPF(MX),KSTATL(NX),DSTA2(NX2,NX2)
      KCOL=KCOL-1
      IF(K.GT.KCOL) RETURN
      DO 10 J=K,KCOL
      J1=J+1
      DO 10 I=1,J1
   10 R(I,J)=R(I,J1)
      DO 100 KK=K,KCOL
      K1=KK+1
      X=R(KK,KK)
      Y=R(K1,KK)
      A=SQRT(X*X+Y*Y)
      C=X/A
      S=Y/A
      R(KK,KK)=C*X+S*Y
      IF(KK.EQ.KCOL) GOTO 60
      DO 50 J=K1,KCOL
      X=R(KK,J)
      Y=R(K1,J)
      R(KK,J)=C*X+S*Y
   50 R(K1,J)=C*Y-S*X
   60 X=RIGHT(KK)
      Y=RIGHT(K1)
      RIGHT(KK)=C*X+S*Y
      RIGHT(K1)=C*Y-S*X
   70 DO 80 I=1,N
      X=U(I,KK)
      Y=U(I,K1)
      U(I,KK)=C*X+S*Y
   80 U(I,K1)=C*Y-S*X
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE HACCUM(N,KCOL)
      PARAMETER (NX=40,MX=500,LX=40,NX2=NX+NX)
      COMMON/L13/X(NX2),X1(NX),R(NX,NX),B(NX,NX),W(NX),H(NX),W2(NX),
     1    ASET(NX),U(NX,NX),KSET(NX),KSET0(NX),KSTATC(LX),KSTATF(MX)
      K1=KCOL+1
      DO 10 I=KCOL,N
   10 W(I)=U(I,KCOL)
      DO 20 I=KCOL,N
   20 U(I,I)=1.0-W(I)**2
      IF(KCOL.EQ.N) GOTO 40
      DO 30 I=K1,N
      I1=I-1
      T=W(I)
      DO 30 J=KCOL,I1
      S=-T*W(J)
      U(I,J)=S
   30 U(J,I)=S
   40 IF(KCOL.EQ.1) RETURN
      DO 100 KK=2,KCOL
      K=K1-KK
      DO 50 I=K,N
   50 W(I)=U(I,K)
      T=W(K)
      KP1=K+1
      U(K,K)=1.0-T**2
      DO 60 I=KP1,N
   60 U(I,K)=-T*W(I)
      DO 90 L=KP1,N
      S=0.0
      DO 70 I=KP1,N
   70 S=S+W(I)*U(I,L)
      U(K,L)=-T*S
      DO 80 I=KP1,N
   80 U(I,L)=U(I,L)-S*W(I)
   90 CONTINUE
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE L1A(N,M,X,MAXF)
      DIMENSION X(N)
      DX=0.0
      DO 10 I=1,N
10    DX=DX+X(I)*X(I)
      DX=0.05*SQRT(DX)
      IF(DX.LT.0.005)DX=0.005
      EPS=1.E-6
      MAXF=200
      CALL L1B(N,M,X,DX,EPS,MAXF)
      RETURN
      END
```

```
SUBROUTINE L1B(N,M,X,DX,EPS,MAXF)
DIMENSION X(N)
L=0
LEQ=0
CALL L1C(N,M,L,LEQ,X,DX,EPS,MAXF)
RETURN
END
```

```
      SUBROUTINE L1C(N,M,L,LEQ,X,DX,EPS,MAXF)
      PARAMETER (NX=40,MX=500,LX=40,NX2=NX+NX)
      DIMENSION X(N)
      COMMON/L11/MARK,F1(MX),DF1(MX,NX),F(MX),DF(MX,NX)
      COMMON/L12/C(LX),DC(LX,NX),CLOC(LX)
      COMMON/L13/XX(NX2),X1(NX),R(NX,NX),B(NX,NX),W(NX),W1(NX),W2(NX),
     1    ASET(NX),U(NX,NX),KSET(NX),KSET0(NX),KSTATC(LX),KSTATF(MX)

      SEPS=1.E-12
      IF(N.GT.NX.OR.M.GT.MX.OR.L.GT.LX)THEN
        WRITE(*,*)' Fatal error: dimensional limits exceeded'
        RETURN
      ENDIF
      IF(DX.LE.SEPS.OR.EPS.LE.SEPS)THEN
        WRITE(*,*)' Fatal error: DX or EPS too small '
        RETURN
      ENDIF

      LI=L-LEQ
      LNN=L+N+N
      KEQSET=0
      NSTEP=0
      NACT0=0
      MARK=1
      NCALL=1
      DX0=DX
      FL1REF=1.E16
      XN=0.0
      DO 20 I=1,N
      XN=XN+X(I)**2
      DO 10 J=1,N
10    B(I,J)=0.0
20    B(I,I)=1.0
      XN=SQRT(XN)
      IF(L.GT.0) THEN
              DO 30 K=1,L
              T=0.0
              DO 25 I=1,N
25            T=T+DC(K,I)*X(I)
30            CLOC(K)=T+C(K)
      ENDIF
      CALL FDF(N,M,X)
      IF(MARK.EQ.0)GOTO 300
      FL10=0.0
      DO 40 J=1,M
      F(J)=F1(J)
      DO 35 I=1,N
35    DF(J,I)=DF1(J,I)
40    FL10=FL10+ABS(F(J))

c     iterative loop

50    NACT=NACT0
      IF(NACT.GT.0) THEN
              DO 60 I=1,NACT
```

```
60              KSET(I)=KSET0(I)
      ENDIF

      CALL L1LP(M,N,L,LEQ,DX,XXN,NACT,FL1,IFALL)
      IF(FL1.GE.FL10) GOTO 300

      DO 70 I=1,N
70    X1(I)=X(I)+XX(I)
      CALL FDF(N,M,X1)
      IF(MARK.NE.1)GOTO 300
      NCALL=NCALL+1
      FL11=0.0
      DO 80 I=1,M
80    FL11=FL11+ABS(F1(I))

      IF((FL10-FL11).GT.0.75*(FL10-FL1)) DX=2.0*XXN
      IF((FL10-FL11).LT.0.25*(FL10-FL1)) DX=0.25*XXN

      K=0
      IF(NACT.GT.0)THEN
              DO 110 I=1,NACT
              II=KSET(I)
              IF(II.GT.LNN)THEN
                      K=K+1
                      KSET(K)=KSET(I)
                      II=II-LNN
                      DO 90 J=1,N
90                    R(J,K)=DF(II,J)
              ELSEIF(II.LE.L)THEN
                      K=K+1
                      KSET(K)=KSET(I)
                      DO 100 J=1,N
100                   R(J,K)=DC(II,J)
              ENDIF
110           CONTINUE
      ENDIF

      DO 120 I=1,N
      W(I)=0.D0
120   W1(I)=0.D0
      DO 150 I=1,M
      IF(KSTATF(I).LT.0) THEN
              DO 130 J=1,N
              W(J)=W(J)-DF(I,J)
130           W1(J)=W1(J)-DF1(I,J)
      ELSEIF(KSTATF(I).GT.0) THEN
              DO 140 J=1,N
              W(J)=W(J)+DF(I,J)
140           W1(J)=W1(J)+DF1(I,J)
      ENDIF
150   CONTINUE
      DO 160 I=1,N
160   ASET(I)=-W(I)
      NACT=0
      IF(K.GT.0) THEN
```

```
                CALL ADDCOL(N,NACT,K,ASET,W2,.FALSE.,.TRUE.,SEPS)
                IF(NACT.LT.K) KEQSET=1
                CALL RSOLV(N,NACT,ASET,ASET)
                DO 180 I=1,NACT
                K=KSET(I)
                IF(K.GT.LNN) THEN
                        KK=K-LNN
                        T=ASET(I)
                        DO 170 J=1,N
                        W1(J)=W1(J)+T*DF1(KK,J)
170                     W(J)=W(J)+T*DF(KK,J)
                ENDIF
180             CONTINUE
        ENDIF

        DO 190 I=1,N
190     W2(I)=W1(I)-W(I)
        CALL BFGS(N,SEPS)

        IF((FL10-FL11).LE.0.01*(FL10-FL1)) GOTO 270

        IF(NACT0.NE.NACT) THEN
                KEQSET=1
        ELSE
                IF(NACT.GT.0) THEN
                        DO 200 I=1,NACT
                        K=KSET(I)
                        DO 210 J=1,NACT
                        IF(K.EQ.KSET0(J)) GOTO 200
210                     CONTINUE
                        KEQSET=1
                        GOTO 220
200                     CONTINUE
                ENDIF
                IF (FL11.GE.0.99*FL10) KEQSET=KEQSET+1
        ENDIF

220     NSTEP=NSTEP+1
        XN=0.0
        FL10=FL11
        NACT0=NACT
        DO 230 I=1,N
        X(I)=X1(I)
        XN=XN+X(I)**2
        DO 230 J=1,M
230     DF(J,I)=DF1(J,I)
        XN=SQRT(XN)
        DO 240 I=1,M
240     F(I)=F1(I)
        IF(L.GT.0) THEN
                DO 250 K=1,L
                T=0.0
                DO 245 I=1,N
245             T=T+DC(K,I)*X(I)
250             CLOC(K)=T+C(K)
```

```
      ENDIF
      IF(NACT.GT.0) THEN
              DO 260 I=1,NACT
260           KSET0(I)=KSET(I)
      ENDIF

270   IF(NCALL.GT.MAXF.OR.XXN.LE.EPS*XN.OR.XXN.LE.SEPS*XN) GOTO 300

      IF(NSTEP.LT.N.OR.FL10.GT.FL1REF.OR.KEQSET.LT.3) GOTO 50
      IF(NACT.GT.LEQ) THEN
              DO 280 I=1,NACT
              K=KSET(I)
              T=ASET(I)
              IF(K.GT.LEQ.AND.K.LE.L.AND.T.GE.0.0) GOTO 50
              IF(K.GT.LNN.AND.ABS(T).GE.1.0) GOTO 50
280           CONTINUE
      ENDIF

      NSHIFT=NSHIFT+1
      FL1REF=FL10*0.999999
      XXNMAX=AMAX1(DX0,10.0*DX)
      CALL L1STA2(N,M,L,LEQ,X,XXNMAX,NACT,EPS,MAXF,NCALL,
     1      XXN,NSTEP,IFALL)
      IF(IFALL.LT.3) GOTO 300
      FL10=0.0
      DO 290 I=1,M
290   FL10=FL10+ABS(F(I))
      DX=AMAX1(DX,XXN*0.5)
      KEQSET=1
      GOTO 50

300   MAXF=NCALL
      EPS=XXN
      RETURN
      END
```

```
      SUBROUTINE L1LP(M,N,L,LE,XNMAX,XN,NACT,FL1,IFALL)
      PARAMETER (NX=40,MX=500,LX=40,NX2=NX+NX)
      COMMON/L11/MARK,F1(MX),DF1(MX,NX),F(MX),DF(MX,NX)
      COMMON/L12/CO(LX),DC(LX,NX),C(LX)
      COMMON/L13/X(NX2),X1(NX),R(NX,NX),B(NX,NX),W(NX),H(NX),W2(NX),
     1    ASET(NX),U(NX,NX),KSET(NX),KSET0(NX),KSTATC(LX),KSTATF(MX)
      COMMON/L14/DL(NX),FUP(MX),DLDF(MX),CUP(LX),DLDC(LX),
     1       STEPF(MX),KSTATL(NX),DSTA2(NX2,NX2)
      LOGICAL ACCUM,LINDEP
      LE1=LE+1
      LI=L-LE
      LN=L+N
      LNN=LN+N
      XN=0.0
      SEPS=1.E-12
      EPS=N*SEPS
      ACCUM=.FALSE.
      LINDEP=.FALSE.
      NCYC=0
      IFALL=0
      FL1=1.E16
      DO 10 I=1,N
      KSTATL(I)=0
   10 X(I)=0.0
      DO 15 I=1,M
      KSTATF(I)=1
      IF(F(I).LT.0.0) KSTATF(I)=-1
   15 FUP(I)=F(I)

c     activate initial active set of constraints

      NACTIN=NACT
      NACT=0
      IF(L.EQ.0) GOTO 100
      DO 20 I=1,L
      KSTATC(I)=0
   20 CUP(I)=C(I)
      IF(LE.EQ.0) GOTO 50
      DO 40 I=1,LE
      DO 30 J=1,N
   30 R(J,I)=DC(I,J)
      KSET(I)=I
   40 KSTATC(I)=1
      CALL ADDCOL(N,NACT,LE,H,W,ACCUM,.FALSE.,EPS)
      IF(NACT.EQ.LE) GOTO 50
      IFALL=3
      RETURN
   50 IF(NACTIN.LT.LE1) GOTO 100
      DO 70 K=LE1,NACTIN
      KK=KSET(K)
      IF(KK.LT.LE1.OR.KK.GT.L) GOTO 70
      NACT1=NACT+1
      IF(NACT1.GT.N) GOTO 100
      DO 60 I=1,N
   60 R(I,NACT1)=DC(KK,I)
```

```
      CALL UTTRNS(N,NACT,ACCUM,R(1,NACT1))
      CALL ADDCOL(N,NACT,1,H,W,ACCUM,.FALSE.,EPS)
      IF(NACT.LT.NACT1) GOTO 70
      EPS=EPS+2.0*SEPS
      KSET(NACT1)=KK
      KSTATC(KK)=1
   70 CONTINUE

c     calculate the restricted gradient

  100 FL1=0.0
      DO 105 I=1,N
  105 H(I)=0.0
      DO 140 I=1,M
      FL1=FL1+ABS(FUP(I))
      K=KSTATF(I)
      IF(K.GT.-1) GOTO 120
      DO 110 J=1,N
  110 H(J)=H(J)-DF(I,J)
  120 IF(K.LT.1) GOTO 140
      DO 130 J=1,N
  130 H(J)=H(J)+DF(I,J)
  140 CONTINUE

c     transform the restricted gradient

      DO 150 I=1,N
  150 DL(I)=-H(I)
      CALL UTTRNS(N,NACT,ACCUM,DL)

c     check if the current point is a dead point

  200 T=0.0
      NACT1=NACT+1
      DLN2=0.0
      IF(NACT.EQ.N) GOTO 220
      DO 210 I=NACT1,N
  210 DLN2=DLN2+DL(I)**2
  220 IF(NACT1.EQ.1) GOTO 240
      DO 230 I=1,NACT
  230 T=T+DL(I)**2
  240 T=T+DLN2
      IF(T.EQ.0.0) GOTO 900
      IF(DLN2.GT.EPS*EPS*T) GOTO 395

c     dead point - calculate multipliers

      IF(NACT.EQ.LE) GOTO 900
      CALL RSOLV(N,NACT,DL,ASET)
      AMAX=-1.E16
      DO 250 K=LE1,NACT
      A=ASET(K)
      IF(KSET(K).GT.LNN) A=ABS(A)-1.0
      IF(A.LE.AMAX) GOTO 250
      KMAX=K
```

```
      AMAX=A
  250 CONTINUE
      IF(AMAX.LE.0.0) GOTO 900

c     delete a function from the active set

      KK=KSET(KMAX)
      IF(KK.LE.LNN) GOTO 310
      KK=KK-LNN
      A=ASET(KMAX)
      IF(A.LT.0.0) GOTO 280
      KSTATF(KK)=1
      DO 260 I=1,N
  260 H(I)=H(I)+DF(KK,I)
      DO 270 I=1,KMAX
  270 DL(I)=DL(I)-R(I,KMAX)
      GOTO 330
  280 KSTATF(KK)=-1
      DO 290 I=1,N
  290 H(I)=H(I)-DF(KK,I)
      DO 300 I=1,KMAX
  300 DL(I)=DL(I)+R(I,KMAX)
      GOTO 330

c     deactivate a constraint

  310 IF(KK.GT.L) GOTO 320
      KSTATC(KK)=0
      GOTO 330
  320 IF(KK.GT.LN) GOTO 325
      KSTATL(KK-L)=0
      GOTO 330
  325 KSTATL(KK-LN)=0

c     remove the corresponding column in r

  330 IF(ACCUM) GOTO 340
      ACCUM=.TRUE.
      CALL HACCUM(N,NACT)
  340 CALL DELCOL(KMAX,N,NACT)
      EPS=EPS+2.0*SEPS
      IF(KMAX.GT.NACT) GOTO 360
      DO 350 I=KMAX,NACT
  350 KSET(I)=KSET(I+1)

c     remove linear dependence labels

  360 IF(.NOT.LINDEP) GOTO 200
      LINDEP=.FALSE.
      DO 370 I=1,M
      IF(KSTATF(I).EQ.2) KSTATF(I)=1
  370 IF(KSTATF(I).EQ.-2) KSTATF(I)=-1
      DO 380 I=1,N
  380 IF(KSTATL(I).EQ.2) KSTATL(I)=0
      IF(LI.EQ.0) GOTO 200
```

```
      DO 390 I=LE1,L
  390 IF(KSTATC(I).EQ.2) KSTATC(I)=0
      GOTO 200

c     calculate the direction of search

  395 IF(NACT.EQ.0) GOTO 405
      DO 400 I=1,NACT
  400 DL(I)=0.0
  405 CALL UTRNS(N,NACT,ACCUM)
      DLH=0.0
      DO 410 I=1,N
  410 DLH=DLH+DL(I)*H(I)

c     project gradients on the direction of search

      DO 415 J=1,M
      T=0.0
      DO 412 I=1,N
412   T=T+DF(J,I)*DL(I)
415   DLDF(J)=T
      IF(L.GT.0) THEN
                 DO 419 K=1,L
                 T=0.0
                 DO 418 I=1,N
418              T=T+DC(K,I)*DL(I)
419              DLDC(K)=T
      ENDIF

c     calculate the step lengths

      DO 420 I=1,M
      S=DLDF(I)
      K=KSTATF(I)
      STEPF(I)=1.E16
      IF(IABS(K).NE.1.OR.K*S.GE.-1.E-12) GOTO 420
      STEPF(I)=-FUP(I)/S
  420 CONTINUE
  425 SLP=1.E16
      SLN=1.E16
      DO 440 I=1,N
      IF(KSTATL(I).NE.0) GOTO 440
      S=DL(I)
      IF(S*S.LE.EPS*EPS*DLN2) GOTO 440
      T=(XNMAX-X(I))/S
      IF(S.LT.0.0.OR.T.GE.SLP) GOTO 430
      SLP=T
      KLP=I
  430 T=-(XNMAX+X(I))/S
      IF(S.GT.0.0.OR.T.GE.SLN) GOTO 440
      SLN=T
      KLN=I
  440 CONTINUE
      SC=1.E16
      IF(LI.EQ.0) GOTO 460
```

```
      DO 450 I=LE1,L
      S=DLDC(I)
      IF(KSTATC(I).NE.0.OR.S.GE.-1.E-12) GOTO 450
      T=-CUP(I)/S
      IF(T.GE.SC) GOTO 450
      KC=I
      SC=T
  450 CONTINUE
  460 STEPLC=AMIN1(SLP,SLN)
      STEPLC=AMIN1(STEPLC,SC)
      NACT1=NACT+1

c     do line search

  500 SF=1.E16
      DO 510 I=1,M
      S=STEPF(I)
      IF(S.GE.SF) GOTO 510
      KF=I
      SF=S
  510 CONTINUE
      STEP=AMIN1(STEPLC,SF)
      IF(SF.GT.STEP) GOTO 630
      T=DLH-2*KSTATF(KF)*DLDF(KF)
      IF(T.GE.0.0) GOTO 600
      DLH=T
      KSTATF(KF)=-KSTATF(KF)
      STEPF(KF)=1.E16
      GOTO 500

c     activate a function

  600 DO 610 I=1,N
  610 R(I,NACT1)=DF(KF,I)
      CALL UTTRNS(N,NACT,ACCUM,R(1,NACT1))
      CALL ADDCOL(N,NACT,1,DL,W,ACCUM,.FALSE.,EPS)
      IF(NACT.EQ.NACT1) GOTO 620
      KSTATF(KF)=2*KSTATF(KF)
      STEPF(KF)=1.E16
      LINDEP=.TRUE.
      GOTO 500
  620 KSET(NACT)=KF+LNN
      KSTATF(KF)=0
      GOTO 700

c     -  or a constraint

  630 IF(SC.GT.STEP) GOTO 660
      DO 640 I=1,N
  640 R(I,NACT1)=DC(KC,I)
      CALL UTTRNS(N,NACT,ACCUM,R(1,NACT1))
      CALL ADDCOL(N,NACT,1,DL,W,ACCUM,.FALSE.,EPS)
      IF(NACT.EQ.NACT1) GOTO 650
      KSTATC(KC)=2
      LINDEP=.TRUE.
```

```
      GOTO 425
  650 KSTATC(KC)=1
      KSET(NACT)=KC
      GOTO 700

c     - or a bound

  660 DO 670 I=1,N
  670 R(I,NACT1)=0.0
      IF(SLN.EQ.SLP) SLN=2.E16
      IF(STEP.EQ.SLN) R(KLN,NACT1)=1.0
      IF(STEP.EQ.SLP) R(KLP,NACT1)=-1.0
      CALL UTTRNS(N,NACT,ACCUM,R(1,NACT1))
      CALL ADDCOL(N,NACT,1,DL,W,ACCUM,.FALSE.,EPS)
      IF(NACT.EQ.NACT1) GOTO 680
      LINDEP=.TRUE.
      IF(STEP.EQ.SLN) KSTATL(KLN)=2
      IF(STEP.EQ.SLP) KSTATL(KLP)=2
      GOTO 425
  680 IF(STEP.EQ.SLN) GOTO 690
      KSET(NACT)=KLP+L
      KSTATL(KLP)=1
      GOTO 700
  690 KSET(NACT)=KLN+LN
      KSTATL(KLN)=-1
  700 EPS=EPS+2.0*SEPS

c     introduce the new point

      XN=0.0
      XXN=0.0
      NCYC=NCYC+1
      IF(NCYC.GT.2*N) GOTO 900
      DO 800 I=1,N
      S=STEP*DL(I)
      XXN=AMAX1(XXN,ABS(S))
      X(I)=X(I)+S
  800 XN=AMAX1(XN,ABS(X(I)))
      IF(XXN.GT.EPS*XN) NCYC=0
      DO 810 I=1,M
  810 FUP(I)=FUP(I)+STEP*DLDF(I)
      IF(L.EQ.0) GOTO 100
      DO 820 I=1,L
  820 CUP(I)=CUP(I)+STEP*DLDC(I)
      GOTO 100

c     return

  900 IF(NACT.LT.N) IFALL=2
      DO 910 I=1,NACT
  910 IF(KSET(I).GT.L.AND.KSET(I).LE.LNN) IFALL=1
      IF(IFALL.EQ.1) XN=XNMAX
      IF(NCYC.GT.2*N) IFALL=4
      RETURN
      END
```

```
      SUBROUTINE L1STA2(N,M,L,LEQ,X,XXNMAX,NACT,EPS,MAXF,NCALL,
     1       XXN,NSTEP,IFALL)
      PARAMETER (NX=40,MX=500,LX=40,NX2=NX+NX)
      DIMENSION X(N)
      COMMON/L11/MARK,F1(MX),DF1(MX,NX),F(MX),DF(MX,NX)
      COMMON/L12/C(LX),DC(LX,NX),CLOC(LX)
      COMMON/L13/ZZ(NX2),X1(NX),R(NX,NX),B(NX,NX),W(NX),W0(NX),W1(NX),
     1    ASET(NX),U(NX,NX),KSET(N),KSET0(NX),KSTATC(LX),KSTATF(MX)
      COMMON/L14/ASET1(NX),WM(MX),DLDF(MX),WL(LX),DLDC(LX),
     1       STEPF(MX),KSTATL(NX),DZ(NX2,NX2)

      LI=L-LEQ
      LE1=LEQ+1
      LNN=L+N+N
      N1=N+1
      IFALL=0
      SEPS=1.E-12
      SSEPS=SQRT(SEPS)
      NZ=N+NACT
      NSTEP2=0
      XXN=0.0
      RESREF=1.E16

c     iterative loop starts here
c     set up the iteration matrix and the righthand side

   50 DO 70 I=1,N
      DO 60 J=1,N
   60 DZ(I,J)=B(I,J)
   70 ZZ(I)=0.0
      DO 100 I=1,M
      IF(KSTATF(I).GE.0) GOTO 80
      DO 75 J=1,N
   75 ZZ(J)=ZZ(J)+DF(I,J)
      GOTO 100
   80 IF(KSTATF(I).EQ.0) GOTO 100
      DO 90 J=1,N
   90 ZZ(J)=ZZ(J)-DF(I,J)
  100 CONTINUE
      IF(NACT.EQ.0) GOTO 200
      DO 190 J=1,NACT
      K=KSET(J)
      JN=J+N
      IF(K.GT.L) GOTO 140
      ZZ(JN)=-CLOC(K)
      DO 130 I=1,N
      DZ(I,JN)=DC(K,I)
  130 DZ(JN,I)=DZ(I,JN)
      GOTO 160
  140 KK=K-LNN
      ZZ(JN)=-F(KK)
      DO 150 I=1,N
      DZ(I,JN)=DF(KK,I)
  150 DZ(JN,I)=DZ(I,JN)
  160 DO 170 I=N1,NZ
```

```
  170 DZ(I,JN)=0.0
      T=ASET(J)
      DO 180 I=1,N
  180 ZZ(I)=ZZ(I)-T*DZ(JN,I)
  190 CONTINUE

c     calculate the quasi-newton step

  200 CALL LINSYS(NZ,K,SEPS)
      IF(K.EQ.NZ) GOTO 205
      IFALL=3
      RETURN

c     control step length

  205 XXN1=0.0
      ALFA=1.0
      DO 210 I=1,N
  210 XXN1=XXN1+ZZ(I)**2
      XXN1=SQRT(XXN1)
      IF(XXN1.GT.XXNMAX) ALFA=XXNMAX/XXN1
      IF(ALFA.LT.0.1) IFALL=4

c     will other constraints or functions become active?

      DO 220 J=1,M
      T=0.0
      DO 215 I=1,N
215   T=T+DF(J,I)*ZZ(I)
220   WM(J)=T
      IF(L.GT.0) THEN
                 DO 225 K=1,L
                 T=0.0
                 DO 222 I=1,N
222              T=T+DC(K,I)*ZZ(I)
225              WL(K)=T
      ENDIF
      STEP=1.E16
      IF(LI.EQ.0) GOTO 240
      DO 230 I=LE1,L
      IF(KSTATC(I).NE.0) GOTO 230
      T=WL(I)
      IF(T.GE.-1.E-12) GOTO 230
      T=-CLOC(I)/T
      IF(T.GT.STEP) GOTO 230
      STEP=T
  230 CONTINUE
  240 DO 250 I=1,M
      T=WM(I)
      K=KSTATF(I)
      IF(IABS(K).NE.1.OR.K*T.GE.-1.E-12) GOTO 250
      T=-2.5*F(I)/T
      IF(T.GE.STEP) GOTO 250
      STEP=T
  250 CONTINUE
```

```
      IF(STEP.GT.ALFA) GOTO 275
      IFALL=4
      ALFA=STEP

c     scale the step

  275 DO 280 I=1,NZ
  280 ZZ(I)=ALFA*ZZ(I)
      XXN1=ABS(ALFA)*XXN1

c     calculate function values in the new point

      XN1=0.0
      DO 290 I=1,N
      X1(I)=X(I)+ZZ(I)
  290 XN1=XN1+X1(I)**2
      XN1=SQRT(XN1)
      NCALL=NCALL+1
      CALL FDF(N,M,X1)
      IF(MARK.NE.1)GOTO 560
      IF(NACT.GT.0) THEN
              DO 295 I=1,NACT
  295         ASET1(I)=ASET(I)+ZZ(I+N)
      ENDIF
      IF(L.GT.0) THEN
                DO 305 K=1,L
                T=0.0
                DO 300 I=1,N
300             T=T+DC(K,I)*X1(I)
305             WL(I)=T+C(I)
      ENDIF

c     calculate residuals

      DO 310 I=1,N
      W(I)=0.0
  310 W1(I)=0.0
      DO 330 I=1,M
      IF(KSTATF(I).GE.0) GOTO 320
      DO 315 J=1,N
      W(J)=W(J)-DF(I,J)
  315 W1(J)=W1(J)-DF1(I,J)
      GOTO 330
  320 IF(KSTATF(I).EQ.0) GOTO 330
      DO 325 J=1,N
      W(J)=W(J)+DF(I,J)
  325 W1(J)=W1(J)+DF1(I,J)
  330 CONTINUE
      RES=0.0
      IF(NACT.EQ.0) GOTO 370
      DO 360 J=1,NACT
      K=KSET(J)
      T=ASET1(J)
      IF(K.GT.L) GOTO 340
      DO 335 I=1,N
```

```
      S=T*DC(K,I)
      W(I)=W(I)+S
  335 Wl(I)=Wl(I)+S
      RES=RES+WL(K)**2
      GOTO 360
  340 KK=K-LNN
      DO 350 I=1,N
      W(I)=W(I)+T*DF(KK,I)
  350 Wl(I)=Wl(I)+T*DFl(KK,I)
      RES=RES+Fl(KK)**2
  360 CONTINUE
  370 DO 380 I=1,N
  380 RES=RES+Wl(I)**2
      RES=SQRT(RES)

c     update the hessian approximation

      IF(XXNl.LT.100.0*SEPS*XNl) GOTO 395
      DO 390 I=1,N
  390 Wl(I)=Wl(I)-W(I)
      CALL BFGS(N,SEPS)

c     test if the residual has decreased

  395 IF(RES.LE.0.999*RESREF) GOTO 480
      IF(IFALL.GT.2) RETURN
      IF(XXNl.GT.SSEPS*(XXNMAX+XNl).OR.NSTEP2.LT.2) GOTO 400
      IFALL=2
      IF(XXNl.GT.EPS*XNl) RETURN
      IFALL=1
      IF(NACT.LT.N) RETURN
      IFALL=0
      RETURN
  400 IFALL=5
      RETURN
  480 NSTEP2=NSTEP2+1
      NSTEP=NSTEP+1
      DO 490 I=1,N
      X(I)=Xl(I)
      DO 490 J=1,M
  490 DF(J,I)=DFl(J,I)
      XN=XNl
      XXN=XXNl
      FLl=0.0
      DO 500 I=1,M
      FLl=FLl+ABS(Fl(I))
  500 F(I)=Fl(I)
      RESREF=RES
      IF(NACT.EQ.0) GOTO 520
      DO 510 I=1,NACT
      K=KSET(I)
      S=ASETl(I)
      ASET(I)=S
      IF(K.GT.L) S=ABS(S)-1.0
  510 IF(K.GT.LEQ.AND.S.GT.0.0) IFALL=6
```

```
520 IF(L.EQ.0) GOTO 535
    DO 530 J=1,L
530 CLOC(J)=WL(J)
535 DO 540 I=1,M
    IF(KSTATF(I).GT.0.AND.F(I).LT.-RES) IFALL=7
540 IF(KSTATF(I).LT.0.AND.F(I).GT.RES) IFALL=7
    IF(IFALL.GT.2) RETURN
    IF(XXN.GT.EPS*XN) GOTO 550
    IF(NACT.LT.N) IFALL=1
    RETURN
550 IF(XXN.GT.SEPS*XN.AND.NCALL.LE.MAXF) GOTO 50
560 IFALL=2
    RETURN
    END
```

```
      SUBROUTINE LINSYS(N,NR,SEPS)
      PARAMETER (NX=40,MX=500,LX=40,NX2=NX+NX)
      COMMON/L13/B(NX2),X1(NX),R(NX,NX),B0(NX,NX),W(NX),W0(NX),W1(NX),
     1    ASET(NX),U(NX,NX),KSET(NX),KSET0(NX),KSTATC(LX),KSTATF(MX)
      COMMON/L14/ASET1(NX),WM(MX),DLDF(MX),WL(LX),DLDC(LX),
     1       STEPF(MX),KSTATL(NX),A(NX2,NX2)
      NR=0

      IF(N-1) 200,1,10
    1 IF(ABS(A(1,1)).LT.1.E-12) RETURN
      NR=1
      B(1)=B(1)/A(1,1)
      RETURN
   10 DO 30 I=1,N
      AM=ABS(A(I,1))
      DO 20 J=2,N
      S=ABS(A(I,J))
   20 IF(AM.LT.S) AM=S
      IF(AM.LT.1.E-12) AM=1.0
      B(I)=B(I)/AM
      DO 30 J=1,N
   30 A(I,J)=A(I,J)/AM
      N1=N-1
      DO 80 K=1,N1
      NR=K-1
      AM=ABS(A(K,K))
      I0=K
      K1=K+1
      DO 40 I=K1,N
      S=ABS(A(I,K))
      IF(S.LE.AM) GOTO 40
      AM=S
      I0=I
   40 CONTINUE
      IF(AM.LT.2*K*SEPS) RETURN
      IF(I0.EQ.K) GOTO 60
      DO 50 J=K,N
      S=A(K,J)
      A(K,J)=A(I0,J)
   50 A(I0,J)=S
      S=B(K)
      B(K)=B(I0)
      B(I0)=S
   60 AM=A(K,K)
      DO 80 I=K1,N
      S=A(I,K)/AM
      DO 70 J=K1,N
   70 A(I,J)=A(I,J)-S*A(K,J)
   80 B(I)=B(I)-S*B(K)
      NR=N1
      IF(ABS(A(N,N)).LT.2*N*SEPS) RETURN
      NR=N
      B(N)=B(N)/A(N,N)
      K=N
      DO 100 I=2,N
```

```
      K1=K
      K=K-1
      S=B(K)
      DO 90 J=K1,N
   90 S=S-A(K,J)*B(J)
  100 B(K)=S/A(K,K)
  200 RETURN
      END
```

```
      SUBROUTINE RSOLV(N,KCOL,RIGHT,X)
      PARAMETER (NX=40,MX=500,LX=40,NX2=NX+NX)
      COMMON/L13/XX(NX2),X1(NX),R(NX,NX),B(NX,NX),W(NX),W1(NX),W2(NX),
     1    ASET(NX),U(NX,NX),KSET(NX),KSET0(NX),KSTATC(LX),KSTATF(MX)
      DIMENSION RIGHT(N),X(N)
      K=KCOL
      K1=K+1
   10 IF(K.EQ.0) RETURN
      T=RIGHT(K)
      IF(K1.GT.KCOL) GOTO 30
      DO 20 J=K1,KCOL
   20 T=T-X(J)*R(K,J)
   30 X(K)=T/R(K,K)
      K1=K
      K=K-1
      GOTO 10
      END
```

```
      SUBROUTINE UTRNS(N,KCOL,ACCUM)
      PARAMETER (NX=40,MX=500,LX=40,NX2=NX+NX)
      COMMON/L13/X(NX2),X1(NX),R0(NX,NX),B(NX,NX),W(NX),H(NX),W2(NX),
     1    ASET(NX),U(NX,NX),KSET(NX),KSET0(NX),KSTATC(LX),KSTATF(MX)
      COMMON/L14/R(NX),FUP(MX),DLDF(MX),CUP(LX),DLDC(LX),
     1       STEPF(MX),KSTATL(NX),DSTA2(NX2,NX2)
      LOGICAL ACCUM
      K1=KCOL+1
      IF(ACCUM) GOTO 100
      IF(KCOL.EQ.0) RETURN
      DO 30 KK=1,KCOL
      K=K1-KK
      T=0.0
      DO 10 J=K,N
   10 T=T+U(J,K)*R(J)
      DO 20 J=K,N
   20 R(J)=R(J)-T*U(J,K)
   30 CONTINUE
      RETURN
  100 DO 110 I=1,N
  110 W(I)=R(I)
      DO 130 I=1,N
      T=0.0
      DO 120 J=1,N
  120 T=T+U(I,J)*W(J)
  130 R(I)=T
      RETURN
      END
```

```
      SUBROUTINE UTTRNS(N,KCOL,ACCUM,R)
      PARAMETER (NX=40,MX=500,LX=40,NX2=NX+NX)
      COMMON/L13/X(NX2),X1(NX),RO(NX,NX),B(NX,NX),W(NX),H(NX),W2(NX),
     1    ASET(NX),U(NX,NX),KSET(NX),KSETO(NX),KSTATC(LX),KSTATF(MX)
      DIMENSION R(N)
      LOGICAL ACCUM

      IF(ACCUM) GOTO 100
      IF(KCOL.EQ.0) RETURN
      DO 30 K=1,KCOL
      T=0.0
      DO 10 I=K,N
   10 T=T+R(I)*U(I,K)
      DO 20 I=K,N
   20 R(I)=R(I)-T*U(I,K)
   30 CONTINUE
      RETURN
  100 DO 110 I=1,N
  110 W(I)=R(I)
      DO 130 K=1,N
      T=0.0
      DO 120 I=1,N
  120 T=T+U(I,K)*W(I)
  130 R(K)=T
      RETURN
      END
```

```
c*************************************************************************

c       file name: QUASI2.FOR

c       for programs FETSC and FETSC1 (VAX version).
c       this group of subroutines constitutes the L2 optimization
c       package

c*************************************************************************



        SUBROUTINE PENALTY(N,L,X,V,G)
        PARAMETER (NX=40,MX=500,LX=40)
        DIMENSION X(N),G(NX)
        COMMON/L12/C(LX),DC(LX,NX),CLOC(LX)
        IF(L.LE.0)RETURN
        DO 30 K=1,L
        CLC=C(K)
        DO 10 I=1,N
10      CLC=CLC+DC(K,I)*X(I)
        IF(CLC.LT.0.0)THEN
                V=V+10.0*CLC*CLC
                DO 20 I=1,N
20              G(I)=G(I)+20.0*CLC*DC(K,I)
        ENDIF
30      CONTINUE
        RETURN
        END
```

```
      SUBROUTINE QUASI2(N,M,L,LEQ,X,DX,EPS,MAXF)
      PARAMETER (NX=40,MX=500,LX=40)
      DIMENSION X(N),H(NX,NX),X1(NX),G(NX),G1(NX)
      DIMENSION W(NX),W1(NX),S(NX)
      LOGICAL CONV
      COMMON/L11/MARK,F(MX),DF(MX,NX),F1(MX),DF1(MX,NX)
      NCALL=1
      MARK=1
      DMIN=1.0
      ALP=0.5
      SEPS=1.E-7
      NP=N+1
      N1=N-1
      NN=N*NP/2
      DO 20 I=1,N
      DO 10 J=1,N
10    H(J,I)=0.
20    H(I,I)=1.
      CALL FDF(N,M,X)
      IF(MARK.EQ.0)GOTO 370
      V=0.0
      DO 30 I=1,N
30    G(I)=0.0
      CALL PENALTY(N,L,X,V,G)
      DO 40 J=1,M
      V=V+F(J)*F(J)
      DO 40 I=1,N
40    G(I)=G(I)+2.0*F(J)*DF(J,I)

c     iterative loop
c     find direction S

60    W1(1)=-G(1)
      IF(N.EQ.1)GOTO 90
      DO 80 I=2,N
      I1=I-1
      Z=-G(I)
      DO 70 J=1,I1
70    Z=Z-H(I,J)*W1(J)
80    W1(I)=Z
90    S(N)=W1(N)/H(N,N)
      IF(N.EQ.1)GOTO 120
      N1=N-1
      DO 110 I=1,N1
      I1=N-I
      I2=I1+1
      Z=0.
      DO 100 J=I2,N
100   Z=Z+H(I1,J)*S(J)
110   S(I1)=W1(I1)/H(I1,I1)-Z

c     check the directional gradient

120   GS=0.
      DO 130 I=1,N
```

```
130     GS=GS+S(I)*G(I)
        IF(GS.GE.-SEPS)GOTO 350
140     GS0=GS
        ALAMDA=-2.*V/GS
        IF(ALAMDA.GT.1.)ALAMDA=1.
        TOT=0.
        VDIFF=V

c       small loop, line search

150     DXX=0.0
        DO 160 I=1,N
        DX=ALAMDA*S(I)
        IF(ABS(DX).GT.DXX)DXX=ABS(DX)
160     X1(I)=X(I)+DX
        CALL FDF(N,M,X1)
        NCALL=NCALL+1
        V1=0.0
        DO 170 I=1,N
170     G1(I)=0.0
        CALL PENALTY(N,L,X1,V1,G1)
        DO 180 J=1,M
        V1=V1+F(J)*F(J)
        DO 180 I=1,N
180     G1(I)=G1(I)+2.0*F(J)*DF(J,I)

        CONV=MARK.EQ.0.OR.V1.LT.SEPS.OR.NCALL.GE.MAXF.OR.DXX.LE.EPS
        IF(CONV)GOTO 350

        GYS=0.
        DO 190 I=1,N
190     GYS=GYS+S(I)*G1(I)
        IF(V1.GE.V)GOTO 220
        IF(ABS(GYS/GS0).LE.0.9)GOTO 230
        IF(GYS.GT.0.)GOTO 220

c       linear extrapolation

        TOT=TOT+ALAMDA
        Z=10.
        IF(GS.GE.GYS)GOTO 200
        Z1=10.0*ABS(GS-GYS)
        Z2=ABS(GYS)
        IF(Z2.GT.Z1)GOTO 200
        Z=10.0*Z2/Z1
200     ALAMDA=ALAMDA*Z
        V=V1
        GS=GYS
        DO 210 I=1,N
210     X(I)=X1(I)
        GOTO 150

c       cubic interpolation

220     Z=3.*(V-V1)/ALAMDA+GYS+GS
```

```
      ZZ=SQRT(Z*Z-GS*GYS)
      GZ=GYS+ZZ
      Z=1.0-(GZ-Z)/(ZZ+GZ-GS)
      ALAMDA=ALAMDA*Z
      GOTO 150

c     line search completed

230   ALAMDA=ALAMDA+TOT
      V=V1
      VDIFF=VDIFF-V
      DGS=GYS-GS0
      LINK=1

c     updating H

      IF(DGS+ALAMDA*GS0.GT.0.)GOTO 250
      DO 240 I=1,N
240   S(I)=G1(I)-G(I)
      SIG=1./(ALAMDA*DGS)
      GOTO 270

250   ZZ=ALAMDA/(DGS-ALAMDA*GS0)
      Z=DGS*ZZ-1.
      DO 260 I=1,N
260   S(I)=Z*G(I)+G1(I)
      SIG=1./(ZZ*DGS*DGS)
270   W1(1)=S(1)
      DO 290 I=2,N
      I1=I-1
      Z=S(I)
      DO 280 J=1,I1
280   Z=Z-H(I,J)*W1(J)
290   W1(I)=Z
      DO 300 I=1,N
      Z=H(I,I)+SIG*W1(I)*W1(I)
      IF(Z.LE.0.)Z=DMIN
      IF(Z.LT.DMIN)DMIN=Z
      H(I,I)=Z
      W(I)=W1(I)*SIG/Z
300   SIG=SIG-W(I)*W(I)*Z
      N1=N-1
      DO 310 I=1,N1
      I1=I+1
      DO 310 J=I1,N
      S(J)=S(J)-H(I,J)*W1(I)
      H(I,J)=H(I,J)+W(I)*S(J)
310   H(J,I)=H(I,J)
      IF(LINK.EQ.2)GOTO 330
      LINK=2
      DO 320 I=1,N
320   S(I)=G(I)
      SIG=1./GS0
      IF(DGS+ALAMDA*GS0.GT.0.)SIG=-ZZ
      GOTO 270
```

```
c       Updating of H completed

330     V=V1
        DO 340 I=1,N
        X(I)=X1(I)
340     G(I)=G1(I)
        GOTO 60
350     IF(V1.GE.V)GOTO 370
        DO 360 I=1,N
360     X(I)=X1(I)
370     MAXF=NCALL
        RETURN
        END
```