BNL-104689-2014-TECH

AGS/AD/Tech Note No. 271;BNL-104689-2014-IR

Guide to Accelerator Physics Program Synch - CDC Version

Z. Parsa

January 1987

Collider Accelerator Department

Brookhaven National Laboratory

U.S. Department of Energy

USDOE Office of Science (SC)

DISCLAIMER

GUIDE to ACCELERATOR PHYSICS PROGRAM SYNCH

– CDC VERSION –

[This is an on line manual in]

BNLDAG::DUA0:[PARSA1.SYNCH]

Accelerator Division
Technical Note No. 271

January 12, 1987

ZOHREH PARSA
&
ERNEST COURANT

BROOKHAVEN NATIONAL LABORATORY

ACCELERATOR DEVELOPMENT DEPARTMENT

GUIDE to

ACCELERATOR PHYSICS

PROGRAM SYNCH

CDC Version

ZOHREH PARSA

and

ERNEST COURANT

BROOKHAVEN NATIONAL LABORATORY

Associated Universities, INC.
Upton, New York 11973

[January 1987]

This Guide is written to accommodate users of Accelerator Physics Data Base BNLDAG::DUA0:[PARSA1]. It describes the contents of the on line Accelerator Physics data base DUA0:[PARSA1.SYNCH]. SYNCH is a computer program used for the design and analysis of synchrotrons, storage rings and beamlines.

TABLE OF CONTENTS

I. INTRODUCTION

This guide describes the contents of the [PARSA1.SYNCH] Directory. In section II, the Directory listing of the files, and [PARSA1.SYNCH]READ.ME are shown.

You may note that, throughout the Data Base, READ.ME files provide you with detail information about the programs in the selected program directory.

Section III, shows [PARSA1.SYNCH]SYNCH.FOR, the source program.

SECTION II

A) Following are the contents of the [PARSA1.SYNCH] Directory:

```
$ DIR [PARSA1.SYNCH]

Directory DUAO:[PARSA1.SYNCH]

CDCDEMO.DAT;1        CDCSHOW.COM;1
CDCSYNCH.DAT;1       DO.LOGA;1
FNLSYNCH.EXE;2       FOR011.DAT;1
FOR012.DAT;1         INPUT.DIR;1
OUTPUT.DIR;1         READ.ME;12
SOURCE.DIR;1         SYN.COM;4
SYNBOOST10.DAT;1     SYNBOOST9.DAT;1
SYNCH.COM;2          SYNCH.EXE;52
SYNCH.EXE;47         SYNCH.GUIDE;1
SYNCH.OUT;6          SYNCH.RNO;1
SYNCHINC.TLB;5       VAXSYN.COM;2
VAXSYN.COM;1         VAXSYNBST9.DAT;3

Total of 24 files.
```

B) $ DIR [PARSA1.SYNCH.SOURCE]

```
Directory DUAO:[PARSA1.SYNCH.SOURCE]

BKLYIII.DIF;2        BNLSYNIII.FOR;42
CYX.POL;1            FROM.ARDITH;1
III.DIF;2            IIIDIF.FOR;42
MAINS.FOR;8          SYNCHINC.TLB;5
SYNDEMO.COM;2        SYNDEMO.DAT;1
SYNDEMO.OUT;1        SYNI.FOR;39
```

```
SYNII.FOR;27        SYNIII.FOR;42
SYNIIIA.FOR;6       SYNRHIC.DAT;1
VAXSYNCH.INF;2

Total of 17 files.
```

C) $ DIR [PARSA1.SYNCH.INPUT]

D) Following are the contents of [PARSA1.SYNCH]READ.ME:

```
     Welcome to DUAO:[PARSA1.SYNCH]

 LOG

 SEPTEMBER 1985    ZP      SET UP
 OCTOBER 1985              UPDATE  NEW VAX-VERSION
 MAY      1986             UPDATE CDC-VERSION
 JANUARY 1987              UPDATE CDC-VERSION

 GENERAL

     Synch is a computer program for use in the
 design and analysis of synchrotrons, storage rings
 and beamlines.  The program has modular organization,
 so the new capabilities can be added.


  HOW TO RUN THE CDC - VERSION OF SYNCH FROM VAX
  ----------------------------------------------

  INPUT:
      To run a SYNCH data file from the VAX on the CDC, you
   must  include 1) The normal job cards (see example), 2)
   The synch input data file.

       Hence,i) create a file and call it SYNCH.DAT on VAX,
    ii) use the CYBERVAX to run it on CDC.  So, [while on the
    VAX] type after the VAX $ prompt

       $ MSUB SYNCH.DAT<cr>

       [See Cybervax Guide for other commands]

   OUTPUT:

   I.  When the job is finished on CDC, you will
       receive a mail message on Vax [that returns
       the output from CDC to VAX].

    EXAMPLE:  (you are on vax and the prompt is $)
```

```
$ edit Synch.dat
ZNAME,STMFZ.    [your cdc outputfile will be ZNAME plus
                                          few more characters]
ACCOUNT,yourname,123,123.     [123 is your problem number]
COPYSP.
REWIND,INPUT.
ATTACH,MN,BSM85,ID=EDC.
ATTACH,L,BSL85,ID=EDC.
ATTACH,L1,PRINTLIB.
ATTACH,L2,CALCOMPLIB.
LIBRARY,L,L1,L2.
MAP,OFF.
MODE,1.
MN.
%%EOR
.
.
"SYNCH data file"
.
.
$
```

[To compute depolarization resonance strength, you may use the programs POLARIZ, (which can give the plot of the result, but is limited to maximum of about 100 resonances), or POLRIZE (which can't give a plot of the result but can find as many resonances as you want). e.g. touse POLARIZE, you must add, and put the followng cards before %%EOR above.

Following are the contents of cdcdemo.dat:

```
ty cdcdemo.dat
...
...
ATTACH,POLRIZE,ID=ZP.
POLRIZE.
REWIND,TAPE20.
COPY,TAPE20.
%%EOR
.
.
  "SYNCH DATA FILE"
.
.

$                  [now you have created your SYNCH.DAT on vax]
$MSUB SYNCH.DAT    [Via cybervax you submit synch.dat and run
                    the program synch on cdc]
$ YOU HAVE MAIL    [you will receive a mail message when the cdc
                    output is completed and returned to the VAX]
```

VAX VERSION INFO.

Please note, that in this dir. we have a [FORTRAN (.FOR)], copy of SYNCH, that can run on VAX, (by using:

```
$FOR SYNI, $FOR SYNII, $FOR SYNIIIA, $FOR MAINS you make
 SYNI.OBJ, SYNII.OBJ, SYNIIIA.OBJ, MAINS.OBJ; then use
$LINK/EXECUTABLE=SYNCH.EXE SYNI,SYNII,SYNIIIA,MAINS and
 then,
$RUN SYNCH.
```

[Without the /EXECUTABLE=SYNCH.EXE, the LINK command would produce the same executable copy but would call it SYN1.EXE).

In addition you are given the information to run the synch on CDC (i.e.the CDC version of SYNCH) from vax through the CDC link. This allows you to edit your input data file on VAX, yet use the CDC version of the program (and the benefit of the CDC speed for running the program).

II. HOW TO USE THE VAX VERSION OF PROGRAM SYNCH:

```
GET SOURCE PROGRAMS
  MAINS.FOR
  SYNI.FOR
  SYNII.FOR
  SYNIIIA.FOR (or use SYNIII.FOR for synch with plot)
  SYNCHINC.TLB
```

Compile MAINS,SYNI,SYNII,SYNIII. SYNCHINC.TLB is a table of the common blocks which are needed to satisfy the 'INCLUDE' STATEMENTS in the program. you never need to refer to SYNCHINC.TLB, But it has to be available for the compiler.

IF compiled OBJ files have the same names, then
LINK/EXECUTABLE = SYNCH.EXE MAINS,SYNI,SYNII,SYNIIIA

Our current executable file that is equivalent to all of this is SYNCH.EXE.

A current test case is SYNDEMO.DAT and its output is SYNDEMO.OUT.

HOW TO RUN THE VAX VERSION OF SYNCH

You may run the vax version of SYNCH (with or without ploting) interactively by creating the following SYNCH.COM file. [Note that, you may add other DCL commands to the following example of SYNCH.COM file]

```
$ ASSIGN SYNCH.DAT FOR002
$ ASSIGN DYNCH.OUT FOR003
$ RUN DUAO:[PARSA1.SYNCH]SYNCH.EXE
```

[Note, SYNCH.EXE, is the executable copy of SYNCH which can be obtained following the above description; for both cases, with or without the ploting options. additionally, the executable copies are provided for your convenience.]

SYNCH.DAT contains the input statements starting with a RUN command and ending with a STOP statement, (see example of input file; SYNDEMO.DAT). The output from the RUN goes to SYNCH.DAT, (for our example, SYNDEMO.OUT]

Having created the SYNCH.COM file, enter:

1. @SYNCH <return> [for interactiv run]

   or to batch the job enter:

2. $ SUBMIT/NOPRINT SYNCH <return> [here you are submitting SYNCH.COM file]

WORK FILES:

[PARSA1.SYNCH]READ.ME

[PARSA1.SYNCH]SYNCH.FOR

[PARSA1.SYNCH]SYNCH.EXE

[PARSA1.SYNCH]SYNCH.COM

READ.ME; gives general information about this directory and the program SYNCH.

SYNCH.FOR; is the source code from which SYNCH.EXE was obtained and a copy of it is given for convenience.

Following are the contents of [PARSA1.SYNCH]*.com files:

```
DUAO:[PARSA1.SYNCH]SYN.COM;4

$  ASSIGN [PARSA1.SYNCH]SYNBOOST9.dat FOR002
$! SYNBOOST9.DAT is a sample VAX input data.
$  ASSIGN SYNCH.OUT FOR003
$! ASSIGN NL: FOR012
$  ASSIGN DEBUG.DAT FOR001
$  RUN   [PARSA1.SYNCH]SYNCH
-------------------------------------

DUAO:[PARSA1.SYNCH]SYNCH.COM;2

$  ASSIGN SYNCH.DAT FOR002
$  ASSIGN SYNCH.OUT FOR003
$! ASSIGN NL: FOR012
$  ASSIGN DEBUG.DAT FOR001
$  RUN SYNCH
--------------------------------------

DUAO:[PARSA1.SYNCH]VAXSYN.COM;2

$  ASSIGN [PARSA1.SYNCH]VAXSYNBST9.dat FOR002
$! VAXSYNBST9.DAT is a sample VAX input data.
$  ASSIGN [PARSA1.SYNCH]SYNCH.OUT FOR003
$! ASSIGN NL: FOR012
$  ASSIGN DEBUG.DAT FOR001
$  RUN   [PARSA1.SYNCH]SYNCH.EXE;47
-----------------------------------------

DUAO:[PARSA1.SYNCH]VAXSYN.COM;1

$  ASSIGN [PARSA1.SYNCH]VAXSYNBST9.dat FOR002
$! VAXSYNBST9.DAT is a sample VAX input data.
$  ASSIGN [PARSA1.SYNCH]SYNCH.OUT FOR003
$! ASSIGN NL: FOR012
$  ASSIGN DEBUG.DAT FOR001
$  RUN   [PARSA1.SYNCH]SYNCH
-------------------------------------------
```

Following are the contents of [PARSA1.SYNCH]*.DAT;*:

DUAO:[PARSA1.SYNCH]CDCDEMO.DAT;1

```
ACCOUNT,YOURNAME,123,123.           (123=YOUR PROB  )
COPYSP.
REWIND(INPUT)
REQUEST(TAPE11,*PF)
REQUEST(TAPE20,*PF)
FILE(TAPE20,RT=W,MBL=5120,BT=I)
REQUEST(L,*PF)
REQUEST(NN,*PF)
ATTACH,MN,BSM85,ID=EDC.
ATTACH,L,BSL85,ID=EDC.
ATTACH(L1,PRINTLIB)
ATTACH(L2,CALCOMPLIB)
LIBRARY(L,L1,L2)
MAP(OFF)
MODE(1)
MN.
ATTACH,POLRIZE,ID=ZP.
POLRIZE.
REWIND,TAPE20.
COPY,TAPE20.
%%EOR
 AGSP  RUN         AGS WITH ERRORS FOR POLARIZATION CALCULATIONS
 BR    =           981.44
 BZ    =           11.4952
 TUNE  SUB
 GFX1  =           7.5
 GDX1  =           -7.5
 GFX2  =           7.5
 GDX2  =           -7.5
 GF    =           47.421958
 GD    =           -47.573705
 S2    DRF         .6096
 S2.5  DRF         .762
 S10   DRF         3.048
 FS    MAG         2.0066    GF        BR        BZ        $
 DS    MAG         2.0066    GD        BR        BZ        $
 FL    MAG         2.3876    GF        BR        BZ        $
 DL    MAG         2.3876    GD        BR        BZ        $
 S5    DRF         1.524
 HPER  BML         S2.5 FL   S2   DL   S5   DL   S2   FS   S2   FS   S10
DS
                   S2   DS   S2   FL   S5   FL   S2   DL   S2.5
 HP2   BML   -1    HPER
 MP    MMM         HPER HP2
       END
 MUX   =           .716666667
 MUY   =           .733333333
 FQ    FITQ        TUNE MP   GF   GD        1    1MUX      MUY
 "P    DRF         0.
```

```
 .AGS  BML         HPER HP2
 V     PVEC
 DX    =           .0002
 DY    =           .0002
 "M    DRF
 K     KICK   1    "M                 1.
 S1    DRF         .3048
 S5K   BML         S2.5 K    S2.5
 S2M   BML         S1   "M   S1
 SX    DRF         .262
 S5MX1 BML         SX   "M   DXE1 SX
 S5X1  BML         SX   FXE1 SX
 S5MX2 BML         SX   "M   FXE2 SX
 S5X2  BML         SX   DXE2 SX
 S5M   BML         S2.5 "M   S2.5
 S2K   BML         S2   K
 DG    =           .1
 GFE   RAND        GF        DG
 GDE   RAND        GD        DG
 FSE   MAG         2.0066    GFE       BR        BZ        $
 DSE   MAG         2.0066    GDE       BR        BZ        $
 FLE   MAG         2.3876    GFE       BR        BZ        $
 DLE   MAG         2.3876    GDE       BR        BZ        $
 .HP   BML         S2.5 FLL  S2K  DLL  S5M  DLL  S2   FSS  S2   FSS  S10
DSS
                   S2M  DSS  S2K  FLL  S5   FLL  S2K  DLL  S2.5
 .HPX1 BML         S2.5 FLL  S2K  DLL  S5MX1DLL  S2   FSS  S2   FSS  S10
DSS
                   S2M  DSS  S2K  FLL  S5X1 FLL  S2K  DLL  S2.5
 .HPX2 BML         S2.5 FLL  S2K  DLL  S5MX2DLL  S2   FSS  S2   FSS  S10
DSS
                   S2M  DSS  S2K  FLL  S5X2 FLL  S2K  DLL  S2.5
 .HP1X BML    -1   .HPX1
 .HP2X BML    -1   .HPX2
 .HPP  BML    -1   .HP
 .AGX  BML         .HP  .HP1X.HP  .HP2X.HP  .HP2X.HP  .HP1X
 .AGP  BML         .HP  "M   .HPP
 .AGM  BML          3( .AGX     )
       RAND 999 111
 FXE1  MAG         1.0000    GFX1      BR
 DXE1  MAG         1.0000    GDX1      BR
 FXE2  MAG         1.0000    GFX2      BR
 DXE2  MAG         1.0000    GDX2      BR
 FSS   MOVE      1 FSE                 DX                  DY
 DSS   MOVE      1 DSE                 DX                  DY
 FLL   MOVE      1 FLE                 DX                  DY
 DLL   MOVE      1 DLE                 DX                  DY
 AGI   FXPT   -2 1 V    .AGM

 AGF   FXPT   -2 1 V    .AGM                          1

       ORBC   1 36 AGF  AGF  "M   K    .0003     .0003
 AGU   FXPT   -2 1 V    .AGM

       FIN
```

```
      STOP
%%EOR
 AGS, INTRINSIC     1.79275   10.
    0   60    1
                 -1
 AGS,  .1 MM ERRORS 1.79275   10.
    0   60    1    1
                 -1
 AGS,  .1 MM ERRORS 1.79275   10.
    0   60    1    1
%%EOR
-----------------------------------------
```

DUAO:[PARSA1.SYNCH]CDCSYNCH.DAT;1

```
ZNAME,STMFZ.
ACCOUNT,PARSA,123,123.
COPYSP.
REWIND,INPUT.
ATTACH,MN,BSM85,ID=EDC.
ATTACH,L,BSL85,ID=EDC.
ATTACH,L1,PRINTLIB.
ATTACH,L2,CALCOMPLIB.
LIBRARY,L,L1,L2.
MAP,OFF.
MODE,1.
MN.
%%EOR
```

III. Following are the contents of the [parsa1.synch]cdcsynch.for

[this is the CDC version of prog. SYNCH.FOR].

```
      OVERLAY(SYNCH,0,0)                                                 MAIN        2
      PROGRAM SYNCH(INPUT=101,OUTPUT=1001,TAPE2=INPUT,TAPE3=OUTPUT,      MAIN        3
     1 TAPE5, TAPE6, TAPE11,TAPE12, TAPE98,TAPE99)                       MY31MAIN    1
                                                                         MAIN        5
C     SYNCH-- A COMPUTER SYSTEM FOR SYNCHROTRON DESIGN AND ORBIT         MAIN        6
*  ANALYSIS.  BY A.A.GARREN AND A.S.KENNEY, LAWRENCE BERKELEY LAB.       MAIN        7
                                                                         MAIN        8
                                                                         MAIN       11
C                                                                        MAIN       12
      COMMON/CLC/REG(109)                                                CLC         2
                                                                         BLANK       2
      LEVEL 2, STORE,INFF,IWORK                                          BLANK       3
      COMMON STORE(48000),IWORK(10)                                      86MARSIZ    1
      DIMENSION INFF(24,2000)                                            86MARSIZ    2
      EQUIVALENCE (INFF,STORE)                                           86MARSIZ    3
                                                                         BLANK       5
      LEVEL 2, XX,Y1,Y2,Y3,Y4                                            BETPTS      2
      COMMON XX(4000),Y1(4000),Y2(4000), Y3(4000),Y4(4000)               RENAM83     1
      COMMON/BPLTCOM/MN,KW,BXX,BYX,NPLT                                  RENAM83     2
      COMMON/CCPOOL/XMIN,XMAX,YMIN,YMAX,CCXMIN,CCXMAX,CCYMIN,CCYMAX      PL6683      1
      COMMON/CCFACT/FACTOR                                               PL6683      2
                                                                         BETPTS      5
      COMMON/INSTR/OPNAME,NAME,OP,KA,KB,OBJA,OBJB,NXTM,TRA,LQ,NQ,LIN,    INSTR       2
     1            NIN,LFL,NFL,LBC,NBC,NTOT,IX,KIND,TYPE,ROWS,COLS,EXTR,  INSTR       3
     2            M,NF,NB,NI,MSUBR,ISIGN                                 INSTR       4
      INTEGER OPNAME,OP,OBJA,OBJB,TRA,TYPE,ROWS,COLS,EXTR                INSTR       5
                                                                         INSTR       6
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,     CONTRL      2
     1         LDFLG,FIN                                                 CONTRL      3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN       CONTRL      4
                                                                         CONTRL      5
      COMMON/STORE/LMAX,LINF,LFILE,MAX,MIN,ICARD(11)                     STORE       2
*  LMAX AND MAX ARE DIMENSIONS OF STORE AND INFF                         STORE       3
                                                                         STORE       4
      COMMON/GRR/IERR,BSW,VSW,RSW,QSW,XSW                                GRR         2
      LOGICAL IERR,BSW,VSW,RSW,QSW,XSW                                   GRR         3
                                                                         GRR         4
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL    DIM         2
                                                                         DIM         3
      COMMON/BMI/MI(5000)                                                BMIL        1
      COMMON/NELS/NELS                                                   BMIL        2
      LEVEL 2,MI                                                         BMIL        3
                                                                         BMIL        4
                                                                         BMI         3
      COMMON/BMI1/MI1(16000)                                             NV3BMI1     1
      COMMON/NELS1/NELS1                                                 BMI1L       2
      LEVEL 2,MI1                                                        BMI1L       3
                                                                         BMI1L       4
C                                                                        BMI1        3
      COMMON /MATCH3/NOPR,NONU,FLAG,IPR,IPP,MINFLG,MIFLG,MATFLG,GLOBAL   MATCH3      2
      LOGICAL FLAG,NOPR,NONU,MIFLG,MATFLG,GLOBAL                         MATCH3      3
      COMMON /MATCH2/ AA(30),BB(30),KBTS,SMINCAL,                        MATCH2      2
     1        DWORD(10),DWORD2(10),DWORD3(10),DWORD7(7,10),MINCOM        MATCH2      3
      LOGICAL SMINCAL                                                    MATCH2      4
      COMMON/FLTN/IFL(15)                                                FLTN        2
```

```
                                                                         FLTN        3
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER    SWTCH       2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                           SWTCH       3
                                                                         SWTCH       4
      COMMON/BCFD/PAR(10),RMUX,RMUY,XGES,YGES,LAM                        BCFD        2
C                                                                        BCFD        3
      COMMON/TRKINT/G,EX,EY,OMSQ,SEND,VMX,VMN,DINT,BEG,DPR,JPR,          TRKINT      2
     1   THET,PLT,NPT,NZ,PMAX,NSIZ,ITITLE(6)                             TRKINT      3
      LOGICAL BEG,DPR,PLT                                                TRKINT      4
      COMMON/TSW/TRSW,MCY(20),JM,LOCC                                    NOV3TSW     1
      LOGICAL TRSW                                                       TSW         3
      COMMON/OPLIST/LIST(200)                                            OPLIST      2
                                                                         OPLIST      3
      COMMON/SVNAM/NAMRUN                                                SVNAM       2
      COMMON/COPY/CPYSW                                                  COPY        2
      LOGICAL CPYSW                                                      COPY        3
                                                                         COPY        4
      COMMON/CORB/TH(84),TV(84),NMON,NNH,NNV,IIH,IIV,                    CORB613     1
     .             NAMONH,NAMONV,NACORH,NACORV                           CORB613     2
      DIMENSION IC(24)                                                   MAIN       33
      EQUIVALENCE (IC(1),OPNAME)                                         MAIN       34
      DIMENSION IFM(15),IW(6)                                            MAIN       35
      COMMON/KINET/KNFLAG,TK,P,GAM,BETT,BETGAM,BRHO                      KINET       2
      LOGICAL KNFLAG                                                     KINET       3
      INTEGER HREM,HPAGE                                                 MAIN       37
      DATA DWORD(1),DWORD2(1),DWORD3(1)/4HPRIN,4HTOUT,2H  /              MAIN       39
      DATA DWORD(2),DWORD2(2),DWORD3(2)/4HMINI,4HMIZE,2H  /              MAIN       40
      DATA DWORD(3),DWORD2(3),DWORD3(3)/4HEND ,4HRETU,2HRN/              MAIN       41
*                    TO ALTER STORAGE, CHANGE LMAX,MAX AND DIMENSION STORE MAIN     42
*     TO INCREASE MATRIX SCRATCH SPACE, CHANGE ISAV AND ISAV7 FOR        MAIN       43
* 3X3 OR 7X7 SIZES.                                                      MAIN       44
      DATA LMAX,MAX,ISAV,ISAV7,KADD,KADDR/48000,2000,10,12,3,7/          86MARSIZ    4
      DATA MASK1 /77000000000000000000B/                                 MAIN       46
      DATA (IW(I),I=1,6) /0,0,0,-1,-1,-1/                                MAIN       47
      DATA (IFM(J),J=1,15)/1H-,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,      MAIN       48
     1  1H0,1H+,1H ,1H.,1H//                                             MAIN       49
      DATA REG/109*0.0/                                                  MAIN       50
      DATA MINZER/77777 77777 77777 77777B/                              MAIN       51
      CALL SYSTEMC( 3,IW)                                                MAIN       52
      CALL SYSTEMC( 6,IW)                                                MAIN       53
      CALL SYSTEMC(13,IW)                                                MAIN       54
      CALL SYSTEMC(30,IW)                                                MAIN       55
      CALL SYSTEMC(36,IW)                                                MAIN       56
      CALL SYSTEMC(39,IW)                                                MAIN       57
      CALL SYSTEMC(42,IW)                                                MAIN       58
      CALL RANGET(RS)                                                    MAIN       59
      CALL RANSET(RS)                                                    MAIN       60
      NPLT=0                                                             MAIN       61
      DD=RANF(DD)                                                        MAIN       62
      DO 10 I=1,84                                                       MN9JN83     1
      TH(I) = 0.                                                         MN9JN83     2
   10 TV(I) = 0.                                                         MN9JN83     3
C                                                                        MAIN       63
      CALL SETOPS                                                        MAIN       64
                                                                         MAIN       65
    9 FIN = .FALSE.                                                      MAIN       66
      MINFLG=1                                                           MAIN       67
```

```
      NONU = .FALSE.
      NOPR = .FALSE.
      MIFLG = .FALSE.
      MATFLG = .FALSE.
      GLOBAL = .FALSE.
      LFILE = 1
      LDFLG=.FALSE.
      KNFLAG = .FALSE.
      EMPTY = .FALSE.
      IERR=.FALSE.
      SMINCAL=.FALSE.
      CPYSW = .TRUE.
      DO 1 I=1,7
      DO 1 J=1,10
    1 DWORD7(I,J)=0.00000001
      MINCOM=3
      NOPR=.FALSE.
      NONU=.FALSE.
      LEND=LMAX
      MEND=MAX
      MIN=MAX
      M7END = MEND - ISAV
*          FILL BLANK PORTION OF ARRAY WITH ZEROES.
*   IFL IS USED MAINLY TO DISTINGUISH SYMBOLIC FL.PT. FROM REAL FL.PT.
      DO 8 J=1,15
      IFL(J) = MASK1.AND.IFM(J)
    8 CONTINUE
    2 CALL RDINST

    4 CALL SWITCH
      IF (FIN) GO TO 9
      IF(ERROR) GO TO 6
C  CHECK FOR REDEFINITION OF ELEMENT BY NAME
      IF (MODE.EQ.1.OR.MODE.EQ.2) CALL NAMCHK(M)
      IF(MODE.EQ.3) GO TO 3
      IF (.NOT.MSSW) GO TO 7
      CALL RDINF(IC,M)
      CALL MESH(M,KA,OPNAME)
      IF (MSSW) GO TO 3
C   IF PAGE OR COMMENT OUTSIDE SUBROUTINE, REUSE INFF SPACE.
    7 IF ((OPNAME.EQ.HPAGE.OR.OPNAME.EQ.HREM).AND.(MODE.EQ.1)) GO TO 2
      MIN = MIN - 1
      NXTM=MIN
      CALL INFW( MIN,8,M )
      M=MIN
      GO TO 2
    3 NXTM = INFF( 8,M )
      M=NXTM
      IF(.NOT.TRASW) GO TO 5
      M=TRA
      TRASW=.FALSE.
    5 CALL RDINF( IC,M )
      GO TO 4
    6 CALL ERRSR(M)
      GO TO 9
      END
```

```
      SUBROUTINE SETPLT                                                 PLT       7
      INTEGER FR80(2),ITEK(4),C835(2)                                   PLT       8
                                                                        BLANK     2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK     3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ  1
      DIMENSION INFF(24,2000)                                           86MARSIZ  2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ  3
                                                                        BLANK     5
      LEVEL 2, XX,Y1,Y2,Y3,Y4                                           BETPTS    2
      COMMON XX(4000),Y1(4000),Y2(4000), Y3(4000),Y4(4000)              RENAM83   1
      COMMON/BPLTCOM/MN,KW,BXX,BYX,NPLT                                 RENAM83   2
      COMMON/CCPOOL/XMIN,XMAX,YMIN,YMAX,CCXMIN,CCXMAX,CCYMIN,CCYMAX     PL6683    1
      COMMON/CCFACT/FACTOR                                              PL6683    2
                                                                        BETPTS    5
      COMMON/TRKINT/G,EX,EY,OMSQ,SEND,VMX,VMN,DINT,BEG,DPR,JPR,         TRKINT    2
     1   THET,PLT,NPT,NZ,PMAX,NSIZ,ITITLE(6)                            TRKINT    3
      LOGICAL BEG,DPR,PLT                                               TRKINT    4
      COMMON/SVNAM/NAMRUN                                               SVNAM     2
      DATA CCXMIN,CCXMAX,CCYMIN,CCYMAX,FACTOR/70.,1070.,80.,1080.,1./   PLT      13
      C835(1)=10HSYNCH PLOT                                             PLT      14
      C835(2)=NAMRUN                                                    PLT      15
      FR80(1)=1 $ FR80(2)=4                                             PLT      16
      ITEK(1)=120 $ ITEK(2)=1 $ ITEK(3)=1 $ ITEK(4)=1024                PLT      17
      CALL PLOTDVC(10,99,C835(1),ITEK(1),FR80(1),PRINT)                 PLT      18
      CALL CALINIT                                                      PLT      19
      RETURN                                                            PLT      20
      END                                                               PLT      21
```

```
      SUBROUTINE FINPL                                          PLT      22
      CALL FIN                                                  PLT      23
      RETURN                                                    PLT      24
      END                                                       PLT      25
```

```
      SUBROUTINE CCLTR(X,Y,KOR,KS,KBCD,NC)                      PLT   26
      DIMENSION KBCD(7)                                         PLT   27
      NCH=10                                                    PLT   28
      CALL NOARG(NA)                                            PLT   29
      IF(NA.GT.4) GO TO 2                                       PLT   30
      REWIND 98                                                 PLT   31
      READ(98,10) KTX                                           PLT   32
    5 KBCD(1)=KTX                                               PLT   33
   10 FORMAT(A10)                                               PLT   34
      REWIND 98                                                 PLT   35
    2 ANG=KOR*90.                                               PLT   36
      HT=KS/10.                                                 PLT   37
      IF(KS.LE.1) HT=.15                                        PL2    1
      XZ=.01*X  $ YZ= .01*Y                                     PLT   38
      IF(NA.EQ.6) NCH=NC                                        PLT   39
      CALL SYMBOL(XZ,YZ,HT,KBCD,ANG,NCH)                        PLT   40
      RETURN                                                    PLT   41
      END                                                       PLT   42
```

```
      SUBROUTINE CCPLOT(X,Y,N,KBCD,NS,NTH,IS)                            PRCH       1
                                                                         BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                          BLANK      3
      COMMON STORE(48000),IWORK(10)                                      86MARSIZ   1
      DIMENSION INFF(24,2000)                                            86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                           86MARSIZ   3
                                                                         BLANK      5
      LEVEL 2, XX,Y1,Y2,Y3,Y4                                            BETPTS     2
      COMMON XX(4000),Y1(4000),Y2(4000), Y3(4000),Y4(4000)               RENAM83    1
      COMMON/BPLTCOM/MN,KW,BXX,BYX,NPLT                                  RENAM83    2
      COMMON/CCPOOL/XMIN,XMAX,YMIN,YMAX,CCXMIN,CCXMAX,CCYMIN,CCYMAX      PL6683     1
      COMMON/CCFACT/FACTOR                                               PL6683     2
                                                                         BETPTS     5
      DIMENSION KBCD(1),X(1),Y(1)                                        PLT       46
      CALL INTENSE(1H*)                                                  PRCH       2
      DX=(XMAX-XMIN )*100./(CCXMAX-CCXMIN)                               PLT       47
      DY=(YMAX-YMIN )*100./(CCYMAX-CCYMIN)                               PLT       48
      X(N+1)=XMIN-.01*CCXMIN*DX                                          PLT       49
      Y(N+1)=YMIN-.01*CCYMIN*DY                                          PLT       50
      X(N+2)=DX                                                          PLT       51
      Y(N+2)=DY                                                          PLT       52
      ISYM=3                                                             PLT       53
      J=0                                                                PLT       54
      K=1                                                                PLT       55
      CALL NOARG(NA)                                                     PLT       56
      IF(NA.LE.4) GO TO 1                                                PLT       57
      J=1                                                                PLT       58
      IF(KBCD(1).EQ.6HNOJOIN) J=-1                                       PLT       59
      IF(NA.EQ.6) J=J*NTH                                                PLT       60
    1 CONTINUE                                                           PLT       61
      IF(N.GT.2) GO TO 3                                                 PL2        2
      CALL LINE(X,Y,N,K,J,ISYM)                                          PLT       62
      RETURN                                                             PLT       63
    3 IF(NA.EQ.7) CALL INTENSE(IS)                                       PRCH       3
      CALL LINE(X,Y,N,1,0,3)                                             PRCH       4
      RETURN                                                             PL2        4
      END                                                                PLT       64
```

```
      SUBROUTINE CCNEXT                                                   PLT        65
                                                                          BLANK       2
      LEVEL 2, STORE,INFF,IWORK                                           BLANK       3
      COMMON STORE(48000),IWORK(10)                                       86MARSIZ    1
      DIMENSION INFF(24,2000)                                             86MARSIZ    2
      EQUIVALENCE (INFF,STORE)                                            86MARSIZ    3
                                                                          BLANK       5
      LEVEL 2, XX,Y1,Y2,Y3,Y4                                             BETPTS      2
      COMMON XX(4000),Y1(4000),Y2(4000), Y3(4000),Y4(4000)                RENAM83     1
      COMMON/BPLTCOM/MN,KW,BXX,BYX,NPLT                                   RENAM83     2
      COMMON/CCPOOL/XMIN,XMAX,YMIN,YMAX,CCXMIN,CCXMAX,CCYMIN,CCYMAX       PL6683      1
      COMMON/CCFACT/FACTOR                                                PL6683      2
                                                                          BETPTS      5
      XN=CCXMAX/100.+5.                                                   PLT        68
      CALL PLOT(XN,0.,-3)                                                 PLT        69
      RETURN                                                              PLT        70
      END                                                                 PLT        71
```

STORAGE ALLOCATION.

```
        ADDRESS   LENGTH              BINARY CONTROL CARDS.

              0       12              IDENT NOARG
             12                       END


                               ENTRY POINTS.

                               NOARG            1+



                                      IDENT NOARG
                               *
                               *  NOARG RETURNS THE NUMBER OF PARAMETERS IN A FORTRAN CALL
                               *  STATEMENT.  SINCE COMPILER CALLING CONVENTIONS DIFFER, NOARG
                               *  IS ASSEMBLED CONDITIONALLY TO PRODUCE CODE FOR EITHER FTN OR RUN
                               *  (COMPASS CALLS ASSEMBLE FTN CODE).
                               *
                               *  CERN LIBRARY PROGRAM Z029
                               *
                               *  A. YULE/L. POPE  CERN  DECEMBER 1971
                               *
                                      ENTRY    NOARG
                               * SUBROUTINE NOARG(N)
    0  16170122070000000001           VFD      42/0LNOARG,18/1
    1  00000000000000000000    NOARG  DATA     0
                               *  CODE FOR FTN FOLLOWS - OR - SKIP TO RUN CODE
                               RUNARG IFNE     *F,1
                               *  FTN CODE
                               *  ROUTINE FOR ZERO NUMBER OF ARGUMENTS
    2  5120000001 +                   SA2      NOARG     EQ00AAAAAA0000000000
                 21236                AX2      30        ALIGN ADDRESS OF CALL+1
    3  5232777776                     SA3      X2-1      GO BACK TO +RJ-TRACE
                 53230                SA2      X3        POINT TO TRACE WORD
                      73720           SX7      X2        LOOK AT LAST 18 BITS
    4  0317000006 +                   NZ       X7,ARGS
                 53710                SA7      X1        STORE AT ARG ADDRESS
    5  0400000001 +                   ZR       B0,NOARG  RETURN
                               *  FTN CODE FOR NON-ZERO NUMBER OF ARGUMENTS
    6  54200                   ARGS   SA2      A0        POINT TO ARG LIST
            6110000001                SB1      1         INIT INCREMENT REG
                      76700           SX7      B0        ZERO OUT X7 FOR COUNT
    7  0302000011 +                   ZR       X2,ENDSCAN  IF FIRST ADDRESS ZERO - END
   10  54221                   VFDSCAN SA2     A2+B1     NEXT ARG ADDRESS
            73771                     SX7      X7+B1     INCREMENT COUNTER
                 0312000010 +         NZ       X2,VFDSCAN  SCAN UNTIL ZERO
   11  53710                   ENDSCAN SA7     X1        STORE AT ARG ADDRESS
            0400000001 +              ZR       B0,NOARG  RETURN
                               RUNARG ELSE
                               RUNARG ENDIF
   12                                 END

      16100B LCM  31300B SCM STORAGE USED         52 STATEMENTS        4 SYMBOLS
                  7600-TYPE CPU ASSEMBLY       0.024 SECONDS          11 REFERENCES
```

SYMBOLIC REFERENCE TABLE.

ARGS	6	PROGRAM*	9/38	9/42 L			
ENDSCAN	11	PROGRAM*	9/45	9/49 L			
NOARG	1	PROGRAM*	9/25 E	9/28 L	9/33	9/40	9/50
VFDSCAN	10	PROGRAM*	9/46 L	9/48			

```
          OVERLAY(SYNCH,1,0)                                                SW1           2
          PROGRAM SW1                                                       SW1           3
      C  PROCESSES DATA FOR CURRENT INSTR.IF IN MODE 1 OR 2.                SW1           4
      C  PROCEEDS WITH CALC.BY CALLING APPROPRIATE ROUTINES IF MODE=1 OR 3. SW1           5
      C                                                                     SW1           6
          DIMENSION INF(24)                                                 SW1           7
          COMMON/INSTR/OPNAME,NAME,OP,KA,KB,OBJA,OBJB,NXTM,TRA,LQ,NQ,LIN,   INSTR         2
         1            NIN,LFL,NFL,LBC,NBC,NTOT,IX,KIND,TYPE,ROWS,COLS,EXTR, INSTR         3
         2            M,NF,NB,NI,MSUBR,ISIGN                                INSTR         4
          INTEGER OPNAME,OP,OBJA,OBJB,TRA,TYPE,ROWS,COLS,EXTR               INSTR         5
                                                                            INSTR         6
                                                                            BLANK         2
          LEVEL 2, STORE,INFF,IWORK                                         BLANK         3
          COMMON STORE(48000),IWORK(10)                                     86MARSIZ      1
          DIMENSION INFF(24,2000)                                           86MARSIZ      2
          EQUIVALENCE (INFF,STORE)                                          86MARSIZ      3
                                                                            BLANK         5
          COMMON/STORE/LMAX,LINF,LFILE,MAX,MIN,ICARD(11)                    STORE         2
      *  LMAX AND MAX ARE DIMENSIONS OF STORE AND INFF                      STORE         3
                                                                            STORE         4
          COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL        2
         1          LDFLG,FIN                                               CONTRL        3
          LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL        4
                                                                            CONTRL        5
          COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL   DIM           2
                                                                            DIM           3
          COMMON/BMI/MI(5000)                                               BMIL          1
          COMMON/NELS/NELS                                                  BMIL          2
          LEVEL 2,MI                                                        BMIL          3
                                                                            BMIL          4
                                                                            BMI           3
          COMMON/BMI1/MI1(16000)                                            NV3BMI1       1
          COMMON/NELS1/NELS1                                                BMI1L         2
          LEVEL 2,MI1                                                       BMI1L         3
                                                                            BMI1L         4
      C                                                                     BMI1          3
          COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER   SWTCH         2
          LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                          SWTCH         3
                                                                            SWTCH         4
          COMMON/GRR/IERR,BSW,VSW,RSW,QSW,XSW                               GRR           2
          LOGICAL IERR,BSW,VSW,RSW,QSW,XSW                                  GRR           3
                                                                            GRR           4
          COMMON/TRKINT/G,EX,EY,OMSQ,SEND,VMX,VMN,DINT,BEG,DPR,JPR,         TRKINT        2
         1   THET,PLT,NPT,NZ,PMAX,NSIZ,ITITLE(6)                            TRKINT        3
          LOGICAL BEG,DPR,PLT                                               TRKINT        4
          LEVEL 2, XX,Y1,Y2,Y3,Y4                                           BETPTS        2
          COMMON XX(4000),Y1(4000),Y2(4000), Y3(4000),Y4(4000)              RENAM83       1
          COMMON/BPLTCOM/MN,KW,BXX,BYX,NPLT                                 RENAM83       2
          COMMON/CCPOOL/XMIN,XMAX,YMIN,YMAX,CCXMIN,CCXMAX,CCYMIN,CCYMAX     PL6683        1
          COMMON/CCFACT/FACTOR                                              PL6683        2
                                                                            BETPTS        5
          COMMON/SVNAM/NAMRUN                                               SVNAM         2
          COMMON/COPY/CPYSW                                                 COPY          2
          LOGICAL CPYSW                                                     COPY          3
                                                                            COPY          4
                                                                            SW1          21
          INTEGER BDAT,PVECS                                                SW1          22
```

```
      DIMENSION PAR(100),KREM(100),MATNM(20)                            SW1      23
      INTEGER R7,ROT,PRD                                                SW1      24
      EQUIVALENCE (OPNAME,INF(1)),(PAR,KREM)                            SW1      25
      DIMENSION BX(20)                                                  SW1      26
      LOGICAL TABSW                                                     SW1      27
      DATA IBLNK/4H    /                                                SW1      28
      DATA ISCL/9/                                                      SW1      29
      DATA IBLNK,MXPMY,PVECS,R7,ROT,PRD/1H ,1,2,5,6,8/                  SW1      30
      DATA (MATNM(I),I=1,12)/2HF ,2HD ,2HS ,2HT ,2HE ,2HK ,2HIA,2HIB,2HB SW1     31
     1 ,2HA ,2HWD,2HWF/                                                 SW1      32
C                                                                       SW1      33
*       IF OPNAME = NULL, IT IS A P OR C CARD.  NEEDS SPECIAL           SW1      34
*     HANDLING TO CONSTRUCT A SYNCH PAGE OR REM INSTRUCTION             SW1      35
                                                                        SW1      36
                                                                        SW1      37
*        GO TO STOP, FIN, RUN, SUB, END, CALL, INCR, REPL, MESH, VPAR   SW1      38
*             =, CRD, BML, DRF, MAG, MAGV, EQU, INV, SHF, REF           SW1      39
*             **, MMM, CYA, CYB, CYC, CELL, ALTC, TAB, PTAB, FITQ       SW1      40
C            FITB,FITR,FITV,PAGE,REM,WMA,SIZE,WBE,FIT,COPY,NCOPY,BEST,BE SW1     41
C          ,BETA,KICK,RAND,SHF7,SOL,VAR,PRNT                            SW1      42
                                                                        SW1      43
      GO TO (1,2,3,4,5,6,19,20,74,74,                                   SW1      44
     1       27,7,86,10,13,42,11,12,16,18,                              SW1      45
     2       21,14,23,24,23,29,30,81,82,25,                             SW1      46
     3        25,25,25,34,35,15,61,26,31,32,                            SW1      47
     4     33,23,43,44,45,46,47,48,49,50) OP                            SW1      48
                                                                        SW1      49
*                     STOP ENTRY                                        SW1      50
    1 IF(NPLT.GE.1) CALL FINPL                                          S1PLT     1
      STOP                                                              S1PLT     2
*                     FIN ENTRY                                         SW1      52
    2  CALL SRUN(2)                                                     SW1      53
      RETURN                                                            SW1      54
*                     RUN ENTRY                                         SW1      55
    3 CALL SRUN(1)                                                      SW1      56
      RETURN                                                            SW1      57
*                     SUB ENTRY                                         SW1      58
    4 CALL SUBR(1)                                                      SW1      59
      RETURN                                                            SW1      60
*                     END ENTRY                                         SW1      61
    5 CALL SUBR(2)                                                      SW1      62
      RETURN                                                            SW1      63
*                     CALL ENTRY                                        SW1      64
    6 CALL SUBR(3)                                                      SW1      65
      RETURN                                                            SW1      66
*                     CRD ENTRY                                         SW1      67
    7 CALL INP(M,MXPMY,MINZER,                                          SW1      68
     .  12,1,0,0,0,0,0,0                                                SW1      69
     .)                                                                 SW1      70
      IF (MODE.EQ.2) RETURN                                             SW1      71
      IF (XSW) RETURN                                                   SW1      72
      CALL DATA (M,1,1,12,PAR)                                          SW1      73
      CALL RXRY(M,PAR)                                                  SW1      74
      RETURN                                                            SW1      75
C                                                                       SW1      76
C  DRIFT ENTRY                                                          SW1      77
C  PROCESSES DRIFT DATA.  SETS UP DRIFT MATRICES IF IN EXEC.MODE.       SW1      78
```

```
 10   IF ( MODE.EQ.3 ) GO TO 101                                        SW1
      CALL DRFRSV( M,NAME,KA,KB,0 )                                     SW1
      IF (KB.EQ.0)  CALL LOAD(M,1,1,0,0,0,0,0,0)                        SW1
      IF (KB.GT.0)  CALL LOAD(M,2,1,KB,2,0,0,0,0)                       SW1
      IF (KA.EQ.0) GO TO 102                                            SW1
*        SET UP MATRIX SPACE FOR KA NUMBER OF DRIFTS, WHOSE LENGTHS WILL SW1
*  BE ELEMENTS OF INPUT VECTOR.                                         SW1
      CALL RSVMAT                                                       SW1
      MIN = MIN - KA                                                    SW1
 102   IF (XSW)  RETURN                                                 SW1
      IF (ERROR) RETURN                                                 SW1
 101  IF (KA.GT.0) GO TO 103                                            SW1
      EL = FDAT(M,1)                                                    SW1
      IF (KB.EQ.0)  GO TO 106                                           SW1
C   VARIABLE DRIFT LENGTH                                               SW1
      FLTOT = 0.                                                        SW1
      DO 104 K=1,KB                                                     SW1
      MM = MDAT(M,K)                                                    SW1
 104  FLTOT = FLTOT + FDAT(MM,1)                                        SW1
      FLSEP = FDAT(M,2)                                                 SW1
      IF (FLSEP.NE.0.) GO TO 105                                        SW1
      FLSEP = EL + FLTOT                                                SW1
      CALL REPFLT (M,2,FLSEP)                                           SW1
      GO TO 106                                                         SW1
 105  EL = FLSEP - FLTOT                                                SW1
      CALL REPFLT(M,1,EL)                                               SW1
 106  IF (MODE.EQ.2)  RETURN                                            SW1
      CALL DRIFT(M,EL)                                                  SW1
      RETURN                                                            SW1
 103  IF (MODE.EQ.2)  RETURN                                            SW1
      CALL DRFMATS(M,KA)                                                SW1
      RETURN                                                            SW1
C                                                                       SW1
C  EQUATE ENTRY                                                         SW1
C  PROCESSES EQU DATA. EQUATES MATRIX M TO N IF IN EXECUTION MODE.      SW1
C  MATRICES COULD HAVE VARIABLE DIMENSIONS.                             SW1
 11   CALL INP(M,0,0,                                                   SW1
     .1,2,0,0,0,0,0,0,0                                                 SW1
     .)                                                                 SW1
      IF (MODE.EQ.2) RETURN                                             SW1
      IF (XSW) RETURN                                                   SW1
      N=MDAT(M,1)                                                       SW1
      CALL EQU(M,N)                                                     SW1
      RETURN                                                            SW1
C                                                                       SW1
C  INVERT ENTRY                                                         SW1
C PROCESSES INVERT DATA. INVERTS IF IN EXECUTION MODE.                  SW1
 12   KN = IBLNK                                                        SW1
      IF (MSIZE.EQ.7) KN=R7                                             SW1
      IF (MSIZE.EQ.3) KN=MXPMY                                          SW1
      CALL INP(M,KN,MINZER,                                             SW1
     .  1,2,0,0,0,0,0,0                                                 SW1
     .)                                                                 SW1
      IF (MODE.EQ.2) RETURN                                             SW1
      IF (XSW) RETURN                                                   SW1
      N=MDAT(M,1)                                                       SW1
C  INVERT MATRIX AT INDEX N AND STORE IN INDEX M                        SW1
```

```
      CALL INV(M,N)
      RETURN
C
C  MAGNET ENTRY
C  PROCESSES MAG DATA. SETS UP MAGNET MATRICES IF IN EXEC. MODE.
 13   IF ( MODE.EQ.3 ) GO TO 131
      CALL MAGRSV( M,NAME,KA,KB,0 )
      CALL LOAD(M,
     .  6,1,0,0,0,0,0,0
     .)
      IF (KA.EQ.0) GO TO 132
*          SET UP MATRIX SPACE FOR KA NUMBER OF MAGNETS.  SOME OF THE
*  INPUT PARAMETERS DERIVE FROM PREVIOUSLY DEFINED VECTOR.
      CALL RSVMAT
      MIN = MIN - KA
 132  IF (MODE.EQ.2) RETURN
      IF (XSW) RETURN
      IF (ERROR) RETURN
 131  IF (KA.GT.0) GO TO 133
      CALL DATA (M,1,1,6,PAR)
      CALL MAGNET(M,PAR)
      RETURN
 133  CALL MAGMATS(M,KA)
      RETURN
C
C  KICK ENTRY
 45   IF (MODE.EQ.3) GO TO 451
      CALL INP(M,R7,MINZER,1,2,3,1,0,0,0,0)
      IF ( (MODE.EQ.2).OR.(XSW).OR.(ERROR) ) RETURN
 451  CALL KICK(M)
      RETURN

C
C  MMM ENTRY
C  PROCESSES MMM DATA. MULTIPLIES MATRICES IF IN EXEC. MODE
 14   IF ( MODE.EQ.3 ) GO TO 141
      IF (MSIZE.NE.3)  KN = R7
      CALL INP(M,KN,MINZER,1,-2,0,0,0,0,0,0)
      IF (MODE.EQ.2) RETURN
      IF (XSW) RETURN
 141  NDAT=INFF(17,M)
      CALL MIFILL(M,1,NDAT,NELS,MI)
      KB = INFF(5,M)
      IF (KB.NE.0.AND.KB.NE.1)  GO TO 142
      CALL MMM(M,NELS,MI)
      RETURN
 142  CALL EXECMM(M,NELS,KB,MI)
      RETURN
C
C  WMA ENTRY
C  PROCESSES WMA DATA. WRITES MATRICES IF IN EXEC.MODE.
 15   CALL INP(M,0,0,
     .  KA,2,0,0,0,0,0,0
     .)
      IF(MODE.EQ.2) RETURN
      IF (XSW) RETURN
      CALL MLIST(M,1,KA,MI)
```

```
      IF(ERROR) RETURN
      CALL WMA(KA)
      RETURN
C
C  SHIFT ENTRY
C  PROCESSES SHF DATA. SETS UP SHIFT MATRIX IF IN EXEC. MODE
 16   CALL INP(M,MXPMY,MINZER,
     .  4,1,0,0,0,0,0,0
     .)
      IF (MODE.EQ.2) RETURN
      IF (XSW) RETURN
      CALL DATA(M,1,1,4,PAR)
      CALL SHF(M,PAR)
      RETURN
C
C  REFLECT ENTRY
C  PROCESSES REF DATA. COMPUTES MATRIX REFLECTION IF IN EXEC.MODE
 18   KN = IBLNK
      IF (MSIZE.EQ.7) KN = R7
      IF (MSIZE.EQ.3) KN = MXPMY
      CALL INP(M,KN,MINZER,
     .  1,2,0,0,0,0,0,0
     .)
      IF (MODE.EQ.2) RETURN
      IF (XSW) RETURN
      N=MDAT(M,1)
      CALL REF(M,N)
      RETURN
C
C  INCR ENTRY
*  DETERMINE TYPE TO BE INCREMENTED-- FL.PT., INTEGER, KA OR KB.
 19   IF (MODE.EQ.3) GO TO 192
      DECODE (10,201,ICARD(3)) KT
      KE = 0
      IF (KT.EQ.1H ) KE = 1
      IF (KT.EQ.1HF) KE = 1
      IF (KT.EQ.1HI) KE = 3
      IF (KT.EQ.2HKA) KE = 3
      IF (KT.EQ.2HKB) KE = 3
      IF (KE.NE.0) GO TO 191
*         ILLEGITIMATE TYPE
      WRITE (3,190) KT
 190  FORMAT (5X,1H*,*ERROR*,2H* ,A5,* IS AN ILLEGITIMATE TYPE OF INCREM
     1ENT.*)
      ERROR = .TRUE.
      RETURN
 191  CONTINUE
      CALL RESRV(M,0,0,
     .  2,2,2,KE,0,0,0,0
     .)
      CALL LOAD(M,
     .  2,2,1,KE,0,0,0,0
     .)
      IF (MODE.EQ.2) GO TO 193
 192  IF (XSW) RETURN
      KO = KA
      CALL INCR(M,KO)
```

```
      RETURN
  193 CALL INFW(MSUBR,19,M)
      RETURN
C
C  REPL ENTRY
 20   CONTINUE
      IF (MODE.EQ.3) GO TO 204
*        DETERMINE TYPE OF REPLACEMENT -- FL. PT., INTEGER, BCD OR KA,KB
      DECODE (10,201,ICARD(3)) KT
  201 FORMAT (5X,A5)
      KE = 0
*                    IF BLANK, DEFAULT IS F
      IF (KT.EQ.1H ) KE =1
      IF (KT.EQ.1HF) KE = 1
      IF (KT.EQ.2HSF)  KE = 1
      IF (KT.EQ.1HQ) KE = 1
      IF (KT.EQ.1HH) KE = 2
      IF (KT.EQ.1HI) KE = 3
      IF (KT.EQ.2HKA) KE = 3
      IF (KT.EQ.2HKB) KE = 3
      IF (KE.NE.0) GO TO 202
*                    ILLEGITIMATE TYPE
      WRITE (3,203) KT
  203 FORMAT (5X,1H*,*ERROR*,2H* ,A5,* IS AN ILLEGITIMATE TYPE OF REPLAC
     1EMENT.*)
      ERROR = .TRUE.
      RETURN
  202 CONTINUE
      CALL INP(M,0,0,
     .  2,2,1,KE,0,0,0,0
     .)
      IF (MODE.EQ.2) RETURN
      IF (XSW) RETURN
  204 CONTINUE
      CALL REPL(M)

      RETURN
C
C  STAR ENTRY
C PROCESSES STAR DATA. COMPUTES IF IN EXEC. MODE
 21   KN = IBLNK
      IF (MSIZE.EQ.7) KN=R7
      IF (MSIZE.EQ.3) KN=MXPMY
      CALL INP(M,KN,MINZER,
     .  1,2,0,0,0,0,0,0
     .)
      IF (MODE.EQ.2) RETURN
      IF (XSW) RETURN
      CALL STAR(M,KA)
      RETURN
C
C
C  CYA AND CYC ENTRY
*          CYA, CYC, BEST ENTRY
 23   IF (MODE.EQ.3) GO TO 543
C  MAKE RESERVATION AND STORE INPUT DATA
      KN = MXPMY
```

```
      CALL INP(M,KN,MINZER,                                   SW1       307
     . 4,2,2,1,0,0,0,0)                                       NOV3SW1     1
      NDAT=1                                                  NOV3SW1     2
      IF (KN.EQ.IBLNK) KN = INFF(20,M)                        SW1       311
      IF (KN.EQ.2HR7) KN = R7                                 SW1       312
      IF (KN.EQ.3HMXY) KN = MXPMY                             SW1       313
      IF (OPNAME.NE.3HCYA.AND.KA.NE.21)  GO TO 521            SW1       314
      IF (KA.GT.999) GO TO 545                                SW1       315
      DECODE (1,520,NAME) NM                                  SW1       316
  520 FORMAT (A1)                                             SW1       317
      CALL MIFILL(M,1,NDAT,NELS1,MI1)                         SW1       318
      IF ( OPNAME.EQ.3HCYA)  GO TO 522                        SW1       319
      NELS2=NELS1                                             SW1       320
      IF (KB.LT.0)  NELS2=NELS1+1                             SW1       321
      NQ3 = 14*(NELS2+1) + 18                                 SW1       322
      LQ3 = LFILE + 1                                         SW1       323
      LFILE = LQ3 + NQ3                                       SW1       324
      NTOT = NTOT + NQ3                                       SW1       325
      INFF(18,M) = NTOT                                       SW1       326
      INFF(24,M) = LQ3                                        SW1       327
      IF (MODE.EQ.2)  GO TO 542                               SW1       328
      IF (XSW) RETURN                                         SW1       329
      GO TO 544                                               SW1       330
  522 CALL DEFSET(M,NM,NELS1,MATNM,0,KN)                      SW1       331
  521 IF (MODE.EQ.2) GO TO 542                                SW1       332
      IF (XSW) RETURN                                         SW1       333
  543 NDAT=1                                                  NOV3SW1     3
      CALL MIFILL(M,1,NDAT,NELS1,MI1)                         SW1       335
  544 CALL CYX(M)                                             SW1       336
      IF (MODE.EQ.3) RETURN                                   SW1       337
  542 IF (OPNAME.NE.4HCYA ) RETURN                            SW1       338
      MIN = MIN - NELS1                                       SW1       339
      RETURN                                                  SW1       340
  545 WRITE (3,546)                                           SW1       341
  546 FORMAT (23H CYA COUNT EXCEEDS 999.)                     SW1       342
      ERROR=.TRUE.                                            SW1       343
      RETURN                                                  SW1       344
C                                                             SW1       345
C  CYB  ENTRY                                                 SW1       346
   24 IF (MODE.EQ.3) GO TO 240                                SW1       347
      KN = MXPMY                                              SW1       348
      CALL INP(M,0,0,                                         SW1       349
     .1,2,1,3,0,0,0,0                                         SW1       350
     .)                                                       SW1       351
      IF (MODE.EQ.2) RETURN                                   SW1       352
      IF (XSW) RETURN                                         SW1       353
C  RETRIEVE ELEMENT NAME FROM DATA AND OBTAIN ITS INDEX.      SW1       354
  240 CONTINUE                                                SW1       355
      CALL SETCYB(M,KA)                                       SW1       356
      IF (ERROR) RETURN                                       SW1       357
  245 CALL CYX(M)                                             SW1       358
      RETURN                                                  SW1       359
C                                                             SW1       360
C                        FITQ,FITB,FITR,FITV ENTRY            SW1       361
   25 CALL SETFIT                                             SW1       362
      RETURN                                                  SW1       363
                                                              SW1       364
```

```
C
C  WBE ENTRY
C   PROCESSES WBE DATA. CALLS  WBE TO WRITE BETATRON FUNCTIONS
C      IF IN EXECUTION MODE
 26   CALL INP(M,0,0,
     .  KA,2,0,0,0,0,0,0
     .)
      IF(MODE.EQ.2) RETURN
      IF (XSW) RETURN
      CALL MLIST(M,1,KA,MI)
      IF(ERROR) RETURN
      CALL WBE(KA,KB)
      RETURN
C
C  = ENTRY
 27   IF (MODE.EQ.3) GO TO 275
      KN = IBLNK
      DECODE (10,5000,ICARD(4)) IOP
 5000 FORMAT (A5,5X)
      IF (IOP.NE.IBLNK) GO TO 271
C  KIND AND TYPE ARE SCALAR AND SINGLE
      KN = ISCL
      CALL INP(M,KN,0,
     .  1,1,1,2,0,0,0,0
     .)
      RETURN
C  LOOK AT FIRST INPUT PARAMETER
 271  CALL CHKD(ICARD(3),KN,ITP)
      IF (KN.NE.4HSCAL) GO TO 272
      IQ = 1
      KN = ISCL
      IF (ITP.NE.4HSNGL) IQ = 2
      CALL INP(M,KN,IQ,
     .  1,1,1,2,2,1,0,0
     .)
      F = FDAT(M,3)
      CALL REPLQ(M,1,1,F)
      IF (MODE.EQ.2.OR.XSW)  RETURN
      GO TO 276
C     MATRIX.  INPUT IS BCD,BUT IN 10 COLUMN FIELDS.
 272  IF (KN.EQ.3HMXY) KN=1
      IF (KN.EQ.2HR7) KN = 5
      CALL  INP(N,KN,MINZER,
     .  3,2,0,0,0,0,0,0
     .)
      IF (MODE.EQ.2.OR.XSW)  RETURN
      GO TO 277
 275  IF (XSW)  RETURN
      KN = INFF(20,M)
      IF (KN.NE.4HSCAL) GO TO 277
      IOP = BDAT(M,1)
C  RETRIEVE OPERATION SYMBOL.  IF KIND NOT SCALAR,IT IS 2ND BCD INPUT
C IF SINGLE SYMBOLIC ARGUMENT, GET ITS VALUE
      IF (IOP.EQ.IBLNK) RETURN
      ITP = INFF(21,M)
 276  IQ = 2
      IF (ITP.NE.4HSNGL) IQ = 4
```

```
      CALL DATA (M,1,1,IQ,PAR)
      CALL AROP(M,PAR,IOP)
      RETURN
  277 KREM(1) = BDAT(M,1)
      KREM(2) = BDAT(M,3)
      IOP = BDAT(M,2)
      CALL AROP(M,KREM,IOP)
      RETURN
C
C  CELL ENTRY
  29   IF(MODE.EQ.3) GO TO 293
C MAKE RESERVATIONS FOR FIRST CYCLED MATRIX AND STORE INPUT DATA
C  RESERVE SPACE IN DATA FOR THE MI LIST
       CALL RESRV (M,MXPMY,MINZER,
     .  7,1,12,3,0,0,0,0
     .)
      CALL LOAD(M,
     .  7,1,0,0,0,0,0,0
     .)
C  SET UP SPACE FOR CYCLED MATRICES
      DECODE (1,520,NAME) NM
      CALL DEFSET (M,NM,12,MATNM,4,MXPMY)
      IF (MODE.EQ.2) GO TO 294
      IF (XSW) RETURN
  293 CALL DATA(M,1,1,7,PAR)
      CALL CELL(M,PAR)
      IF (MODE.EQ.3) RETURN
  294 MIN=MIN-16
      RETURN
C
C  ALTC ENTRY
  30   IF (MODE.EQ.3) GO TO 301
C  MAKE RESERVATIONS FOR FIRST CYCLED MATRIX AND STORE INPUT
C  RESERVE SPACE IN DATA FOR MI LIST
      CALL RESRV(M,MXPMY,MINZER,
     .  8,1,12,3,0,0,0,0
     .)
      CALL LOAD(M,
     .8,1,0,0,0,0,0,0
     .)
      DECODE (1,520,NAME) NM
C SET UP SPACE FOR CYCLED MATRICES AND  QF,QD,QS,QT
      CALL DEFSET (M,NM,12,MATNM,4,MXPMY)
      IF (MODE.EQ.2) GO TO 302
      IF (XSW) RETURN
  301 CALL DATA(M,1,1,8,PAR)
      CALL ALTC(M,KA,PAR)
      IF(MODE.EQ.3) RETURN
  302 MIN=MIN-16
      RETURN
C
*                   FIT ENTRY
  31   IF (MODE.EQ.3) GO TO 311
      K1 = KA*2
      K2 = KB*2
      K = K1 + K2 + 2
      CALL RESRV(M,0,0,
```

```
     .  K,2,1,3,0,0,0,0
     .)
      CALL LOAD(M,
     .  2,2,1,3,0,0,0,0
     .)
      CALL LOAD(M,
     .  K1,2,0,0,0,0,0,0
     .)
      CALL LOAD(M,
     .  K2,2,0,0,0,0,0,0
     .)
      IF (MODE.EQ.2) RETURN
      IF (XSW) RETURN
 311  CALL DOFIT(M)
      RETURN

C              COPY ENTRY
 32    IF (MODE.EQ.3)  GO TO 320
      CALL INP(M,0,0,0,0,0,0,0,0,0,0)
 320  IF (MODE.EQ.2)  RETURN
      IF (XSW)  RETURN
      CPYSW = .TRUE.
      RETURN
C              NCPY ENTRY
 33   IF (MODE.EQ.3)  GO TO 330
      CALL INP(M,0,0,0,0,0,0,0,0,0,0)
 330  IF (MODE.EQ.2)  RETURN
      IF (XSW)  RETURN
      CPYSW = .FALSE.
      RETURN
C  PAGE ENTRY
C   RESTORES PAGE IF IN EXECUTION MODE
 34   IF (MODE.EQ.3) GO TO 341
*                    STORE DATA FIELD TO PRINT AS A COMMENT
      CALL INP(M,0,0,
     .  12,2,0,0,0,0,0,0
     .)
      IF (MODE.EQ.2) RETURN
      IF (XSW) RETURN
 341  K0 = INFF(17,M)
      CALL DATA (M,2,1,K0,KREM)
      IF (CPYSW)  WRITE (3,342) (KREM(I),I=1,K0)
 342  FORMAT (1H1/8X,16A5)
      RETURN
C
C  REM ENTRY
C   PRINTS REMARKS
 35   IF (MODE.EQ.3) GO TO 352
      CALL INP(M,0,0,
     .  12,2,0,0,0,0,0,0
     .)
 352  IF (XSW) RETURN
      K0 = INFF(17,M)
      CALL DATA (M,2,1,K0,KREM)
      IF (CPYSW)  WRITE (3,353) (KREM(I),I=1,K0)
 353  FORMAT (/8X,16A5)
      RETURN
```

```
C
*     MAGV ENTRY
 42    CONTINUE
       IF (MODE.EQ.3) GO TO 421
       CALL INP(M,R7,MINZER,
     .  6,1,0,0,0,0,0,0
     .)
       IF (MODE.EQ.2) RETURN
       IF (XSW) RETURN
 421   CALL DATA (M,1,1,6,PAR)
       CALL MAGV(M,PAR)
       RETURN

*                     BETP ENTRY  -- PLOT BETATRON FUNCTIONS
 43    IF (MODE.EQ.3) GO TO 431
       CALL INP(M,0,0,
     .  1,2,2,3,3,1,0,0
     .)
       IF (MODE.EQ.2) RETURN
       IF (XSW) RETURN
 431   CALL PLOTBET(M)
       RETURN
C
C  SIZE ENTRY, SWITCH DEFINING MATRIX SIZE
 61    IF (MODE.EQ.3) RETURN
C SET MSZ TO KA. IF KA = 7 FOR 7X7 , 3 FOR 3X3, 37 IF NOT KNOWN
       MSIZE = KA
       IF (CPYSW)  WRITE (3,610) NAME,OPNAME,KA,KB
 610   FORMAT (6H  *** ,1X,A5,2X,A5,1X,I3,1X,I3)
       RETURN
*                     MESH AND VPAR ENTRY
 74    CONTINUE
       IF (MODE.EQ.3) GO TO 741
       KL = KA + 1
       KM = 3*KA
       CALL RESRV(M,0,0,
     .  KL,2,KA,3,KM,1,0,0
     .)
         CALL LOAD(M,
     .  1,2,0,0,0,0,0,0,0
     .)
       DO 740 IJ=1,KA
       CALL LOAD(M,
     .  1,3,1,2,3,1,0,0
     .)
 740   CONTINUE
       IF (MODE.EQ.2) RETURN
       IF (XSW) RETURN
 741   CONTINUE
       CALL MESH(M,KA,OPNAME)
       RETURN
C  TAB ENTRY
 81    IF (MODE.EQ.3) GO TO 810
       IF (KA.GT.10) KA = 10
       KS = KA*60+1
       CALL INP(M,0,KS,
     .  KA,2,0,0,0,0,0,0
```

```
     .)
C  ZERO ARRAY AT LQ
      DO 811 I=1,60
 811   PAR(I) = 0.
      IK = 2
      DO 812 I=1,KA
      CALL STDAT(M,5,IK,60,PAR)
      IK = IK + 60
 812   CONTINUE
C    SET CURSOR = 2 AT LQ
      CALL REPLQ(M,1,1,2)
      TABSW = .FALSE.
C  SET UP INDICES TABLE
      CALL TABULAT(M,TABSW)
      IF (MODE.EQ.2) RETURN
 810  TABSW = .TRUE.
      CALL TABULAT(M,TABSW)
      RETURN
C  PTAB ENTRY
 82   IF (MODE.EQ.3) GO TO 820
      CALL INP(M,0,0,
     .  KA,2,0,0,0,0,0,0
     .)
      IF (MODE.EQ.2) RETURN
 820  CALL PRNTAB(M)
      RETURN
*                   BML ENTRY -- BEAM LINE
 86   IF (MODE.EQ.3) RETURN
      CALL INP(M,0,0,
     .  1,-2,0,0,0,0,0,0
     .)
      RETURN

C   BETA ENTRY
 44   IF(MODE.EQ.3) GO TO 441
      CALL INP(M,ISCL,MINZER,
     .  1,2,1,1,0,0,0,0
     .)
      IF(MODE.EQ.2) RETURN
 441  MM=MDAT(M,1)
      CALL BET(MM,BX(1),BX(11),0)
      X=BX(KA)
      CALL REPFLT(M,0,X)
      RETURN
C  RAND ENTRY
C
 46   IF(MODE.NE.3) CALL INP(M,ISCL,0,3,1,0,0,0,0,0,0)
      IF(MODE.EQ.2) RETURN
      RNS=FDAT(M,3)
      IF(RNS.GT.0..AND.RNS.LT.1) CALL RANSET(RNS)
      KB=MAX0(KB,1)
      IF(KA.LE.0) RETURN
C  RUN RANDOM GENERATOR KA*KB TIMES
      KB = MAX0(KB,1)
      DO461 I=1,KA
      DO461 J=1,KB
 461  X=RANF(D)
```

```
      RETURN

C   SHF7 ENTRY  7X7 SHIFT MATRIX
 47   IF (MODE.EQ.3)  GO TO 471
      CALL INP (M,R7,MINZER,
     .  6,1,0,0,0,0,0,0
     .)
      IF (MODE.EQ.2)  RETURN
 471  CALL DATA (M,1,1,6,PAR)
      CALL SHF7(M,PAR)
      RETURN

C   SOLENOID ENTRY
 48   IF (MODE.EQ.3)  GO TO 481
      CALL INP (M,R7,MINZER,
     . 4,1,0,0,0,0,0,0
     .)
      IF (MODE.EQ.2)  RETURN
 481  CALL DATA (M,1,1,4,PAR)
      CALL SOL(M,PAR)
      RETURN
C        VAR ENTRY
C    DEFINES VARIABLE BY INSTRUCTION NAME AND POSITION.
C...........................................................
C  VNAM   VAR   KA    NAME
C.....................................................
C   EXAMPLE 1
C     QF   MAG          1.5      0.06        1.
C     GQF  VAR    1     QF
C     QFH  MAG          0.75     GQF         1.
C--------------------------------------------------------------------------
C.....................................................
C
C   IN THIS EXAMPLE QFH HAS HALF THE LENGTH OF QF, BUT THE SAME
C  GRADIENT, 0.06.
C
C   EXAMPLE 2 -- A MORE USEFUL EXAMPLE
C     L      DRF          2.
C     SR     SUB
C     QF     MAG          1.       0.05       1.
C     QD     MAG          1.      -0.05       1.
C     C      MMM          QF   L   QD   L
C            END
C            FITQ         SR   C   QF   QD      2      2     .25   .25
C
C     GF     VAR   2      QF
C     GD     VAR   1      QD
C     QFH    MAG          .75       GF            1.
C     QDH    MAG          .75       GD            1.
C            CYC       -1 QFH   L   QDH
C--------------------------------------------------------------------------
C
 49   IF (MODE.EQ.3) GO TO 491
      CALL INP (M,0,0,1,2,0,0,0,0,0,0)
      IF (MODE.EQ.2) RETURN
 491  FVAR = FLDAT(M,1)
      NAM=MNAME(M)
```

```
                WRITE (3,490) NAM,FVAR                          SW1       707
          490   FORMAT (30X,A5,* VALUE ---- *,E15.8)            SW1       708
        C                                                       SW1       709
                RETURN                                          SW1       710
        C   PRNT ENTRY ---- PRINT SELECTED DATA                 SW1       711
           50   IF (MODE.EQ.3) GO TO 51                         SW1       712
                CALL INP(M,0,0,1,-2,0,0,0,0,0,0)                SW1       713
                IF (MODE.EQ.2)  RETURN                          SW1       714
           51   IF (XSW) RETURN                                 SW1       715
                CALL PRNT(M)                                    SW1       716
                RETURN                                          SW1       717
                                                                SW1       718
                END                                             SW1       719
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
90	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
93	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
96	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
99	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
102	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
107	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
108	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
111	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
124	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
125	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
141	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
143	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
144	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
146	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
154	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
155	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
158	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
168	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
169	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
173	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
187	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
188	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
189	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
193	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
195	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
200	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
202	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
210	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
211	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
217	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
219	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
226	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
227	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
229	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
231	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
238	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
239	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
242	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
252	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
253	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
256	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
270	I	TAPE3#	I/O FILE NOT DEFINED.
274	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
283	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
286	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
288	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
312	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
317	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
318	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
322	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
332	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
333	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
335	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
364	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
368	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
372	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
373	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
375	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
379	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
387	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
388	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
392	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
394	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
398	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
407	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
408	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
410	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
412	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
425	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
437	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
445	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
447	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
453	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
459	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
464	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
480	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
483	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
485	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
501	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
504	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
506	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
525	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
526	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
528	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
533	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
534	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
536	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
540	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
541	I		*** DUE TO THE MANY ERRORS NOTED, ONLY THOSE WHICH ARE FATAL WILL BE LISTED HEREAFT

```
          OVERLAY(SYNCH,3,0)                                               SW3        2
          PROGRAM SW3                                                      SW3        3
C     PROCESSES DATA FOR CURRENT INSTR.IF IN MODE 1 OR 2.                  SW3        4
C     PROCEEDS WITH CALC.BY CALLING APPROPRIATE ROUTINES IF MODE=1 OR 3.   SW3        5
C                                                                          SW3        6
          DIMENSION INF(24)                                                SW3        7
          COMMON/INSTR/OPNAME,NAME,OP,KA,KB,OBJA,OBJB,NXTM,TRA,LQ,NQ,LIN,  INSTR      2
         1              NIN,LFL,NFL,LBC,NBC,NTOT,IX,KIND,TYPE,ROWS,COLS,EXTR, INSTR   3
         2              M,NF,NB,NI,MSUBR,ISIGN                             INSTR      4
          INTEGER OPNAME,OP,OBJA,OBJB,TRA,TYPE,ROWS,COLS,EXTR              INSTR      5
                                                                           INSTR      6
                                                                           BLANK      2
          LEVEL 2, STORE,INFF,IWORK                                        BLANK      3
          COMMON STORE(48000),IWORK(10)                                    86MARSIZ   1
          DIMENSION INFF(24,2000)                                          86MARSIZ   2
          EQUIVALENCE (INFF,STORE)                                         86MARSIZ   3
                                                                           BLANK      5
          COMMON/STORE/LMAX,LINF,LFILE,MAX,MIN,ICARD(11)                   STORE      2
*     LMAX AND MAX ARE DIMENSIONS OF STORE AND INFF                        STORE      3
                                                                           STORE      4
          COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,   CONTRL     2
         1         LDFLG,FIN                                               CONTRL     3
          LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN     CONTRL     4
                                                                           CONTRL     5
          COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL  DIM        2
                                                                           DIM        3
          COMMON/BMI/MI(5000)                                              BMIL       1
          COMMON/NELS/NELS                                                 BMIL       2
          LEVEL 2,MI                                                       BMIL       3
                                                                           BMIL       4
                                                                           BMI        3
          COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER  SWTCH      2
          LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                         SWTCH      3
                                                                           SWTCH      4
          COMMON/GRR/IERR,BSW,VSW,RSW,QSW,XSW                              GRR        2
          LOGICAL IERR,BSW,VSW,RSW,QSW,XSW                                 GRR        3
                                                                           GRR        4
                                                                           SW3       16
          INTEGER BDAT,PVECS                                               SW3       17
          INTEGER R7,ROT,PRD                                               SW3       18
          EQUIVALENCE (OPNAME,INF(1)),(PAR,KREM)                           SW3       19
          DIMENSION MATNM(20),MXLIST(20)                                   SW3       20
          DIMENSION MATNM(20),MXLIST(20)                                   SW3       21
          DIMENSION PAR(100),KREM(100),NCELL(5),JMAG(7)                    SW3       22
          DATA (JMAG(K),K=1,7)/2HIF,2HID,2HOF,2HOD,1HE,1HS,1HT/            SW3       23
          DATA MXPMY,PVECS,R7,ROT,PRD/1,2,5,6,8/                           SW3       24
          DATA (MATNM(I),I=1,12)/2HF ,2HD ,2HS ,2HT ,2HE ,2HK ,2HIA,2HIB,2HB SW3     25
         1 ,2HA ,2HWD,2HWF/                                                SW3       26
C                                                                          SW3       27
*           IF OPNAME = NULL, IT IS A P OR C CARD.  NEEDS SPECIAL          SW3       28
*         HANDLING TO CONSTRUCT A SYNCH PAGE OR REM INSTRUCTION            SW3       29
                                                                           SW3       30
          NOP = OP - 50                                                    SW3       31
                                                                           SW3       32
          GO TO (36,80,80,49,22,17,28,37,38,39,                            SW3       33
         1    47,44,48,46) NOP                                             SW3       34
                                                                           SW3       35
```

```
     *        GO TO SUM, ACT, DELE, SCOP, TEST, WFL, MAGS, CELC, CFD, DCFD
     *                  STR2,STR4,STRP,STRN

      17  CALL WFLSR
          RETURN
     C
      22  CALL TEST(M)
          RETURN
     C
     C
     C  MAGS ENTRY
     C   PROCESSES MAGS DATA.CALCULATES TRANSFER MATRIX FOR NEW MAGNET IF IN
     C     EXECUTION MODE
      28  IF ( MODE.EQ.3 ) GO TO 281
          CALL MAGSRV( M,NAME,KA,KB,0 )
          IF(MODE.EQ.2) RETURN
          IF (XSW) RETURN
      281 CALL DATA (M,1,1,5,PAR)
          N=MDAT(M,1)
          CALL MAGS(M,N,PAR)
          RETURN
     C  SUM ENTRY
      36  IF(MODE.EQ.3) GO TO 361
          CALL INP(M,0,1,
         .  KA,1,0,0,0,0,0,0
         .)
          IF(MODE.EQ.2) RETURN
          IF (XSW) RETURN
      361 CALL DATA (M,1,1,KA,PAR)
          CALL SUM(KA,PAR,RES)
          CALL STDAT(M,5,1,1,RES)
          RETURN
     C
     C   CELC ENTRY
      37  IF (MODE.EQ.3) GO TO 371
     C MAKE RESERVATIONS FOR FIRST CYCLED MATRIX AND STORE INPUT DATA
     C RESERVE SPACE IN DATA FOR THE MI LIST
            CALL RESRV(M,MXPMY,MINZER,
         .  8,1,12,3,0,0,0,0
         .)
          CALL LOAD(M,
         .  7,1,0,0,0,0,0,0
         .)
          DECODE(1,520,NAME) NM
     C SET UP SPACE FOR CYCLED MATRICES AND QF,QD,QS,QT,QE
          CALL DEFSET(M,NM,12,MATNM,5,MXPMY)
          IF (MODE.EQ.2) GO TO 372
          IF (XSW) RETURN
      371 CALL DATA (M,1,1,8,PAR)
          CALL CELC(M,PAR)
          IF(MODE.EQ.3) RETURN
      372 MIN=MIN-17
          RETURN
     C
     C  CFD ENTRY
      38  IF (MODE.EQ.3) GO TO 381
     C  MAKE RESERVATIONS FOR FIRST CYCLED MATRIX AND STORE INPUT DATA
```

```
      C  RESERVE SPACE IN DATA FOR THE MILIST
            CALL RESRV(M,MXPMY,MINZER,
           .  8,1,12,3,0,0,0,0
           .)
            CALL LOAD(M,
           .  8,1,0,0,0,0,0,0
           .)
            DECODE(1,520,NAME) NM
      C  SET UP SPACE FOR CYCLED MATRICES AND QF,QD,QS,QT
            CALL DEFSET(M,NM,12,MATNM,4,MXPMY)
            IF (MODE.EQ.2) GO TO 382
            IF (XSW) RETURN
        381 CALL DATA(M,1,1,8,PAR)
            CALL CFD(M,PAR)
            IF (MODE.EQ.3) RETURN
        382 MIN=MIN-16
            RETURN
      C
      C  DCFD ENTRY
        39  IF(MODE.EQ.3) GO TO 391
            CALL RESRV(M,MXPMY,MINZER,
           .  9,1,12,3,0,0,0,0
           .)
            CALL LOAD(M,
           .  9,1,0,0,0,0,0,0
           .)
            DECODE(1,520,NAME) NM
      C  SET UP SPACE FOR CYCLED MATRICES AND QF,QD,QS,QT
            CALL DEFSET(M,NM,12,MATNM,4,MXPMY)
            IF (MODE.EQ.2) GO TO 392
            IF (XSW) RETURN
        391 CALL DATA(M,1,1,9,PAR)
            CALL DCFD(M,PAR)
            IF (MODE.EQ.3) RETURN
        392 MIN = MIN-16
            RETURN
      C  STR4 ENTRY.  PROCESSES STR4 DATA
        44  IF (MODE.EQ.3) GO TO 442
            CALL INP(M,MXPMY,MINZER,
           .  3,2,4,1,0,0,0,0
           .)
            M1 = M - 1
            DO 441 I=1,4
            CALL MAGRSV (-M1,JMAG(I),0,0,NAME)
        441 M1 = M1 - 1
            DO 440 I=5,7
            CALL DRFRSV(-M1,JMAG(I),0,0,NAME)
        440 M1 = M1 - 1
            IF (MODE.EQ.2) GO TO 443
            IF (XSW) RETURN
        442 CALL DATA(M,2,1,3,NCELL)
            CALL DATA(M,1,1,4,PAR)
            CALL STR4(M,NCELL,PAR)
            IF (MODE.EQ.3) RETURN
        443 MIN = MIN - 7
            RETURN
      C
```

```
C
C  STRN ENTRY.
 46   IF (MODE.EQ.3) GO TO 462
      IF (KA.NE.2) KA=1
      KF = 4
      IF(KA.EQ.2) KF=8
      CALL RESRV(M,MXPMY,MINZER,
     .  2,2,KF,1,0,0,0,0
     .)
      CALL LOAD (M,
     . 2,2,4,1,0,0,0,0
     .)
      IF (KA.EQ.2)  CALL LOAD (M,
     .  4,1,0,0,0,0,0,0
     .)
      DECODE (1,520,NAME) NM
      DO 461 I = 1,6
 461  MXLIST(I) = MATNM(I + 6)
      MXLIST(7)= MATNM(5)
      MXLIST(8) = MATNM(3)
      M1 = M - 1
      DO 460 I=1,6
      CALL MAGRSV(-M1,MXLIST(I),0,0,NAME)
 460  M1 = M1 - 1
      DO 464 I=7,8
      CALL DRFRSV(-M1,MXLIST(I),0,0,NAME)
 464  M1 = M1 - 1
      IF (MODE.EQ.2) GO TO 463
      IF (XSW) RETURN
 462  CALL DATA(M,2,1,2,NCELL)
      KF = 4
      IF (KA.EQ.2) KF = 8
      CALL DATA(M,1,1,KF,PAR)
      CALL STRN(M,NCELL,PAR)
      IF (MODE.EQ.3) RETURN
 463  MIN = MIN - 8
      RETURN
C  STR2 ENTRY. DESIGN COLLINS STRAIGHT SECTION FOR INSERTION.
 47   IF (MODE.EQ.3) GO TO 502
      CALL RESRV(M,MXPMY,MINZER,
     .  1,2,4,1,6,3,0,0
     .)
      CALL LOAD(M,
     .  1,2,4,1,0,0,0,0
     .)
      DO 501 I = 1,3
 501  MXLIST(I) = MATNM(I)
      MXLIST(4) = MATNM(5)
      DECODE (1,520,NAME) NM
      CALL DEFSET(M,NM,0,MXLIST,4,MXPMY)
      M1 = BDAT(M,1)
      M2 = M - 5
      ENCODE (10,504,INFF(2,M2)) M1,MATNM(6)
 504  FORMAT (A1,A4,5X)
      CALL RESRV(M-5,MXPMY,MINZER,
     .  13,3,0,0,0,0,0,0
     .)
```

```
      IF (MODE.EQ.2) GO TO 503
      IF (XSW) RETURN
  502 CALL STR2(M)
      IF (MODE.EQ.3) RETURN
  503 MIN = MIN - 5
      RETURN
C  STRPI ENTRY. DESIGN PI-TYPE STRAIGHT SECTION FOR INSERTION.
   48 IF (MODE.EQ.3) GO TO 512
      CALL RESRV(M,MXPMY,MINZER,
     .  1,2,5,1,10,3,0,0
     .)
      CALL LOAD(M,
     .  1,2,5,1,0,0,0,0
     .)
      DECODE (1,520,NAME) NM
      CALL DEFSET(M,NM,0,MATNM,4,MXPMY)
      IF (MODE.EQ.2) GO TO 513
      IF (XSW) RETURN
  512 CALL STRPI(M)
      IF (MODE.EQ.3) RETURN
  513 MIN = MIN - 4
      RETURN
C   SCOPE ENTRY
   49 CALL INP(M,0,0,
     .  0,0,0,0,0,0,0,0
     .)
      IF (MODE.EQ.2) RETURN
      IF (XSW) RETURN
      CALL SCOPCN
      RETURN
C  ACT OR DELE ENTRY
   80 CONTINUE
      IF (MODE.EQ.3) GO TO 800
C  IF KB=0, INPUT CONTAINS KA ELEMENTS
C  IF KB=1, KA ELEMENTS SHOULD BE CHANGED BEGINNING WITH ONE INPUT
      IF (KB.NE.0)  CALL INP(M,0,0,1,2,0,0,0,0,0,0)
      IF (KB.EQ.0)  CALL INP(M,0,0,1,-2,0,0,0,0,0,0)
      IF (MODE.EQ.2) RETURN
      IF(XSW) RETURN
  800 CALL XEQCON(M)
      RETURN
  520 FORMAT(A1)
      END
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
43	I	MATNM	PREVIOUSLY DIMENSIONED ARRAY. FIRST DIMENSIONS WILL BE RETAINED.
43	I	MXLIST	PREVIOUSLY DIMENSIONED ARRAY. FIRST DIMENSIONS WILL BE RETAINED.
62	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
65	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
73	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
74	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
78	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
84	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
85	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
89	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
105	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
108	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
110	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
126	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
129	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
131	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
145	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
148	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
150	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
164	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
168	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
170	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
200	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
206	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
208	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
230	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
232	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
234	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
246	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
248	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
250	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
255	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
256	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
258	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
266	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
267	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
269	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.

```
      OVERLAY(SYNCH,4,0)                                                  SW4        2
      PROGRAM SW4                                                         SW4        3
C  PROCESSES DATA FOR CURRENT INSTR.IF IN MODE 1 OR 2.                    SW4        4
C  PROCEEDS WITH CALC.BY CALLING APPROPRIATE ROUTINES IF MODE=1 OR 3.     SW4        5
C                                                                         SW4        6
      DIMENSION INF(24)                                                   SW4        7
      COMMON/INSTR/OPNAME,NAME,OP,KA,KB,OBJA,OBJB,NXTM,TRA,LQ,NQ,LIN,     INSTR      2
     1            NIN,LFL,NFL,LBC,NBC,NTOT,IX,KIND,TYPE,ROWS,COLS,EXTR,   INSTR      3
     2            M,NF,NB,NI,MSUBR,ISIGN                                  INSTR      4
      INTEGER OPNAME,OP,OBJA,OBJB,TRA,TYPE,ROWS,COLS,EXTR                 INSTR      5
                                                                          INSTR      6
                                                                          BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                           BLANK      3
      COMMON STORE(48000),IWORK(10)                                       86MARSIZ   1
      DIMENSION INFF(24,2000)                                             86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                            86MARSIZ   3
                                                                          BLANK      5
      COMMON/STORE/LMAX,LINF,LFILE,MAX,MIN,ICARD(11)                      STORE      2
*  LMAX AND MAX ARE DIMENSIONS OF STORE AND INFF                          STORE      3
                                                                          STORE      4
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,      CONTRL     2
     1         LDFLG,FIN                                                  CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN        CONTRL     4
                                                                          CONTRL     5
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL     DIM        2
                                                                          DIM        3
      COMMON/BMI/MI(5000)                                                 BMIL       1
      COMMON/NELS/NELS                                                    BMIL       2
      LEVEL 2,MI                                                          BMIL       3
                                                                          BMIL       4
                                                                          BMI        3
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER     SWTCH      2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                            SWTCH      3
                                                                          SWTCH      4
      COMMON/GRR/IERR,BSW,VSW,RSW,QSW,XSW                                 GRR        2
      LOGICAL IERR,BSW,VSW,RSW,QSW,XSW                                    GRR        3
                                                                          GRR        4
C                                                                         MOVTOG     2
      COMMON/MOVTOG/TOG                                                   MOVTOG     3
      LOGICAL TOG                                                         MOVTOG     4
C                                                                         MOVTOG     5
                                                                          SW4       16
      DIMENSION PAR(100),KREM(100)                                        SW4       17
      DIMENSION T(49),RW(3)                                               SW4       18
      INTEGER R7,ROT,PRD,HSXTP,HMAP                                       SW4       19
      EQUIVALENCE(OPNAME, INF(1)),(PAR,KREM)                              SW4       20
      DIMENSION V0(7)                                                     SW4       21
      DATA R7,ROT,MAP,PRD/5,6,7,8/                                        SW4       22
      DATA HSXTP,HMAP/4HSXTP,3HMAP/                                       SW4       23
      DATA RADEG/.0174532925/                                             SW4       24
      DATA (V0(I),I=1,7)/7*0./                                            SW4       25
C                                                                         SW4       26
*       IF OPNAME = NULL, IT IS A P OR C CARD.  NEEDS SPECIAL             SW4       27
*     HANDLING TO CONSTRUCT A SYNCH PAGE OR REM INSTRUCTION               SW4       28
                                                                          SW4       29
      NOP = OP - 64                                                       SW4       30
C     GO TO FXPT,TRK,PRD,SXTP,MAP,BMIS,EMIS,ROTZ,ROT,INV2,                SW4       31
```

```
      C         ELQ,MOVE,MODE,CON,VAR,NPOL,ORBC                                SW4683    1
            GO TO (40,35,34,33,1000,8,9,41,42,43,44,10,10,39,45,               SW4      33
           1       50,55,56,57) NOP                                            SW4683    2
                                                                               SW4      35
       8    CALL BMIS                                                          SW4      36
       10   RETURN                                                             SW4      37
       9    CALL EMIS                                                          SW4      38
            RETURN                                                             SW4      39
      C                                                                        SW4      40
      C                                                                        SW4      41
      *                      SXTP ENTRY                                        SW4      42
       33   IF (MODE.EQ.3) GO TO 331                                           SW4      43
            CALL INP(M,MAP,MINZER,                                             SW4      44
           .  4,1,0,0,0,0,0,0                                                  SW4      45
           .)                                                                  SW4      46
            IF (MODE.EQ.2) RETURN                                              SW4      47
            IF (XSW) RETURN                                                    SW4      48
       331  CALL DATA(M,1,1,4,PAR)                                             SW4      49
            IF (MODE.EQ.1)  PAR(4)=0.                                          SW4      50
            DO 332 I=1,6                                                       SW4      51
       332  V0(I)=0.                                                           SW4      52
            CALL SXTP(M,V0,PAR)                                                SW4      53
            RETURN                                                             SW4      54
                                                                               SW4      55
      C  PRD ENTRY  DEFINE A PRODUCT OF LINEAR AND NON-LINEAR TRANSFORMATION   SW4      56
       34   IF (MODE.EQ.3)  RETURN                                             SW4      57
            CALL INP (M,PRD,MINZER,                                            SW4      58
           .  KA,2,0,0,0,0,0,0                                                 SW4      59
           .)                                                                  SW4      60
            RETURN                                                             SW4      61
                                                                               SW4      62
      C  TRK ENTRY                                                             SW4      63
       35   IF (MODE.EQ.3) GO TO 352                                           SW4      64
            CALL INP(M,R7,MINZER,4,2,6,3,1,4,4,1)                              MY3SW4    2
            IF(MODE.EQ.2) RETURN                                               SW4      66
            IF (XSW) RETURN                                                    SW4      67
       352  CALL TRK(M)                                                        SW4      68
            RETURN                                                             SW4      69
                                                                               SW4      70
      C  FXPT ENTRY                                                            SW4      71
       40   IF (MODE.EQ.3) GO TO 401                                           SW4      72
      C  RESERVE SETS UP SPECIAL LQ LOCATIONS FOR FXPT                         SW4      73
            CALL RESRV(M,R7,0,                                                 SW4      74
           .5,1,2,2,5,3,0,0                                                    SW4683    3
           .)                                                                  SW4      76
            CALL LOAD(M,                                                       SW4      77
           .2,2,5,3,0,0,0,0                                                    SW4683    4
           .)                                                                  SW4      79
            CALL LOAD(M,                                                       SW4      80
           .  5,1,0,0,0,0,0,0                                                  SW4      81
           .)                                                                  SW4      82
            IF(MODE.EQ.2) GO TO 402                                            SW4      83
            IF (XSW) RETURN                                                    SW4      84
        401 CALL FXPT(M)                                                       SW4      85
       402  RETURN                                                             SW4      86
                                                                               SW4      87
      C  ROTZ ENTRY                                                            SW4      88
```

```
 41   IF(MODE.EQ.3) RETURN
      CALL INP(M,ROT,MINZER,
     .  1,1,0,0,0,0,0,0
     .)
      RETURN
C
C  ROT ENTRY-----ROTATE MATRIX
 42   IF (MODE.EQ.3) GO TO 420
      CALL INP(M,R7,MINZER,
     .  1,2,1,1,0,0,0,0
     .)
      IF (MODE.EQ.2) RETURN
      IF (XSW) RETURN
 420  MB = MDAT(M,1)
      THETA = FDAT(M,1)
      THETA = THETA*RADEG
      CALL ROTM(M,MB,THETA)
      RETURN

*                    INV2 ENTRY  ROTATE MATRIX 180 DEGREES AND REFLECT
 43   IF (MODE.EQ.3) GO TO 430
      CALL INP(M,R7,MINZER,
     .  1,2,0,0,0,0,0,0
     .)
      IF (MODE.EQ.2) RETURN
      IF (XSW) RETURN
 430  MB = MDAT(M,1)
      CALL INV2(M,MB)
      RETURN

C  ELQ ENTRY
 44   IF (MODE.EQ.3) RETURN
      CALL INP (M,MAP,MINZER,
     .    1,2,5,1,0,0,0,0)
      RETURN
C
C  MAP ENTRY
 1000 IF (MODE.EQ.3) RETURN
      CALL INP(M,MAP,MINZER,
     .  KA,1,0,0,0,0,0,0
     .)
      RETURN
C
C  MOVE ENTRY
 39   IF (MODE.EQ.3) GO TO 390
      CALL RESRV(M,R7,MINZER,3,2,10,1,0,0,0,0)
      CALL LOAD(M,3,2,4,1,0,0,0,0)
      MQ = MDAT(M,1)
      NOPQ = INFF(1,MQ)
      IF (NOPQ.EQ.HSXTP)  INFF(20,M) = HMAP
      IF (MODE.EQ.2) RETURN
      IF (XSW) RETURN
 390  DO 391 I=1,7
 391  V0(I)=0.
      CALL RANGET(NRN)
      CALL MOVE(M,M,V0)
      TOG = .TRUE.
```

```
      CALL RANSET(NRN)                                                  MY3SW4     5
      RETURN                                                            SW4      144
                                                                        SW4      145
C     MOD ENTRY                                                         SW4      146
C                                                                       SW4      147
C       THIS INSTRUCTION MAY BE USED TO SEE EXPLICITLY THE EFFECT OF    SW4      148
C  SUBROUTINE REVMAT, WHICH MAKES A NEW MATRIX WHIS IS THE TRANSFER     SW4      149
C  MATRIX FOR RAYS IN THE NEIGHBORHOOD OF A REFERENCE TRAJECTORY.       SW4      150
C  EXAMPLE                                                              SW4      151
C                                                                       SW4      152
C   V    PVEC         .005       0.       0.      0.     0.     0.     0.   SW4      153
C   QF   MAG          1.25       .075     1.                            SW4      154
C   QFV  MOD          QF    V                                           SW4      155
C                                                                       SW4      156
 45   IF (MODE.EQ.3)  GO TO 450                                         SW4      157
      CALL INP(M,R7,MINZER,2,2,0,0,0,0,0,0)                             SW4      158
      IF (MODE.EQ.2) RETURN                                             SW4      159
      IF (XSW)  RETURN                                                  SW4      160
 450  MB = MDAT(M,1)                                                    SW4      161
      MV = MDAT(M,2)                                                    SW4      162
      CALL DATA (MV,1,1,7,V0)                                           SW4      163
      CALL REVMAT(M,MB,V0)                                              SW4      164
      RETURN                                                            SW4      165
C                                                                       SW4      166
C             CON ENTRY                                                 SW4      167
 50   IF (MODE.EQ.3)  RETURN                                            SW4      168
      CALL INP(M,0,0,2,2,2,1,0,0,0,0)                                   SW4      169
      RETURN                                                            SW4      170
C                                                                       SW4      171
C             PAR ENTRY                                                 SW4      172
 55   IF (MODE.EQ.3)  RETURN                                            SW4      173
      CALL INP(M,0,0,KA,1,0,0,0,0,0,0)                                  SW4      174
      RETURN                                                            SW4      175
C                                                                       SW4      176
C               NPOL ENTRY                                              SW4      177
C  Q    NPOL  N  J     L      CM      BRHO                              SW4      178
 56   IF (MODE.EQ.3)  GO TO 561                                         SW4      179
      CALL INP (M,MAP,MINZER,3,1,0,0,0,0,0,0)                           SW4      180
      IF (MODE.EQ.2)  RETURN                                            SW4      181
      IF (XSW)  RETURN                                                  SW4      182
C   PRESET MATRIX = UNIT MATRIX                                         SW4      183
 561  CALL RTRV7(MUNIT,T,RW)                                            SW4      184
      CALL STOR7(M,T,RW)                                                SW4      185
      RETURN                                                            SW4      186
C                                                                       SW4      187
C  ORBC ENTRY                                                           SW4683     5
 57   IF(MODE.EQ.3) RETURN                                              SW4683     6
      CALL INP (M,0,0,4,2,2,1,0,0,0,0)                                  SW4JN983   1
      IF(MODE.EQ.2) RETURN                                              SW4683     8
      IF(XSW) RETURN                                                    SW4683     9
 571  CALL ORBIT(M)                                                     SW4683    10
      RETURN                                                            SW4683    11
C                                                                       SW4683    12
       END                                                              SW4      188
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
63	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
65	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
73	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
74	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
80	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
83	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
87	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
92	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
93	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
95	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
110	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
112	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
115	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
119	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
126	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
127	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
132	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
139	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
140	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
143	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
146	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
149	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
152	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
156	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
165	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
166	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
173	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
188	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
189	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
194	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
197	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
199	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
202	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
204	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
210	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
211	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
215	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
218	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
220	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
221	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
223	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.

```
      OVERLAY(SYNCH,7,0)                                                SW7        2
      PROGRAM SW7                                                       SW7        3
C  PROCESSES DATA FOR CURRENT INSTR.IF IN MODE 1 OR 2.                  SW7        4
C  PROCEEDS WITH CALC.BY CALLING APPROPRIATE ROUTINES IF MODE=1 OR 3.   SW7        5
C                                                                       SW7        6
      DIMENSION INF(24)                                                 SW7        7
      COMMON/INSTR/OPNAME,NAME,OP,KA,KB,OBJA,OBJB,NXTM,TRA,LQ,NQ,LIN,   INSTR      2
     1             NIN,LFL,NFL,LBC,NBC,NTOT,IX,KIND,TYPE,ROWS,COLS,EXTR, INSTR     3
     2             M,NF,NB,NI,MSUBR,ISIGN                               INSTR      4
      INTEGER OPNAME,OP,OBJA,OBJB,TRA,TYPE,ROWS,COLS,EXTR               INSTR      5
                                                                        INSTR      6
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/STORE/LMAX,LINF,LFILE,MAX,MIN,ICARD(11)                    STORE      2
*  LMAX AND MAX ARE DIMENSIONS OF STORE AND INFF                        STORE      3
                                                                        STORE      4
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1        LDFLG,FIN                                                 CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL   DIM        2
                                                                        DIM        3
      COMMON/BMI1/MI1(16000)                                            NV3BMI1    1
      COMMON/NELS1/NELS1                                                BMI1L      2
      LEVEL 2,MI1                                                       BMI1L      3
                                                                        BMI1L      4
C                                                                       BMI1       3
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER   SWTCH      2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                          SWTCH      3
                                                                        SWTCH      4
      COMMON/GRR/IERR,BSW,VSW,RSW,QSW,XSW                               GRR        2
      LOGICAL IERR,BSW,VSW,RSW,QSW,XSW                                  GRR        3
                                                                        GRR        4
                                                                        SW7       16
      DIMENSION PAR(100),KREM(100)                                      SW7       17
      INTEGER R7,ROT,PRD,HSXTP,HMAP,PVECS                               SW7       18
      INTEGER BDAT                                                      86SW7      1
      DIMENSION ZZ(1)                                                   86SW7      2
      EQUIVALENCE(OPNAME, INF(1)),(PAR,KREM)                            SW7       19
      DIMENSION V0(7)                                                   SW7       20
      DATA R7,ROT,MAP,PRD/5,6,7,8/                                      SW7       21
      DATA HSXTP,HMAP/4HSXTP,3HMAP/                                     SW7       22
      DATA RADEG/.0174532925/                                           SW7       23
      DATA (V0(I),I=1,7)/7*0./                                          SW7       24
      DATA IBLNK,MXPMY,PVECS,R7,ROT,PRD/1H ,1,2,5,6,8/                  SW7       25
C                                                                       SW7       26
*       IF OPNAME = NULL, IT IS A P OR C CARD.  NEEDS SPECIAL           SW7       27
*     HANDLING TO CONSTRUCT A SYNCH PAGE OR REM INSTRUCTION             SW7       28
C                                                                       SW7       29
      NOP = OP - 160                                                    SW7       30
      ZZ(1)=0.                                                          86SW7      3
      GO TO (36,37,38,39,40,41) NOP                                     86SW7      4
C     GO TO TRKB,SOLV,SMIN,TRKE,TRKM,IBET                               86SW7      5
```

```
C
C  TRKB ENTRY
 36   IF (MODE.EQ.3)  GO TO 360
      CALL RESRV(M,MXPMY,MINZER,
     . 4,2,6,3,2,1,0,0
     .)
      CALL LOAD(M,
     . 4,2,4,3,2,1,0,0)
      CALL STDAT(M,3,6,1,ZZ)
C    6TH INTEGER IS A FLAG TO TELL TRKB IF IT HAS BEEN CALLED YET.
      IP1 = IDAT(M,1)
      IP2 = IDAT(M,2)
      NAMVEC = BDAT(M,3)
      NAMOP = INFF(1,M)
      NCOL = 20
      IF(NAMVEC.EQ.4H    ) NCOL = 14
      CALL STDAT(M,3,5,1,NCOL)
      MBML = MDAT(M,1)
      NDAT = INFF(17,MBML)
      CALL MIFILL(MBML,1,NDAT,NEL,MI1)
      IF (KA.NE.0.OR.KB.NE.0)  GO TO 361
      NQ3 = NCOL*(NEL+1)
      KB = NEL
      GO TO 362
 361  IF (KA.LE.-1)  GO TO 363
      NQ3 = NCOL*(KB-KA+1)
 362  LQ3 = LFILE + 1
      LFILE = LQ3 + NQ3
      NTOT = NTOT + NQ3
      INFF(5,M) = NEL
      INFF(18,M) = NTOT
      INFF(24,M) = LQ3
 363  IF (IP1.NE.0.OR.IP2.NE.0)  GO TO 364
      CALL REPINT(M,1,0)
      CALL REPINT(M,2,NEL)
 364  IF (MODE.EQ.2)  RETURN
      IF (XSW)  RETURN
  360 CALL TRKB(M)
      RETURN
C
C  TRKE ENTRY
 39   IF (MODE.EQ.3)  GO TO 390
      CALL RESRV(M,MXPMY,MINZER,
     . 4,2,6,3,2,1,0,0
     .)
      CALL LOAD(M,
     . 4,2,4,3,2,1,0,0)
      CALL STDAT(M,3,6,1,ZZ)
C    6TH INTEGER IS A FLAG TO TELL TRKE IF IT HAS BEEN CALLED YET.
      IP1 = IDAT(M,1)
      IP2 = IDAT(M,2)
      NAMVEC = BDAT(M,3)
      NAMOP = INFF(1,M)
      NCOL = 20
      IF(NAMVEC.EQ.4H    ) NCOL = 14
      CALL STDAT(M,3,5,1,NCOL)
      MBML = MDAT(M,1)
```

```
      NDAT = INFF(17,MBML)
      CALL MIFILL(MBML,1,NDAT,NEL,MI1)
      IF (KA.NE.0.OR.KB.NE.0)  GO TO 391
      NQ3 = NCOL*(NEL+1)
      KB = NEL
      GO TO 392
  391 IF (KA.LE.-1)  GO TO 393
      NQ3 = NCOL*(KB-KA+1)
  392 LQ3 = LFILE + 1
      LFILE = LQ3 + NQ3
      NTOT = NTOT + NQ3
      INFF(5,M) = NEL
      INFF(18,M) = NTOT
      INFF(24,M) = LQ3
  393 IF (IP1.NE.0.OR.IP2.NE.0)  GO TO 394
      CALL REPINT(M,1,0)
      CALL REPINT(M,2,NEL)
  394 IF (MODE.EQ.2)  RETURN
      IF (XSW)  RETURN
  390 CALL TRKE(M)
      RETURN
C
C  TRKM ENTRY
 40   IF (MODE.EQ.3)  GO TO 400
      CALL RESRV(M,MXPMY,MINZER,
     . 4,2,6,3,2,1,0,0
     .)
      CALL LOAD(M,
     . 4,2,4,3,2,1,0,0)
      CALL STDAT(M,3,6,1,ZZ)
C    6TH INTEGER IS A FLAG TO TELL TRKM IF IT HAS BEEN CALLED YET.
      IP1 = IDAT(M,1)
      IP2 = IDAT(M,2)
      NAMVEC = BDAT(M,3)
      NAMOP = INFF(1,M)
      NCOL = 20
      IF(NAMVEC.EQ.4H    ) NCOL = 14
      CALL STDAT(M,3,5,1,NCOL)
      MBML = MDAT(M,1)
      NDAT = INFF(17,MBML)
      CALL MIFILL(MBML,1,NDAT,NEL,MI1)
      IF (KA.NE.0.OR.KB.NE.0)  GO TO 401
      NQ3 = NCOL*(NEL+1)
      KB = NEL
      GO TO 402
  401 IF (KA.LE.-1)  GO TO 403
      NQ3 = NCOL*(KB-KA+1)
  402 LQ3 = LFILE + 1
      LFILE = LQ3 + NQ3
      NTOT = NTOT + NQ3
      INFF(5,M) = NEL
      INFF(18,M) = NTOT
      INFF(24,M) = LQ3
  403 IF (IP1.NE.0.OR.IP2.NE.0)  GO TO 404
      CALL REPINT(M,1,0)
      CALL REPINT(M,2,NEL)
  404 IF (MODE.EQ.2)  RETURN
```

```
      IF (XSW)  RETURN
  400 CALL TRKM(M)
      RETURN
C
C  IBET ENTRY
 41   IF (MODE.EQ.3) RETURN
      CALL INP(M,0,0,12,1,0,0,0,0,0,0)
      RETURN
C  SOLV ENTRY
C
C  SOLV ENTRY
 37   IF (MODE.EQ.3) GO TO 370
      NF1 = 1 +  2*KA + 3*KB
C   NB1 = 4 + 2*KA + 6*KB
C   NI1 = 5 + 6*KA
      NB1 = 4 + 8*KA + 6*KB
      NI1 = 5
      CALL RESRV(M,0,0,
     . NF1,1,NB1,2,NI1,3,0,0
     .)
      CALL LOAD(M,4,2,5,3,1,1,0,0)
      DO 371 I=1,KA
      CALL LOAD(M,
     .8,2,2,1,0,0,0,0
     .)
 371  CONTINUE
 372  DO 373 J=1,KB
      CALL LOAD(M,
     .  6,2,3,1,0,0,0,0
     .)
 373  CONTINUE
      IF (MODE.EQ.2)  RETURN
      IF (XSW)  RETURN
 370  CALL SOLV(M)
      RETURN

C  SMIN ENTRY
 38   IF (MODE.EQ.3) GO TO 380
      NF1 = 5*KA
      NB1 = 2*KA
      CALL RESRV(M,0,0,
     .  NF1,1,NB1,2,0,0,0,0
     .)
      DO 381 I=1,KA
      CALL LOAD(M,
     .  2,2,5,1,0,0,0,0
     .)
 381  CONTINUE
      IF (XSW) RETURN
 380  CALL SMIN(M)
      RETURN
       END
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
93	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
94	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
96	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
132	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
133	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
135	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
171	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
172	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
174	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
177	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
179	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
203	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
204	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
206	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
220	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
222	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.

```
      OVERLAY(SYNCH,5,0)                                              SW5        2
      PROGRAM SW5                                                     SW5        3
C  PROCESSES DATA FOR CURRENT INSTR.IF IN MODE 1 OR 2.                SW5        4
C  PROCEEDS WITH CALC.BY CALLING APPROPRIATE ROUTINES IF MODE=1 OR 3. SW5        5
C                                                                     SW5        6
      DIMENSION INF(24)                                               SW5        7
      COMMON/INSTR/OPNAME,NAME,OP,KA,KB,OBJA,OBJB,NXTM,TRA,LQ,NQ,LIN, INSTR      2
     1           NIN,LFL,NFL,LBC,NBC,NTOT,IX,KIND,TYPE,ROWS,COLS,EXTR, INSTR     3
     2           M,NF,NB,NI,MSUBR,ISIGN                               INSTR      4
      INTEGER OPNAME,OP,OBJA,OBJB,TRA,TYPE,ROWS,COLS,EXTR             INSTR      5
                                                                      INSTR      6
                                                                      BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                       BLANK      3
      COMMON STORE(48000),IWORK(10)                                   86MARSIZ   1
      DIMENSION INFF(24,2000)                                         86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                        86MARSIZ   3
                                                                      BLANK      5
      COMMON/STORE/LMAX,LINF,LFILE,MAX,MIN,ICARD(11)                  STORE      2
*  LMAX AND MAX ARE DIMENSIONS OF STORE AND INFF                      STORE      3
                                                                      STORE      4
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,  CONTRL     2
     1        LDFLG,FIN                                               CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN    CONTRL     4
                                                                      CONTRL     5
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL DIM        2
                                                                      DIM        3
      COMMON/BMI/MI(5000)                                             BMIL       1
      COMMON/NELS/NELS                                                BMIL       2
      LEVEL 2,MI                                                      BMIL       3
                                                                      BMIL       4
                                                                      BMI        3
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER SWTCH      2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                        SWTCH      3
                                                                      SWTCH      4
      COMMON/GRR/IERR,BSW,VSW,RSW,QSW,XSW                             GRR        2
      LOGICAL IERR,BSW,VSW,RSW,QSW,XSW                                GRR        3
                                                                      GRR        4
                                                                      SW5       16
      INTEGER BDAT,PVECS                                              SW5       17
      DIMENSION T(49),RW(3)                                           SW5       18
      DIMENSION PAR(100),KREM(100)                                    SW5       19
      INTEGER R7                                                      SW5       20
      EQUIVALENCE (OPNAME,INF(1)),(PAR,KREM)                          SW5       21
      DATA PVECS,MAT,R7/2,3,5/                                        SW5       22
      DATA ISCL/9/                                                    SW5       23
C                                                                     SW5       24
*       IF OPNAME = NULL, IT IS A P OR C CARD.  NEEDS SPECIAL         SW5       25
*     HANDLING TO CONSTRUCT A SYNCH PAGE OR REM INSTRUCTION           SW5       26
                                                                      SW5       27
      IF (OP.EQ.123) GO TO 123                                        SW5       28
      NOP = OP - 99                                                   MY3SW5     1
      IF(OP.GT.111) GO TO 75                                          SW5       30
      GO TO (40,43,42,64,65,71,66,68,67,62,72,74) NOP                 MY3SW5     2
* GO TO PBML,MAT,VEC,PVEC,MXV,CPLX,EQIL,EVEC,PRTV,PRV7,CVEC           MY3SW5     3
C         ,LIST,SIN,COS,EXP,ALOG,SQ,SQRT,TAN,ASIN,ACOS,ATAN,ABS       SW5       33
C        ,CALC                                                        SW5       34
C                                                                     SW5       35
```

```
C      PBML ENTRY     PRINT BEAM LINE                                      MY3SW5      4
 40    IF (MODE.EQ.3)  GO TO 401                                           MY3SW5      5
       CALL INP(M,0,0,1,-2,0,0,0,0,0,0)                                    MY3SW5      6
       IF (MODE.EQ.2)  RETURN                                              MY3SW5      7
 401   IF (XSW)  RETURN                                                    MY3SW5      8
       CALL PBML(M)                                                        MY3SW5      9
       RETURN                                                              MY3SW5     10
C                                                                          MY3SW5     11
C  VEC ENTRY -- KB VECTORS EACH WITH KA COMPONENTS                         SW5        36
 42    IF (MODE.EQ.3)  RETURN                                              SW5        37
       IF (KB.NE.0) GO TO 421                                              SW5        38
       KB = 1                                                              SW5        39
       INFF(5,M) = KB                                                      SW5        40
 421   ROWS = KA                                                           SW5        41
       COLS = KB                                                           SW5        42
       MN = KA*KB                                                          SW5        43
       CALL RESRV(M,MAT,-1,                                                SW5        44
      .    MN,1,0,0,0,0,0,0)                                               SW5        45
       DO 420 J=1,KB                                                       SW5        46
 420   CALL LOAD(M,KA,1,0,0,0,0,0,0)                                       SW5        47
       RETURN                                                              SW5        48
C  MAT ENTRY---MATRIX OF KA ROWS, KB COLS (INPUT BY ROWS)                  SW5        49
 43    IF (MODE.EQ.3) GO TO 430                                            SW5        50
       MN = KA*KB                                                          SW5        51
       CALL RESRV(M,MAT,MN,                                                SW5        52
      .  MN,1,0,0,0,0,0,0                                                  SW5        53
      .)                                                                   SW5        54
 431   DO 432 J=1,KA                                                       SW5        55
 432   CALL LOAD(M,                                                        SW5        56
      .   KB,1,0,0,0,0,0,0)                                                SW5        57
 430   CALL TRNSPOS                                                        SW5        58
       RETURN                                                              SW5        59
C                                                                          SW5        60
C  PRINT PARTICLE VECTORS                                                  SW5        61
 62    IF (MODE.EQ.3) GO TO 620                                            SW5        62
       CALL INP(M,0,0,                                                     SW5        63
      .  KA,2,0,0,0,0,0,0                                                  SW5        64
      .)                                                                   SW5        65
       IF (MODE.EQ.2) RETURN                                               SW5        66
       IF (XSW) RETURN                                                     SW5        67
 620   CALL PRNTV7(M,KA)                                                   SW5        68
       RETURN                                                              SW5        69
                                                                           SW5        70
C  PVEC ENTRY.  READ IN KA VECTORS (COL) OF 6 ROWS.  SET 7TH TO 1.0        SW5        71
 64    IF(MODE.EQ.3) RETURN                                                SW5        72
       IF (KA.EQ.0) KA = 1                                                 SW5        73
       V7 = 0.0                                                            SW5        74
       DECODE (10,641,ICARD(2))KH                                          SW5        75
 641   FORMAT (6X,A3,1X)                                                   SW5        76
       IF (KB.EQ.1.OR.KH.EQ.1H ) V7=1.                                     SW5        77
C  PVECS KIND RESERVES 7 X KA                                              SW5        78
       KC = 7*KA                                                           SW5        79
       COLS = KA                                                           SW5        80
       CALL RESRV(M,PVECS,MINZER,KC,1,0,0,0,0,0,0)                         SW5        81
       DO 640 I=1,KA                                                       SW5        82
       CALL LOAD(M,                                                        SW5        83
      .  6,1,0,0,0,0,0,0                                                   SW5        84
```

```
     .)
      NF = NF + 1
      NPL = 7*I
      CALL REPFLT(M,NPL,V7)
  640 CONTINUE
      RETURN
C  MXV ENTRY.  MATRIX TIMES VECTOR
 65   IF(MODE.EQ.3) GO TO 650
      ROWS = 7
      COLS = 1
      CALL INP(M,PVECS,7,2,2,0,0,0,0,0,0,0)
      IF (MODE.EQ.2) RETURN
      IF (XSW) RETURN
C  RETRIEVE ELEMENTNUMBER OF INPUT MATRIX AND VECTOR.FIND LQ OF VECTORS
  650 M1 = MDAT(M,1)
      NM = BDAT(M,2)
      M2 = ELNUM(NM)
      LOC = 1
      IF(.NOT.VCSW) GO TO 651
      DECODE (10,652,NM) NUM
  652 FORMAT (1X,I1,8X)
      LOC = (NUM-1)*7+1
  651 CALL DATA (M2,5,LOC,7,PAR(8))
      CALL RTRV7(M1,T,RW)
      CALL MXV7(T,PAR(8),PAR(1))
      CALL STDAT(M,5,1,7,PAR)
      RETURN
C  EQIL ENTRY
 66   IF (MODE.EQ.3) GO TO 660
      KA = 3
      CALL INP(M,PVECS,MINZER,
     .  2,2,0,0,0,0,0,0
     .)
C  SET KA FOR NUMBER OF COLUMN VECTORS
      IF (MODE.EQ.2) RETURN
      IF (XSW) RETURN
  660 CALL EQIL(M)
      RETURN
C  PRINT VECTORS   ANY SIZE
 67   IF(MODE.EQ.3) GO TO 670
      CALL INP(M,0,0,
     .  KA,2,0,0,0,0,0,0
     .)
      IF (MODE.EQ.2) RETURN
      IF (XSW) RETURN
  670 CALL PRINTV(M,KA)
      RETURN
C  EVEC ENTRY
 68   IF (MODE.EQ.3) GO TO 680
      CALL INP(M,0,112,
     .  2,2,0,0,0,0,0,0
     .)
      IF (MODE.EQ.2) RETURN
      IF (XSW) RETURN
  680 CALL EVEC(M)
      RETURN
```

```
      C  CPLX ENTRY -----A COMPLEX NUMBER
       71   IF (MODE.EQ.3) RETURN
            TYPE = 4HCPLX
            KN = ISCL
            CALL INP(M,KN,0,
          .  2,1,0,0,0,0,0,0
          .)
            RETURN
      C  CVEC ENTRY -----COMPLEX PARTICLE VECTOR,KA, IN NUMBER
      C  IF KB=BLANK OR 1, ROW 7 IS 1.  IF KB = 0, ROW7 IS 0.
       72   IF (MODE.EQ.3) RETURN
            V7R = 0.
            V7I=0.
            IF (KB.EQ.1.OR.KB.EQ.1H ) V7R = 1.
            IF(KA.EQ.0) KA=1
            KC = 14*KA
            ROWS = 7
            COLS = KA
            TYPE = 4HCPLX
            CALL RESRV(M,PVECS,0,
          .  KC,1,0,0,0,0,0,0
          .)
            DO 721 I=1,KA
            CALL LOAD(M,
          .  12,1,0,0,0,0,0,0
          .)
            NF = NF + 2
            NPL = 14*(I-1) + 13
            CALL REPFLT(M,NPL,V7R)
            NPL = NPL + 1
            CALL REPFLT (M,NPL,V7I)
       721  CONTINUE
            RETURN
      C  LIST ENTRY
      C
      C   A      LIST    KA     A1    A2     A3      A4    A5
      C
      C    THE LIST INSTRUCTION SPECIFIES A SET OF ELEMENTS THAT ARE TO BE
      C  USED SUCCESSIVELY.  THE NAME APPEARS IN A BML INSTRUCTION.   EACH
      C  TIME THE SAME LIST NAME IS ENCOUNTERED, THE NEXT AI IS USED.
      C
      C  EXAMPLE
      C----------------------------------------------------------------------
      C   QQ      LIST        Q1    Q2    Q3    Q4    Q5
      C   L      DRF         3.
      C   .S     BML            5(Q     L        )
      C----------------------------------------------------------------------
      C   .S IS EQUIVALENT TO
      C   .S     BML         Q1    L     Q2    L     Q3    L     Q4    L     Q5    L
      C
       74   IF (MODE.EQ.3) GO TO 741
            CALL INP(M,0,0,1,-2,0,0,0,0,0,0)
       741  IF (KB.NE.0)  RETURN
             KB=1
             INFF(5,M)=1
            RETURN
```

```
      C  SCALAR FUNCTIONS
       75    IF (MODE.EQ.3) GO TO 750
             KN = ISCL
             CALL INP(M,KN,1,1,1,0,0,0,0,0,0)
             IF (MODE.EQ.2.OR.XSW) RETURN
       750   XX = FDAT(M,1)
             NNOP = OP - 111
             GO TO (76,77,78,79,80,81,82,83,84,85,86) NNOP
       76    YY = SIN(XX)
             GO TO 99
       77    YY = COS(XX)
             GO TO 99
       78    YY = EXP(XX)
             GO TO 99
       79    YY = ALOG(XX)
             GO TO 99
       80    YY = XX*XX
             GO TO 99
       81    YY = SQRT(XX)
             GO TO 99
       82    YY = TAN(XX)
             GO TO 99
       83    YY = ASIN(XX)
             GO TO 99
       84    YY = ACOS(XX)
             GO TO 99
       85    YY = ATAN(XX)
             GO TO 99
       86    YY = ABS(XX)
       99    CALL STDAT(M,5,1,1,YY)
             RETURN

      C  CALC ENTRY
       123   IF (MODE.EQ.3) GO TO 1231
             CALL INP(M,0,5,1,-2,0)
             IF (MODE.EQ.2.OR.XSW) RETURN
       1231   CALL CALC(M)
             RETURN
             END
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
61	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
62	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
64	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
67	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
78	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
89	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
96	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
97	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
99	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
102	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
120	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
126	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
127	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
141	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
149	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
150	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
152	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
158	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
159	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
161	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
167	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
168	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
170	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
173	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
179	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
182	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
204	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
224	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
227	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
233	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
259	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
263	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
264	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
266	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.

```
      OVERLAY(SYNCH,6,0)                                                SW6        2
      PROGRAM SW6                                                       SW6        3
C  PROCESSES DATA FOR CURRENT INSTR.IF IN MODE 1 OR 2.                  SW6        4
C  PROCEEDS WITH CALC.BY CALLING APPROPRIATE ROUTINES IF MODE=1 OR 3.   SW6        5
C                                                                       SW6        6
      DIMENSION INF(24)                                                 SW6        7
      COMMON/INSTR/OPNAME,NAME,OP,KA,KB,OBJA,OBJB,NXTM,TRA,LQ,NQ,LIN,   INSTR      2
     1            NIN,LFL,NFL,LBC,NBC,NTOT,IX,KIND,TYPE,ROWS,COLS,EXTR, INSTR      3
     2            M,NF,NB,NI,MSUBR,ISIGN                                INSTR      4
      INTEGER OPNAME,OP,OBJA,OBJB,TRA,TYPE,ROWS,COLS,EXTR               INSTR      5
                                                                        INSTR      6
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/STORE/LMAX,LINF,LFILE,MAX,MIN,ICARD(11)                    STORE      2
*  LMAX AND MAX ARE DIMENSIONS OF STORE AND INFF                        STORE      3
                                                                        STORE      4
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1         LDFLG,FIN                                                CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL   DIM        2
                                                                        DIM        3
      COMMON/BMI/MI(5000)                                               BMIL       1
      COMMON/NELS/NELS                                                  BMIL       2
      LEVEL 2,MI                                                        BMIL       3
                                                                        BMIL       4
                                                                        BMI        3
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER   SWTCH      2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                          SWTCH      3
                                                                        SWTCH      4
      COMMON/GRR/IERR,BSW,VSW,RSW,QSW,XSW                               GRR        2
      LOGICAL IERR,BSW,VSW,RSW,QSW,XSW                                  GRR        3
                                                                        GRR        4
                                                                        SW6       16
       EQUIVALENCE(OPNAME,INF(1))                                       SW6       17
C                                                                       SW6       18
*       IF OPNAME = NULL, IT IS A P OR C CARD.  NEEDS SPECIAL           SW6       19
*     HANDLING TO CONSTRUCT A SYNCH PAGE OR REM INSTRUCTION             SW6       20
                                                                        SW6       21
      NOP = OP - 150                                                    SW6       22
      GO TO (76,77,78,79,83,84,85) NOP                                  SW6       23
                                                                        SW6       24
C   GO TO BEAM,DKE,EDRF,SOL,CYEM,BVAL,CYAE                              SW6       25
                                                                        SW6       26
C  BEAM ENTRY. DEFINE A BEAM PARTICLE                                   SW6       27
 76   CONTINUE                                                          SW6       28
      IF (MODE.EQ.3) GO TO 761                                          SW6       29
      CALL RESRV(M,0,10,                                                SW6       30
     .  8,1,1,3,0,0,0,0                                                 SW6       31
     .)                                                                 SW6       32
C  LOAD 4 COMPONENTS OF VECTOR FROM CARD 1                              SW6       33
      CALL LOAD(M,                                                      SW6       34
     .  4,1,0,0,0,0,0,0                                                 SW6       35
```

```
     .)
C  LOAD SECOND CARD
      CALL LOAD(M,
     .  4,1,0,0,0,0,0,0
     .)
      IF (MODE.EQ.2) RETURN
      IF (XSW) RETURN
 761  CALL BEAM(M)
      RETURN
C  DKE ENTRY.  ENERGY GAIN
 77   CONTINUE
      IF (MODE.EQ.3) RETURN
      CALL INP(M,0,0,
     .  1,1,0,0,0,0,0,0
     .)
      RETURN
C  EDRF ENTRY.
 78   CONTINUE
      IF (MODE.EQ.3) RETURN
      CALL INP(M,0,0,
     .  1,1,0,0,0,0,0,0
     .)
      RETURN
C  SOL ENTRY
 79   CONTINUE
      IF (MODE.EQ.3) RETURN
      CALL INP(M,0,0,
     .  3,1,0,0,0,0,0,0
     .)
      RETURN
C

*                    CYEM ENTRY
 83   IF (MODE.EQ.3) GO TO 831
      CALL INP(M,0,4,
     .  1,2,5,1,0,0,0,0
     .)
      IF (MODE.EQ.2) RETURN
      IF (XSW) RETURN
 831  CALL CYEM(M)
      RETURN
C          BVAL ENTRY
 84   IF (MODE.EQ.3) GO TO 841
      CALL INP(M,0,4,6,1,0,0,0,0,0,0)
      IF (MODE.EQ.2) RETURN
      IF (XSW) RETURN
 841  CALL BVAL(M)
      RETURN

*                    CYAE ENTRY
 85   IF (MODE.EQ.3) GO TO 851
      CALL INP(M,0,0,
     .  2,2,3,1,0,0,0,0
     .)
      IF (MODE.EQ.2) RETURN
      IF (XSW) RETURN
 851  CALL CYAE(M)
```

```
115          RETURN                                                SW6      93
             END                                                   SW6      94
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
63	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
64	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
66	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
69	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
73	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
76	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
80	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
83	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
87	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
95	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
96	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
98	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
102	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
103	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
105	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
112	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
113	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.
115	I		RETURN STATEMENT APPEARS IN MAIN PROGRAM.

```
*** YANK, SELYANK, OR YANKDECK IDENT   MVR      NOT KNOWN ***

*** YANK, SELYANK, OR YANKDECK IDENT   EDC77    NOT KNOWN ***

*** YANK, SELYANK, OR YANKDECK IDENT   EDC77    NOT KNOWN ***
```

```
      SUBROUTINE AFD(M,J,DELTA)                                          AFD      2
C  INCREMENTS BY DELTA THE J-TH FL PT PARAMETER OF ELEMENT M AND        AFD      3
C  RETURNS NEW VALUE TO STORAGE.                                        AFD      4
      F=FDAT(M,J)                                                        AFD      5
      F=F+DELTA                                                          AFD      6
      CALL REPFLT(M,J,F)                                                 AFD      7
      RETURN                                                             AFD      8
      END                                                                AFD      9
```

```
      SUBROUTINE ALTC(M,KA,P1)                                          ALTC       2
                                                                        ALTC       3
*        DESIGN A SYMMETRIC CELL TO OBTAIN A SPECIFIED MUX VALUE        ALTC       4
      DIMENSION P1(1)                                                   ALTC       5
                                                                        ALTC       6
      COMMON/GRR/IERR,BSW,VSW,RSW,QSW,XSW                               GRR        2
      LOGICAL IERR,BSW,VSW,RSW,QSW,XSW                                  GRR        3
                                                                        GRR        4
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER   SWTCH      2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                          SWTCH      3
                                                                        SWTCH      4
                                                                        ALTC       9
      EXTERNAL AUXC                                                     ALTC      10
      COMMON/MAGL/P(10),D,K                                             ALTC      11
                                                                        ALTC      12
      DO 100 I=1,8                                                      ALTC      13
  100 P(I)=P1(I)                                                        ALTC      14
      K=KA                                                              ALTC      15
      D=COS(6.283185307*P(8))                                           ALTC      16
      CYCSWT=.TRUE.                                                     ALTC      17
      GUESS=P(K)                                                        ALTC      18
      CALL GRT (1,GUESS,1,AUXC)                                         ALTC      19
      IF(IERR) RETURN                                                   ALTC      20
      CALL HED                                                          ALTC      21
      NAM = MNAME(M)                                                    ALTC      22
      WRITE (3,1000) NAM,(P(I),I=1,6)                                   ALTC      23
      CALL HED                                                          ALTC      24
      CYCSWT=.FALSE.                                                    ALTC      25
C  STORE THE FINAL VALUE OF THE ALTERED PARAMETER AS DATA FOR CELL      ALTC      26
      CALL REPFLT(M,K,P(K))                                             ALTC      27
      CALL CELL(M,P)                                                    ALTC      28
 1000 FORMAT (//19H NEW VALUES OF ALTC,2X,A5,11H PARAMETERS/            ALTC      29
     13X,6(3X,F12.6))                                                   ALTC      30
      RETURN                                                            ALTC      31
      END                                                               ALTC      32
```

```
      SUBROUTINE ARGINP(ND,I,NSETS,                                   ARGINP   2
     . N1,I1,N2,I2,N3,I3,N4,I4                                        ARGINP   3
     .)                                                               ARGINP   4
      DIMENSION ND(10),I(10)                                          ARGINP   5
      NSETS=0                                                         ARGINP   6
      IF (N1.EQ.0) GO TO 8                                            ARGINP   7
      NSETS=1                                                         ARGINP   8
      ND(1)=N1                                                        ARGINP   9
      I(1)=I1                                                         ARGINP  10
      IF (N2.EQ.0) GO TO 8                                            ARGINP  11
      NSETS=2                                                         ARGINP  12
      ND(2)=N2                                                        ARGINP  13
      I(2)=I2                                                         ARGINP  14
      IF (N3.EQ.0) GO TO 8                                            ARGINP  15
      NSETS=3                                                         ARGINP  16
      ND(3)=N3                                                        ARGINP  17
      I(3)=I3                                                         ARGINP  18
      IF (N4.EQ.0) GO TO 8                                            ARGINP  19
      NSETS=4                                                         ARGINP  20
      ND(4)=N4                                                        ARGINP  21
      I(4)=I4                                                         ARGINP  22
    8 RETURN                                                          ARGINP  23
      END                                                             ARGINP  24
```

```
      SUBROUTINE AROP(M,INAM,IOP)                                   AROP       2
                                                                    BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                     BLANK      3
      COMMON STORE(48000),IWORK(10)                                 86MARSIZ   1
      DIMENSION INFF(24,2000)                                       86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                      86MARSIZ   3
                                                                    BLANK      5
      DIMENSION T(49),T1(49)                                        AROP       4
      DIMENSION RX(2,3),RY(2,3),RW(3),MI(2),RX1(2,3),RY1(2,3),RW1(3) AROP      5
      COMPLEX CA,CB,CC                                              AROP       6
      COMMON /MA/ ML(3) $        LEVEL 2, ML                        AROPLL     1
      EQUIVALENCE(ML,MI)                                            AROPLL     2
                                                                    AROPLL     3
      DOUBLE DA,DB,DC                                               AROP       7
      DIMENSION INAM(4),NAM(4),PAR(4)                               AROP       8
      EQUIVALENCE (NAM,PAR),(CA,PAR(1)),(CB,PAR(3)),(A,PAR(1))      AROP       9
      EQUIVALENCE (B,PAR(2))                                        AROP      10
      EQUIVALENCE (T(1),RX(1,1)),(T(7),RY(1,1))                     AROP      11
      EQUIVALENCE (T1(1),RX1(1,1)),(T1(7),RY1(1,1))                 AROP      12
      EQUIVALENCE (CA,DA),(CB,DB),(CC,DC)                           AROP      13
      LOGICAL PLUS,MINUS,MULT,DIV,POWER                             AROP      14
      INTEGER ELNUM                                                 AROP      15
                                                                    AROP      16
      DO 1 I=1,4                                                    AROP      17
    1 NAM(I) = INAM(I)                                              AROP      18
      KN = INFF(20,M)                                               AROP      19
      ITYP = INFF(21,M)                                             AROP      20
      PLUS = .FALSE.                                                AROP      21
      MINUS = .FALSE.                                               AROP      22
      MULT = .FALSE.                                                AROP      23
      DIV = .FALSE.                                                 AROP      24
      POWER = .FALSE.                                               AROP      25
      IF (IOP.EQ.1H+) PLUS = .TRUE.                                 AROP      26
      IF (IOP.EQ.1H-) MINUS = .TRUE.                                AROP      27
      IF(IOP.EQ.1H*) MULT = .TRUE.                                  AROP      28
      IF (IOP.EQ.1H/) DIV = .TRUE.                                  AROP      29
      IF (IOP.EQ.2H**) POWER = .TRUE.                               AROP      30
      IF (KN.NE.4HSCAL) GO TO 10                                    AROP      31
C  SCALAR OPERATION                                                 AROP      32
      IF (ITYP.NE.4HSNGL) GO TO 6                                   AROP      33
      IF (.NOT.MINUS) GO TO 5                                       AROP      34
      B = -B                                                        AROP      35
    5 IF (MINUS.OR.PLUS) C = A + B                                  AROP      36
      IF (MULT) C = A*B                                             AROP      37
      IF (DIV) C = A/B                                              AROP      38
      IF (POWER) C = A**B                                           AROP      39
      CALL STDAT(M,5,1,1,C)                                         AROP      40
      RETURN                                                        AROP      41
    6 IF (ITYP.EQ.3HDBL) GO TO 8                                    AROP      42
C  COMPLEX NUMBERS                                                  AROP      43
      IF (.NOT.MINUS) GO TO 7                                       AROP      44
      CB = - CB                                                     AROP      45
    7 IF (MINUS.OR.PLUS) CC=CA+CB                                   AROP      46
      IF (MULT) CC = CA*CB                                          AROP      47
      IF (DIV) CC = CA/CB                                           AROP      48
   11 CALL STDAT(M,5,1,2,CC)                                        AROP      49
      RETURN                                                        AROP      50
```

```
      8   IF (.NOT.MINUS) GO TO 9
          DB = - DB
      9   IF (MINUS.OR.PLUS) DC = DA + DB
          IF (MULT) DC=DA*DB
          IF(DIV) DC = DA/DB
          GO TO 11
C   OPERATE ON MATRICES.  CONSIDER ONLY +,-,* FOR PRESENT
     10   MA = ELNUM(INAM(1))
          MB = ELNUM(INAM(2))
          IF (.NOT.MULT) GO TO 15
          MI(1) = MA
          MI (2) = MB
          CALL MMM(M,2,MI)
          RETURN
     15   IF (KN.NE.3HMXY) GO TO 30
          CALL RXY(MA,RX,RY,RW)
          CALL RXY(MB,RX1,RY1,RW1)
          IF (.NOT.MINUS) GO TO 20
          DO 17 J=1,3
          DO 16 I=1,2
          RX1(I,J) = -RX1(I,J)
          RY1(I,J) = - RY1(I,J)
     16   CONTINUE
          RW1(J) = -RW1(J)
     17   CONTINUE
     20   DO 25 J=1,3
          DO 22 I=1,2
          RX(I,J) = RX(I,J) + RX1(I,J)
          RY(I,J) = RY(I,J) + RY1(I,J)
     22   CONTINUE
          RW(J) = RW(J) + RW1(J)
     25   CONTINUE
          CALL STXY(M,RX,RY,RW)
          RETURN
     30   CALL RTRV7(MA,T,RW)
          CALL RTRV7(MB,T1,RW1)
          IF (.NOT.MINUS) GO TO 35
          DO 31 I=1,49
     31   T1(I)= - T1(I)
          DO 32 I=1,3
     32   RW1(I) = -RW1(I)
     35   DO 37 I=1,49
     37   T(I) = T(I) + T1(I)
          DO 38 I=1,3
     38   RW(I) = RW(I) + RW1(I)
          CALL STOR7(M,T,RW)
          RETURN
          END
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
EQV/COMM	I	MA	NOT ALL ITEMS IN THIS COMMON BLOCK OCCUR IN LEVEL STATEMENTS.

```
      SUBROUTINE AUXC(X,FMUX)                                       AUXC     2
C  FUNCTION EVALUATION FOR CELL DESIGN                              AUXC     3
                                                                    AUXC     4
      COMMON/INSTR/OPNAME,NAME,OP,KA,KB,OBJA,OBJB,NXTM,TRA,LQ,NQ,LIN,  INSTR   2
     1             NIN,LFL,NFL,LBC,NBC,NTOT,IX,KIND,TYPE,ROWS,COLS,EXTR,  INSTR   3
     2             M,NF,NB,NI,MSUBR,ISIGN                           INSTR    4
      INTEGER OPNAME,OP,OBJA,OBJB,TRA,TYPE,ROWS,COLS,EXTR           INSTR    5
                                                                    INSTR    6
      COMMON/MAGL/P(10),DESMU,K                                     AUXC     6
      DIMENSION RX(2,3),RY(2,3),RW(3)                               AUXC     7
      EQUIVALENCE (DESMU,FMU)                                       AUXC     8
                                                                    AUXC     9
      P(K)=X                                                        AUXC    10
      CALL CELL(M,P)                                                AUXC    11
      CALL RXY(M,RX,RY,RW)                                          AUXC    12
      COSMUX = (RX(1,1) + RX(2,2)) * 0.5                            AUXC    13
      FMUX = COSMUX -FMU                                            AUXC    14
      RETURN                                                        AUXC    15
      END                                                           AUXC    16
```

```
      SUBROUTINE AUXD(Y,FMUY)                                   AUXD     2
      RETURN                                                    AUXD     3
      END                                                       AUXD     4
```

```
      SUBROUTINE AUXF(X,FMUX)                                           AUXF       2
      RETURN                                                            AUXF       3
      END                                                               AUXF       4
```

```
      INTEGER FUNCTION BDAT(M,J)                                        BDAT       2
C  RETRIEVES ONE PIECE OF BCD DATA FROM INDEX J OF ELEMENT J            BDAT       3
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1          LDFLG,FIN                                               CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      EQUIVALENCE(IDATA,FDATA)                                          BDAT       6
                                                                        BDAT       7
      LOC=INFF(16,M)+J-1                                                BDAT       8
      IF (LOC.GT.0)  GO TO 1                                            BDAT       9
      ERROR = .TRUE.                                                    BDAT      10
      RETURN                                                            BDAT      11
    1 FDATA = STORE(LOC)                                                BDAT      12
      BDAT=IDATA                                                        BDAT      13
      RETURN                                                            BDAT      14
      END                                                               BDAT      15
```

```
      SUBROUTINE BEAM(M)
      RETURN
      END
```

```
                                                                      BEND       2
      SUBROUTINE BEND(M,V,PAR,IOP)                                    BEND       3
                                                                      BEND       4
C  3X3 MATRICES FOR RECTANGULAR,ZERO-GRADIENT MAGNET, MODIFIED TO ACT BEND       5
C  ON OFF-MOMENTUM ORBITS.  TRANSFORMATION IS RELATIVE TO A RAY V,    BEND       6
C  GIVEN IN THE ORIGINAL COORDINATES, WHICH WILL BE TRACKED THROUGH   BEND       7
C  THE MAGNET.  THE RESULTING MATRIX IS TO BE USED BY CYX TO FIND     BEND       8
C  BETATRON FUNCTIONS RELATIVE TO A NEW CLOSED ORBIT.                 BEND       9
C  IF IOP = 1, ACTS LIKE A MAG FOR THE SAME PARAMETERS.               BEND      10
                                                                      BEND      11
C.................................................................... BEND      12
C B    BEND          L          0.          BRHO       BZ         $   BEND      13
C.................................................................... BEND      14
                                                                      BEND      15
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL DIM        2
                                                                      DIM        3
      COMMON /MB/ ML    $          LEVEL 2, ML                        BENDLL     1
      DIMENSION V(1),PAR(1),TAR(6),ML(3),T(49),RW(3)                  BEND      17
      COMPLEX R(6),Z,XI                                               BEND      18
      REAL L                                                          BEND      19
      DATA XI/(0.,1.)/                                                BEND      20
                                                                      BEND      21
      IF (IOP.EQ.1) GO TO 2                                           BEND      22
      PIO2 = ACOS(0.)                                                 BEND      23
      DEGRAD = 90./PIO2                                               BEND      24
                                                                      BEND      25
      S0 = PAR(1)                                                     BEND      26
      BR0 = PAR(3)                                                    BEND      27
      B0 = PAR(4)                                                     BEND      28
      R0 = ABS(BR0/B0)                                                BEND      29
      PHI0 = S0 / (2.*R0)                                             BEND      30
      SINPHI0 = SIN(PHI0)                                             BEND      31
      L = 2. * R0 * SINPHI0                                           BEND      32
                                                                      BEND      33
      X0 = V(1)                                                       BEND      34
      AL0 = V(2)                                                      BEND      35
      IF (B0.GE.0.)  GO TO 3                                          BEND      36
       X0 = -X0                                                       BEND      37
       AL0 = -AL0                                                     BEND      38
    3 CONTINUE                                                        BEND      39
      DPP = V(6)                                                      BEND      40
      FP = 1. + DPP                                                   BEND      41
      BR = FP * BR0                                                   BEND      42
      RHO = FP * R0                                                   BEND      43
                                                                      BEND      44
      ARG0 = PIO2 + PHI0                                              BEND      45
      R(1) = X0 * CEXP(XI*ARG0)                                       BEND      46
                                                                      BEND      47
      ARG1 = PHI0 + AL0                                               BEND      48
      D1 = X0 * SINPHI0 / COS(ARG1)                                   BEND      49
      R(2) = D1 * CEXP(XI*ARG1)                                       BEND      50
                                                                      BEND      51
      GAM0 = ASIN( (L/RHO) - SIN(ARG1) )                              BEND      52
      THETA = ARG1 + GAM0                                             BEND      53
      ARG2 = ARG1 - PIO2                                              BEND      54
      R(3) = RHO * CEXP(XI*ARG2)                                      BEND      55
                                                                      BEND      56
```

```
      ARG3 = PIO2 - GAM0                                   BEND      57
      R(4) = RHO * CEXP(XI*ARG3)                           BEND      58
                                                           BEND      59
      Z = R(1) + R(2) + R(3) + R(4)                        BEND      60
      H = AIMAG(Z)                                         BEND      61
      COSPHGM = COS(PHI0-GAM0)                             BEND      62
      D4 = H * SINPHI0 / COSPHGM                           BEND      63
      X0F = H * COS(GAM0) / COSPHGM                        BEND      64
      AL0F = AL0 + 2.*PHI0 - THETA                         BEND      65
      S = RHO * THETA                                      BEND      66
                                                           BEND      67
      TAR(1) = S                                           BEND      68
      TAR(2) = 0.                                          BEND      69
      TAR(3) = BR                                          BEND      70
      TAR(4) = B0                                          BEND      71
      TAR(5) = ARG1 * DEGRAD                               BEND      72
      TAR(6) = GAM0 * DEGRAD                               BEND      73
                                                           BEND      74
      DO 1 I=1,3                                           BEND      75
    1 ML(I) = MEND - 2 - I                                 BEND      76
      CALL DRIFT(MEND-3,D1)                                BENDLL     2
            CALL MAGNET(MEND-4,TAR)                        BENDLL     3
                        CALL DRIFT(MEND-5,D4)              BENDLL     4
                                                           BENDLL     5
      CALL MMM(M,3,ML)                                     BEND      80
                                                           BEND      81
      CALL RTRV7(M,T,RW)                                   BEND      82
      CALL MXV7(T,V,V)                                     BEND      83
      V(1) = X0F                                           BEND      84
      V(2) = AL0F                                          BEND      85
      V(5) = V(5) - D1 - S - D4 + S0                       BEND      86
      IF (B0.GE.0.)  GO TO 4                               BEND      87
       V(1) = -X0F                                         BEND      88
       V(2) = -AL0F                                        BEND      89
    4 RETURN                                               BEND      90
                                                           BEND      91
    2 CALL MAGNET(M,PAR)                                   BEND      92
      RETURN                                               BEND      93
      END                                                  BEND      94
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
28	I	PAR	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
29	I	PAR	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
36	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
41	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
87	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
88	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
88	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
91	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

```
                                                                   BEND      95
      SUBROUTINE BESIS(X,FI,NN)                                    BESIS      2
C >>> FOR EVALUATION OF BESSEL FUNCTION \FI\ ONLY                  BESIS      3
C >>> DIMENSION OF FI IN CALLING PROGRAM MUST BE GE 2              BESIS      4
*                                                                  BESIS      5
      DIMENSION FI(1), PI(600)                                     BESIS      6
      NMAX = MAX0((NN+1),2)                                        BESIS      7
      KMAX=NMAX                                                    BESIS      8
      SUM=0.                                                       BESIS      9
      I=X                                                          BESIS     10
      JMAX=I+31                                                    BESIS     11
      TZ=2./X                                                      BESIS     12
      JM2=JMAX+2                                                   BESIS     13
      DO 10 J=JM2,NMAX                                             BESIS     14
   10 PI(J)=0.                                                     BESIS     15
      PI(JMAX+1)=1.E-50                                            BESIS     16
      DO 11 J=1,JMAX                                               BESIS     17
      K=JMAX+2-J                                                   BESIS     18
      DK=K-1                                                       BESIS     19
      PI(K-1)=DK*TZ*PI(K)+PI(K+1)                                  BESIS     20
   11 SUM=SUM+PI(K)                                                BESIS     21
      SUM=SUM+SUM                                                  BESIS     22
      A=EXP(X)/(PI(1)+SUM)                                         BESIS     23
      DO 12 N=1,NMAX                                               BESIS     24
   12 FI(N)=A*PI(N)                                                BESIS     25
      RETURN                                                       BESIS     26
      END                                                          BESIS     27
```

```
      SUBROUTINE BET(M,BETX,BETY,IF)                                    BET        2
C  COMPUTES BETATRON FUNCTIONS                                          BET        3
C                       BETX(I)=MU,BETA,ALPHA,GAMMA,XEQ,DXEQ    FOR RX  BET        4
C                       BETY(I)=MU,BETA,ALPHA,GAMMA,YEQ,DYEQ    FOR RY  BET        5
                                                                        BET        6
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER   SWTCH      2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                          SWTCH      3
                                                                        SWTCH      4
      DIMENSION RX(2,3),RY(2,3),BETX(1),BETY(1),RXI(2,3),RYI(2,3)       BET        8
      DIMENSION RW(3),RWI(3)                                            BET        9
      DATA PI /3.141592654/                                             BET       10
                                                                        BET       11
      TWOPI = 2.*PI                                                     BET       12
      QINV = 0.                                                         BET       13
      NN = -M                                                           BET       14
      CALL RXY(M,RX,RY,RW)                                              BET       15
      TR= RX(1,1) + RX(2,2)                                             BET       16
      CS=TR/2.                                                          BET       17
      BETX(9) = CS                                                      BET       18
      IF(TR.EQ.2.)GO TO  1                                              BET       19
      IF(TR.EQ.(-2.))GO TO  2                                           BET       20
      IF (ABS(TR).GT.2.)GO TO  3                                        BET       21
      GO TO 7                                                           BET       22
    3 IF( IF.EQ.0 ) GO TO 33                                            BET       23
      BETX(1)=0.                                                        BET       24
      GO TO 4                                                           BET       25
    1 BETX(1)=0.                                                        BET       26
      GO TO 4                                                           BET       27
    2 BETX(1)=PI                                                        BET       28
    4 DO 5 I=2,6                                                        BET       29
    5 BETX(I)=0.                                                        BET       30
      GO TO 6                                                           BET       31
   33 SN = SIGN( 1., TR )*SQRT( CS**2 - 1. )                            BET       32
      S = ALOG( ABS( CS ) + SN )                                        BET       33
      BS = RX(1,2)                                                      BET       34
      GO TO 8                                                           BET       35
    7 BS = RX(1,2)                                                      BET       36
      SN=SQRT(1.-CS*CS)*BS/ABS(BS)                                      BET       37
      S = ATAN2(SN,CS) + TWOPI                                          BET       38
    8 B=BS/SN                                                           BET       39
      IF (ABS(TR).GT.2.) B = -ABS(B)                                    BET       40
      G=-RX(2,1)/SN                                                     BET       41
      A=(RX(1,1)-RX(2,2))/(2.*SN)                                       BET       42
      BETX(1)=S                                                         BET       43
      BETX(2)=B                                                         BET       44
      BETX(3)=A                                                         BET       45
      BETX(4)=G                                                         BET       46
      CALL RXY(NN,RXI,RYI,RWI)                                          BET       47
      QINV=1.                                                           BET       48
      CSS = 2.*( 1. - CS )                                              BET       49
      BETX(5) = (RX(1,3)+RXI(1,3)) / CSS                                BET       50
      BETX(6) = (RX(2,3)+RXI(2,3)) / CSS                                BET       51
    6 TR = (RY(1,1) + RY(2,2))                                          BET       52
      CS=TR/2.                                                          BET       53
      BETY(9) = CS                                                      BET       54
      IF(TR.EQ.2.)GO TO 11                                              BET       55
      IF(TR.EQ.(-2.))GO TO 12                                           BET       56
```

```
                IF (ABS(TR).GT.2.)GO TO 13                               BET      57
                GO TO 17                                                 BET      58
 60          13 IF( IF.EQ.0 ) GO TO 313                                  BET      59
                BETY(1)=0.                                               BET      60
          C  INDEX M HAS ITS OWN INFO, WHICH REFERENCES DATA AND MATRIX LOC.OF N.  BET   61
                GO TO 14                                                 BET      62
             11 BETY(1)=0.                                               BET      63
 65             GO TO 14                                                 BET      64
             12 BETY(1)=PI                                               BET      65
             14 DO 15 I=2,6                                              BET      66
             15 BETY(I)=0.                                               BET      67
                GO TO 16                                                 BET      68
 70         313 SN = SIGN( 1., TR )*SQRT( CS**2 - 1. )                   BET      69
                S = ALOG( ABS( CS ) + SN )                               BET      70
                BS = RY(1,2)                                             BET      71
                GO TO 20                                                 BET      72
             17 BS = RY(1,2)                                             BET      73
 75             SN=SQRT(1.-CS*CS)*BS/ABS(BS)                             BET      74
                S = ATAN2(SN,CS) + TWOPI                                 BET      75
             20 B=BS/SN                                                  BET      76
                IF (ABS(TR).GT.2.) B = -ABS(B)                           BET      77
                G = -RY(2,1)/SN                                          BET      78
 80             A = (RY(1,1)-RY(2,2))/(2.*SN)                            BET      79
                BETY(1)=S                                                BET      80
                BETY(2)=B                                                BET      81
                BETY(3)=A                                                BET      82
                BETY(4)=G                                                BET      83
 85             IF(QINV)18,18,19                                         BET      84
             18 CALL RXY(NN,RXI,RYI,RWI)                                 BET      85
             19 CSS = 2.*( 1. - CS )                                     BET      86
                BETY(5) = (RY(1,3)+RYI(1,3)) / CSS                       BET      87
                BETY(6) = (RY(2,3)+RYI(2,3)) / CSS                       BET      88
 90          16 J=2                                                      BET      89
                BETX(7) = SQRT( ABS( BETX(J) ) )                         BET      90
                BETY(7) = SQRT( ABS( BETY(J) ) )                         BET      91
                BETX(8)=BETX(1)*180./PI                                  BET      92
                BETY(8)=BETY(1)*180./PI                                  BET      93
 95             RETURN                                                   BET      94
                END                                                      BET      95
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
19	I	BETX	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
45	I	BETX	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
46	I	BETX	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
47	I	BETX	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
51	I	BETX	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
52	I	BETX	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
55	I	BETY	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
82	I	BETY	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
83	I	BETY	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
84	I	BETY	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
88	I	BETY	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
89	I	BETY	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
91	I	BETX	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
92	I	BETY	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
93	I	BETX	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
94	I	BETY	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

```
      SUBROUTINE BMIS                                                  BMIS        2
C  BEGIN MISALIGNMENT                                                  BMIS        3
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER  SWTCH       2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                         SWTCH       3
                                                                       SWTCH       4
      COMMON/INSTR/OPNAME,NAME,OP,KA,KB,OBJA,OBJB,NXTM,TRA,LQ,NQ,LIN,  INSTR       2
     1             NIN,LFL,NFL,LBC,NBC,NTOT,IX,KIND,TYPE,ROWS,COLS,EXTR, INSTR     3
     2             M,NF,NB,NI,MSUBR,ISIGN                              INSTR       4
      INTEGER OPNAME,OP,OBJA,OBJB,TRA,TYPE,ROWS,COLS,EXTR              INSTR       5
                                                                       INSTR       6
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,   CONTRL      2
     1          LDFLG,FIN                                              CONTRL      3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN     CONTRL      4
                                                                       CONTRL      5
                                                                       BMIS        7
C                                                                      BMIS        8
      IF (MODE.EQ.3) GO TO 10                                          BMIS        9
      ISGN = 1H                                                        BMIS       10
      IF (OP.LT.0) ISGN=1H-                                            BMIS       11
      WRITE (3,20) ISGN                                                BMIS       12
   20 FORMAT (6H  *** ,A1,7X,4HBMIS)                                   BMIS       13
      IF (OP.LT.0) RETURN                                              BMIS       14
      IF (MODE.EQ.2) RETURN                                            BMIS       15
   10 BEND=0.                                                          BMIS       16
      RETURN                                                           BMIS       17
      END                                                              BMIS       18
```

```
      SUBROUTINE BVAL(M)                                                BVAL       2
C   PARTICLE BEAM DEFINITION                                            BVAL       3
C   COMPUTES KINEMATICS AND UNNORMALIZED EMITTANCES.                    BVAL       4
C   IF INPUT EMITTANCES ARE NORMALIZED, UNNORMALIZED VALUES             BVAL       5
C WILL BE CALCULATED.                                                   BVAL       6
C   UNNORMALIZED VALUES WILL BE STORED FOR LATER USE.                   BVAL       7
C.......................................................                BVAL       8
C   NAM  BVAL    KA    KB// PT    A  EPSX    EPSY  EPSL    SIGL         BVAL       9
C......................................................                 BVAL      10
C   NAM    BVAL        //            EPSX    EPSY  EPSL    SIGL         BVAL      11
C   KIN    BVAL             P     A  EPSX    EPSY  EPSL    SIGL         BVAL      12
C   KNM    BVAL   1    1//  T     A  EPSX    EPSY  EPSL    SIGL         BVAL      13
C                                                                       BVAL      14
C       PT = EITHER P, T OR BLANK WHERE                                 BVAL      15
C         P = MOMENTUM GEV/C                                            BVAL      16
C         T = KINETIC ENERGY GEV                                        BVAL      17
C         A = ATOMIC WEIGHT                                             BVAL      18
C             IF A = 0 OR BLANK, ELECTRON MASS IS ASSUMED.              BVAL      19
C         EMITTANCES (EPS) IN MM-MRAD                                   BVAL      20
C         SIGL (MM)                                                     BVAL      21
C                                                                       BVAL      22
C       IF KA=0 OR BLANK, 1ST PARAMETER IS P.                           BVAL      23
C       IF KA NOT 0 (OR BLANK), 1ST PARAMETER IS T.                     BVAL      24
C                                                                       BVAL      25
C       IF KB = 0 (OR BLANK), THE NEXT CALL TO CYC WILL                 BVAL      26
C       PRINT ETAS = -VP(5)                                             BVAL      27
C       IF KB NOT 0 , THE NEXT CALL TO CYC WILL                         BVAL      28
C       PRINT ETAT =VP(5) -S/GAMMASQ.                                   BVAL      29
C                                                                       BVAL      30
C   UNNORMALIZED INPUT -- WHEN PT=0, THE EPSX ETC. ON THE INPUT CARD    BVAL      31
C                           REPRESENT UNNORMALIZED EMITTANCES AND NO    BVAL      32
C                            KINEMATIC CALCULATIONS ARE DONE.           BVAL      33
C    NORMALIZED INPUT -- WHEN PT NOT 0 AND EPSX,ETC. ARE PRESENT.       BVAL      34
C                                                                       BVAL      35
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/KINET/KNFLAG,TK,P,GAM,BETT,BETGAM,BRHO                     KINET      2
      LOGICAL KNFLAG                                                    KINET      3
C                                                                       BVAL      38
      DIMENSION PAR(6),EM(4)                                            BVAL      39
      EQUIVALENCE (EPSX,EM(1)),(EPSY,EM(2)),(EPSL,EM(3)),(SIGP,EM(4))   BVAL      40
      DATA E0,RME,CONK/.9382796,.0005110034,3.33564095/                 BVAL      41
C                                                                       BVAL      42
      KA = INFF(4,M)                                                    BVAL      43
      KB = INFF(5,M)                                                    BVAL      44
      CALL DATA (M,1,1,6,PAR)                                           BVAL      45
      IF (PAR(1).EQ.0.)  GO TO 22                                       BVAL      46
      A = PAR(2)                                                        BVAL      47
      IF (A.EQ.0.)  A = RME                                             BVAL      48
      AMPC2 = A*E0                                                      BVAL      49
      AMPC2R = 1./AMPC2                                                 BVAL      50
      IF (KA.NE.0)  GO TO 15                                            BVAL      51
C  1ST PARAM IS P                                                       BVAL      52
```

```
      P = PAR(1)
      BETGAM = P*AMPC2R
      GAM = SQRT(BETGAM*BETGAM + 1.)
      TK = AMPC2*(GAM-1.)
      GO TO 20
C  1ST PARAM IS T
 15   TK = PAR(1)
      GAM = TK*AMPC2R + 1.
      BETGAM = SQRT(GAM*GAM - 1.)
      P = AMPC2*BETGAM
 20   BETT = BETGAM/GAM
      BRHO = P*CONK
      IF (KB.NE.0)  KNFLAG = .TRUE.
      WRITE (3,101)
      CALL DASH
      WRITE (3,100)  P,TK,A,BRHO,BETT,GAM,BETGAM
      IF ((PAR(3).EQ.0.).AND.(PAR(4).EQ.0.).AND.(PAR(5).EQ.0.).AND.
     1  (PAR(6).EQ.0.))  RETURN
      BETGAMR = 1./BETGAM
      EPSX = PAR(3)*BETGAMR
      EPSY = PAR(4)*BETGAMR
      EPSL = PAR(5)*BETGAMR
      GO TO 25
C   ALREADY UNNORMALIZED CASE
 22   EPSX = PAR(3)
      EPSY = PAR(4)
      EPSL = PAR(5)
 25   SIGL = PAR(6)
      SIGP = EPSL/SIGL
      CALL STDAT (M,5,1,4,EM)
      WRITE (3,105) EPSX,EPSY,EPSL,SIGL,SIGP
      CALL DASH
C
 100  FORMAT (8X,*P = *,F10.5,* GEV/C*,6X,*T = *,
     1  F10.5,* GEV*,12X,*A = *,F10.5,6X,*BRHO = *,F10.5,/
     2  5X,*BETA = *,E15.8,3X,*GAMMA = *,E15.8,3X,
     3  *BETAGAMMA = *,E15.8)
 101   FORMAT (1H )
 105  FORMAT (/5X,*EMITTANCES UNNORMALIZED  (MM-MRAD)*/
     1   8X,*EPSX = *,F10.6,4X,*EPSY = *,F10.6,4X,*EPSL = *,
     2   F10.6,10X,*SIGL = *,F10.6,1X,*MM*,4X,*SIGP = *,F10.6,
     3  * (0/00)*)
      RETURN
      END
```

```
      SUBROUTINE CADJ6(R,S,DET)                                      CADJ6      2
      RETURN                                                         CADJ6      3
      END                                                            CADJ6      4
```

```
              SUBROUTINE STOREC(M,I,NREG,NSY)                STOREC      2
                                                             BLANK       2
              LEVEL 2, STORE,INFF,IWORK                      BLANK       3
              COMMON STORE(48000),IWORK(10)                  86MARSIZ    1
              DIMENSION INFF(24,2000)                        86MARSIZ    2
              EQUIVALENCE (INFF,STORE)                       86MARSIZ    3
                                                             BLANK       5
              INTEGER BDAT                                   STOREC      4
              DATA MASK/77000 00000 00000 00000B/            STOREC      5
              IREG = BDAT(M,I+1)                             STOREC      6
              LC = IREG.AND.MASK                             STOREC      7
              IF (LC.EQ.1H ) GO TO 1                         STOREC      8
              NREG = -1                                      STOREC      9
              MN = MDAT(M,I+1)                               STOREC     10
              NSY = INFF(10,MN)    $ RETURN                  STOREC     11
          1   DECODE(5,2,IREG) NR                            STOREC     12
          2   FORMAT(I5)                                     STOREC     13
              NREG = NR + 10                                 STOREC     14
              RETURN                                         STOREC     15
              END                                            STOREC     16
```

```
      SUBROUTINE STKLFT(X,Y,Z,T)                                        STKLFT     2
      T = Z  $ Z = Y  $ Y = X                                           STKLFT     3
      RETURN                                                            STKLFT     4
      END                                                               STKLFT     5
```

```
      SUBROUTINE CALC(M)                                                CALC       2
C  INSTRUCTION CALC -- SIMULATES AN H-P CALCULATOR.                     CALC       3
C  COMMANDS ARE ENTERED AS 5 CHARACTER BCD DATA.  COMMANDS ALLOWED ---- CALC       4
                                                                        CALC       5
C             1/X  SQRT      LN                 Y**X X*X       EXP      CALC       6
C         X-Y RDN  SIN  COS  TAN                RUP  ASIN ACOS ATAN     CALC       7
C                  STO  RCL  ABS                                        CALC       8
C       ENTER CHS  EEX  CLX                                             CALC       9
C           -                              X=Y                          CALC      10
C           +      P-R  R-P                XLEY                         CALC      11
C           *                              XGTY                         CALC      12
C           /  PI  LASTX                   X=0                          CALC      13
                                                                        CALC      14
C  KA CONTROLS PRINTING AS FOLLOWS -- KA = 0 -- NO PRINT                CALC      15
C                                     KA = 1, PRINT STACK AT END        CALC      16
C                                     KA = 2, PRINT STACK EVERY STEP    CALC      17
                                                                        CALC      18
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1          LDFLG,FIN                                               CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      COMMON /MATCH3/NOPR,NONU,FLAG,IPR,IPP,MINFLG,MIFLG,MATFLG,GLOBAL  MATCH3     2
      LOGICAL FLAG,NOPR,NONU,MIFLG,MATFLG,GLOBAL                        MATCH3     3
      COMMON/CLC/REG(109)                                               CLC        2
      DIMENSION CMND(16,5),LX(16,5),NUM(14),STK(5)                      CALC882    1
      LOGICAL  LIFT,SKIP,NPRT                                           CALC      24
      INTEGER CMND,ALPH,BDAT                                            CALC      25
      COMPLEX CX                                                        CALC      26
      EQUIVALENCE (STK,REG),(X,STK(1)),(Y,STK(2)),(Z,STK(3)),(T,STK(4)),CALC      27
     .                      (XL,STK(5)),(CX,X)                          CALC      28
      DATA MASK/ 77000 00000 00000 00000 B/,                            CALC      29
     .     IBLANK/00555 55555 55555 55555 B/                            CALC      30
      DATA NUM/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H ,1H.,1H+,1H-/ CALC882    2
      DATA (LX(1,L),L=1,5)/0,1,1,0,1/                                   CALC882    3
      DATA (LX(2,L),L=1,5)/0,0,1,1,1/                                   CALC882    4
      DATA (LX(3,L),L=1,5)/0,0,0,0,1/                                   CALC882    5
      DATA (LX(4,L),L=1,5)/5*0/                                         CALC882    6
      DATA (LX(5,L),L=1,5)/1,4*0/                                       CALC882    7
      DATA (LX(6,L),L=1,5)/1,0,1,1,0/                                   CALC882    8
      DATA (LX(7,L),L=1,5)/1,4*0/                                       CALC882    9
      DATA (LX(8,L),L=1,5)/1,4*0/                                       CALC882   10
      DATA (LX(9,L),L=1,5)/0,1,1,0,1/                                   CALC882   11
      DATA (LX(10,L),L=1,5)/0,0,1,1,1/                                  CALC882   12
      DATA ((LX(K,L), K=11,16),L=1,5)/30*0/                             CALC882   13
      DATA (CMND(1,L),L=1,5)/4H    ,4H1/X ,4HSQRT,4H    ,4HLN  /,       CALC      34
     2     (CMND(2,L),L=1,5)/4HX-Y ,4HRDN ,4HSIN ,4HCOS ,4HTAN /,       CALC      35
     3     (CMND(3,L),L=1,5)/4H    ,4H    ,4HSTO ,4HRCL ,4HABS /,       CALC      36
     4    (CMND(4,L),L=1,4)/5HENTER,4HCHS ,4HEEX ,4HCLX       /,        CALC      37
     5      CMND(5,1)       /4H-                              /,        CALC      38
     6     (CMND(6,L),L=1,4)/4H+    ,4H    ,4HP-R ,4HR-P      /,        CALC      39
     7      CMND(7,1)       /4H*                              /,        CALC      40
```

```
            8     (CMND(8,L),L=1,3)/4H/    ,4HPI   ,5HLASTX               /,     CALC        41
            9     (CMND(9,L),L=1,5)/4H     ,4HY**X,4HX*X  ,4H     ,4HEXP /,     CALC        42
            A    (CMND(10,L),L=1,5)/4H     ,4HRUP ,4HASIN,4HACOS,4HATAN/,     CALC        43
            C     CMND(12,4)        /                  4HCLST         /,     CALC        44
            D   (CMND(K,1),K=13,16)/4HX=Y ,4HXLEY,4HXGTY,4HX=0         /      CALC        45
                                                                              CALC        46
             SKIP = .FALSE.    $ LIFT = .TRUE.    $ NPRT = .FALSE.            CALC        47
             NBC = INFF(17,M)   $ KA = INFF(4,M)                              CALC        48
             PI = 2.*ACOS(0.)    $ I = 0    $ ICMD = 5H                       CALC        49
             IF (NOPR.OR.KA.EQ.0) NPRT = .TRUE.                               CALC        50
             IF (.NOT.NPRT) CALL DASH                                         CALC        51
             IF (.NOT.NPRT) WRITE(3,901)                                      CALC        52
             IF (.NOT.NPRT) CALL DASH                                         CALC        53
             IF (.NOT.NPRT.AND.KA.EQ.2)  WRITE(3,902) I,ICMD,X,Y,Z,T,XL       CALC        54
                                                                              CALC        55
                                                                              CALC        56
             DO 300 I=1,NBC                                                   CALC        57
             ICMD = BDAT(M,I)                                                 CALC        59
             IF(ICMD.EQ.5HLSTX ) ICMD=5HLASTX                                 86CALC       1
             IF(ICMD.EQ.5HENTR ) ICMD=5HENTER                                 86CALC       2
             IF (ICMD.EQ.5H     ) GO TO 300                                   CALC        60
             IF (.NOT.SKIP) GO TO 2                                           CALC        61
             SKIP = .FALSE.    $ GO TO 3                                      CALC        62
                                                                              CALC        63
        2    DO 6 K=1,16                                                      CALC        64
             DO 6 L=1,5                                                       CALC        65
             KL = 5*(K-1) + L                                                 CALC        66
             IF(ICMD.EQ.CMND(K,L)) GO TO 77                                   CALC882     14
        6    CONTINUE                                                         CALC        68
                                                                              CALC        69
             IC1 = (MASK.AND.ICMD).OR.IBLANK                                  CALC        70
                                                                              CALC        71
             DO 8 K=1,14                                                      CALC882     15
             IF(IC1.EQ.NUM(K)) GO TO 500                                      CALC882     16
        8    CONTINUE                                                         CALC        74
                                                                              CALC        76
        9    WRITE (3,10) ICMD                                                CALC        77
        10   FORMAT(1X,5H**** ,A5,*IS NOT A RECOGNIZED CALC COMMAND*)         CALC        78
             ERROR = .TRUE.    $ RETURN                                       CALC        79
                                                                              CALC        80
       C  FLOATING POINT NUMBER                                               CALC        81
        500  DECODE(5,900,ICMD) F                                             CALC        82
        900  FORMAT(F5.0)                                                     CALC        83
             IF (LIFT) CALL STKLFT(X,Y,Z,T)                                   CALC        84
             X = F  $  GO TO 1                                                CALC        85
                                                                              CALC        86
        77   IF(LX(K,L).EQ.1) XL=X                                            CALC882     17
        7    GO TO ( 1,12,13, 1,15,  21,22,23,24,25,   1, 1,33,34,35,         CALC        87
           .        41,42,43,44, 1,  51, 1, 1, 1, 1,  61, 1,63,64, 1,         CALC        88
           .        71, 1, 1, 1, 1,  81,82,83, 1, 1,   1,92,93, 1,95,         CALC        89
           .      1,102,103,104,105,  1, 1, 1, 1, 1,   1, 1, 1,124, 1,        CALC        90
           .   131, 1, 1, 1, 1,    141, 1, 1, 1, 1, 151, 1, 1, 1, 1,          CALC        91
           .   161, 1, 1, 1, 1                                      ) KL      CALC        92
                                                                              CALC        93
       *  1/X                                                                 CALC        94
        12   X=1./X $ GO TO 1                                                 CALC882     18
       *  SQRT                                                                CALC        96
```

```
 13   X = SQRT(X)  $  GO TO 1                                          CALC        97
*  LN                                                                  CALC        98
 15   X = ALOG(X)  $  GO TO 1                                          CALC        99
*  X-Y                                                                 CALC       100
 21   XX=X $ X=Y $ Y=XX $ GO TO 1                                      CALC882     19
*  RDN                                                                 CALC       102
 22   XX=X $ X=Y $ Y=Z $ Z=T $ T=XX $ GO TO 1                          CALC882     20
*  SIN                                                                 CALC       104
 23   X = SIN(X)  $ GO TO 1                                            CALC       105
*  COS                                                                 CALC       106
 24   X = COS(X)  $ GO TO 1                                            CALC       107
*  TAN                                                                 CALC       108
 25   X = TAN(X)  $ GO TO 1                                            CALC       109
*  STO                                                                 CALC       110
 33   CALL STOREC(M,I,NREG,NSY)                                        CALC       111
      IF (NREG.LT.0) GO TO 331                                         CALC       112
      REG(NREG) = X  $ GO TO 5                                         CALC       113
 331  STORE(NSY) = X  $ GO TO 5                                        CALC       114
*  RCL                                                                 CALC       115
 34   CALL STOREC(M,I,NREG,NSY)                                        CALC       116
      IF (LIFT) CALL STKLFT(X,Y,Z,T)                                   CALC       117
      IF (NREG.LT.0) GO TO 341                                         CALC       118
      X = REG(NREG)  $ GO TO 5                                         CALC       119
 341  X = REALNUM(NSY)                                                 JUN83CALC    1
 5    SKIP = .TRUE.    $ GO TO 1                                       CALC       121
*  ABS                                                                 CALC       122
 35   X = ABS(X)    $ GO TO 1                                          CALC       123
*  ENT                                                                 CALC       124
 41   CALL STKLFT(X,Y,Z,T)                                             CALC       125
      LIFT = .FALSE.   $ GO TO 3                                       CALC       126
*  CHS                                                                 CALC       127
 42   X = -X $ GO TO 1                                                 CALC882     21
*  EEX                                                                 CALC       129
 43   ICMD2 = BDAT(M,I+1)     $ DECODE(5,900,ICMD2) XEXP               CALC       130
      X = X*10.**XEXP    $ SKIP = .TRUE.    $ GO TO 1                  CALC       131
*  CLX                                                                 CALC       132
 44   X = 0.  $ LIFT = .FALSE.  $ GO TO 3                              CALC       133
*  -                                                                   CALC       134
 51   X = Y - X  $ GO TO 4                                             CALC       135
*  +                                                                   CALC       136
 61   X = Y + X  $ GO TO 4                                             CALC       137
*  P-R                                                                 CALC       138
 63   X1 = X*COS(Y)    $ Y = X*SIN(Y)    $ X = X1    $ GO TO 1         CALC       139
*  R-P                                                                 CALC       140
 64   TH = ATAN2(Y,X)    $ R = CABS(CX)    $ X = R    $ Y = TH  $ GO TO 1 CALC    141
*  *                                                                   CALC       142
 71   X = Y * X  $ GO TO 4                                             CALC       143
*  /                                                                   CALC       144
 81   X = Y / X    $ GO TO 4                                           CALC       145
*  PI                                                                  CALC       146
 82   IF (LIFT) CALL STKLFT(X,Y,Z,T)                                   CALC       147
      X = PI            $ GO TO 1                                      CALC       148
*  LASTX                                                               CALC       149
 83   IF(LIFT) CALL STKLFT(X,Y,Z,T) $ X=XL $ GO TO 1                   CALC882     22
*  Y**X                                                                CALC       151
 92   X = Y**X          $ GO TO 4                                      CALC       152
*  X*X                                                                 CALC       153
```

```
       93   X = X*X          $ GO TO 1                           CALC     154
      *  EXP                                                      CALC     155
       95   X = EXP(X)       $ GO TO 1                           CALC     156
      *  RUP                                                      CALC     157
       102  XX=X $ X=T $ T=Z $ Z=Y $ Y=XX $ GO TO 1              CALC882   23
      *  ASIN                                                     CALC     159
       103  X = ASIN(X)      $ GO TO 1                           CALC     160
      *  ACOS                                                     CALC     161
       104  X = ACOS(X)      $ GO TO 1                           CALC     162
      *  ATAN                                                     CALC     163
       105  X = ATAN(X)      $ GO TO 1                           CALC     164
      *  CLST                                                     CALC     165
       124  X = 0.    $ Y = 0.    $ Z = 0.    $ T = 0.           CALC     166
            GO TO 1                                               CALC882   24
      *  X=Y                                                      CALC     168
       131  IF(X.NE.Y) SKIP = .TRUE. $ GO TO 1                   CALC882   25
      *  XLEY                                                     CALC     170
       141  IF(X.GT.Y) SKIP = .TRUE. $ GO TO 1                   CALC882   26
      *  XGTY                                                     CALC     172
       151  IF(X.LE.Y) SKIP = .TRUE. $ GO TO 1                   CALC882   27
      *  X=0                                                      CALC     174
       161  IF(X.NE.0.)SKIP = .TRUE. $ GO TO 1                   CALC882   28
                                                                  CALC     176
      C  DROP STACK                                               CALC     177
       4    Y = Z  $ Z = T                                        CALC     178
                                                                  CALC     179
       1    LIFT = .TRUE.                                         CALC     180
       3    IF (.NOT.NPRT.AND.KA.EQ.2)  WRITE(3,902) I,ICMD,X,Y,Z,T,XL   CALC     181
       300  CONTINUE                                              CALC     182
                                                                  CALC     183
                                                                  CALC     184
            CALL STDAT(M,5,1,5,STK)                               CALC     185
            IF (.NOT.NPRT.AND.KA.EQ.1)  WRITE(3,902) I,ICMD,X,Y,Z,T,XL   CALC     186
            IF (.NOT.NPRT) CALL DASH                              CALC     187
                                                                  CALC     188
       901  FORMAT (4X,1HI,5X,5HCMND ,9X,1HX,14X,1HY,14X,1HZ,14X,1HT,14X,1HL)  CALC     189
       902  FORMAT (1X,I4,5X,A5,5F15.5)                           CALC     190
            RETURN                                                CALC     191
            END                                                   CALC     192
```

```
      SUBROUTINE CALFUN(NC,NV,F,X,INIT)                 CALFUN     2
      RETURN                                            CALFUN     3
      END                                               CALFUN     4
```

```
      SUBROUTINE CDET5(C,DET)                                       CDET5     2
      RETURN                                                        CDET5     3
      END                                                           CDET5     4
```

```
      SUBROUTINE CELC(M,PAR)                                           CELC   2
      RETURN                                                           CELC   3
      END                                                              CELC   4
```

```
      SUBROUTINE CELL(M,PAR)                                            CELL       2
                                                                        CELL       3
C MAGNET PARAMETER GAMMA IS DEPENDENT ON E. GAMMA IS P(5)                CELL       4
C SET UP MATRICES FOR AN FDDF CELL (WITH DRIFT SPACES).                 CELL       5
C PARAMETER LIST...LMAG,K,RHO,LSTR,LSEP,E,OMEGA                         CELL       6
C  CYCLE MATRIX PRODUCT (N+1-1)=(I-1)(I-2)...(K)(K-1)...(I),I=1,K       CELL       7
C  COMPUTE THE PRODUCT (FIRST CYCLED MATRIX IF L.NE.0)                  CELL       8
C QF = FOCUSING MAGNET = MATRIX M-13                                    CELL       9
C QD = DEFOCUSING MAGNET = MATRIX M-14                                  CELL      10
C QS = HALF THE CENTER DRIFT SPACE = MATRIX M-15                        CELL      11
C QT = HALF THE DRIFT SPACE BETWEEN A D AND AN F MAGNET = MATRIX M-16   CELL      12
C SUBROUTINE GENERATES 17 DRIFT MATRICES AS FOLLOWS...                  CELL      13
C THE FDDF CELL = Q = THE PRODUCT OF...QS QF QT QT QD QS QS QD QT QT QF CELL      14
C QS = MATRIX M                                                         CELL      15
C QI = THE I-TH CYCLED TRANSFER MATRIX = MATRIX M-1, WHERE              CELL      16
C I = 1,12, WHERE Q1 = Q.                                               CELL      17
C MAGNET PARAMETER DELTA IS SET = 0.                                    CELL      18
                                                                        CELL      19
      COMMON/BMI/MI(5000)                                               BMIL       1
      COMMON/NELS/NELS                                                  BMIL       2
      LEVEL 2,MI                                                        BMIL       3
                                                                        BMIL       4
                                                                        BMI        3
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER   SWTCH      2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                          SWTCH      3
                                                                        SWTCH      4
      DIMENSION MMI(12)                                                 CELLL      1
      DIMENSION PAR(7),P(7)                                             CELL      22
                                                                        CELL      23
      P(1)=PAR(1)                                                       CELL      24
      P(3)=PAR(3)                                                       CELL      25
      SOV2=PAR(4)/2.                                                    CELL      26
      TOV2=PAR(5)/2.                                                    CELL      27
      P(2)=ABS(PAR(2))                                                  CELL      28
      P(4)=PAR(7)                                                       CELL      29
      P(5)=PAR(6)                                                       CELL      30
*  SET ENTRANCE AND EXIT ANGLES EQUAL TO EACH OTHER.                    CELL      31
*  SAME CONVENTION AS FOR MAGNET HOLDS.                                 CELL      32
      P(6) = P(5)                                                       CELL      33
*           COMPUTE MAGNETS, SET UP MI LIST, STORE MAGNET PARAMETERS    CELL      34
C   SET UP QF                                                           CELL      35
      L=M-13                                                            CELL      36
      CALL STDAT (L,1,1,6,P)                                            CELL      37
      CALL MAGNET(L,P)                                                  CELL      38
      MI(2)=L                                                           CELL      39
      MI(11)=L                                                          CELL      40
C SET UP QD                                                             CELL      41
      L=M-14                                                            CELL      42
      P(2)=-P(2)                                                        CELL      43
      CALL STDAT(L,1,1,6,P)                                             CELL      44
      CALL MAGNET(L,P)                                                  CELL      45
      MI(5)=L                                                           CELL      46
      MI(8)=L                                                           CELL      47
C SET UP QS                                                             CELL      48
      L=M-15                                                            CELL      49
      CALL DRIFT(L,SOV2)                                                CELL      50
      MI(1)=L                                                           CELL      51
```

```
          MI(6)=L                                                        CELL      52
          MI(7)=L                                                        CELL      53
          MI(12)=L                                                       CELL      54
C SET UP QT                                                              CELL      55
          L=M-16                                                         CELL      56
          CALL DRIFT(L,TOV2)                                             CELL      57
          MI(3)=L                                                        CELL      58
          MI(4)=L                                                        CELL      59
          MI(9)=L                                                        CELL      60
          MI(10)=L                                                       CELL      61
          L=M-1                                                          CELL      62
          CALL MMM(M,12,MI)                                              CELL      63
C IF CYCSWT IS ON, COMPUTE CELL MATRIX ONLY FOR TEMPORARY USE            CELL      64
C                                       (FOR DESIGN INSTRUCTIONS).       CELL      65
          IF(CYCSWT) RETURN                                              CELL      66
C COMPUTE 12 CYCLED MATRICES                                             CELL      67
          CALL CYCLE (M,12,MI,0,0)                                       CELL      68
C STORE THE MILIST AS DATA FOR Q                                         CELL      69
          CALL MOVLEV(MI,MMI,12)                                         CELLL      2
          CALL STDAT(M,3,1,12,MMI)                                       CELLL      3
                                                                         CELLL      4
          RETURN                                                         CELL      71
          END                                                            CELL      72
```

```
      SUBROUTINE CFD(M,PAR)                                               CFD       2
C                                                                         CFD       3
C  SET UP MATRICES FOR FDDF CELL (WITH DRIFT SPACES)                      CFD       4
C  THE LENGTHS OF THE F AND D MAGNETS MAY BE DIFFERENT                    CFD       5
C                   PAR = INPUT PARAMETERS                                CFD       6
C  INPUT CARDS...                                                         CFD       7
C  Q     CFD       LFMAG    LDMAG    K        RHO      LSTR     LSEP      CFD       8
C                  GAMMOP                                                 CFD       9
C                                                                         CFD      10
C  THIS SUBROUTINE GENERATES 17 MATRICES AS FOLLOWS...                    CFD      11
C  QF = THE FOCUSSING MAGNET = MATRIX M - 13                              CFD      12
C  QD = THE DEFOCUSSING MAGNET = MATRIX M - 14                            CFD      13
C  QS = HALF THE CENTER DRIFT SPACE = MATRIX M - 15                       CFD      14
C  QT = HALF THE DRIFT SPACE BETWEEN A D AND AN F MAGNET = MATRIX M       CFD      15
C  THE FDDF CELL=Q=THE PRODUCT OF QS QF QT QT QD QS QS QD QT QF QS        CFD      16
C               = MATRIX M                                                CFD      17
C  QI = THE I-TH CYCLED TRANSFER MATRIX = MATRIX M - I, WHERE             CFD      18
C  I = 1, 12., AND Q1 = 0.                                                CFD      19
C  OMEGAX IS SET = 1., AND DELTA IS SET = 0. FOR THE CFD MAGNETS.         CFD      20
C                                                                         CFD      21
      COMMON/BMI/MI(5000)                                                 BMIL      1
      COMMON/NELS/NELS                                                    BMIL      2
      LEVEL 2,MI                                                          BMIL      3
                                                                          BMIL      4
                                                                          BMI       3
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER     SWTCH     2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                            SWTCH     3
                                                                          SWTCH     4
      DIMENSION PAR(8),P(8)                                               CFD      24
      DIMENSION MMI(12)                                                   CFDL      1
C                                                                         CFD      25
      DO 1 I=1,8                                                          CFD      26
    1 P(I) = PAR(I)                                                       CFD      27
      SLFMAG = P(1)                                                       CFD      28
      SLDMAG = P(2)                                                       CFD      29
      SOV2 = P(5)/2.                                                      CFD      30
      TOV2 = P(6)/2.                                                      CFD      31
      P(2) = ABS(P(3))                                                    CFD      32
      P(3) = P(4)                                                         CFD      33
      P(4) = P(8)                                                         CFD      34
      P(5) = P(7)                                                         CFD      35
      P(6) = P(5)                                                         CFD      36
C  SET UP QF                                                              CFD      37
      L = M - 13                                                          CFD      38
      CALL STDAT(L,1,1,6,P)                                               CFD      39
      CALL MAGNET(L,P)                                                    CFD      40
      MI(2) = L                                                           CFD      41
      MI(11) = L                                                          CFD      42
C  SET UP QD                                                              CFD      43
      L = M - 14                                                          CFD      44
      P(1) = SLDMAG                                                       CFD      45
      P(2) = -P(2)                                                        CFD      46
      CALL STDAT(L,1,1,6,P)                                               CFD      47
      CALL MAGNET(L,P)                                                    CFD      48
      MI(5) = L                                                           CFD      49
      MI(8) = L                                                           CFD      50
C  SET UP QS                                                              CFD      51
```

```
      L = M - 15                                                        CFD       52
      CALL DRIFT(L,SOV2)                                                CFD       53
      MI(1) = L                                                         CFD       54
      MI(6) = L                                                         CFD       55
      MI(7) = L                                                         CFD       56
      MI(12) = L                                                        CFD       57
C  SET UP QT                                                            CFD       58
      L = M - 16                                                        CFD       59
      CALL DRIFT(L,TOV2)                                                CFD       60
      MI(3) = L                                                         CFD       61
      MI(4) = L                                                         CFD       62
      MI(9) = L                                                         CFD       63
      MI(10) = L                                                        CFD       64
C  COMPUTE THE 12 CYCLED MATRICES.                                      CFD       65
      L = M - 1                                                         CFD       66
      CALL MMM(M,12,MI)                                                 CFD       67
C  IF THE SWITCH CYCSWT IS ON, WE ONLY COMPUTE THE CELL MATRICES        CFD       68
C FOR TEMPORARY USE (FOR INSTRUCTION DCFD)                              CFD       69
      IF (CYCSWT)  RETURN                                               CFD       70
      CALL CYCLE (M,12,MI,0,0)                                          CFD       71
C  LENGTH OF QF, QD = LMAG.                                             CFD       72
C  LENGTH OF QS = LSTR/2.                                               CFD       73
C  LENGTH OF QT = LSEP/2.                                               CFD       74
C  LENGTH OF Q = 2LFMAG + 2LDMAG + 2LSTR + 2LSEP.                       CFD       75
C  LENGTH OF QI = LENGTH OF Q.  I = 1,12.                               CFD       76
C  STORE THE MILIST AS DATA FOR Q                                       CFD       77
      CALL MOVLEV(MI,MMI,12)                                            CFDL       2
      CALL STDAT(M,3,1,12,MMI)                                          CFDL       3
                                                                        CFDL       4
      RETURN                                                            CFD       79
      END                                                               CFD       80
```

```
      SUBROUTINE CHKD(NFT,KN,ITP)                                       CHKD       2
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/FLTN/IFL(15)                                               FLTN       2
                                                                        FLTN       3
      COMMON/INSTR/OPNAME,NAME,OP,KA,KB,OBJA,OBJB,NXTM,TRA,LQ,NQ,LIN,   INSTR      2
     1             NIN,LFL,NFL,LBC,NBC,NTOT,IX,KIND,TYPE,ROWS,COLS,EXTR, INSTR     3
     2             M,NF,NB,NI,MSUBR,ISIGN                               INSTR      4
      INTEGER OPNAME,OP,OBJA,OBJB,TRA,TYPE,ROWS,COLS,EXTR               INSTR      5
                                                                        INSTR      6
      INTEGER ELNUM                                                     CHKD       6
      DATA MASKF/77000000000000000000B/                                 CHKD       7
      DATA MASKL/   77777 77777 00000 00000 B/,                         CHKD       8
     .     MASKBLK/ 00000 00000 55555 55555 B/,                         CHKD       9
     .     MASKLBL/ 55555 55555 00000 00000 B/,                         CHKD      10
     .     IBLANK/1H /                                                  CHKD      11
                                                                        CHKD      12
C  DETERMINE KIND AND TYPE OF NFT                                       CHKD      13
      MSK = MASKF.AND.NFT                                               CHKD      14
      IF (MSK.NE.IFL(1).AND.MSK.NE.IFL(12))  GO TO 12                   CHKD      15
      MSK= SHIFT(NFT,6)                                                 CHKD      16
      MSK = MASKF.AND.MSK                                               CHKD      17
 12   DO 15 L=2,14                                                      CHKD      18
      IF (MSK.EQ.IFL(L)) GO TO 25                                       CHKD      19
 15   CONTINUE                                                          CHKD      20
C  SYMBOLIC.  DETERMINE KIND                                            CHKD      21
      NFT1 = MASKL.AND.NFT                                              CHKD      22
      NFT2 = MASKBLK.OR.NFT1                                            CHKD      23
      IF (NFT2.EQ.NAME)  GO TO 25                                       CHKD      24
      NFT3 = .NOT.MASKL.AND.NFT                                         CHKD      25
      NFT4 = MASKLBL.OR.NFT3                                            CHKD      26
      IF (NFT4.NE.IBLANK)  GO TO 25                                     CHKD      27
      IK   = ELNUM(NFT2)                                                CHKD      28
      KN = INFF(20,IK)                                                  CHKD      29
      ITP = INFF(21,IK)                                                 CHKD      30
      RETURN                                                            CHKD      31
C  FLOATING POINT.  SCALAR AND SINGLE                                   CHKD      32
 25   KN = 4HSCAL                                                       CHKD      33
      ITP = 4HSNGL                                                      CHKD      34
      RETURN                                                            CHKD      35
      END                                                               CHKD      36
```

```
      SUBROUTINE CHRM(GK,RI,SS,TAN1, TAN2,BETX,BETY,BXP,BYP,CRX,CRY)
      DIMENSION BETX(6),BETY(6),BXP(6),BYP(6),
     1 BX1(6),BY1(6),BX2(6),BY2(6)
      CRX=0.
      CRY=0.
      IF(GK.EQ.0..AND.RI.EQ.0.) RETURN
      IF(RI.NE.0.) GO TO 11
      CRX = (BETX(4)+GK*BETX(2))*SS + BETX(3) - BXP(3)
      CRY = (BETY(4)-GK*BETY(2))*SS + BETY(3) - BYP(3)
      RETURN
   11 CONTINUE
      DO 1 I=1,6
      BX1(I)=BXP(I)
      BY1(I)=BYP(I)
      BX2(I)=BETX(I)
    1 BY2(I)=BETY(I)
      BX1(3)=BX1(3)-TAN1*RI*BX1(2)
      BX2(3)=BX2(3)+TAN2*RI*BX2(2)
      BY1(3)=BY1(3)+TAN1*RI*BY1(2)
      BY2(3)=BY2(3)-TAN2*RI*BY2(2)
      BX1(6)=BX1(6)+TAN1*RI*BX1(5)
      BX2(6)=BX2(6)-TAN2*RI*BX2(5)
      RI2=RI**2
      GKX=GK+RI2
      BKGX = GKX*BX1(2) + (1.+BX1(3)**2)/BX1(2)
      BKGY = -GK*BY1(2) + (1.+BY1(3)**2)/BY1(2)
      F2 = (SS*BKGY+BY2(3)-BY1(3) )/2.
      IF (GKX.NE.0.) GO TO 12
      F5 = SS*( BX1(5)+BX2(5)-SS**2*RI/6. )/2.
      F1 = SS*( BX1(2)+BX2(2)+SS*(BX2(3)-BX1(3))/3.) /2.
      F3 = ( (BX1(2)+BX2(2))*(BX1(5)+BX2(5))
     1      + BX1(2)*BX2(5) + BX2(2)*BX1(5) )/3.
     2      + (BX2(3)-BX1(3))*(5.*(BX1(5)+BX2(5))-RI*SS**2)*SS/30.
     3      - (BX1(2)+BX2(2))*RI*SS**2/12.
      F3 = F3*SS/2.
      GO TO 13
   12 F1 = (SS*BKGX+BX2(3)-BX1(3))/GKX/2.
      F5 = (RI*SS-BX2(6)+BX1(6))/GKX
      F3= (2.*BKGX*F5-RI*F1 + BX2(6)*BX2(2) + 2.*BX2(3)*BX2(5)
     1                    - BX1(6)*BX1(2) - 2.*BX1(3)*BX1(5) )/3./GKX
   13 CRX = - (GKX+RI2)*F1 + RI*(GK+GK)*F3 + RI*BKGX*F5
     1       - RI*(BXP(2)*BXP(6)-BETX(2)*BETX(6))
      CRY = - F2 + RI*BKGY*F5 + RI*(BYP(2)*BXP(6)-BETY(2)*BETX(6))
      IF (TAN1.EQ.0..AND.TAN2.EQ.0.) GO TO 14
      T1 = RI*TAN1
      T2 = RI*TAN2
      CRX = CRX + BXP(2)*( T1-TAN1*((GK+GKX)*BXP (5)-T1*BXP (6)) )
     1          + BETX(2)*( T2-TAN2*((GK+GKX)*BETX(5)+T2*BETX(6)) )
     2 +(BXP(3)+BX1(3))*BXP(5)*T1*TAN1 -(BETX(3)+BX2(3))*BETX(5)*T2*TAN2
      CRY = CRY + BYP (2)*(-T1+TAN1*(2.*GK*BXP (5)+T1*BXP (6)) )
     1           + BETY(2)*(-T2+TAN2*(2.*GK*BETX(5)-T2*BETX(6)) )
     2 -(BYP(3)+BY1(3))*BXP(5)*T1*TAN1+(BETY(3)+BY2(3))*BETX(5)*T2*TAN2
   14 CONTINUE
      CRX= -CRX - CRX
      CRY = -CRY - CRY
      RETURN
      END
```

```
      SUBROUTINE CMATX6(A,B,C)                                        CMATX6     2
      RETURN                                                          CMATX6     3
      END                                                             CMATX6     4
```

```
      SUBROUTINE CMXDIV(V,IN,IM,DET,IDIM)
C     REPLACES B BY (A INVERSE)*B.  DESTROYS A.
C     A IS N BY N, B IS N BY M.  A AND B ARE STORED IN V(N,N+M).
      COMPLEX SAVE,PIVOT,DETERM,DET,V,CABS
      DIMENSION V(IDIM,1)
      INTEGER P
      N=IN
      M=IM
C
      NM1=N-1
      NP1=N+1
      NPM=N+M
      DETERM=(1.0E0,0.0E0)
      P=1
  202 CONTINUE
      IMIN=P+1
C                                FIND PIVOT
      RPIVOT=0.0E0
      DO 209 I=P,N
      RSAVE=CABS(V(I,P))
      IF (RSAVE .LE. RPIVOT) GO TO 209
      RPIVOT=RSAVE
      IBIG=I
  209 CONTINUE
      IF (RPIVOT .GT. 0.0E0) GO TO 210
      DETERM=(0.0E0,0.0E0)
      GO TO 290
C                                ROW EXCHANGE AND ROW REDUCTION
  210 PIVOT=V(IBIG,P)
      DETERM=DETERM*PIVOT
      DO 219 J=P,NPM
      SAVE=V(IBIG,J)/PIVOT
      V(IBIG,J)=V(P,J)
      V(P,J)=SAVE
  219 CONTINUE
      IF(IBIG .NE. P) DETERM=-DETERM
      IF (P .GE. N) GO TO 250
      V(P,P) =(1.0E0,0.0E0)
C                                BLOCK REDUCTION
      DO 239 I=IMIN,N
      DO 238 J=IMIN,NPM
      V(I,J)=V(I,J)-V(I,P)*V(P,J)
  238 CONTINUE
      V(I,P)=(0.0E0,0.0E0)
  239 CONTINUE
      P=P+1
  249 GO TO 202
C                                BACK SUBSTITUTION
  250 CONTINUE
      IF (M .LE. 0) GO TO 290
      DO 259 K=NP1,NPM
      DO 253 P=1,NM1
      I=N-P
      DO 252 J=I,NM1
      V(I,K)=V(I,K)-V(J+1,K)*V(I,J+1)
  252 CONTINUE
  253 CONTINUE
```

```
  259 CONTINUE
C
  290 CONTINUE
      DET=DETERM
      RETURN
      END
```

```
      SUBROUTINE CONJ(A,ABAR)                                            CONJ      2
      RETURN                                                             CONJ      3
      END                                                                CONJ      4
```

```
      SUBROUTINE CONTYP(K,NB,ICTYP)                                  CONTYP   2
      RETURN                                                         CONTYP   3
      END                                                            CONTYP   4
```

```
        SUBROUTINE CONVMAT(KN,T,RX,RY,RW,RPQ)                             CONVMAT   2
                                                                          CONVMAT   3
*     IF KN=3, CONVERTS 7X7 MATRIX TO A 3X3                               CONVMAT   4
*     IF KN=7, CONVERTS 3X3 MATRIX TO A 7X7                               CONVMAT   5
        COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL    2
       1           LDFLG,FIN                                              CONTRL    3
        LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL    4
                                                                          CONTRL    5
        DIMENSION T(1),T1(49),T2(7,7),RX(2,3),RY(2,3),RW(3),RPQ(4)        CONVMAT   7
        EQUIVALENCE(T1,T2)                                                CONVMAT   8
                                                                          CONVMAT   9
        DO 1 I=1,49                                                       CONVMAT  10
        T1(I) = T(I)                                                      CONVMAT  11
    1   CONTINUE                                                          CONVMAT  12
        IF (KN.EQ.7) GO TO 15                                             CONVMAT  13
        IF (KN.NE.3) GO TO 30                                             CONVMAT  14
*                    CONVERT T ARRAY OF 7X7 TO RX,RY OF 3X3               CONVMAT  15
        DO 3 J=1,3                                                        CONVMAT  16
        DO 2 I=1,2                                                        CONVMAT  17
        RX(I,J) = 0.                                                      CONVMAT  18
        RY(I,J) = 0.                                                      CONVMAT  19
    2   CONTINUE                                                          CONVMAT  20
    3   CONTINUE                                                          CONVMAT  21
        DO 8 I=1,2                                                        CONVMAT  22
        DO 7 J=1,2                                                        CONVMAT  23
        RX(I,J) = T2(I,J)                                                 CONVMAT  24
        RY(I,J) = T2(I+2,J+2)                                             CONVMAT  25
    7   CONTINUE                                                          CONVMAT  26
    8   CONTINUE                                                          CONVMAT  27
        DO 9 I=1,2                                                        CONVMAT  28
        RX(I,3) = T2(I,6)                                                 CONVMAT  29
        RY(I,3) = T2(I+2,6)                                               CONVMAT  30
    9   CONTINUE                                                          CONVMAT  31
        RW(3) = T2(5,6)                                                   CONVMAT  32
*          EXTRACT P1,P2,Q1,Q2                                            CONVMAT  33
        DO 10 I=1,4                                                       CONVMAT  34
   10   RPQ(I) = T2(5,I)                                                  CONVMAT  35
        RETURN                                                            CONVMAT  36
*                    CONVERT RX,RY OF 3X3 TO T OF 7X7                     CONVMAT  37
   15   CONTINUE                                                          CONVMAT  38
        DO 17 I=1,49                                                      CONVMAT  39
        T(I) = 0.                                                         CONVMAT  40
        T1(I) = 0.                                                        CONVMAT  41
   17   CONTINUE                                                          CONVMAT  42
        DO 20 I=1,2                                                       CONVMAT  43
        DO 19 J=1,2                                                       CONVMAT  44
        T2(I,J) = RX(I,J)                                                 CONVMAT  45
        T2(I+2,J+2) = RY(I,J)                                             CONVMAT  46
   19   CONTINUE                                                          CONVMAT  47
   20   CONTINUE                                                          CONVMAT  48
        DO 21 I=1,2                                                       CONVMAT  49
        T2(I,6) = RX(I,3)                                                 CONVMAT  50
        T2(I+2,6) = RY(I,3)                                               CONVMAT  51
   21   CONTINUE                                                          CONVMAT  52
        T2(5,6) = RW(3)                                                   CONVMAT  53
        T2(5,5) = 1.                                                      CONVMAT  54
        T2(6,6) = 1.                                                      CONVMAT  55
```

```
      T2(7,7) = 1.
*        SET P1,P2,Q1,Q2
      DO 22 I=1,2
       T2(5,I) = T2(1,6)*T2(2,I) - T2(2,6)*T2(1,I)
      T2(5,I+2) = T2(3,6)*T2(4,I+2) - T2(4,6)*T2(3,I+2)
      RPQ(I) = T2(5,I)
   22 RPQ(I+2) = T2(5,I+2)
      DO 26 I=1,49
      T(I) = T1(I)
   26 CONTINUE
      RETURN

*                    ERROR IF KN IS NOT MXY OR R7
   30 WRITE (3,100) KN
  100 FORMAT (5H *** ,*ERROR.  CANNOT CONVERT MATRIX OF KIND  *,I5)
      ERROR = .TRUE.
      RETURN
      END
```

```
      SUBROUTINE CUBSOL(PP,QQ,RR,Y)                                     CUBSOL     2
      RETURN                                                            CUBSOL     3
      END                                                               CUBSOL     4
```

```
      SUBROUTINE CYAE(M)                                                CYAE       2
                                                                        CYAE       3
*        USING BETATRON FUNCTIONS, COMPUTES BEAM ENVELOPES ALONG BEAM   CYAE       4
*  LINE FROM PREVIOUSLY CALCULATED EMITTANCES, EPSX,EPSY,DP/P.          CYAE       5
                                                                        CYAE       6
C.....................................................                  CYAE       7
* NAME CYAE  KA    /BL  BEAM  KSQ       EPXCO     EPYCO                 CYAE       8
C.......................................................                CYAE       9
*  BL = BEAMLINE NAME                                                   CYAE      10
*  BEAM = NAME OF BVAL OR CYEM THAT GIVES EMITTANCES AND DP/P           CYAE      11
*  KSQ = EPS/EPS0, WHERE EPS0 IS EMITTANCE FROM BVAL (SIG**2/BETA)      CYAE      12
*                  AND EPS IS EMITTANCE TO BE USED BY CYAE.             CYAE      13
*                  XB AND XP ARE THUS MULTIPLIED BY SQRT(KSQ).          CYAE      14
*  EPXCO, EPYCO = EQUIVALENT EMITTANCES THAT GIVE CLOSED ORBIT DISPLACEM CYAE      15
*  KA = 0 OR NON-0 ---  ADD BETATRON AND MOMENTUM DISPLACEMENTS IN QUADR CYAE      16
*                   OR ALGEBRAICALLY.                                   CYAE      17
                                                                        CYAE      18
                                                                        CYAE      19
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/BMI/MI(5000)                                               BMIL       1
      COMMON/NELS/NELS                                                  BMIL       2
      LEVEL 2,MI                                                        BMIL       3
                                                                        BMIL       4
                                                                        BMI        3
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL   DIM        2
                                                                        DIM        3
                                                                        CYAE      23
      COMMON /MC/ ML    $        LEVEL 2, ML                            CYAEL      1
                                                                        CYAEL      2
      DIMENSION P(4),BETX(9),BETY(9),ML(3)                              CYAE      24
      EQUIVALENCE (EPX,P(1)),(EPY,P(2)),(EPL,P(3)),(DPP,P(4))           CYAE      25
      EQUIVALENCE (BX,BETX(2)),(GX,BETX(4)),(DX,BETX(5)),(DDX,BETX(6)), CYAE      26
     1            (BY,BETY(2)),(GY,BETY(4)),(DY,BETY(5)),(DDY,BETY(6))  CYAE      27
                                                                        CYAE      28
      DATA IBLNK/1H /                                                   CYAE      29
                                                                        CYAE      30
                                                                        CYAE      31
*                 GET DATA  EPX,EPY AND DP/P                            CYAE      32
      MD = MDAT(M,2)                                                    CYAE      33
      CALL DATA (MD,5,1,4,P)                                            CYAE      34
                                                                        CYAE      35
*     KA DETERMINES CALCULATION FOR XTOT AND XTOT PR                    CYAE      36
      KA = INFF(4,M)                                                    CYAE      37
                                                                        CYAE      38
*  GET BEAM LINE, MAKE MILIST AND MULTIPLY MATRICES                     CYAE      39
*      K IS NUMBER OF ELEMENTS IN BEAM                                  CYAE      40
      MB = MDAT(M,1)                                                    CYAE      41
      K = INFF(17,MB)                                                   CYAE      42
      MS = MEND - 1                                                     CYAE      43
      CALL MIFILL (MB,1,K,NELS,MI)                                      CYAE      44
      CALL MMM (MS,NELS,MI)                                             CYAE      45
      KW = NELS + 1                                                     CYAE      46
```

```
      MI(KW) = MI(1)
      I = 0
      EL = 0.
      FL = 0.
      MAT = IBLNK

      XKSQ=FDAT(M,1)
      RK=SQRT(XKSQ)
      EPXCO=FDAT(M,2)
      EPYCO=FDAT(M,3)
      RX=SQRT(EPXCO/EPX)
      RY=SQRT(EPYCO/EPY)

*    PRINT TABLE HEADING AND SIGX AND SIGY, AS DEFINED BY FACTK
      CALL DASH
      WRITE(3,1) P(1),EPXCO,P(3),P(2),EPYCO,P(4)
      IG = 1
      IF (KA.EQ.0) IG = 2
      GO TO (104,105) IG
  104 WRITE (3,6)
      GO TO 106
  105 WRITE (3,8)
  106 WRITE(3,2) RK
      WRITE(3,11)

  115  CONTINUE
      CALL BET(MS,BETX,BETY,1)
*                   COMPUTE AND PRINT BEAM ENVELOPES
      X=SQRT(EPX*BX)*RK
      Y=SQRT(EPY*BY)*RK
      XP=DX*DPP
      YP=DY*DPP
      XCO=RX*X
      YCO=RY*Y
      XPR=SQRT(EPX*GX)*RK
      YPR=SQRT(EPY*GY)*RK
      XPPR=DDX*DPP*RK
      YPPR=DDY*DPP*RK
      XCOPR=RX*XPR
      YCOPR=RY*YPR
      GO TO (116,117) IG
  116 XTOT = X + ABS(XP)
      XPRTOT = XPR + ABS(XPPR)
      YTOT = Y + ABS(YP)
      YPRTOT = YPR + ABS(YPPR)
      GO TO 118
  117 CONTINUE
      XTOT = SQRT( X * X + XP * XP )
      XPRTOT = SQRT( XPR*XPR + XPPR*XPPR)
      YTOT = SQRT(Y*Y + YP*YP)
      YPRTOT = SQRT(YPR*YPR + YPPR*YPPR)
  118 CONTINUE
      XTOT=XTOT+XCO
      YTOT=YTOT+YCO
      XPRTOT=XPRTOT+XCOPR
      YPRTOT=YPRTOT+YCOPR
      EL = EL + FL
```

```
          WRITE(3,12) I,MAT,EL,X,XP,XCO,XTOT,Y,YP,YCO,YTOT,XPRTOT,YPRTOT     CYAE     104
                                                                             CYAE     105
          I = I + 1                                                          CYAE     106
          IF (I.EQ.KW) GO TO 125                                             CYAE     107
          NI = MI(I)                                                         CYAE     108
          CALL DATA(NI,5,1,1,FL)                                             CYAE     109
          MAT = MNAME(NI)                                                    CYAE     110
          ML(1) = - NI                                                       CYAE     111
          ML(2) = MS                                                         CYAE     112
          ML(3) = NI                                                         CYAE     113
          CALL MMM(MS,3,ML)                                                  CYAE     114
          GO TO 115                                                          CYAE     115
                                                                             CYAE     116
      125 CALL DASH                                                          CYAE     117
          RETURN                                                             CYAE     118
                                                                             CYAE     119
C********************************************************************** CYAE     120
                                                                             CYAE     121
        1  FORMAT (3X,*BEAM ENVELOPES (MM,MRAD)*/                            CYAE     122
         1  5X,*EMITTANCES (MM-RAD) --- EPSX = *,F10.6,5X,*EPSXCO = *,      CYAE     123
         2  F10.6,5X,*EPSL = *,F13.6/29X,*EPSY = *,F10.6,5X,*EPSYCO = *,     CYAE     124
         3  F10.6,5X,*SIGP = *,F10.6,* (0/00)*)                              CYAE     125
      2   FORMAT(100X,*DISPLACEMENT = *,F5.2,1H*,*SIGMA*)                    CYAE     126
      6   FORMAT (1H+,13X,*XTOT = X + XP + XCO*,5X,                          CYAE     127
         1            *XTOTPR = XPR + XPPR +XCOPR*)                          CYAE     128
      8   FORMAT (1H+,13X,30HXTOT = SQRT(X*X + XP*XP + XCO),5X,              CYAE     129
         1  42HXTOTPR = SQRT(XPR*XPR + XPPR*XPPR + XCOPR))                   CYAE     130
       9   FORMAT (I4,1X,A5,F9.4,12F9.4)                                     CYAE     131
      10  FORMAT (/2X,*POS*,10X,*S*,8X,*X*,7X,*X PR*,6X,                     CYAE     132
         1 *XP*,5X,*XP PR*,4X,*XTOT*,3X,*XTOT PR*,5X,*Y*,7X,                 CYAE     133
         2 *Y PR*,5X,*YP*,6X,*YP PR*,5X,*YTOT*,3X,*YTOT PR*/1X,131(1H-))     CYAE     134
      11  FORMAT(/2X,*POS*,10X,*S*,13X,*XB*,7X,*XP*,7X,*XCO*,6X,*XTOT*,      CYAE     135
         1                         10X,*YB*,7X,*YP*,7X,*YCO*,6X,*YTOT*,      CYAE     136
         2                         10X,*XPRTOT*,3X,*YPRTOT*/1X,131(1H-))     CYAE     137
      12  FORMAT(I4,1X,A5,F9.4,5X,4F9.4,5X,4F9.4,5X,2F9.4)                   CYAE     138
                                                                             CYAE     139
          END                                                                CYAE     140
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
76	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT
98	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT

```
            SUBROUTINE CYCLE(M,K,MI,L,WS)                                  CYCLE     2
                                                                           CYCLE     3
            COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL DIM      2
                                                                           DIM       3
            DIMENSION M3(3)                                                CYCLE     5
            DIMENSION MI(1)                                                CYCLE     6
            COMMON/MD/M3     $ LEVEL 2,MI,M3                               CYCLEL    1
                                                                           CYCLEL    2
            INTEGER WS                                                     CYCLE     7
                                                                           CYCLE     8
            IF (WS.EQ.0) GO TO 5                                           CYCLE     9
            NO=MEND-3                                                      CYCLE    10
            NC=K                                                           CYCLE    11
            GO TO 6                                                        CYCLE    12
       5    NO=M-1                                                         CYCLE    13
            NC=K-1                                                         CYCLE    14
       6    IF (L.EQ.0) GO TO 1                                            CYCLE    15
            CALL MMM(M,K,MI)                                               CYCLE    16
       1    MI(K+1)=MI(1)                                                  CYCLE    17
            DO 2 I=1,NC                                                    CYCLE    18
            IF (I.NE.1) GO TO 11                                           CYCLE    19
            M3(2)=M-1                                                      CYCLE    20
            NJ=NO - 1                                                      CYCLE    21
            GO TO 12                                                       CYCLE    22
      11    NJ=NO - I                                                      CYCLE    23
            M3(2) = NJ + 1                                                 CYCLE    24
      12    M3(3) = MI(I)                                                  CYCLE    25
            M3(1) =-M3(3)                                                  CYCLE    26
            CALL MMM (NJ,3,M3)                                             CYCLE    27
       2    CONTINUE                                                       CYCLE    28
            RETURN                                                         CYCLE    29
            END                                                            CYCLE    30
```

```
      SUBROUTINE CYEM(M)                                                CYEM       2
                                                                        CYEM       3
*  CALCULATES ELECTRON INTEGRALS, RF OUTPUT, ELECTRON EMITTANCE FACTORS CYEM       4
                                                                        CYEM       5
C.......................................................................CYEM       6
C EPS  CYEM  KA  KB // BL     E         R         F         V         K  CYEM       7
C.......................................................................CYEM       8
                                                                        CYEM       9
C  KA = OUTPUT OPTION ---- SEE BELOW                                    CYEM      10
C  KB = NUMBER OF SUPERPERIODS                                          CYEM      11
C       IF BL IS A FXPT INSTRUCTION, BEAM LINE AND KB ARE TAKEN FROM    CYEM      12
C       THOSE OF THE FXPT INSTRUCTION.                                  CYEM      13
C  BL = NAME OF BEAM LINE (BML INSTRUCTION) OR OF FXPT INSTRUCTION      CYEM      14
C   E = ELECTRON BEAM ENERGY (GEV)                                      CYEM      15
C   R = MACHINE RADIUS (M)                                              CYEM      16
C   F = RF FREQUENCY (MHZ)                                              CYEM      17
C   V = RF VOLTS/TURN (MV)                                              CYEM      18
C    K = KAPPA = COUPLING COEFFICIENT (SQRT(EPY/EPN)                    CYEM      19
                                                                        CYEM      20
C   --------------------------------------------------------------------CYEM      21
C         KA OPTIONAL OUTPUT CAN ONLY BE DONE WHEN BL IS FXPT INSTR.    CYEM      22
C                                                                       CYEM      23
C  KA = 0,1  PRINT ONLY AT START OF BEAM LINE                           CYEM      24
C  KA = 2,3,4  PRINT AT EVERY ELEMENT                                   CYEM      25
C  KA = EVEN, BETATRON PART OF SIGMA MATRIX                             CYEM      26
C  KA = ODD,  TOTAL SIGMA MATRIX                                        CYEM      27
C  KA = 2,3  SIGMA MATRIX LONG OUTPUT                                   CYEM      28
C  KA = 4    AIJ, BIJ LONG OUTPUT                                       CYEM      29
                                                                        CYEM      30
C  KA   POSITIVE ---- PRINTS RII, TIJ.       RII = SQRT(SIGMA(I,I))     CYEM      31
C                          TIJ = ARCTAN( SIJ/SII-SJJ ) / 2              CYEM      32
C                     RII IS THE PROJECTION OF BEAM ELLIPSE ON I AXIS   CYEM      33
C                     TIJ IS TILT OF ELLIPSE PROJECTION IN I-J PLANE    CYEM      34
C  KA   NEGATIVE ---- PRINTS  SIJ = SIGMA(I,J)  FOR ALL ELEMENTS        CYEM      35
                                                                        CYEM      36
C        AIJ = 4 X REAL( F1I* X F1J ),    BIJ = 4 X REAL( F3I* X F3J )  CYEM      37
C        F = INVERSE(E), E = MATRIX OF EIGENVECTORS, FIJ* = CONJ(FIJ)   CYEM      38
C        SIGMA = INVERSE( A/EP1 + B/EP3 )                               CYEM      39
C        EP1, EP3 = GENERALIZED EMITTANCES (SIGMA**2/BETA)              CYEM      40
C        STOTIJ = SBETIJ + ETAI*ETAJ*SIGDP**2                           CYEM      41
C        IF EP3=0, SETS EP3=EPY                                         CYEM      42
C    IF KAPPA=0 AND BL=FXPT INSTR., KAPPA IS SET TO 1.E-08              CYEM      43
C  -------------------------------------------------------------------- CYEM      44
C                                                                       CYEM      45
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/BMI/MI(5000)                                               BMIL       1
      COMMON/NELS/NELS                                                  BMIL       2
      LEVEL 2,MI                                                        BMIL       3
                                                                        BMIL       4
                                                                        BMI        3
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL   DIM        2
                                                                        DIM        3
```

```
      C                                                                   MOVTOG       2
            COMMON/MOVTOG/TOG                                             MOVTOG       3
            LOGICAL TOG                                                   MOVTOG       4
      C                                                                   MOVTOG       5
            DIMENSION VWW(7)                                              MY3CYEM      2
            LOGICAL TOG1                                                  MY3CYEM      3
            DIMENSION BETX(9),BETY(9),PAR(5),P(6),EM(4),ML(3)             CYEM        49
            COMMON /MF/ ML   $   LEVEL 2,ML                               CYEMLL       1
                                                                          CYEMLL       2
            DIMENSION EV(4,8),VZ(7,7),VW(7),RW(3),AA(4,4),BB(4,4),CC(4,8) CYEM        50
            DIMENSION RX(2,3),RY(2,3),HMOV(4)                             CYEM        51
            DIMENSION EI(2)                                               CYEM        52
            COMPLEX GAMT                                                  CYEM        53
            COMPLEX EV,DET,EJ1,EJ3,EK1,EK3,EI,XM2I                        CYEM        54
            COMPLEX DETX,DETY                                             CYEM        55
            REAL I41,I43,I51,I53,J1,J3,J6,K,KSQ,LH                        CYEM        56
                                                                          CYEM        57
            EQUIVALENCE (L,P(1)),(DB,P(2)),(BRHO,P(3)),(BZ,P(4)),(OM,P(4)) CYEM       58
            EQUIVALENCE (PSI,XI)                                          CYEM        59
            EQUIVALENCE (EPSX,EM(1)),(EPSY,EM(2)),(EPSL,EM(3)),(SIGP,EM(4)) CYEM      60
            EQUIVALENCE (B,BETX(2)),(A,BETX(3)),(E,BETX(5)),(F,BETX(6))   CYEM        61
            EQUIVALENCE (GAM,BETX(4))                                     CYEM        62
            EQUIVALENCE (PC,PAR(1)),(R,PAR(2)),(FREQ,PAR(3)),(V0,PAR(4)), CYEM        63
           1    (FKAP,PAR(5))                                             CYEM        64
            EQUIVALENCE (ANG0,IANG0),  (ANG1,IANG1)                       CYEM        65
                                                                          CYEM        66
            REAL I1,I2,I3,I4,I5,L,KX,KK,N,LMTOT,K3R,JE,JX,JZ              CYEM        67
            LOGICAL BSW,MOVSW,LONG,FLAG,SIGFLG                            CYEM        68
            DATA RME/.0005110041/                                         CYEM        69
            DATA PI,CGAM,CQP,C,JE,JX,JZ                                   CYEM        70
           1/3.1415926536,8.846002206E-5,1.467471313E-6,2.997925E8,2.,1.,1./ CYEM     71
            DATA RADEG/.0174532925/                                       CYEM        72
            DATA RW/3*0.0/                                                CYEM        73
                                                                          CYEM        74
            CALL RANGET(NRN)                                              MY3CYEM      4
                                                                          CYEM        75
            XM2I = (0.,-2.)                                               CYEM        76
      *     GET DATA  P/C,R,FREQ,VRF,KAPPA                                CYEM        77
      *     UNITS ---    GEV/C, METERS, MHZ, MV                           CYEM        78
            CALL DATA (M,1,1,5,PAR)                                       CYEM        79
            FMHZ = FREQ                                                   CYEM        80
            V0MV = V0                                                     CYEM        81
            FREQ = FREQ*1.E6                                              CYEM        82
            V0 = V0*1.E-3                                                 CYEM        83
                                                                          CYEM        84
      *     GET BEAM LINE, MAKE MI LIST AND MULTIPLY MATRICES             CYEM        85
      *     K IS NUMBER OF ELEMENTS IN BEAM LINE                          CYEM        86
            MB = MDAT(M,1)                                                CYEM        87
            KA = 0                                                        CYEM        88
            KB = INFF(5,M)                                                CYEM        89
            FLAG = .FALSE.                                                CYEM        90
            MS = MEND - 1                                                 CYEM        91
            IOPMB = INFF(1,MB)                                            CYEM        92
            NGO=1                                                         CYEM        93
            IF (IOPMB.EQ.4HFXPT) NGO=2                                    CYEM        94
            CALL DASH                                                     CYEM        95
            WRITE (3,13)                                                  CYEM        96
```

```
      GO TO (51,52) NGO

   52 MFX=MB
      MB=MDAT(MFX,2)
      NP=MDAT(MFX,1)
      KA = INFF(4,M)
      SIGFLG = .FALSE.
      IF (KA.GE.0) GO TO 55
      SIGFLG = .TRUE.
      KA = -KA
   55 KAF = INFF(4,MFX)
      LONG = .FALSE.
      IF(KA.GT.1) LONG =.TRUE.
      KB = INFF(5,MFX)
      MS7 = M7END - 3

   53 CONTINUE
      CALL DATA(MFX,7,1,49,VZ)
      DO 60 I=1,4
      EV(I,1) = CMPLX(VZ(I,2),VZ(I,3))
      EV(I,3) = CMPLX(VZ(I,4),VZ(I,5))
      EV(I,2) = CONJG(EV(I,1))
   60 EV(I,4) = CONJG(EV(I,3))
      DETX = EV(1,1)*EV(2,2) - EV(2,1)*EV(1,2)
     1     + EV(3,1)*EV(4,2) - EV(4,1)*EV(3,2)
      DETY = EV(1,3)*EV(2,4) - EV(2,3)*EV(1,4)
     1     + EV(3,3)*EV(4,4) - EV(4,3)*EV(3,4)
      CALL STOR7(MS7,VZ,RW)
      CALL DATA(MFX,7,1,7,VW)
      CALL RXY(MFX,RX,RY,RW)
      CALL STXY(MS,RX,RY,RW)
      IF (.NOT.FLAG) GO TO 54
      II = 0
      CALL DASH
      IF (KA.EQ.1.OR.KA.EQ.3) GO TO 56
      WRITE (3,19)
      GO TO 57
   56 WRITE (3,20)
   57 IF (SIGFLG) GO TO 58
      WRITE (3,16)
      GO TO 503
   58 WRITE (3,18)
      GO TO 503
   54 WRITE (3,14)

   51 CONTINUE
      I = INFF(17,MB)
      CALL MIFILL (MB,1,I,NELS,MI)
      IF(NGO.EQ.2) GO TO 59
      CALL MMM(MS,NELS,MI)
      IF (KB.GE.0) GO TO 59
      NELS=NELS+1
      MI(NELS)=MEND-6
      CALL REF(MEND-6,MS)
      ML(1)=MS
      ML(2)=MEND-6
      CALL MMM(MS,2,ML)
```

```
   59   IF(KB.LT.0) KB=-2*KB
        KW = NELS + 1
        MI(KW) = MI(1)
        BSW = .FALSE.
        MOVSW = .FALSE.
        I1 = 0.
        I2 = 0.
        I3 = 0.
        I4 = 0.
        I5 = 0.
        I41 = 0.
        I43 = 0.
        I51 = 0.
        I53 = 0.
        II = 0

*************************************************************************

  100   CALL BET(MS,BETX,BETY,1)
        IF (II.EQ.0) BX0=BETX(2)
        IF (NGO.EQ.2) GO TO 503
  101   II = II + 1
*       IF I = KW, END OF BEAM LINE
        IF (II.GT.KW) GO TO 400
        NI = MI(II)
        IF (FLAG) GO TO 121
*       IF NOT AT BEGINNING OR END OF A BENDING MAGNET, CONTINUE
  110   IOP = INFF(1,NI)
        IF (BSW) GO TO 300
  111   IF (IOP.NE.4HMOVE)  GO TO 115
        DO 105 IV=1,7
  105   VWW(IV) = VW(IV)
        TOG1 = TOG
        CALL RANGET(NRN1)
        NN1 = M7END - 5
        CALL MOVE(NI,NN1,VWW)
        CALL RANSET(NRN1)
        TOG = TOG1
        CALL DATA(NI,1,7,4,HMOV)
        MOVSW = .TRUE.
        MQ = MDAT(NI,1)
        IOP = INFF(1,MQ)
        IF (IOP.NE.4HMAG ) GO TO 120
        CALL DATA (MQ,1,1,6,P)
        GO TO 116
  115   IF (IOP.NE.4HMAG ) GO TO 120
*            GET MAGNET DATA.  L,K,RHO,OMEGA,GAMMA,DELTA
        CALL DATA (NI,1,1,6,P)
  116   IF ( (BZ.NE.0.).OR.(NGO.EQ.2) )   GO TO 200
  120   CONTINUE

        GO TO (122,121) NGO
  121   NF = M7END - 5
        CALL TRKCN(MS7,NI,VW,NP,0)
        NI = NF
        IF (KAF.EQ.0) GO TO 122

```

```
C  PRINT AIJ, BIB
  900  CONTINUE
       IPOS = II-1
       IF(IPOS.NE.0) GO TO 901
       MATI=1H
       IF(FLAG) GO TO 903
       GO TO 902
  901  IF (.NOT.LONG) GO TO 122
       MK=MI(IPOS)
       MATI=INFF(2,MK)
       IF(FLAG) GO TO 903
       IF(KA.NE.4) GO TO 122
  902  WRITE(3,15) IPOS,MATI,
     1  AA(1,1),AA(1,2),AA(2,2),AA(1,3),AA(1,4),
     2  AA(2,3),AA(2,4),AA(3,3),AA(3,4),AA(4,4),
     3  BB(1,1),BB(1,2),BB(2,2),BB(1,3),BB(1,4),
     4  BB(2,3),BB(2,4),BB(3,3),BB(3,4),BB(4,4)
       GO TO 122
  903  WRITE (3,17) IPOS,MATI,
     1              CC(1,1),CC(1,2),CC(2,2),CC(1,3),CC(1,4),
     2              CC(2,3),CC(2,4),CC(3,3),CC(3,4),CC(4,4)
       IF (.NOT.LONG.OR.KA.EQ.4) GO TO 400
       GO TO 503

  122  CONTINUE
       ML(1) = -NI
       ML(2) = MS
       ML(3) = NI
       CALL MMM(MS,3,ML)
       GO TO 100

*****************************************************************************

C  CALCULATE BILINEAR FORMS ---- GENERALIZED ELLIPSE MATRICES
  503  CALL RTRV7(MS7,VZ,RW)
C      H = GAM*E*E + 2.*A*E*F + B*F*F
       ASSIGN 504 TO IRET
  500  DO 501 I=1,4
       EV(I,1) = CMPLX( VZ(I,2),VZ(I,3) )
       EV(I,3) = CMPLX( VZ(I,4),VZ(I,5) )
       EV(I,2) = CONJG( EV(I,1) )
       EV(I,4) = CONJG( EV(I,3) )
       DO 502 J=5,8
  502  EV(I,J) = CMPLX(0.,0.)
  501  EV(I,I+4) = CMPLX(1.,0.)
  600  DO 601 J=1,2
       I=2*J-1
  601  EI(J) = CONJG( VZ(1,6)*EV(2,I) - VZ(2,6)*EV(1,I)
     1               + VZ(3,6)*EV(4,I) - VZ(4,6)*EV(3,I) )
       EJ1 = EI(1)*EV(1,1)*XM2I/DETX
       EJ3 = EI(1)*EV(3,1)*XM2I/DETX
       EK1 = EI(2)*EV(1,3)*XM2I/DETY
       EK3 = EI(2)*EV(3,3)*XM2I/DETY
       CALL CMXDIV(EV,4,4,DET,4)
       GO TO IRET,(53,504)
  504  DO 505 I=1,4
       DO 505 J=1,4
```

```
      AA(I,J) = 4.0 * REAL( CONJG(EV(1,I+4)) * EV(1,J+4) )
      BB(I,J) = 4.0 * REAL( CONJG(EV(3,I+4)) * EV(3,J+4) )
      IF (.NOT.FLAG) GO TO 505
      CC(I,J) = REP1*AA(I,J) + REP3*BB(I,J)
      UU = 0.
      IF (J.EQ.I) UU = 1.
      CC(I,J+4) = UU
  505 CONTINUE

      IF (.NOT.FLAG) GO TO 101
      CALL MXDIV(CC,4,4,DCC,4)
      IF (.NOT.SIGFLG) GO TO 510
      DO 511 I=1,4
      DO 511 J=1,4
  511 CC(I,J) = CC(I,J+4)
      GO TO 101
  510 DO 506 I=1,4
      DO 506 J=1,4
      SPII = 0.
      SPJJ = 0.
      SPIJ = 0.
      IF(KA.NE.0.AND.KA.NE.3) GO TO  509
      SPII = VZ(I,6)*VZ(I,6)*SIGPSQ
      SPJJ = VZ(J,6)*VZ(J,6)*SIGPSQ
      SPIJ = VZ(I,6)*VZ(J,6)*SIGPSQ
  509 CONTINUE
      SII = CC(I,I+4) + SPII
      IF (J.EQ.I) GO TO 507
      SIJ = CC(I,J+4) + SPIJ
      SJJ = CC(J,J+4) + SPJJ
      YY = 2.*SIJ
      XX = SII - SJJ
      TIJ = 0.
      IF (YY.EQ.0..AND.XX.EQ.0.) GO TO 508
      TIJ = ATAN2( YY,XX ) / 2.
  508 CC(I,J) = TIJ
      GO TO 506
  507 CC(I,I) = SQRT(SII)
  506 CONTINUE
      GO TO 101

C                          MAGNET —— ENTRANCE
  200 LH = L/2.
      BR = P(3)
      IF (NGO.EQ.2)  BR = BR*(1.+VW(6))
      IF (BZ.EQ.0.)  GO TO 262
      G = BZ/BR
      RHO = 1./G
      KX = P(2) + OM*G
      KK = SQRT(ABS(KX/P(3)))
      PHI = KK*L
      N = - P(2) * P(3) / (OM*OM)
      G2 = G*G
      G3 = G*G*G
      AG3 = ABS(G3)
      AG3L = AG3*LH
      DFAC = G3*(1.-2.*N)
```

```
      EL2G = DFAC*L*L*L*G/24.
      Q10= DFAC*LH
      Q3 = 0.
      THET = L * ABS(G)

*                      E0,E1 ARE ENTRANCE EXIT ANGLES
      ANG0=P(5)
      ANG1=P(6)
      IF ( (IANG0.EQ.1H$).OR.(IANG1.EQ.1H$) )  GO TO 201
      E0 = P(5)*RADEG
      E1 = P(6)*RADEG
      GO TO 203
  201 E0 = .5*THET
      E1=E0
  203 CONTINUE
      TE0 = TAN(E0)  $  TE1 = TAN(E1)
      GAM =(1. + A*A)/ B

      IF (E0.EQ.0.) GO TO 202
      F = F + E*TE0/RHO
      A = A - B*TE0/RHO
      GAM = (1. + A*A)/B
  202 K3R = 1. / (KK*KK*KK*RHO)

      IF (KX) 230,210,220

*                      COMPUTE V.  DO U AT END OF MAGNET

*                      FOR KX = 0
  210 CONTINUE
      RHO2 = RHO*RHO
      XL2=L*L
      XL3=L*XL2
      XL4=L*XL3
      XL5=L*XL4
      V = ( GAM*E*E + 2.*A*E*F + B*F*F ) * L
     1   + ( A*E + B*F ) * XL2 / RHO
     2   + ( B/RHO - GAM*E - A*F ) * XL3 / ( 3.*RHO )
     4   -   A * XL4 / ( 4.*RHO2 )  +  GAM * XL5 / ( 20.*RHO2 )
      GO TO 250

*                      FOR KX.GT.0,  F MAGNET
  220 CONTINUE
      T = TAN(.5*PHI)
      CS = COS(PHI)
      S = SIN(PHI)
      SC = S*CS
      VC1 = 1. - CS
      VC2 = PHI - S
      VC3 = 2.*K3R
      VC4 = 1./(KK*B)
      V1 = L* (GAM*E*E + 2.*A*E*F + B*F*F)
      V2 = VC3*E* (KK*A*VC1 - GAM*VC2)
      V3 = VC3*F* (KK*B*VC1 - A*VC2)
      VC5 = .5*(PHI-SC) -2.*A*VC4*(VC1 - .5*S*S)
      VC6 = 1.5*PHI - 2.*S + .5*SC
      V4 = (B*K3R / RHO)* (VC5 + GAM*VC4*VC6/KK)
```

```
      V = V1 + V2 + V3 + V4
      GO TO 250

*                    FOR KX.LT.0    D MAGNET
  230 CONTINUE
      EX = EXP(PHI)
      EXM = EXP(-PHI)
      SH = .5*(EX - EXM)
      CH = .5 * (EX + EXM)
      SC = SH*CH
      VC1 = 1. - CH
      VC2 = PHI - SH
      VC3 = 2.*K3R
      VC4 = 1./(KK*B)
      VC5 = PHI - SC
      VC6 = VC1 + .5*SH*SH
      VC7 = 1.5*PHI - 2.*SH + .5*SC
      V1 = L* (GAM*E*E + 2.*A*E*F + B*F*F)
      V2 = VC3*E*(KK*A*VC1 - GAM*VC2)
      V3 = VC3*F* (KK*B*VC1 - A*VC2)
      V4 = (B*K3R/RHO) *(.5*VC5 + 2.*A*VC4*VC6 - GAM*VC4*VC7/KK)
      V = V1 - V2 - V3 - V4
      GO TO 250

C                     QUADRUPOLE
  262 K = DB/BR
      EL2G = 0.
      KSQ = K*K
      XK2L = KSQ*LH
      AK3L = ABS(K*KSQ*LH)
      TKSQL = 2.*KSQ*LH
      IF (.NOT.MOVSW)  GO TO 265
      X = HMOV(1)
      Y = HMOV(2)
      GO TO 266

  250 GO TO (251,260)  NGO
  260 IF (BZ.NE.0.)  GO TO 264
      IF (.NOT.MOVSW)  GO TO 265
      X = HMOV(3)
      Y = HMOV(4)
      GO TO 266
  265 X = VW(1)
      Y = VW(3)
  266 RSQ = X*X + Y*Y
      RQ = SQRT(RSQ)
      G2L = RSQ*XK2L
      AG3L = RQ*RSQ*AK3L
      I2 = I2 + G2L
      I3 = I3 + AG3L
      Q1 = TKSQL*X
      Q3 = TKSQL*Y
      GO TO 263
  264 Q1 = Q10 - TE0*G2
      IF (BSW)  Q1 = Q10 - TE1*G2
  263 I41 = I41 - Q1*AIMAG(EJ1) - Q3*AIMAG(EJ3) - EL2G
      I43 = I43 - Q1*AIMAG(EK1) - Q3*AIMAG(EK3)
```

```
      DO 261 I=1,4
      DO 261 J=1,4
      I51 = I51 + AA(I,J)*VZ(I,6)*VZ(J,6)*AG3L
  261 I53=I53 + BB(I,J)*VZ(I,6)*VZ(J,6)*AG3L
      IF (BSW)  GO TO 302

  251 CONTINUE
      BSW = .TRUE.
*     SAVE E TO COMPUTE U AT END OF MAGNET
      ESAV = E
      GO TO 120

*                    BENDING MAGNET EXIT.  CALCULATE U AND DO SUMMING.
  300 CONTINUE
      GO TO (301, 260 ) NGO
  302 IF (OM.EQ.0.)  GO TO 550
  301 IF (KX) 330,310,320
  310 U = (ESAV + E ) * .5*L - (L*L*L)/(12.*RHO)
      GO TO 340
  320 T = TAN(.5*PHI)
      U = (ESAV + E) *T/KK + K3R*(PHI - 2.*T)
      GO TO 340
  330 T = TANH(.5*PHI)
      U=(ESAV + E)*T/KK - K3R* (PHI - 2.*T)
  340 CONTINUE
      I1 = I1 + G*U
      I2 = I2 + G*G*L
      I3 = I3 + AG3*L
      TR = (ESAV*TAN(E0) + E*TAN(E1))/(RHO*RHO)
      I4 = I4 + DFAC*U - TR
      I5 = I5 + AG3*V
  550 CONTINUE
      MOVSW = .FALSE.
      BSW = .FALSE.
*     GO BACK TO FIND OUT IF THIS ALSO BEGINS A BENDING MAGNET.
      GO TO 111

************************************************************************

*                    RF OUTPUT
  400 CONTINUE
      CALL RANSET(NRN)
      IF (NGO.EQ.1) GO TO 561
      IF (FLAG) GO TO 563
      PC = PC*(1.+VW(6))
      FLAG =.TRUE.
  561 CONTINUE
      I1 = I1 * KB
      I2 = I2 * KB
      I3 = I3 * KB
      I4 = I4 * KB
      I5 = I5 * KB
      GO TO (552,553) NGO
  553 I51 = I51 * KB
      I53 = I53 * KB
      I41 = I41*KB
      I43 = I43*KB
```

```
      D1 = I41/I2
      D3 = I43/I2
      J1 = 1.0 - D1
      J3 = 1.0 - D3
      J6 = 2.0 + D1 + D3
  552 CONTINUE
      PI2 = 2.*PI
      LMTOT = PI2*RHO
      EL = PI2*R
      PC4 = PC**4
      ALPHA = I1/EL
      GAMT = (1./ALPHA)
      GAMT = CSQRT(GAMT)
      U0 = CGAM*PC4*I2/PI2
      UM = U0*1.E03
      D = I4 /I2
      JX = 1.0 - D
      JE = 2.0 + D
      T0 = EL/C
      TAU0 = 4.*PI *T0 /(CGAM*PC**3)
      TAUY = TAU0/I2
      TAUX = TAU0 / (I2 - I4)
      TAUE = TAU0 /(2.*I2 + I4)
      SQSIGE= ABS(CQP*PC4*I3/(JE*I2))
      SIGE = SQRT(SQSIGE)
      FK = FREQ*T0
      IF (V0.LT.U0)  V0 = U0/.86603
      CPHIS = U0/V0
      PHIS = ACOS(CPHIS)
      TE = SQRT(PI2*PC*T0*T0 / (ALPHA*FK*V0*SIN(PHIS)))
      SQE = 2.*U0*(TAN(PHIS) - PHIS)/(PI*FK*ALPHA*PC)
      ASQE = ABS(SQE)
      EPSM = PC*SQRT(ASQE)
      PSI = (EPSM/SIGE)**2/2.
      IF (PSI.LE.740.3) GO TO 556
      PSITRU=PSI
      PSI=740.3
      WRITE (3,1) PSITRU,PSI
  556 CONTINUE
      TAUQ =(TAUE/2.)*EXP(PSI) /PSI
      SIGPHI = SIGE*SQRT (PI2*FK*ALPHA / (V0*SIN(PHIS)*PC))
      SIGL = SIGPHI*R/FK
      PHISD = PHIS*180./PI

*                     COMPUTE EPSILONS FOR EMITTANCE AND STORE
      BETX(2)=BX0
      IF (JX.GT.0.)  GO TO 574
      SIGXOB = 0.
      WRITE (3,21)
      GO TO 575
  574 CONTINUE
      SIGXOB = SQRT (CQP*PC*PC*I5 / (JX*I2))
  575 CONTINUE
      SIGX = SIGXOB * SQRT(BETX(2))
      EPX0 = SIGX*SIGX/BETX(2)
      IF (FLAG.AND.FKAP.EQ.0.0)  FKAP = 1.E-08
      EPX = (1. - FKAP*FKAP)*EPX0
```

```
      EPY = FKAP*FKAP*EPX0
      GO TO (554,555)NGO
  555 CK = CQP*PC*PC/I2
      EP1 = CK*I51/J1
      EP3 = CK*I53/J3
      EP3S = EP3
      IF(EP3.EQ.0.)EP3S=EPY
      REP1=1./EP1
      REP3=1./EP3S
  554 CONTINUE
      SIGBG = SIGE/RME
      EPLN = SIGBG * SIGL
      BETGAM = PC/RME
      EPL = EPLN/BETGAM
C     SIGY = SQRT(EPY*BETY(2))
      SIGDP = SIGE/PC
      SIGPSQ = SIGDP*SIGDP
C     DPP = SIGDP
      EPXN = EPX * BETGAM
      EPYN = EPY * BETGAM

C    SCALE AND STORE
      EPSX = EPX*1.E06
      EPSY = EPY*1.E06
      EPSL = EPL*1.E06
      SIGP = SIGDP*1.E03
      CALL STDAT(M,5,1,4,EM)
      IF (NGO.EQ.1)  GO TO 562
      IF ((J1.GE.0.).AND.(J3.GE.0.).AND.(J6.GE.0.))  GO TO 562
      WRITE (3,21)
      FLAG = .FALSE.

  562 IF (FLAG) GO TO 53
  563 CONTINUE

      CALL DASH
      WRITE (3,2)  PC,GAMT,R,UM
      WRITE (3,4) FMHZ,T0,V0MV,TAUX
      WRITE (3,6) FKAP,TAUY,TAUE
      WRITE (3,3) D,SIGX,PHISD,TAUQ
      WRITE (3,5) TE,SIGPHI,EPSM,SIGL
      WRITE (3,7) XI,SIGDP
      WRITE (3,11) SIGE,SIGBG
      WRITE (3,10) FK,BETGAM,ALPHA,BX0
      WRITE(3,9)I1,I2,I3,I4,I41,I43,I5,I51,I53
      WRITE (3,12) JX,JZ,JE
      IF (NGO.EQ.2)  WRITE (3,22) J1,J3,J6
      WRITE (3,8) EPXN,EPYN,EPLN,EPX,EPY,EPL,EPX0
      IF (NGO.EQ.2)  WRITE (3,23) EP1,EP3
      CALL DASH

**********************************************************************
*                     FORMATS
 1    FORMAT (1X,*PSI = *E13.5* ——— TAUQ IS GREATER THAN VALUE PRINTED*
     1 * BELOW, WHICH CORRESPONDS TO PSI = *E13.5)
 2    FORMAT (* MOMENTUM*,25X,*P=*,E12.5,* GEV/C*,11X,*TRANSITION *,
     1*GAMMA*,10X,*GAMT=(*,E12.5,*,*,E12.5,*)*/
```

```
          . * AVERAGE RADIUS*,17X,*  R=*,E12.5,
          2 * METERS*,9X,* SYNCHROTRON RADIATION*,7X,*U0=*,E12.5,* MEV/TURN*)
    3      FORMAT (* DAMPING PARTITION*16X*D=*E12.5,16X,
          1  * SIGMAX(0-COUPLING)*,7X,*SIGX0=*,E12.5,* METERS*,
          A/,* PHASE ANGLE*,19X,
          2*PHIS=*,E12.5,* DEGREES*,8X,* QUANTUM LIFETIME*,10X,*TAUQ=*,E12.5,
          3 * SECONDS*)
    4      FORMAT (* R F FREQUENCY*,18X,*  F=*,E12.5,* MHZ*,12X,* CIRCULATION
          1PERIOD*,10X,*T0=*,E12.5,* SECONDS*,/,* R F PEAK VOLTS*,18X,*V0=*,
          2E12.5,* MV*,14X,*HORIZONTAL DAMPING*,8X,*TAUX=*,E12.5,* SECONDS*)
    5      FORMAT (* ENERGY OSCILLATION PERIOD*,7X,*TE=*,E12.5,* SECONDS*,
          1 9X,*PHASE WIDTH*,13X,*SIGPHI=*,E12.5,* RADIANS*/
          2* ENERGY APERTURE*,15X,*EPSM=*,E12.5,* GEV  *,10X,
          3* BUNCH LENGTH*,14X,*SIGL=*,E12.5,* METERS*)
    6      FORMAT (* COUPLING CONSTANT*,16X,*K=*,E12.5,17X,*VERTICAL *
          1 *DAMPING*,10X,*TAUY=*,E12.5,* SECONDS*/
          2  65X,*ENERGY DAMPING*,12X,*TAUE=*,E12.5,* SECONDS*)
    7      FORMAT(* (EPSM*,2H**,*2/SIGE*,2H**,*2)/2*,11X,*  XI=*,E12.5,17X,
          2 *SIGMA(DP/P)*,14X,*SIGDP=*,E12.5)
    9      FORMAT(/* I1=*,E12.5,/,* I2=*,E12.5,/,* I3=*,E12.5,/,* I4=*,E12.5,
          A 14X*I41 = *E12.5,14X,*I43 = *E12.5,
          1 /,* I5=*,E12.5,14X,*I51 = *,E12.5,14X,*I53 = *E12.5 )
    11     FORMAT (* ENERGY SPREAD*,17X,*SIGE=*,E12.5,* GEV*,13X,*MOMENTUM *,
          1*SPREAD*,8X,*SIGBGAM=*,E12.5)
    12     FORMAT (/* JX = *,F12.5,14X,*JY = *,F12.5,14X,*JE = *,F12.5)
    22     FORMAT (
          1         * J1 = *F12.5,14X,*J3 = *F12.5,14X,*J6 = *F12.5)
    10     FORMAT (* HARMONIC NUMBER*,18X,*H=*,E12.5,17X,*MOMENTUM*,13X,
          1 *BETAGAMMA=*,E12.5/
          2  * MOMENTUM COMPACTION*,10X,*ALPHA=*,E12.5,17X,*BETA FUNCTION*,
          3  11X,*BETAX0=*,E12.5)
    8      FORMAT (/10X,*EMITTANCES (METER RADIANS)*/10X,*EPXN = *,E12.5,
          1 7X,*EPYN = *,E12.5,7X,*EPLN = *,E12.5/10X,* EPX = *,E12.5,7X,
          2 * EPY = *,E12.5,8X,*EPL = *,E12.5,7X,*EPX0 = *,E12.5)
    23     FORMAT (11X,*EP1 = *,E12.5,7X,* EP3 = *,E12.5/)
    13     FORMAT (1X,*CYEM*/)
    14     FORMAT(2X*POS*11X*A11*9X*A12*9X*A22*9X*A13*9X*A14*9X*A23*9X*A24*
          1                9X*A33*9X*A34*9X*A44*/
          2 16X*B11*9X*B12*9X*B22*9X*B13*9X*B14*9X*B23*9X*B24*9X*B33*9X*B34*
          3 9X*B44*/ )
    15     FORMAT(/1X,I4,1X,A5,1X,10F12.7/ 12X,10F12.7)
    16     FORMAT (1X*BEAM ELLIPSE MATRIX --- SIJ = SIGMA(I,J),*
          1*  RII = SQRT(SII),  TIJ = ATAN( 2.SIJ / (SII-SJJ) ) / 2 * /
          2                          2X*POS*11X*R11*9X*T12*9X*R22*9X*T13*
          3        9X*T14*9X*T23*9X*T24*9X*R33*9X*T34*9X*R44*/ )
    18     FORMAT (1X*BEAM ELLIPSE MATRIX --- SIJ = SIGMA(I,J) * /
          1                          2X*POS*11X*S11*9X*S12*9X*S22*9X*S13*
          2        9X*S14*9X*S23*9X*S24*9X*S33*9X*S34*9X*S44*/ )
    17     FORMAT (1X,I4,1X,A5,1X,10E12.4)
    19     FORMAT(1X*  (BETATRON PART OF SIGMA MATRIX) * )
    20     FORMAT(1X*  (TOTAL SIGMA MATRIX) * )
    21     FORMAT (1X,*NEGATIVE DAMPING CONSTANT*)

           RETURN
           END
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
115	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
223	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
436	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
471	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
509	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
572	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

```
      SUBROUTINE CYX(PROD)                                              CYX        2
                                                                        CYX        3
C  THIS ROUTINE...                                                      CYX        4
C  1. COMPUTES K CYCLED PRODUCTS OF A MATRIX PRODUCT, PROD, WHOSE       CYX        5
C     FACTORS ARE STORED IN THE ARRAY, MI.                              CYX        6
C  2. COMPUTES AND PRINTS OUT THE BETATRON FUNCTIONS OF THE CYCLED      CYX        7
C     PRODUCTS.                                                         CYX        8
*     ITP -- WRITE TAPE5 OPTION IF GT 0.                                CYX        9
                                                                        CYX       10
C  IF THE ORIGINAL INSTRUCTION WAS A CYB, THEN OPTIONAL OUTPUT MAY BE   CYX       11
C  OBTAINED ON TAPE A5.                                                 CYX       12
                                                                        CYX       13
C  IF KA NEGATIVE, ONLY PRINT POSITIONS OF ELEMENTS WHOSE FIRST         CYX       14
C  CHARACTER IS " (4-8 PUNCH)                                           CYX       15
                                                                        CYX       16
C  FOR CYA,CYC WRITE TAPE 5 OPTIONS CONTROLLED BY KA AS FOLLOWS ---     CYX       17
C  ABS(KA) = 11,12 --- BETA FUNCTIONS IN BINARY, AGS SEQUENCE           CYX       18
C  ---- NPOS,POS,NAME,L,LP,LPP,THETA,KV,KVP,KV2,BV,BH,AV,AH,QV,QH,X,XP  CYX       19
C  ABS(KA) = 13,14 --- XEQ,YEQ IN BCD                                   CYX       20
C ABS(KA)=12,14,16 --- SUPPRESS PRINTING                                CYX       21
C ABS(KA)=15,16 ---- POLARIZATION PARAMETERS ON TAPE 11                 CYX       22
C FOR BEST PLOT PARAMETRS (BX, XEQ, BY, YEQ) ON TAPE 12                 CYX       23
C FOR FXPT BX, XCO, BY, YCO ON 12 AND POLARIZATION PARAMETERS ON 11     CYX       24
                                                                        CYX       25
C  IF KB.LT.0, A MATRIX CORRESPONDING TO THE REFLECTION OF THE INPUT    CYX       26
C  BEAM LINE IS ADDED TO THAT BEAM LINE.  THUS ONLY HALF OF A SYMMETRIC CYX       27
C  SUPERPERIOD NEED BE SPECIFIED.  TUNES AND CHROMATICITIES ARE         CYX       28
C  CALCULATED FOR THE COMPLETE SUPERPERIOD.                             CYX       29
                                                                        CYX       30
C  **NOTE** ROUTINE USES ALL 7 SPECIAL 3X3 LOCATIONS, MEND THRU MEND-6. CYX       31
C   AND M7END-5.                                                        CYX       32
                                                                        CYX       33
      INTEGER OP,BDAT,ELNUM                                             CYX       34
                                                                        CYX       35
      COMMON/TSW/TRSW,MCY(20),JM,LOCC                                   NOV3TSW    1
      LOGICAL TRSW                                                      TSW        3
      LOGICAL PFLAG                                                     CYX       37
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      LEVEL 2, XX,Y1,Y2,Y3,Y4                                           BETPTS     2
      COMMON XX(4000),Y1(4000),Y2(4000), Y3(4000),Y4(4000)              RENAM83    1
      COMMON/BPLTCOM/MN,KW,BXX,BYX,NPLT                                 RENAM83    2
      COMMON/CCPOOL/XMIN,XMAX,YMIN,YMAX,CCXMIN,CCXMAX,CCYMIN,CCYMAX     PL6683     1
      COMMON/CCFACT/FACTOR                                              PL6683     2
                                                                        BETPTS     5
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1          LDFLG,FIN                                               CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      COMMON/BMI1/MI1(16000)                                            NV3BMI1    1
      COMMON/NELS1/NELS1                                                BMI1L      2
      LEVEL 2,MI1                                                       BMI1L      3
                                                                        BMI1L      4
```

```
C                                                                          BMI1        3
      DIMENSION MI(1)                                                      CYX        42
      EQUIVALENCE (MI,MI1),(NELS,NELS1)                                    CYX        43
      COMMON/MG/ML       $ LEVEL 2,ML                                      CYXL        1
                                                                           CYXL        2
      COMMON /MATCH3/NOPR,NONU,FLAG,IPR,IPP,MINFLG,MIFLG,MATFLG,GLOBAL     MATCH3      2
      LOGICAL FLAG,NOPR,NONU,MIFLG,MATFLG,GLOBAL                           MATCH3      3
      DIMENSION RX(2,3),RY(2,3),BETX(9),BETY(9),ML(3),GL(18)               CYX        45
      DIMENSION PRM(6)                                                     CYX        46
      DIMENSION TAB(10),T(49),RW(3),VW(7),VP(7)                            CYX        47
      DIMENSION MBX(2),MBY(2),MXQ(2),MYQ(2)                                CYX        48
      DIMENSION MCXX(2),MCYY(2)                                            CYX        49
      DIMENSION BXP(6),BYP(6)                                              CYX        50
      DIMENSION AS(1000),BS(1000),PSX(1000),PSY(1000),UU(3),LS(3)          CYXMAY85    1
      DIMENSION ABX(2,2),ABXI(2,2),B(2),S(2),ISFD(2),XI(2),MSFD(2)         NOV3CYX     1
      DIMENSION SVAL(2)                                                    NOV3CYX     2
      LOGICAL CHFLG                                                        NOV3CYX     3
      EQUIVALENCE (GL,TAB)                                                 CYX        51
      EQUIVALENCE (IX1,XI1),(IX2,XI2)                                      CYX        52
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL      DIM         2
                                                                           DIM         3
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER      SWTCH       2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                             SWTCH       3
                                                                           SWTCH       4
      COMMON/KINET/KNFLAG,TK,P,GAM,BETT,BETGAM,BRHO                        KINET       2
      LOGICAL KNFLAG                                                       KINET       3
      COMPLEX TGAM,TGAM2                                                   CYX        56
      COMPLEX II,ZG,ZCO,ZP                                                 CYXMPL      1
      REAL KV                                                              CYX        57
      INTEGER PROD,PR,HMAGV                                                CYX        58
      LOGICAL MVF                                                          CYX        59
      LOGICAL PLOT,PRINT,FLG,BTABL,SKIP                                    CYX        60
      EQUIVALENCE (GL(4),XNUX),(GL(5),YNUY),(GL(8),CHRX),                  CYX        61
     1            (GL(9),CHRY),(GL(10),ALPHA),(GL(1),BXMX),                CYX        62
     2            (GL(2),BYMX),(GL(3),XQMX),(GL(11),YQMX),                 CYX        63
     3            (GL(12),BXMN),(GL(13),BYMN),(GL(14),XQMN),               CYX        64
     6            (GL(15),YQMN),(GL(16),CIRC),(GL(17),THETH),              CYX        65
     5            (GL(18),THETY),(GL(7),TTGAM)                             CYX        66
      INTEGER HNPOL,HZEQ,HTEQ,HMAX,HMIN                                    CYX        67
      DATA HNPOL,HZEQ,HTEQ/4HNPOL,3HZEQ,3HTEQ/                             CYX        68
      DATA HMAX,HMIN/6HMAXIMA,6HMINIMA/                                    CYX        69
      DATA CC/2.997925E8/                                                  CYX        70
                                                                           CYX        71
      DATA IBLNK/1H /                                                      CYX        72
      DATA MASK/77000000000000000000B/,ZE/0./                              CYX        73
      DATA HMAGV/4HMAGV/                                                   CYX        74
      DATA II/(0.,1.)/                                                     CYXMPL      2
                                                                           CYX        75
      PI2=8.*ATAN(1.)                                                      CYX        76
      RADEG=PI2/360.                                                       CYX        77
      FLG = .TRUE.                                                         CYX        78
      SKIP = .FALSE.                                                       CYX        79
      ABX(1,1)=ABX(1,2)=ABX(2,1)=ABX(2,2)=0.                               NOV3CYX     4
      CHFLG=.F.                                                            NOV3CYX     5
      IF (NOPR.AND..NOT.GLOBAL)  SKIP = .TRUE.                             CYX        80
      PRINT=.TRUE.                                                         CYX        81
      PFLAG = .FALSE.                                                      CYX        82
```

```
              MVF = .FALSE.                                               CYX         83
              BTABL = .FALSE.                                             CYX         84
              IF(NELS.GT.44) PFLAG = .TRUE.                               JAN86CYX     1
              ITP=0                                                       CYX         86
              KS=0                                                        CYXMAY85     2
              ITAPE=0                                                     NOV3CYX      6
              CALL RANGET(NRN)                                            CYX         87
              LNUM=0                                                      CYX         88
                                                                          CYX         89
              IF(ERROR) RETURN                                            CYX         90
        C     USES WORKING STORAGE MEND-1                                 CYX         91
              K = INFF(4,PROD)                                            CYX         92
              IF (K.GE.0) GO TO 1                                         CYX         93
              K=-K                                                        CYX         94
              FLG = .FALSE.                                               CYX         95
         1    KA=K                                                        CYX         96
              KB = INFF(5,PROD)                                           CYX         97
              OP = INFF(1,PROD)                                           CYX         98
              IF (OP.NE.3HCYB) K=NELS                                     CYX         99
              IPP = INFF(6,PROD)                                          CYX        100
        C   ZERO TAB                                                      CYX        101
              DO 27 IC=1,10                                               CYX        102
         27   TAB(IC) = 0.                                                CYX        103
              CHX=0.                                                      CYX        104
              CHY=0.                                                      CYX        105
              IA=0                                                        CYX        106
              EL=0.                                                       CYX        107
              THETH = 0.                                                  CYX        108
              THETV = 0.                                                  CYX        109
              KK=K                                                        CYX        110
              ISK=0                                                       CYX        111
              NK=0                                                        CYX        112
              ETAST = 0.                                                  CYX        113
        C     FKK = KK                                                    CYX        114
              PR = PROD                                                   CYX        115
              NGO = 1                                                     CYX        116
              YCO = DXCO = DYCO = 0.                                      CYX620       1
              XCO = 0.                                                    CYX        117
              NMVAR = HZEQ                                                CYX        118
              IF (KNFLAG) NMVAR = HTEQ                                    CYX        119
              IF (OP.EQ.3HCYB) GO TO 32                                   CYX        120
              IF(OP.EQ.4HFXPT) GO TO 33                                   CYX        121
              CALL DATA(PR,2,3,2,ISFD)                                    NOV3CYX      7
        C SET CHFLG FOR SEXTUPOLE CHROM CORRECTION                         NOV3CYX      8
              IF(ISFD(1).NE.IBLNK.OR.ISFD(2).NE.IBLNK) CHFLG=.T.          NOV3CYX      9
              IF(CHFLG) CALL DATA(PR,1,1,2,XI)                            NOV3CYX     10
              IF (OP.NE.4HCYA ) GO TO 29                                  CYX        122
                                                                          CYX        123
        *                  CYA                                            CYX        124
              IA=1                                                        CYX        125
              NK = PR - K - 1                                             CYX        126
         29   MN = MNAME(PROD)                                            CYX        127
         30   IF (KA.GE.10) ITP=KA-10                                     CYX        128
              IR = PROD                                                   CYX        129
              CALL MMM(PR,KK,MI)                                          CYX        130
              KWP = KK                                                    CYX        131
              IF (KB.GE.0) GO TO 301                                      CYX        132
```

```
            IF (OP.EQ.3HCYA) NK=NK-1                                      CYX       133
            KK = KK+1                                                     CYX       134
            MI(KK)=MEND-6                                                 CYX       135
            CALL REF(MEND-6,PR)                                           CYX       136
            ML(1)=PR                                                      CYX       137
            ML(2)=MEND-6                                                  CYX       138
            CALL MMM(PR,2,ML)                                             CYX       139
       301  CONTINUE                                                      CYX       140
            IF (OP.EQ.3HCYA) PR = PROD - 1                                CYX       141
            IF (OP.EQ.3HCYC)  GO TO 26                                    CYX       142
            IF (OP.EQ.4HBEST) GO TO 28                                    CYX       143
       310  CALL BET(PR,BETX,BETY,1)                                      CYX       144
            VP(1) = BETX(5)                                               CYX       145
            VP(2) = BETX(6)                                               CYX       146
            VP(3) = BETY(5)                                               CYX       147
            VP(4) = BETY(6)                                               CYX       148
            VP(5) = 0.                                                    CYX       149
            VP(6) = 1.                                                    CYX       150
            VP(7) = 0.                                                    CYX       151
        31  KW=KK+1                                                       CYX       152
            MI(KW)=MI(1)                                                  CYX       153
            IF(ITP.EQ.1) PRINT = .FALSE.                                  CYXMAY85    3
            IF(ITP.EQ.2.OR.ITP.EQ.4.OR.ITP.EQ.6) PRINT=.FALSE.            CYX       154
      *                    SET UP PLOT OPTION                             CYX       155
            PLOT = .FALSE.                                                CYX       156
            IF(OP.NE.4HBEST.AND.OP.NE.4HFXPT) GO TO 39                    CYXNOV28    1
            PLOT = .TRUE.                                                 CYX       158
            WRITE(12) MN,KW                                               CYX583      1
            GO TO 39                                                      CYX       160
      *                    CYB                                            CYX       161
        32  ITP=IDAT(PROD,1)                                              CYX       162
            MN=BDAT(PROD,1)                                               CYX       163
            PR = ELNUM(MN)                                                CYX       164
            IR = PR                                                       CYX       165
            GO TO 310                                                     CYX       166
      C            CYC                                                    NOV3CYX    11
        26  IF (KA.NE.21)  GO TO 28                                       CYX       167
            LQ3 = INFF(24,PR)                                             CYX       168
            LINE = LQ3                                                    CYX       169
            BTABL = .TRUE.                                                CYX       170
      C            BEST                                                   NOV3CYX    12
        28  CALL RXY(PR,RX,RY,RW)                                         CYX       171
            PR=MEND-1                                                     CYX       172
            CALL STXY(PR,RX,RY,RW)                                        CYX       173
            GO TO 310                                                     CYX       174
      *                    FXPT                                           CYX       175
        33  ISK = 1                                                       CYX       176
            NGO = 2                                                       CYX       177
            KAF=INFF(4,PR)                                                CYX       178
            IF (KAF.LT.0)  FLG = .FALSE.                                  CYX       179
            KAF = IABS(KAF)                                               CYX       180
            CALL DATA(PR,7,1,7,VW)                                        CYX       181
            IRAYS = IDAT(PR,4)                                            CYX       182
            ITAPE=IDAT(PR,5)                                              CYX683      1
            FAC = 1.                                                      CYX       183
            IF (IRAYS.EQ.0.OR.IRAYS.EQ.2)  FAC = 1000.                    CYX       184
            X0 = VW(1)                                                    CYX       185
```

```
          XP0 = VW(2)                                                  CYX     186
          Y0=VW(3)                                                     CYX     187
          YP0=VW(4)                                                    CYX     188
          DPP=VW(6)                                                    CYX     189
          CALL DATA(PR,8,1,7,VP)                                       CYX     190
          EX0=VP(1)                                                    CYX     191
          EXP0=VP(2)                                                   CYX     192
          EY0=VP(3)                                                    CYX     193
          EYP0=VP(4)                                                   CYX     194
          MN = MNAME(PROD)                                             CYX     195
          ITP = 0                                                      CYX     196
          IF( (KA.LE.1).OR.(KA.EQ.3) ) PRINT=.FALSE.                   CYX     197
          IR = PROD                                                    CYX     198
          GO TO 31                                                     CYX     199
   39     PSIX = 0.                                                    CYX     200
          PSIY = 0.                                                    CYX     201
          DELX = 0.                                                    CYX     202
          DELY = 0.                                                    CYX     203
          I = 1                                                        CYX     204
          SLENI=0.                                                     CYX     205
          THETI=0.                                                     CYX     206
          CALL RXY(PR,RX,RY,RW)                                        CYX     207
*                         COMPUTE TRACE                                CYX     208
          TRX = RX(1,1) + RX(2,2)                                      CYX     209
          TRY = RY(1,1) + RY(2,2)                                      CYX     210
          IF((ABS(TRX).GE.2.).OR.(ABS(TRY).GE.2.)) GO TO 34            CYX     211
          GO TO 35                                                     CYX     212
*   TURN ON SWITCH TO TERMINATE CALCULATION OF ANGLE FIT               CYX     213
   34     WRITE (3,5000) TRX,TRY                                       CYX     214
          TRSW = .TRUE.                                                CYX     215
   35     CONTINUE                                                     CYX     216
 5000     FORMAT (2X,*TRACE OF X OR Y GE 2.  VALUES SET TO 0.*/        CYX     217
        1  2X,* X TRACE = *,E12.5,5X,* Y TRACE = *,E12.5)              CYX     218
          IF ((ABS(TRX).GE.2.).AND.(ABS(TRY).GE.2.).AND.(KA.GT.0)) GO TO 125 CYX     219
          IF (IPP.GT.0) GO TO 75                                       CYX     220
          IF ((ISK.EQ.0).AND.(MODE.EQ.1).AND..NOT.NOPR)  CALL DASH     CYX     221
          GO TO (40,41) NGO                                            CYX     222
   41     IF (.NOT.PRINT) GO TO 75                                     CYX     223
C       WRITE FXPT HEADING                                             CYX     224
          NAMFX = INFF(2,PROD)                                         CYX     225
          IF (PFLAG)  WRITE(3,1035) NAMFX                              CYX     226
          IF (.NOT.PFLAG)  WRITE (3,1029)  NAMFX                       CYX     227
          WRITE(3,1020)                                                CYX     228
          IF (IRAYS.NE.0.AND.IRAYS.NE.2)  GO TO 75                     CYX     229
          WRITE(3,1022)                                                CYX     230
          GO TO 75                                                     CYX     231
C     WRITE CYX HEADING                                                CYX     232
   40     IF(PFLAG) WRITE(3,1035) MN                                   JUN27CYX  1
          IF(.NOT.PFLAG) WRITE(3,1029) MN                              CYXJUL84  1
          IF(ITP.EQ.0.OR.MOD(ITP,2).EQ.1) WRITE(3,1000)NMVAR           CYXJUL84  2
          IF(ITP.EQ.2) WRITE(3,2002)                                   CYXJUL84  3
 2002     FORMAT(T8,*IPOS  NAME*,T23,*BETY*,T33,*BETX*,T43,*ALPHAY*,   CYXJUL84  4
        1 T53,*ALPHAX*,T63,*NUY*,T73,*NUX*,T83,*XEQ*,T93,*DXEQ*,T103,  CYXJUL84  5
        2 *YEQ*,T113,*DYEQ*)                                           CYXJUL84  6
          GO TO 75                                                     CYX     235
********************************************************************** CYX     236
                                                                       CYX     237
```

```
      *                    ITERATES FROM STATEMENTS 50 TO 90                CYX     238
       50  NI=MI(I)                                                         CYX     239
           NISAVE=NI                                                        CYX     240
           MKN = INFF(20,NI)                                                CYX     241
           GO TO (505,506) NGO                                              CYX     242
       506 NF = M7END - 5                                                   CYX     243
           XCO = VW(1)*FAC                                                  CYX     244
           DXCO = VW(2)*FAC                                                 CYX     245
           YCO = VW(3)*FAC                                                  CYX     246
           DYCO = VW(4)*FAC                                                 CYX     247
           EX=VP(1)                                                         CYX     248
           EXP=VP(2)                                                        CYX     249
           EY=VP(3)                                                         CYX     250
           EYP=VP(4)                                                        CYX     251
           IF(I.GT.1) CALL RANSET(NRR)                                      CYX     252
           CALL REVMAT(NF,NI,VW)                                            CYX     253
           CALL RANGET(NRR)                                                 CYX     254
           NI = NF                                                          CYX     255
      C CORRECT CALCULATION OF TRANSITION GAMMA                             CYX     256
       505 IF (I.EQ.KW)  GO TO 5051                                         CYX     257
           CALL RTRV7(NI,T,RW)                                              CYX     258
           CALL MXV7(T,VP,VP)                                               CYX     259
           EST = -VP(5)                                                     CYX     260
           IF (KNFLAG) EST = EST - (EL/(GAM*GAM))                           CYX     261
       5051 CALL RXY(NI,RX,RY,RW)                                           CYX     262
           IF ( ( KB .GE.0).OR.(I.LT.KW) ) GO TO 507                        CYX     263
           PSIX = 2.*PSIX                                                   CYX     264
           PSIY = 2.*PSIY                                                   CYX     265
           IPOS = I - 1                                                     CYX     266
           MATI = 4HREFL                                                    CYX     267
           GO TO (52,508) NGO                                               CYX     268
       508 XCO = X0*FAC                                                     CYX     269
           DXCO = -XP0*FAC                                                  CYX     270
           YCO = Y0*FAC                                                     CYX     271
           DYCO = -YP0*FAC                                                  CYX     272
           EX=EX0                                                           CYX     273
           EXP=-EXP0                                                        CYX     274
           EY=EY0                                                           CYX     275
           EYP=-EYP0                                                        CYX     276
           GO TO 52                                                         CYX     277
       507 CONTINUE                                                         CYX     278
           PSIX = PSIX + DELX/PI2                                           JUN27CYX  3
           PSIY = PSIY + DELY/PI2                                           JUN27CYX  4
           DENX = BETX(2)*RX(1,1)-BETX(3)*RX(1,2)                           CYX     281
           IF(DENX.EQ.0.) GO TO 502                                         JUN27CYX  5
           DELX = ATAN2(RX(1,2),DENX)                                       JUN27CYX  6
           IF(DELX.LT.0..AND.RW(1).GE.0.) DELX = DELX+PI2                   JUN27CYX  7
                                                                            CYX     292
       502 CONTINUE                                                         CYX     293
           DENY = BETY(2)*RY(1,1)-BETY(3)*RY(1,2)                           CYX     294
           IF(DENY.EQ.0.) GO TO 504                                         JUN27CYX  8
           DELY=ATAN2(RY(1,2),DENY)                                         JUN27CYX  9
           IF(DELY.LT.0..AND.RW(1).GE.0.) DELY = DELY+PI2                   JUN27CYX 10
                                                                            CYX     303
       504 CONTINUE                                                         CYX     304
           IPOS = I-1                                                       CYX     305
           IF (IPOS.NE.0) GO TO 51                                          CYX     306
```

```
          MATI = IBLNK                                                    CYX      307
          BXMX = BETX(2)                                                  CYX      308
          BYMX = BETY(2)                                                  CYX      309
          XQMX = BETX(5)                                                  CYX      310
          YQMX = BETY(5)                                                  CYX      311
          BXMN = BETX(2)                                                  CYX      312
          BYMN = BETY(2)                                                  CYX      313
          XQMN = BETX(5)                                                  CYX      314
          YQMN = BETY(5)                                                  CYX      315
          IF(NGO.EQ.1) GO TO 511                                          CYX      316
          XCMX=XCMN=XCO $ YCMX=YCMN = YCO                                 CYX      317
      511 CONTINUE                                                        CYX      318
          DO 510 LI = 1,2                                                 CYX      319
          MCXX(LI)=MCYY(LI)=I                                             CYX      320
          MBX(LI) = I                                                     CYX      321
          MBY(LI) = I                                                     CYX      322
          MYQ(LI) = I                                                     CYX      323
      510 MXQ(LI) = I                                                     CYX      324
          GO TO 52                                                        CYX      325
       51 MK = MI(IPOS)                                                   CYX      326
          MATI = INFF(2,MK)                                               CYX      327
          MATJ = MASK.AND.MATI                                            CYX983     1
          IF (NGO.EQ.1) GO TO 52                                          CYX      328
          BETX(5)=EX                                                      CYX      329
          BETY(5)=EY                                                      CYX      330
       52 IF(IPOS.EQ.0) GO TO 522                                         NOV3CYX   13
          IF(.NOT.CHFLG)GO TO 522                                         NOV3CYX   14
     C      CALCULATE CHROM CORRECTIONS                                   NOV3CYX   15
          B5=BETX(5)                                                      NOV3CYX   16
          IF (MATI.NE.ISFD(1)) GO TO 521                                  NOV3CYX   17
          ABX(1,1)=ABX(1,1)+B5*BETX(2)                                    NOV3CYX   18
          ABX(2,1)=ABX(2,1)+B5*BETY(2)                                    NOV3CYX   19
      521 IF(MATI.NE.ISFD(2)) GO TO 522                                   NOV3CYX   20
          ABX(1,2)=ABX(1,2)+B5*BETX(2)                                    NOV3CYX   21
          ABX(2,2)=ABX(2,2)+B5*BETY(2)                                    NOV3CYX   22
      522 IF(.NOT.PRINT) GO TO 53                                         NOV3CYX   23
          IF (IPP.GT.0) GO TO 53                                          CYX      332
                                                                          CYX      333
     ******************************************************************** CYX      334
     *                PRINT BETATRON FUNCTIONS                            CYX      335
          IF (.NOT.FLG.AND.MATJ.NE.1L") GO TO 53                          CYX      337
          IF (.NOT.PFLAG)  GO TO (571,58) NGO                             CYX      341
          IF(MOD(LNUM,45).NE.0.OR.LNUM.EQ.0.) GO TO (571,58) NGO          DEC85CYX   1
     C  NEW PAGE.  WRITE HEADING.                                         CYX      343
          GO TO (57,59) NGO                                               CYX      344
       57 IF (.NOT.NOPR)  WRITE(3,1040) NMVAR                             CYX      345
      571 CONTINUE                                                        CYX      346
          IF (.NOT.BTABL)  GO TO 573                                      CYX      347
          STORE(LINE) = PSIX                                              CYX      348
          STORE(LINE+6) = PSIY                                            CYX      349
          STORE(LINE+12) = EL                                             CYX      350
          STORE(LINE+13) = THETH                                          CYX      351
          LST=LINE                                                        CYX      352
          DO 572 ICY=2,6                                                  CYX      353
          LST = LST+1                                                     CYX      354
          STORE(LST) = BETX(ICY)                                          CYX      355
      572 STORE(LST+6) = BETY(ICY)                                        CYX      356
```

```
            LINE = LINE + 14                                                 CYX         357
      573   IF (NOPR) GO TO 574                                              CYX         358
            WRITE (3,1001)  IPOS,MATI,EL,PSIX,PSIY,BETX(2),BETY(2),          CYX         359
     1        BETX(5),BETY(5),ETAST,BETX(3),BETY(3),BETX(6),BETY(6)          CYX         360
            LNUM=LNUM+1                                                      CYX         361
            IF(MOD(LNUM,5).EQ.0.) WRITE (3,1006)                             DEC85CYX      2
      574   ETAST = EST                                                      CYX         362
            GO TO 53                                                         CYX         363
      59     IF (IPOS.EQ.0)  GO TO 58                                        CYX         364
      591   WRITE(3,1030)                                                    NOV3CYX      24
            WRITE(3,1020)                                                    NOV3CYX      25
      592   IF (IRAYS.NE.0.AND.IRAYS.NE.2)  GO TO 58                         MY3CYX        8
            WRITE (3,1022)                                                   CYX         367
      58    WRITE(3,1021) IPOS,MATI,EL,PSIX,PSIY,BETX(2),BETY(2),            CYX         368
     1                BETX(3),BETY(3),EX,EXP,EY,EYP,XCO,DXCO,YCO,DYCO        CYX         369
            LNUM=LNUM+1                                                      CYX         370
            IF(MOD(LNUM,5).EQ.0.) WRITE (3,1006)                             DEC85CYX      3
      53    CONTINUE                                                         CYXJN83       1
            IF(ITAPE.LT.1) GO TO 5333                                        CYXNOV28      2
            KV=0.                                                            CYXNOV28      3
            IF (NAM.EQ.3HMAG) KV=-GK                                         CYXNOV28      4
            DKV=0.                                                           CYXNOV28      5
            IF(NAM.EQ.4HSXTP) DKV=-GK2                                       CYXNOV28      6
            WRITE (5) KK,IPOS,MATI,SLENI,ZE,ZE,THETI,KV,DKV,ZE,              CYXNOV28      7
     1            BETY(2),BETX(2),BETY(3),BETX(3),PSIY,PSIX,A,AP             CYXNOV28      8
     2           ,AV,AVP                                                     CYXNOV28      9
            WRITE(5) EX,EXP,EY,EYP,XCO,DXCO,YCO,DYCO                         CYXNOV28     10
      5333  CONTINUE                                                         CYXNOV28     11
            IF (IPOS.EQ.0) GO TO 54                                          CYXJN83       2
            IF (SKIP)  GO TO 54                                              CYX         372
            CALL MAXMIN (BXMX,BXMN,BETX(2),MBX,IPOS)                         CYX         373
            CALL MAXMIN (BYMX,BYMN,BETY(2),MBY,IPOS)                         CYX         374
            CALL MAXMIN(XQMX,XQMN,BETX(5),MXQ,IPOS)                          CYX         375
            CALL MAXMIN(YQMX,YQMN,BETY(5),MYQ,IPOS)                          CYX         376
            IF(NGO.EQ.1) GO TO 54                                            CYX         377
            CALL MAXMIN(XCMX,XCMN,XCO,MCXX,IPOS)                             CYX         378
            CALL MAXMIN(YCMX,YCMN,YCO,MCYY,IPOS)                             CYX         379
            XX(I)=EL $ Y1(I)=BETX(2) $ Y2(I)=XCO $ Y3(I)=BETY(2) $ Y4(I)=YCO CYX         380
      54    ELL = EL                                                         CYX         381
            IF(NGO.EQ.2) GO TO 622                                           CYX         382
                                                                             CYX         383
     *                     STORE FOR PLOT FILE ON OPTION                     CYX         384
            IF (.NOT.PLOT) GO TO 56                                          CYX         385
            XX(I) = EL                                                       CYX         386
            Y1(I) = BETX(2)                                                  CYX         387
            Y2(I) = BETX(5)                                                  CYX         388
            Y3(I) = BETY(2)                                                  CYX         389
            Y4(I) = BETY(5)                                                  CYX         390
            GO TO 55                                                         CYX         391
      56    CONTINUE                                                         CYX         392
            IF (ITP.EQ.0) GO TO 55                                           CYX         393
            IF(ITP.GT.2) GO TO 62                                            CYX         394
            KV=0.                                                            CYX         395
            IF (NAM.EQ.3HMAG) KV=-GK                                         CYX         396
            DKV=0.                                                           CYX         397
            IF(NAM.EQ.4HSXTP) DKV=-GK2                                       CYX         398
            IF(NGO.EQ.2) GO TO 63                                            CYX         399
```

```
      A = BETX(5)                                                       CYX      400
      AP = BETX(6)                                                      CYX      401
      AV = BETY(5)                                                      CYX      402
      AVP = BETY(6)                                                     CYX      403
      GO TO 64                                                          CYX      404
 63   A = VW(1)/DPP                                                     CYX      405
      AP = VW(2)/DPP                                                    CYX      406
 64   CONTINUE                                                          CYX      407
      WRITE (5) KK,IPOS,MATI,SLENI,ZE,ZE,THETI,KV,DKV,ZE,               CYX      408
     1          BETY(2),BETX(2),BETY(3),BETX(3),PSIY,PSIX,A,AP          CYX      409
     2         ,AV,AVP                                                  CYX      410
      IF(ITP.EQ.2) GO TO 55                                             CYXMAY85   4
      IF(.NOT.FLG.AND.MATJ.NE.1L".AND.IPOS.NE.KK) GO TO 55              CYXJUL84   7
      WRITE(3,2) KK,IPOS,MATI,BETY(2),BETX(2),BETY(3),BETX(3),          CYXJUL84   8
     1 PSIY,PSIX,A,AP,AV,AVP                                            CYXJUL84   9
 2    FORMAT(1X2I5,1XA5,10F10.4)                                        CYXJUL84  10
      GO TO 55                                                          CYX      415
 62   IF(ITP.GE.5) GO TO 622                                            CYX      416
      IF(I.EQ.KW) GO TO 55                                              CYX      417
      WRITE (5,1002) MN,IPOS,KK,EL,BETX(5),BETY(5),BETX(7),             CYX      418
     X        BETY(7)                                                   CYX      419
      GO TO 55                                                          CYX      420
C WRITE POLARIZATION PARAMETERS ON TAPE 11                              CYX      421
 622  RHOI = 0.                                                         CYX      422
      GK=0.                                                             CYX      423
      XI1=0.                                                            CYX      424
      XI2=0.                                                            CYX      425
      IF(IPOS.EQ.0) GO TO 3333                                          CYX      426
      NAM=INFF(1,MK)                                                    CYX      427
      IF(NAM.EQ.4HMOVE) GO TO 3335                                      CYX883     4
      IF(NAM.EQ.3HMAG) GO TO 3336                                       CYX      429
      GO TO 3333                                                        CYX      430
 3335 MMK=MDAT(MK,1)                                                    CYX      431
      IF(INFF(1,MMK).NE.4HMAG ) GO TO 3333                              CYX      432
      MK=MMK                                                            CYX      433
 3336 CONTINUE                                                          CYX      434
      CALL DATA(MK,1,1,6,PRM)                                           CYX      435
      RHOI=PRM(4)/PRM(3)                                                CYX      436
      GK=-PRM(2)/PRM(3)                                                 CYX      437
      XI1=PRM(5)    $  XI2=PRM(6)                                       CYX      438
      IF(IX1.EQ.1H$.OR.IX2.EQ.1H$) GO TO 3334                           CYX      439
      XI1=XI1*RADEG                                                     CYX      440
      XI2=XI2*RADEG                                                     CYX      441
      GO TO 3333                                                        CYX      442
 3334 XI1=PRM(1)*RHOI/2.                                                CYX      443
      XI2=XI1                                                           CYX      444
 3333 CONTINUE                                                          CYX      445
      IF(I.EQ.1) WRITE(11) KW                                           CYX      446
      ZZ=0.                                                             CYX      447
      IF (NGO.EQ.2)  ZZ = YCO/FAC                                       CYX      448
      WRITE (11) ELL,PSIY,BETY(2),RHOI,GK,XI1,XI2,ZZ                    CYX      449
*                                COMPUTE LENGTH                         CYX      450
 55   EL = EL + RW(1)                                                   CYX      451
 60   I = I + 1                                                         CYX      452
      IF (I.GT.KW)  GO TO 90                                            CYX      453
      SLENI = RW(1)                                                     CYX      454
      THETI = RW(2)                                                     CYX      455
```

```
      NAM = INFF(1,NI)                                              CYX      456
      IF ((I.EQ.KW).AND.(KB.LT.0))  GO TO 65                        CYX      457
      IF (NAM.NE.HMAGV)  THETH = THETH + RW(2)                      CYX      458
      IF (NAM.EQ.HMAGV)  THETV = THETV + RW(2)                      CYX      459
      GO TO 66                                                      CYX      460
   65  THETH = THETH + THETH                                        CYX      461
      THETV = 2.*THETV                                              CYX      462
   66  CONTINUE                                                     CYX      463
                                                                    CYX      464
C  COMPUTE BETATRON FUNCTIONS FOR THE I-TH CYCLED MATRIX            CYX      465
C  IN CYA CASE, DO SIMILARITY TRANSFORMATION AND CALL BET           JUN27CYX  13
C     IN OTHER CASES, COMPUTE NEW BETA FUNCTIONS DIRECTLY           JUN27CYX  14
      IF(OP.NE.3HCYA) GO TO 7503                                    JUN27CYX  15
      ML(1) = -NI                                                   CYX      466
      ML(2) = PR + IA                                               CYX      467
      ML(3) = NI                                                    CYX      468
      IF (PR.EQ.NK) PR=MEND-1                                       CYX      469
      CALL MMM (PR,3,ML)                                            CYX      470
 7503  CONTINUE                                                     JUN27CYX  16
      DO 555 JJ=1,6                                                 CYX      471
      BXP(JJ)=BETX(JJ)                                              CYX      472
  555 BYP(JJ)=BETY(JJ)                                              CYX      473
      AXS=BETX(3)                                                   CYX      474
      AYS=BETY(3)                                                   CYX      475
      SS=RW(1)                                                      CYX      476
   75 IF(I.EQ.1.OR.OP.EQ.3HCYA) CALL BET(PR,BETX,BETY,1)            JUN27CYX  17
                                                                    CYX      478
C  COMPUTE CHROMATICITY IF ELEMENT IS MAGNET                        CYX      479
      IF(I.EQ.1) GO TO 775                                          CYX      480
      IF (SKIP)  GO TO 775                                          CYX      481
      IF(OP.EQ.3HCYA) GO TO 7501                                    JUN27CYX  18
      IF(BXP(2).EQ.0.) GO TO 7502                                   JUN27CYX  19
      BETX(3)=(1.+2.*RX(1,2)*RX(2,1))*BXP(3)-RX(1,1)*RX(2,1)*BXP(2) JUN27CYX  20
     1-RX(1,2)*RX(2,2)*BXP(4)                                       JUN27CYX  21
       BETX(2)=(RX(1,2)**2+(RX(1,1)*BXP(2)-RX(1,2)*BXP(3))**2)/BXP(2) JUN27CYX  22
      BETX(4)= (1.+BETX(3)**2)/BETX(2)                              JUN27CYX  23
      BETX(5) = RX(1,1)*BXP(5)+RX(1,2)*BXP(6)+RX(1,3)               JUN27CYX  24
      BETX(6)=RX(2,1)*BXP(5)+RX(2,2)*BXP(6)+RX(2,3)                 JUN27CYX  25
 7502  IF(BYP(2).EQ.0.) GO TO 7501                                  JUN27CYX  26
      BETY(3)=(1.+2.*RY(1,2)*RY(2,1))*BYP(3)-RY(1,1)*RY(2,1)*BYP(2) JUN27CYX  27
     1-RY(1,2)*RY(2,2)*BYP(4)                                       JUN27CYX  28
      BETY(2)=(RY(1,2)**2+(RY(1,1)*BYP(2)-RY(1,2)*BYP(3))**2)/BYP(2) JUN27CYX  29
      BETY(4)=(1.+BETY(3)**2)/BETY(2)                               JUN27CYX  30
      BETY(5)=RY(1,1)*BYP(5)+RY(1,2)*BYP(6)+RY(1,3)                 JUN27CYX  31
      BETY(6)= RY(2,1)*BYP(5)+RY(2,2)*BYP(6)+RY(2,3)                JUN27CYX  32
 7501  CONTINUE                                                     JUN27CYX  33
      NI = NISAVE                                                   CYX      482
      NAM=INFF(1,NI)                                                CYX      483
      IF(NAM.EQ.4HSXTP) GO TO 785                                   CYX      484
  760 CONTINUE                                                      CYX      488
      IF (NAM.NE.4HKICK.AND.NAM.NE.4HMOVE) GO TO 765                CYX      489
      NKIK=MDAT(NI,1)                                               CYX      490
      NOP=INFF(1,NKIK)                                              CYX      491
      IF (NOP.NE.3HMAG.AND.NOP.NE.4HMAGV) GO TO 765                 CYX      492
      IF (MKN.EQ.3HMAP)  GO TO 775                                  CYX      493
      CALL DATA(NKIK,1,1,6,PRM)                                     CYX      494
      GK=PRM(2)/PRM(3)                                              CYX      495
```

```
          IF(NOP.EQ.4HMAGV) MVF=.TRUE.                                     CYX       496
          GO TO 770                                                        CYX       497
      765 IF(NAM.NE.4HMAG .AND.NAM.NE.4HMAGV) GO TO 775                    CYX       498
          CALL DATA(NI,1,1,6,PRM)                                          CYX       499
          GK=PRM(2)/PRM(3)                                                 CYX       500
          IF(NAM.EQ.4HMAGV) MVF=.TRUE.                                     CYX       501
      770 CONTINUE                                                         CYX       502
          NZ=NI                                                            CYX       503
          IF (NAM.EQ.4HKICK.OR.NAM.EQ.4HMOVE) NZ=NKIK                      CYX       504
          RI=PRM(4)/PRM(3)                                                 CYX       505
          XI1=PRM(5)                                                       CYX       506
          XI2=PRM(6)                                                       CYX       507
          IF (IX1.EQ.1H$.OR.IX2.EQ.1H$) GO TO 7701                         CYX       508
          TH1=XI1*RADEG                                                    CYX       509
          TH2=XI2*RADEG                                                    CYX       510
          GO TO 7702                                                       CYX       511
     7701 TH1=PRM(1)*RI/2.                                                 CYX       512
          TH2=TH1                                                          CYX       513
     7702 TAN1=TAN(TH1)                                                    CYX       514
          TAN2=TAN(TH2)                                                    CYX       515
          IF(MVF) GO TO 7703                                               CYX       516
          CALL       CHRM(GK,RI,SS,TAN1,TAN2,BETX,BETY,BXP,BYP,CRX,CRY)    CYX       517
          GO TO 786                                                        CYX       518
     7703 MVF = .FALSE.                                                    CYX       519
          CALL CHRM(GK,RI,SS,TAN1,TAN2,BETY,BETX,BYP,BXP,CRY,CRX)          CYX       520
          GO TO 786                                                        CYX       521
      785 CALL DATA(NI,1,1,4,PRM)                                          CYX       522
          SXLEN=PRM(1)                                                     CYX       523
          GK2=PRM(2)/PRM(3)                                                CYX       526
          IF(NAM.NE.4HSXTP) GO TO 5601                                     CYXMAY85    5
    C STORE PARAMETERS FOR NONLINEAR AMPLITUDE CALCULATION                 CYXMAY85    6
          DKV=-GK2                                                         CYXMAY85    7
          KS=KS+1                                                          CYXMAY85    8
          AF=DKV*SQRT(BETX(2))                                             CYXMAY85    9
          IF(SLENI.NE.0.) AF=AF*SLENI                                      CYXMAY85   10
          AS(KS)= AF*BETX(2) $ BS(KS)=AF*BETY(2)                           CYXMAY85   11
          PSX(KS)=PSIX $ PSY(KS)=PSIY                                      CYXMAY85   12
     5601 CONTINUE                                                         CYXMAY85   13
          CRX = 2.*(BETX(3)-AXS)  $  CRY=2.*(BETY(3)-AYS)                  CYXMPL      3
          KAS=INFF(4,NI) $  KBS = INFF(5,NI)                               CYXMPL      4
          IF (KAS.EQ.0) KBS= 0 $ IF(KAS.EQ.0) KAS = 3                      CYXMPL      5
          MP=KAS-3                                                         CYXMPL      6
          IF(MP.LT.0) GO TO 786                                            CYXMPL      7
          ZCO = ((XCO + SXLEN*DXCO/2.) + II*(YCO + SXLEN*DYCO/2.))/1000.   CYXMPL      8
          ZP= BETX(5)-SXLEN*BETX(6)/2. + II*(BETY(5)-SXLEN*BETY(6)/2.)     CYXMPL      9
          BXC = BETX(2) + SXLEN*(BETX(3)+BETX(4)*SXLEN/4.)                 CYXMPL     10
          BYC = BETY(2) + SXLEN*(BETY(3)+BETY(4)*SXLEN/4.)                 CYXMPL     11
          ZG=-2.*ZP*GK2 $ IF(KBS.EQ.1) ZG=II*ZG                            CYXMPL     12
          IF(SXLEN.NE.0.) ZG=ZG*SXLEN                                      CYXMPL     13
          IF(MP.EQ.0) GO TO 7852                                           CYXMPL     14
          DO 7851 MM=1,MP                                                  CYXMPL     15
     7851 ZG=ZG*ZCO/MM                                                     CYXMPL     16
     7852  CRX=   CRX           + REAL(ZG)*BXC                             CYXMPL     17
          CRY = CRY - REAL(ZG)*BYC                                         CYXMPL     18
      786 CHX = CHX + CRX                                                  CYX       530
          CHY = CHY + CRY                                                  CYX       531
      775 CONTINUE                                                         CYX       532
```

```
      C  CHANGE INDEX TO SAVE CYCLED MATRICES IF CYA INSTRUCTION.          CYX       533
            IF (OP.NE.3HCYA.OR.I.EQ.KW) GO TO 50                           CYX       534
            PR=PR-1                                                        CYX       535
            GO TO 50                                                       CYX       536
                                                                           CYX       537
      *                COMPLETES LOOP FROM STATEMENTS 50 TO 90             CYX       538
      ******************************************************************** CYX       539
       90   CONTINUE                                                       CYX       540
            CALL RANSET(NRN)                                               CYX       541
            IF (SKIP)  GO TO 210                                           CYX       542
            IF(.NOT.PLOT) GO TO 91                                         CYX       543
            WRITE(12) (XX(J),Y1(J),Y2(J),Y3(J),Y4(J),J=1,KW)               CYX583      2
             WRITE(12) BXMX, BYMX                                          CYX583      3
            BXX = BXMX                                                     CYX       546
            BYX = BYMX                                                     CYX       547
       91   CONTINUE                                                       CYX       548
            TKB = IABS(KB)                                                 CYX       549
            IF(KB.EQ.0) TKB=1.                                             CYX       550
            XNUX=TKB*PSIX                                                  CYX       551
            YNUY=TKB*PSIY                                                  CYX       552
            CALL RXY(IR,RX,RY,RW)                                          CYX       553
            THETH = THETH * TKB                                            CYX       554
            THETV=THETV*TKB                                                CYX       555
            RLEN=ELL/PI2                                                   CYX       556
            RLEN = RLEN * TKB                                              CYX       557
            CIRC = ELL*TKB                                                 CYX       558
            STOTI = 1./ELL                                                 CYX       559
            ETASTOT = ETAST*STOTI                                          CYX       560
            ALPHA = -VP(5)*STOTI                                           CYX       561
      C    IPP.GT.0 PRINTS ONLY TABULAR VALUES.                            CYX       562
            IF (IPP.GT.0)  GO TO 100                                       CYX       563
            IF (.NOT.NOPR)  WRITE (3,1006)                                 CYX       564
            CHM =-TKB*0.25/PI2                                             CYX       565
            IF (KB.LT.0)  CHM = 2.*CHM                                     CYX       566
            CHRX = CHM*CHX                                                 CYX       567
            CHRY = CHM*CHY                                                 CYX       568
            IF(.NOT.CHFLG) GO TO 100                                       NOV3CYX    26
      C          CALCULATE CHROM CORRECTIONS                               NOV3CYX    27
            DT=ABX(1,1)*ABX(2,2)-ABX(1,2)*ABX(2,1)                         NOV3CYX    28
             ABXI(1,1)=ABX(2,2)/DT $ ABXI(1,2)=-ABX(1,2)/DT                NOV3CYX    29
             ABXI(2,2)=ABX(1,1)/DT $ ABXI(2,1)=-ABX(2,1)/DT                NOV3CYX    30
            SPN=IABS(KB)                                                   NOV3CYX    31
            IF (KB.EQ.0)SPN=1.                                             NOV3CYX    32
            IF(KB.LT.0) SPN=SPN+SPN                                        NOV3CYX    33
            CON=2.*PI2/SPN                                                 NOV3CYX    34
            B(1)=CON*(XI(1)-CHRX)                                          NOV3CYX    35
            B(2)=CON*(CHRY-XI(2))                                          NOV3CYX    36
            DO 99 I=1,2                                                    NOV3CYX    37
            S(I)=0.                                                        NOV3CYX    38
               DO 99 J=1,2                                                 NOV3CYX    39
       99   S(I)=S(I)+ABXI(I,J)*B(J)                                       NOV3CYX    40
      C                                                                    NOV3CYX    41
      C     APPLY CHROM CORRECTION TO SECOND SEXT PARAMETER                NOV3CYX    42
            DO 95 I=1,2                                                    NOV3CYX    43
            MSFD(I)=ELNUM(ISFD(I))                                         NOV3CYX    44
            MSS=MSFD(I) $ FF=FDAT(MSS,1)                                   APR84CYX    1
            IF (FF.EQ.0.) FF= 1.                                           APR84CYX    2
```

```
          S(I)=S(I)*FDAT(MSS,3)/FF                                     CY4AP       1
          SVAL(I)=FDAT(MSFD(I),2) + S(I)                               NOV3CYX    45
   95     STORE(LOCC)=SVAL(I)                                          NOV3CYX    46
  100      IF (BEND.EQ.0.)  GO TO 101                                  CYX       569
C   COMPUTE TRANSITION GAMMA                                           CYX       570
          IF (ALPHA.EQ.0.)  GO TO 101                                  CYX       571
          TGAM2 = 1./ALPHA                                             CYX       572
          TGAM = CSQRT(TGAM2)                                          CYX       573
          TTGAM = AIMAG(TGAM)                                          CYX       574
          IF (TTGAM.NE.0.) TAB(6) = -TTGAM                             CYX       575
          TTGAM = REAL(TGAM)                                           CYX       576
          IF (TTGAM.NE.0.) TAB(6) = TTGAM                              CYX       577
          GO TO 102                                                    CYX       578
  101     TGAM = 0.                                                    CYX       579
  102     IF (IPP.GT.0)  GO TO 103                                     CYX       580
          IF (NOPR)  GO TO 104                                         CYX       581
          WRITE(3,1008) CIRC,THETH,XNUX,CHRX,RLEN,THETV,YNUY,CHRY      CYX       582
          IF (BEND.NE.0)  WRITE (3,1026) ALPHA,TGAM                    CYX       583
          IF (BEND.EQ.0)  WRITE (3,1025) ALPHA                         CYX       584
  104     IF (.NOT.KNFLAG)  GO TO 105                                  CYX       585
          CCBET = CC * BETT                                            CYX       586
          T0 = 0.                                                      CYX       587
          IF (CCBET.NE.0.)  T0 = ELL/CCBET                             CYX       588
          ETA = ALPHA - (1./(GAM*GAM))                                 CYX       589
          IF (NOPR)  GO TO 210                                         CYX       590
           WRITE (3,1028) ETASTOT,T0,ETA                               CYX       591
  105     IF (NOPR)  GO TO 210                                         CYX       592
          LABEL = HMAX                                                 CYX       593
          WRITE (3,1005)                                               CYX       594
          WRITE (3,1007) LABEL,MBX(1),BXMX,MBY(1),BYMX,MXQ(1),XQMX,MYQ(1),  CYX  595
        1  YQMX                                                        CYX       596
            LABEL = HMIN                                               CYX       597
          WRITE (3,1007) LABEL,MBX(2),BXMN,MBY(2),BYMN,MXQ(2),XQMN,MYQ(2),  CYX  598
        1  YQMN                                                        CYX       599
          IF(NGO.EQ.2) WRITE(3,1051) MCXX(1),XCMX,MCYY(1),YCMX,MCXX(2),XCMN, CYX 600
        1MCYY(2),YCMN                                                  CYX       601
 1051     FORMAT(* MAXIMA*6X*XCO(*I4,*)=*F10.5,5X*YCO(*I4,*)=*F10.5/   CYX       602
        1        * MINIMA*6X*XCO(*I4,*)=*F10.5,5X*YCO(*I4,*)=*F10.5)   CYX       603
          IF(CHFLG) WRITE(3,1045) S(1),S(2),SVAL(1),SVAL(2)            NOV3CYX    47
 1045      FORMAT(//*  SEXTUPOLE CORRECTIONS ------ DKSF = *E15.8,     NOV3CYX    48
        1 *    DKSD = *E15.8,*    KSF = *E15.8,*    KSD = *E15.8)      NOV3CYX    49
          IF (ISK.EQ.0)  CALL DASH                                     CYX       604
          IF(KS.EQ.0) GO TO 210                                        CYXMAY85   14
          IF(KB.GE.0) GO TO 410                                        CYXMAY85   15
          DO 411 I=1,KS                                                CYXMAY85   16
             KT=KS+KS+1-I                                              CYXMAY85   17
             AS(KT)=AS(I) $ BS(KT)=BS(I)                               CYXMAY85   18
             PSX(KT)=PSIX-PSX(I) $ PSY(KT)=PSIY-PSY(I)                 JAN86CYX    2
  411      CONTINUE                                                    CYXMAY85   20
          KS=KS+KS                                                     CYXMAY85   21
  410      PI=PI2/2.                                                   CYXMAY85   22
          PX=PI*PSIX $ PY=PI*PSIY                                      CYXMAY85   23
          PX3=3.*PX $ PCP=PY+PY+PX $ PCM=PY+PY-PX                      CYXMAY85   24
          SX1=SIN(PX) $ SX3=SIN(PX3) $ SP=SIN(PCP) $ SM=SIN(PCM)       CYXMAY85   25
          H11=H12=H22=0.                                               CYXMAY85   26
          DO 402 I=1,KS                                                CYXMAY85   27
             DO 402 J=1,KS                                             CYXMAY85   28
```

```
            PHX=PX-PI2*ABS(PSX(I)-PSX(J))                              CYXMAY85   29
            PHY=PY-PI2*ABS(PSY(I)-PSY(J))                              CYXMAY85   30
            C1=COS(PHX)/SX1 $ C3=COS(3.*PHX)/SX3                       CYXMAY85   31
            CP=COS(PHY+PHY+PHX)/SP $ CM=COS(PHY+PHY-PHX)/SM            CYXMAY85   32
            AA=AS(I)*AS(J) $ AB=AS(I)*BS(J) $ BB=BS(I)*BS(J)           CYXMAY85   33
            H11=H11+AA*(C3+3.*C1)                                      CYXMAY85   34
            H12=H12+BB*(CP+CM)-2.*AB*C1                                CYXMAY85   35
  402 H22=H22+BB*(CP-CM+4.*C1)                                         CYXMAY85   36
      UU(1)=TKB*H11/PI2/64.                                            CYXMAY85   37
      UU(2)=TKB*H12/PI2/32.                                            CYXMAY85   38
      UU(3)=TKB*H22/PI2/64.                                            CYXMAY85   39
      DO 420 I=1,3                                                     CYXMAY85   40
         LS(I)=2H - $ IF(UU(I).LE.0.)LS(I)=2H +                        CYXMAY85   41
  420  UU(I)=ABS(UU(I))                                                CYXMAY85   42
      WRITE(3,404) XNUX,LS(1),UU(1),LS(2),UU(2),YNUY,LS(2),UU(2),LS(3) CYXMAY85   43
     1 ,UU(3)                                                          CYXMAY85   44
  404 FORMAT(* AMPLITUDE DEPENDENCE OF TUNES DUE TO SEXTUPOLES*/       CYXMAY85   45
     1* NU-X = *F10.6,A2,E9.3,*EX*A2,E9.3,*EY*/                        CYXMAY85   46
     2* NU-Y = *F10.6,A2,E9.3,*EX*A2,E9.3,*EY*)                        CYXMAY85   47
      GO TO 210                                                        CYX       605
C  PRINT TAB VALUES                                                    CYX       606
  103  IF (NOPR)  GO TO 210                                            CYX       607
      WRITE (3,1112) TAB(1),TAB(3),TAB(4),TAB(2),TAB(5),TGAM           CYX       608
  210 KNFLAG = .FALSE.                                                 CYX       609
      IF (.NOT.BTABL) GO TO 216                                        CYX       610
      LINE = LINE - 1                                                  CYX       611
      DO 215 ICY = 1,18                                                CYX       612
  215 STORE(LINE+ICY) = GL(ICY)                                        CYX       613
  216 IF (NGO.EQ.2)  RETURN                                            CYX       614
      CALL STDAT(PROD,7,1,10,TAB)                                      CYX       615
      RETURN                                                           CYX       616
C   ERROR EXIT                                                         CYX       617
  125   CONTINUE                                                       CYX       618
      IF (IPP.GT.0)  GO TO 210                                         CYX       619
      WRITE (3,1009) MN,TRX,TRY                                        CYX       620
*   COMPUTE AND PRINT LENGTH                                           CYX       621
      KE = KW - 1                                                      CYX       622
      EL = 0.                                                          CYX       623
      DO 127 IK=1,KE                                                   CYX       624
      M1 = MI(IK)                                                      CYX       625
      CALL DATA(M1,5,1,1,FL)                                           CYX       626
      EL = EL + FL                                                     CYX       627
  127 CONTINUE                                                         CYX       628
      WRITE (3,1010) EL                                                CYX       629
      GO TO 210                                                        CYX       630
                                                                       CYX       631
********************************************************************** CYX       632
*                  FORMATS                                             CYX       633
 1009 FORMAT (//1H ,4X,43H ***** BETATRON FUNCTIONS UNSTABLE THROUGH , CYX       634
     1  A5/5X,*X TRACE = *,E12.5,10X,*Y TRACE = *,E12.5)               CYX       635
 1010  FORMAT (10X,*TOTAL BEAM LENGTH = *,F10.5)                       CYX       636
 1000 FORMAT (                                                         CYX       637
     1  2X,*POS*,10X,*S(M) *,4X,*NUX*,5X,*NUY*,6X,*BETAX(M)*,4X,       CYX       638
     2  *BETAY(M)*,5X,*XEQ(M)*,5X,*YEQ(M)*,3X,A4,*(M)*,3X,*ALPHAX*,    CYX       639
     3 4X,*ALPHAY*,5X,*DXEQ*,5X,*DYEQ*/1X,131(1H-))                    CYX       640
 1001 FORMAT(1XI4,1XA5,F10.3,2F8.4,2F12.5,2F11.5,3F10.5,2F9.5)         CYXMAY85   48
 1002 FORMAT(A5,2I3,5F14.8)                                            CYX       642
```

```
      1005 FORMAT (2H  )                                                 CYX        643
      1006 FORMAT (1X)                                                   CYX        644
      1007 FORMAT (2X,A10,*---*,1X,*BETX(*,I4,*) =*,F12.5,4X,*BETY(*,I4,  CYX        645
         1 *) =*,F12.5,5X,*XEQ(*,I4,*) =*,F12.5,5X,*YEQ(*,I4,             CYX        646
         2  *) =*,F12.5)                                                 CYX        647
      1008 FORMAT (/                                                     CYX        648
         1    2X,*CIRCUMFERENCE = *F11.4,* M*,9X,*THETX = *,F12.8,        CYXMAY85    49
         2    1X,*RAD*,6X,*NUX = *,F10.5,6X,*DNUX/(DP/P) = *,F10.5/       CYX        650
         3    9X*RADIUS = *F10.4,* M*10X*THETY = *F12.8,                  CYX        651
         4    1X,*RAD*,6X,*NUY = *,F10.5,6X,*DNUY/(DP/P) = *,F10.5)       CYX        652
      1111 FORMAT (15X,*TGAM=(*,F10.5,*,*,F10.5,*)*)                      CYX        653
      1112 FORMAT (5X,*BETATRON FUNCTIONS*,7X,*MAX BETX = *,F10.5,4X,*MAX XEQ CYX     654
         1 = *,F10.5,4X,*NUX = *,F10.5/30X,*MAX BETY = *,F10.5,28X,*NUY = *, CYX      655
         2  F10.5/30X,*TRANSITION GAMMA = (*,F10.5,*,*,F10.5,*)*)        CYX        656
      1900 FORMAT (2X,*CHROMATICITIES FROM QUADRUPOLES EXPLICITLY APPEARING* CYX      657
         1  * IN *A5, * -- DNUX/(DP/P) =*,F10.4,3X,*DNUY/(DP/P) =*,F10.4)  CYX        658
     1020  FORMAT(* POS          S       QX      QY       BX          BY        AX   CYXF   1
         1  AY      EX        EXP       EY       EYP       XCO       DXCO      YCO      D CYXF   2
         2YCO*)                                                          CYXF         3
     1022  FORMAT(*               (M)                      (M)        (M)    CYXF         4
         1         (M)                (M)              (MM)      (MR)    (MM)    CYXF    5
         2(MR)*)                                                         CYXF         6
      1021 FORMAT(1XI4,1XA5,F8.2,2F6.2,2F10.4,2F6.2,8F8.4)               CYXF         7
      1025  FORMAT (*  (DS/S)/(DP/P)= *,F10.7)                           CYX        664
      1026  FORMAT (*  (DS/S)/(DP/P)= *,F10.7,                           CYX        665
         1          14X,*TGAM=(*,F10.5,1H,F10.5,1H))                     CYX        666
      1028  FORMAT (2X,*(DT/T)/(DP/P) = *,F10.5,14X,*T0 = *,E12.5, *SEC*, CYX       667
         1  6X,*ETA =  1/TGAMSQ  -  1/GAMSQ   = *,F10.5)                 CYX        668
      1029   FORMAT (/1X,*BETATRON FUNCTIONS OF *,A5/)                   CYX        669
      1030  FORMAT (1H1)                                                 CYX        670
      1035  FORMAT (1H1,*BETATRON FUNCTIONS OF *,A5)                     CYX        671
      1040 FORMAT (1H1,1X,                                               CYX        672
         1  1X,*POS*,10X,*S(M) *,4X,*NUX*,5X,*NUY*,6X,*BETAX(M)*,4X,     CYX        673
         2 *BETAY(M)*,5X,*XEQ(M)*,5X,*YEQ(M)*,3X,A4,*(M)*,3X,*ALPHAX*,   CYX        674
         3 4X,*ALPHAY*,5X,*DXEQ*,4X,*DYEQ*/1X,131(1H-))                  CYX        675
           END                                                           CYX        679
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
EQV/COMM	I	BMI1	NOT ALL ITEMS IN THIS COMMON BLOCK OCCUR IN LEVEL STATEMENTS.
265	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
290	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
316	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
384	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
385	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
387	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

```
                                                              CYX      680
          SUBROUTINE DASH                                     DASH       2
C  WRITES ONE LINE OF DASHES                                  DASH       3
          WRITE (3,2)                                         DASH       4
 2        FORMAT (1X,131(1H-))                                DASH       5
          RETURN                                              DASH       6
          END                                                 DASH       7
```

```
              SUBROUTINE DATA(M,I,J,K,A)                                  DATA         2
        C  READS K DATA OF TYPE I STARTING AT INDEX J INTO ARRAY A        DATA         3
        *    IF I=1,FLOATING POINT.   I=2, BCD.  I=3, INTEGER             DATA         4
        C                                                                 DATA         5
        C  IF I=4,SKIP.  I=5,PICK UP FROM LQ,I=7 FROM LQ2, I=8 FROM LQ3   DATA         6
        C   IF I=6, PICK UP FROM FL. PT. STORAGE, BUT RETURN NUMBER IN    DATA         7
        *  STORAGE FORM.  DO NOT TEST FOR FL. PT. OR SYMBOLIC.            DATA         8
                                                                          BLANK        2
              LEVEL 2, STORE,INFF,IWORK                                   BLANK        3
              COMMON STORE(48000),IWORK(10)                               86MARSIZ     1
              DIMENSION INFF(24,2000)                                     86MARSIZ     2
              EQUIVALENCE (INFF,STORE)                                    86MARSIZ     3
                                                                          BLANK        5
              COMMON/FLTN/IFL(15)                                         FLTN         2
                                                                          FLTN         3
              DIMENSION A(1),LOC(10)                                      DATA        11
        C                                                                 DATA        12
              IF (I.EQ.4) RETURN                                          DATA        13
              LOC(1)=INFF(14,M)                                           DATA        14
              LOC(2)=INFF(16,M)                                           DATA        15
              LOC(3)=INFF(12,M)                                           DATA        16
              LOC(5) = INFF(10,M)                                         DATA        17
              LOC(6) = INFF(14,M)                                         DATA        18
              LOC(7)=INFF(19,M)                                           DATA        19
              LOC(8)=INFF(24,M)                                           DATA        20
              LOCI=LOC(I)+J-2                                             DATA        21
                                                                          DATA        22
        *         BEGIN LOOP                                              DATA        23
               DO 10 L=1,K                                                DATA        24
              LOCI=LOCI+1                                                 DATA        25
              A(L) = STORE(LOCI)                                          DATA        26
              IF (I.EQ.1.OR.I.EQ.5.OR.I.GE.7)  A(L) = REALNUM(LOCI)       DATA        27
           10 CONTINUE                                                    DATA        28
              RETURN                                                      DATA        29
              END                                                         DATA        30
```

```
      SUBROUTINE DCFD(M,P)                                                DCFD     2
      COMMON/BCFD/PAR(10),RMUX,RMUY,XGES,YGES,LAM                         BCFD     2
C                                                                         BCFD     3
      COMMON/GRR/IERR,BSW,VSW,RSW,QSW,XSW                                 GRR      2
      LOGICAL IERR,BSW,VSW,RSW,QSW,XSW                                    GRR      3
                                                                          GRR      4
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER     SWTCH    2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                            SWTCH    3
                                                                          SWTCH    4
C                                                                         DCFD     6
      DIMENSION P(9)                                                      DCFD     7
      EQUIVALENCE (FM,PAR(2)),(DM,PAR(3))                                 DCFD     8
      REAL LAM                                                            DCFD     9
      EXTERNAL AUXF                                                       DCFD    10
C                                                                         DCFD    11
      DO 1 I=1,6                                                          DCFD    12
    1 PAR(I+2) = P(I)                                                     DCFD    13
      PAR(1) = PAR(3)                                                     DCFD    14
      CYCSWT = .TRUE.                                                     DCFD    15
      PAR(9) = P(7)                                                       DCFD    16
      RMUY = COS(6.283185307 * P(9))                                      DCFD    17
      RMUX = COS(6.283185307 * P(8))                                      DCFD    18
C  INITIAL GUESS FOR LAMBDA = 1/2.                                        DCFD    19
      XGES = 0.5                                                          DCFD    20
      YGES = PAR(4)                                                       DCFD    21
      CALL GRT(1,XGES,7,AUXF)                                             DCFD    22
      IF (IERR) RETURN                                                    DCFD    23
      CALL HED                                                            DCFD    24
      WRITE(3,1000) PAR(1),(PAR(I),I=4,8),FM,DM                           DCFD    25
      CALL HED                                                            DCFD    26
      CYCSWT = .FALSE.                                                    DCFD    27
      CALL CFD(M,FM)                                                      DCFD    28
C  STORE FINAL VALUE OF PROFILE PARAMETER AS DATA FOR Q.                  DCFD    29
      CALL REPFLT(M,2,PAR(4))                                             DCFD    30
      RETURN                                                              DCFD    31
 1000 FORMAT(1X,*NEW VALUES...*,6F14.8/9X,*LF = *,F12.8,9X,               DCFD    32
     1   *LD = *,F12.8)                                                   DCFD    33
      END                                                                 DCFD    34
```

```
            SUBROUTINE DEFSET(M,NM,KC,MXLIST,IOP,KN)                          DEFSET      2
      *     RESERVES SPACE FOR KC NUMBER OF CYCLED MATRICES  WHOSE NAMES      DEFSET      3
      * CONSIST OF THE FIRST LETTER OF NM , FOLLOWED BY NUMBERS8 IN SEQUENCE  DEFSET      4
      *    SET UP SPACE FOR TWO MAGNETS AND OPTIONAL NUMBER OF DRIFTS.        DEFSET      5
      *           IOP = TOTAL NUMBER OF MAGS AND DRIFTS.                      DEFSET      6
      *     KN = KIND FOR INFO                                                DEFSET      7
                                                                              DEFSET      8
            COMMON/BMI/MI(5000)                                               BMIL        1
            COMMON/NELS/NELS                                                  BMIL        2
            LEVEL 2,MI                                                        BMIL        3
                                                                              BMIL        4
                                                                              BMI         3
            COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER   SWTCH       2
            LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                          SWTCH       3
                                                                              SWTCH       4
            DIMENSION LIST (100)                                              DEFSETL     1
            DIMENSION MXLIST(1)                                               DEFSET     13
                                                                              DEFSET     14
            IF (KC.EQ.0) GO TO 536                                            DEFSET     15
            M1=M-1                                                            DEFSET     16
            CALL MOVLEV(MI,LIST,KC)                                           DEFSETL     2
                                                                              DEFSETL     3
            CALL EQU(M1,M)                                                    DEFSET     17
      *                  GENERATE NUMBERED NAMES.                             DEFSET     18
            CALL GENNUM(NM,KC,LIST)                                           DEFSET     19
      *             STORE NAMES AND RESERVE SPACE FOR MATRIX AND LENGTH       DEFSET     20
            DO 535 JA=1,KC                                                    DEFSET     21
            CALL INFW(LIST(JA),2,M1)                                          DEFSET     22
        530 IF (JA.EQ.1) GO TO 531                                            DEFSET     23
            CALL RESRV(M1,KN,MINZER,                                          DEFSET     24
           .  0,0,0,0,0,0,0,0                                                 DEFSET     25
           .)                                                                 DEFSET     26
        531 M1=M1-1                                                           DEFSET     27
        535 CONTINUE                                                          DEFSET     28
      C  IF IOP=0,RESERVE FOR CYCLED MATRICES ONLY                             DEFSET     29
        536 IF (IOP.EQ.0) RETURN                                              DEFSET     30
            M1 = M - KC                                                       DEFSET     31
                                                                              DEFSET     32
      *                        RESERVE FOR 2 MAGNETS                          DEFSET     33
            I1 = 2                                                            DEFSET     34
            IF (IOP.EQ.1) I1 = 1                                              DEFSET     35
            DO 292 I=1,I1                                                     DEFSET     36
            M1 = M1 -    1                                                    DEFSET     37
            CALL MAGRSV (-M1,MXLIST(I),0,0,NM)                                DEFSET     38
        292 CONTINUE                                                          DEFSET     39
            IF (IOP.EQ.I1) RETURN                                             DEFSET     40
                                                                              DEFSET     41
      *                        RESERVE FOR DRIFTS                             DEFSET     42
            I1 = I1 + 1                                                       DEFSET     43
            DO 293 I=I1,IOP                                                   DEFSET     44
            M1 = M1 - 1                                                       DEFSET     45
            CALL DRFRSV (-M1,MXLIST(I),0,0,NM)                                DEFSET     46
        293 CONTINUE                                                          DEFSET     47
            RETURN                                                            DEFSET     48
            END                                                               DEFSET     49
```

```
      SUBROUTINE DELQ(S,W,DW)                                    DELQ       2
                                                                 BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                  BLANK      3
      COMMON STORE(48000),IWORK(10)                              86MARSIZ   1
      DIMENSION INFF(24,2000)                                    86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                   86MARSIZ   3
                                                                 BLANK      5
      COMMON/CDERIV/IFLAG,MR,NR,DP,NV(11),DAT(20),UZ,STOT,       CDERIV     2
     1  DAT2(7),MP2FLG,NF,H0,LOCALPH,NUMALPH                     CDERIV     3
      EQUIVALENCE (L,DAT(1)),(NU,DAT(2)),(EPS,DAT(3)),(Q,DAT(4)) DELQ       5
      EQUIVALENCE (X,T(1)),(Y,T(2)),(Z,T(3)),                    DELQ       6
     1            (GX,G(1)),(GY,G(2)),(GZ,G(3)),(GV,G(4))        DELQ       7
      DIMENSION T(3),G(4)                                        DELQ       8
      DIMENSION W(4,1),DW(4,1)                                   DELQ       9
C                                                                DELQ      10
C   ENVELOPES                                                    DELQ      11
       EPS2=EPS*EPS                                              DELQ      12
      AX=W(1,1) $  AY=W(3,1) $  AX3=AX*AX*AX $  AY3=AY*AY*AY     DELQ      13
      AXY=AX+AY $  EX=EPS2/AX3 $ EY=EPS2/AY3  $  QXY=Q/AXY       DELQ      14
      FX=EX+QXY $  FY=EY+QXY                                     DELQ      15
C    EL. QUAD.                                                   DELQ      16
      X=AX $       Y=AY $        Z=S                             DELQ      17
      CALL ELQ (LOCALPH,NUMALPH,NU,L,T,G)                        DELQ      18
C                                                                DELQ      19
      DW(1,1) = W(2,1)                                           DELQ      20
      DW(3,1) = W(4,1)                                           DELQ      21
      DW(2,1) = FX + GX                                          DELQ      22
      DW(4,1) = FY + GY                                          DELQ      23
C                                                                DELQ      24
      RETURN                                                     DELQ      25
      END                                                        DELQ      26
```

```
      SUBROUTINE DER1(Z,W,DW)                                           DER1       2
C  INTEGRATION THROUGH WIGGLER MAGNET                                   DER1       3
      COMMON/CDERIV/IFLAG,MR,NR,DP,NV(11),DAT(20),UZ,STOT,              CDERIV     2
     1  DAT2(7),MP2FLG,NF,H0,LOCALPH,NUMALPH                            CDERIV     3
      DIMENSION W(4),DW(4),A(3),B(3)                                    DER1       5
      EQUIVALENCE (L,DAT(1)),(LAMBDA,DAT(2)),(BRHO,DAT(3)),             DER1       6
     1            (BV,DAT(4)),(A0,DAT(5))                               DER1       7
      REAL L,LAMBDA,K                                                   DER1       8
      DATA NH/3/                                                        DER1       9
                                                                        DER1      10
      IF (IFLAG) 1,1,2                                                  DER1      11
    1 J = 0                                                             DER1      12
      DO 12 I=2,6,2                                                     DER1      13
      J = J + 1                                                         DER1      14
      A(J) = DAT(5+I)                                                   DER1      15
   12 B(J) = DAT(6+I)                                                   DER1      16
                                                                        DER1      17
      BR=BRHO*(1.+DP)                                                   DER1      18
      BVBR=BV/BR                                                        DER1      19
      K=4.*ACOS(0.)/LAMBDA                                              DER1      20
                                                                        DER1      21
      DW(2) = 0.                                                        DER1      22
    2 PX = W(2)                                                         DER1      23
      PY = W(4)                                                         DER1      24
C     X = W(1)                                                          DER1      25
      Y = W(3)                                                          DER1      26
      DX = DW(1)                                                        DER1      27
C     DPX = DW(2)                                                       DER1      28
      DY = DW(3)                                                        DER1      29
      DPY = DW(4)                                                       DER1      30
      UY=0.                                                             DER1      31
      UZ = A0*Z                                                         DER1      32
                                                                        DER1      33
      DO 3 N=1,NH                                                       DER1      34
      XK=FLOAT(N)*K                                                     DER1      35
      PHIY=XK*Y                                                         DER1      36
      PHIZ=XK*Z                                                         DER1      37
      AB=A(N)*SIN(PHIZ)-B(N)*COS(PHIZ)                                  DER1      38
      E=EXP(PHIY)                                                       DER1      39
      EI=1./E                                                           DER1      40
      CH=(E+EI)/2.                                                      DER1      41
      SH=(E-EI)/2.                                                      DER1      42
      UZ = UZ + AB*CH/XK                                                DER1      43
    3 UY=UY-AB*SH                                                       DER1      44
                                                                        DER1      45
      UZ = UZ*BVBR                                                      DER1      46
      UY=UY*BVBR                                                        DER1      47
                                                                        DER1      48
      PXM = PX - UZ                                                     DER1      49
      HAM=-SQRT(1.-PXM*PXM-PY*PY)                                       DER1      50
                                                                        DER1      51
      DX=-PXM/HAM                                                       DER1      52
      DY=-PY/HAM                                                        DER1      53
      DPY=PXM*UY/HAM                                                    DER1      54
      DW(1) = DX                                                        DER1      55
      DW(3) = DY                                                        DER1      56
      DW(4) = DPY                                                       DER1      57
```

```
                                                            DER1      58
      RETURN                                                DER1      59
      END                                                   DER1      60
```

```
      SUBROUTINE DER2(S,W,DW)
C
C ENVELOPES AND DISPERSIONS OF IMPLODING BUNCH        11/79
C   L0 = TOTAL LENGTH OF SYSTEM
C   ZT = BUNCH HALF LENGTH AT TARGET
C   DELT = (DP/P) HALF WIDTH OF TARGET
C   QT = 4*ZSQ*RP*N/(A*BETASQ*GAMACUBE)    AT TARGET
C   EPS = TRANSVERSE EMITTANCE/PI  --- NOT NORMALIZED
C   E = ETA = DISPERSION FOR BUNCH CENTER
C   E0= ETA = DISPERSION FOR BUNCH END
C
      COMMON/CDERIV/IFLAG,MR,NR,DP,NV(11),DAT(20),UZ,STOT,
     1  DAT2(7),MP2FLG
      DIMENSION W(8,1),DW(8,1)
      EQUIVALENCE (LMAG,DAT(1)),(GRAD,DAT(2)),(BRHO,DAT(3)),
     1            (BZ,  DAT(4))
      EQUIVALENCE (EPS, DAT2(1)),(QT,  DAT2(2)),(ZT,  DAT2(3)),
     1            (DELT,DAT2(4)),(GAM, DAT2(5)),(L0,  DAT2(6))
      REAL L,L0,K,KBAR,LMAG
C
      IF (IFLAG) 1,1,2
    1 K = GRAD/BRHO
      RHOI = BZ/BRHO
      ZTG2 = ZT*GAM*GAM
      QTDT = QT/DELT
      EPS2 = EPS*EPS
      RETURN
C
    2 AX = W(1,1)
      AY = W(3,1)
      E  = W(5,1)
      E0 = W(7,1)
      L = L0 - S - STOT
      IF (L.LT.ZT.OR.L0.LT.0.)  L = ZT
      DEL0 = ZTG2/L
      Q = QTDT*DEL0
      AP = E*DEL0
      AAX = AX + AP
      AX3 = AX*AX*AX
      AY3 = AY*AY*AY
      A = AAX + AY
      QA = Q/A
       QAXA = QA/AX
      EPSAX = EPS2/AX3
      EPSAY = EPS2/AY3
      KBAR = K - QAXA
C
      DW(1,1) = W(2,1)
      DW(3,1) = W(4,1)
      DW(5,1) = W(6,1)
      DW(7,1) = W(8,1)
      DW(2,1) = -KBAR*AX + EPSAX
      DW(4,1) = K*AY + QA + EPSAY
      DW(6,1) = -KBAR*E + RHOI
      DW(8,1) = -K*E0 + RHOI
      RETURN
```

```
      END                                                               DER2      58
```

```
      SUBROUTINE DER3(S,W,DW)
C  INTEGRATE TRANSVERSE AND LONGITUDINAL ENVELOPES.
      COMMON/CDERIV/IFLAG,MR,NR,DP,NV(11),DAT(20),UZ,STOT,
     1  DAT2(7),MP2FLG
      EQUIVALENCE (L,DAT(1)),   (GRAD,DAT(2)),  (BRHO,DAT(3)),
     1            (EPS,DAT(4)),  (Q1,DAT(5)),  (H0,DAT(6)),
     2            (KZ,DAT(8)),   (EPSL,DAT(10)),(G,DAT(11))
      DIMENSION W(7,1),DW(7,1)
      REAL K,KZ,L
C
      IF (IFLAG) 1,1,2
    1 BRHO=BRHO*(1.+ DP)
      K = GRAD/BRHO
      EPS2 = EPS*EPS
      EPSL2 = EPSL*EPSL
      Q1G = Q1*G
      RETURN
C
    2 AX = W(1,1)
      AY = W(3,1)
      AZ = W(5,1)
      AX3 = AX*AX*AX
      AY3 = AY*AY*AY
      AZ3 = AZ*AZ*AZ
      AXYZ = AZ*(AX+AY)
      AZ2 = 2.*AZ*AZ
      EAX3 = EPS2/AX3
      EAY3 = EPS2/AY3
      EAZ3 = EPSL2/AZ3
      QAXYZ = Q1/AXYZ
      QAZ = Q1G/AZ2
C
      DW(1,1) = W(2,1)
      DW(3,1) = W(4,1)
      DW(5,1) = W(6,1)
C
      DW(2,1) = -K*AX + EAX3 + QAXYZ
      DW(4,1) =  K*AY + EAY3 + QAXYZ
      DW(6,1) = KZ*AZ + EAZ3 + QAZ
C
      RETURN
      END
```

```
      SUBROUTINE DER4(S,W,DW)                                           DER4       2
                                                                        DER4       3
      COMMON/CDERIV/IFLAG,MR,NR,DP,NV(11),DAT(20),UZ,STOT,              CDERIV     2
     1  DAT2(7),MP2FLG,NF,H0,LOCALPH,NUMALPH                            CDERIV     3
      EQUIVALENCE (L,DAT(1)),(GRAD,DAT(2)),(BRHO,DAT(3)),               DER4       5
     1            (EPS,DAT(4)),(Q,DAT(5))                               DER4       6
      DIMENSION W(4,1),DW(4,1)                                          DER4       7
      REAL L,K,KX,KY,KBX,KBY                                            DER4       8
                                                                        DER4       9
      IF (IFLAG) 1,1,2                                                  DER4      10
    1 BRHO = BRHO*(1.+DP)                                               DER4      11
      K = GRAD/BRHO                                                     DER4      12
      EPS2 = EPS*EPS                                                    DER4      13
      RETURN                                                            DER4      14
                                                                        DER4      15
C  ENVELOPE EQUATIONS                                                   DER4      16
    2 AX=W(1,1)                                                         DER4      17
      AY=W(3,1)                                                         DER4      18
      AX3=AX*AX*AX                                                      DER4      19
      AY3=AY*AY*AY                                                      DER4      20
      A=AX+AY                                                           DER4      21
      EAX3=EPS2/AX3                                                     DER4      22
      EAY3=EPS2/AY3                                                     DER4      23
      QA=Q/A                                                            DER4      24
                                                                        DER4      25
      DW(1,1) = W(2,1)                                                  DER4      26
      DW(3,1) = W(4,1)                                                  DER4      27
      DW(2,1) =-K*AX + EAX3 + QA                                        DER4      28
      DW(4,1) = K*AY + EAY3 + QA                                        DER4      29
                                                                        DER4      30
      GO TO (7,3,3,5) IFLAG                                             DER4      31
                                                                        DER4      32
C  LINEARIZED ENVELOPE EQUATIONS                                        DER4      33
    3 EAX4=3.*EAX3/AX                                                   DER4      34
      EAY4=3.*EAY3/AY                                                   DER4      35
      QA2=QA/A                                                          DER4      36
      KBX=-K-EAX4-QA2                                                   DER4      37
      KBY= K-EAY4-QA2                                                   DER4      38
                                                                        DER4      39
      DO 4 J=2,NR                                                       DER4      40
      DW(1,J) = W(2,J)                                                  DER4      41
      DW(3,J) = W(4,J)                                                  DER4      42
      DW(2,J) = KBX*W(1,J) - QA2*W(3,J)                                 DER4      43
    4 DW(4,J) = KBY*W(3,J) - QA2*W(1,J)                                 DER4      44
                                                                        DER4      45
    7 RETURN                                                            DER4      46
                                                                        DER4      47
C  SINGLE PARTICLE EQUATIONS                                            DER4      48
    5 QAX=QA/AX                                                         DER4      49
      QAY=QA/AY                                                         DER4      50
      KX=-K+QAX                                                         DER4      51
      KY= K+QAY                                                         DER4      52
                                                                        DER4      53
      DO 6 J=2,NR                                                       DER4      54
      DW(1,J) = W(2,J)                                                  DER4      55
      DW(3,J) = W(4,J)                                                  DER4      56
      DW(2,J) = KX*W(1,J)                                               DER4      57
```

```
    6   DW(4,J) = KY*W(3,J)

60      RETURN
        END
```

```
      SUBROUTINE DER5(S,W,DW)                                           DER5      2
C                                                                       DER5      3
C     DIFFERENTIAL EQUATION OF THE MOTION IN A SEXTUPOLE                DER5      4
C                                                                       DER5      5
      COMMON/CDERIV/IFLAG,MR,NR,DP,NV(11),DAT(20),UZ,STOT,              CDERIV    2
     1  DAT2(7),MP2FLG,NF,H0,LOCALPH,NUMALPH                            CDERIV    3
      DIMENSION W(4,1),DW(4,1)                                          DER5      7
      REAL K                                                            DER5      8
C                                                                       DER5      9
      IF (IFLAG) 1,1,2                                                  DER5     10
    1 BRHO=(1.+DP)*DAT(3)                                               DER5     11
      K=DAT(2)/BRHO                                                     DER5     12
      RETURN                                                            DER5     13
C                                                                       DER5     14
    2 DW(1,1)=W(2,1)                                                    DER5     15
      DW(2,1)=-K*(W(1,1)*W(1,1)-W(3,1)*W(3,1))/2.                       DER5     16
      DW(3,1)=W(4,1)                                                    DER5     17
      DW(4,1)=K*W(1,1)*W(3,1)                                           DER5     18
                                                                        DER5     19
      IF(IFLAG.EQ.1) RETURN                                             DER5     20
                                                                        DER5     21
      DO 10 J=2,5                                                       DER5     22
      DW(1,J)=W(2,J)                                                    DER5     23
      DW(2,J)=-K*(W(1,1)*W(1,J)-W(3,1)*W(3,J))                          DER5     24
      DW(3,J)=W(4,J)                                                    DER5     25
   10 DW(4,J)=K*(W(1,1)*W(3,J)+W(3,1)*W(1,J))                           DER5     26
C                                                                       DER5     27
      RETURN                                                            DER5     28
      END                                                               DER5     29
```

```
      SUBROUTINE DER6(S,W,DW)                                           DER6       2
      RETURN                                                            DER6       3
      END                                                               DER6       4
```

```
      SUBROUTINE DER7(S,W,DW)
      RETURN
      END
```

```
 1             SUBROUTINE DERIV(S,Y,DY)                                      DERIV      2
               DIMENSION Y(1),DY(1)                                          DERIV      3
               COMMON/TRKINT/G,EX,EY,OMSQ,SEND,VMX,VMN,DINT,BEG,DPR,JPR,     TRKINT     2
              1   THET,PLT,NPT,NZ,PMAX,NSIZ,ITITLE(6)                        TRKINT     3
 5             LOGICAL BEG,DPR,PLT                                           TRKINT     4
         C  ENVELOPE EQUATIONS                                               DERIV      5
               DY(1) =Y(2)                                                   DERIV      6
               F1 = G/(Y(1)+Y(3))                                            DERIV      7
               F2 = EX**2/Y(1)**3                                            DERIV      8
10             DY(2) = F1 + F2 - OMSQ*Y(1)                                   DERIV      9
               DY(3) = Y(4)                                                  DERIV     10
               F2 = EY**2/Y(3)**3                                            DERIV     11
               DY(4) = F1 + F2 - OMSQ*Y(3)                                   DERIV     12
               RETURN                                                        DERIV     13
15             END                                                           DERIV     14
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
7	I	Y	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
8	I	Y	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
10	I	DY	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
11	I	DY	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
11	I	Y	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
12	I	Y	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
13	I	DY	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
13	I	Y	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

```
      SUBROUTINE DET( C, X )
C  CALCULATE THE DETERMINANT OF A 3X3 COMPLEX MATRIX.
      COMPLEX C, X
      DIMENSION C(3,3)
      X = C(1,1)*( C(2,2)*C(3,3) - C(2,3)*C(3,2) ) -
     1       C(1,2)*( C(2,1)*C(3,3) - C(2,3)*C(3,1) ) +
     2       C(1,3)*( C(2,1)*C(3,2) - C(2,2)*C(3,1) )
      RETURN
      END
```

```
      SUBROUTINE DIFEQ(NN,V,DERI)                                        DIFEQ      2
                                                                         BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                          BLANK      3
      COMMON STORE(48000),IWORK(10)                                      86MARSIZ   1
      DIMENSION INFF(24,2000)                                            86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                           86MARSIZ   3
                                                                         BLANK      5
      DIMENSION T(8,28)                                                  DIFEQ      4
      DOUBLE PRECISION H,T                                               DIFEQ      5
      LOGICAL SWPR,OUT                                                   DIFEQ      6
      DIMENSION V(7,1),W(28),DW(28)                                      DIFEQ      7
      EXTERNAL DERI                                                      DIFEQ      8
      COMMON/CDERIV/IFLAG,MR,NR,DP,NV(11),DAT(20),UZ,STOT,               CDERIV     2
     1  DAT2(7),MP2FLG,NF,H0,LOCALPH,NUMALPH                            CDERIV     3
      INTEGER HDRF,HELQ                                                  DIFEQ     10
      DATA HDRF,HELQ/3HDRF,3HELQ/                                        DIFEQ     11
                                                                         DIFEQ     12
C  INTEGRATE   FIRST ORDER DIFFERENTIAL EQUATIONS USING ZAM.             DIFEQ     13
C  S=INDEPENDENT VARIABLE, GOES FROM 0 TO SMAX.                          DIFEQ     14
C  H0=MAXIMUM INTEGRATION STEP SIZE (WILL BE ADJUSTED TO BE AN INTGRAL   DIFEQ     15
C  FRACTION OF SMAX).                                                    DIFEQ     16
C  NPR=NUMBER OF INTERVALS H0 BETWEEN PRINTING OF S,W.                   DIFEQ     17
C  W= DEPENDENT VARIABLE VECTOR, DW=DERIVATIVE VECTOR.                   DIFEQ     18
C  DERI IS THE NAME OF THE SUBROUTINE THAT CALCULATES DW(I).             DIFEQ     19
C  ITS FORM IS --- SUBROUTINE DERI(S,W,DW)                               DIFEQ     20
                                                                         DIFEQ     21
C       NDAT = INFF(4,NN)                                                DIFEQ     22
      NPR=INFF(5,NN)                                                     DIFEQ     23
      NF = INFF(15,NN)                                                   DIFEQ     24
                                                                         DIFEQ     25
      MN = MR*NR                                                         DIFEQ     26
      CALL DATA (NN,1,1,NF,DAT)                                          DIFEQ     27
      NAMOP = INFF(1,NN)                                                 DIFEQ     28
      IF (NAMOP.NE.HDRF)  GO TO 1                                        DIFEQ     29
      DAT2(2) = 0.                                                       DIFEQ     30
      DAT2(3) = 1.                                                       DIFEQ     31
      DAT2(4) = 0.                                                       DIFEQ     32
    1 SMAX = DAT(1)                                                      DIFEQ     33
      H0 = DAT(6)                                                        DIFEQ     34
      IF(NAMOP.EQ.HELQ) CALL SETALPH(NN)                                 DIFEQ     35
      IF (MP2FLG.EQ.1) H0 = DAT2(7)                                      DIFEQ     36
       IF (H0.EQ.0.)  H0 = .01                                           DIFEQ     37
                                                                         DIFEQ     38
      DO 2 J=1,NR                                                        DIFEQ     39
       JV=NV(J)                                                          DIFEQ     40
       JM = MR*(J-1)                                                     DIFEQ     41
      DO 2 I=1,MR                                                        DIFEQ     42
    2   W(I+JM) = V(I,JV)                                                DIFEQ     43
                                                                         DIFEQ     44
      JV1=NV(1)                                                          DIFEQ     45
      DP = V(6,JV1)                                                      DIFEQ     46
                                                                         DIFEQ     47
      S = 0.                                                             DIFEQ     48
      ISTEP=0                                                            DIFEQ     49
      IOSTP=0                                                            DIFEQ     50
      NSTP=SMAX/H0+.000001                                               DIFEQ     51
      H=SMAX/FLOAT(NSTP)                                                 DIFEQ     52
```

```
      OUT=.FALSE.
      IF (NPR.GT.0) OUT=.TRUE.

      IFLAG=-IFLAG
      UZ = 0.
      CALL DERI(S,W,DW)
      W(2) = W(2) + UZ
      IFLAG = -IFLAG
      CALL ZAM0(MN,S,DERI,W,DW,T,H)

      IF (.NOT.OUT)  GO TO 3
      PRINT 101,ISTEP,S, (W(I),I=1,MN)

3     CALL ZAM(S,DERI,W,DW,T,SWPR)

       ISTEP=ISTEP+1
      IF (SWPR)  GO TO 4
       GO TO 5

4     IF (.NOT.OUT)  GO TO 5
      IOSTP=IOSTP+1
      IF (IOSTP.LT.NPR)  GO TO 5
      PRINT 101,ISTEP,S, (W(I),I=1,MN)
      IOSTP = 0

5     IF (S.LT.SMAX-.0000000001)  GO TO 3
      W(2) = W(2) - UZ
      IF (.NOT.OUT)  GO TO 7
      PRINT 101,ISTEP,S, (W(I),I=1,MN)

7       DO 6 J=1,NR
       JV=NV(J)
       JM = MR*(J-1)
      DO 6 I=1,MR
6       V(I,JV) = W(I+JM)

      RETURN
101   FORMAT(5X,I5,10F11.6/(10X,10F11.6))
      END
```

```
1                                                                    DIFEQ      92
                    SUBROUTINE DKE(NBSY,V,NBP)                       DKE         2
                    RETURN                                           DKE         3
                    END                                              DKE         4
```

```
      SUBROUTINE DOFIT(M)                                          DOFIT       2
      RETURN                                                       DOFIT       3
      END                                                          DOFIT       4
```

```
      SUBROUTINE DOIT(MSR)                                              DOIT       2
      INTEGER OPNAME,BDAT                                               DOIT       3
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/BMI/MI(5000)                                               BMIL       1
      COMMON/NELS/NELS                                                  BMIL       2
      LEVEL 2,MI                                                        BMIL       3
                                                                        BMIL       4
                                                                        BMI        3
      COMMON/BMI1/MI1(16000)                                            NV3BMI1    1
      COMMON/NELS1/NELS1                                                BMI1L      2
      LEVEL 2,MI1                                                       BMI1L      3
                                                                        BMI1L      4
C                                                                       BMI1       3
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1          LDFLG,FIN                                               CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      COMMON /MATCH3/NOPR,NONU,FLAG,IPR,IPP,MINFLG,MIFLG,MATFLG,GLOBAL  MATCH3     2
      LOGICAL FLAG,NOPR,NONU,MIFLG,MATFLG,GLOBAL                        MATCH3     3
C                                                                       MOVTOG     2
      COMMON/MOVTOG/TOG                                                 MOVTOG     3
      LOGICAL TOG                                                       MOVTOG     4
C                                                                       MOVTOG     5
      EQUIVALENCE(INDEF,IINDEF)                                         DOIT       9
       DIMENSION RW(3),V0(7),T(50),PAR(50),KREM(50),MIM(100)            DOIT      10
      COMMON/MIM/MIM      $  LEVEL 2,MIM                                DOITL      1
                                                                        DOITL      2
      EQUIVALENCE (PAR,KREM)                                            DOIT      11
                                                                        DOIT      12
      COMMON/TSW/TRSW,MCY(20),JM,LOCC                                   NOV3TSW    1
      LOGICAL TRSW                                                      TSW        3
      LOGICAL TABSW                                                     NOV30DOIT  2
                                                                        DOIT      15
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER   SWTCH      2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                          SWTCH      3
                                                                        SWTCH      4
      DIMENSION N(200)                                                  DOIT      17
      DATA N(5),N(51),N(161),N(102),N(104)                              DOIT      18
     1  /1,17,18,19,20/                                                 DOIT      19
      DATA N(47),N(78) /21,22/                                          DOIT      20
      DATA N(28),N(123)/23,24/                                          DOIT      21
      DATA (N(I),I=11,25)/2,3,4,5,6,7,8,9,10,11,12,13,14,15,16/         DOIT      22
      JMT = 0                                                           DOIT      23
      MS = MSR - 1                                                      DOIT      24
                                                                        DOIT      25
  100 OPNAME = INFF(1,MS)                                               DOIT      26
      IF (ERROR) RETURN                                                 DOIT      27
      IOPMS = INFF(3,MS)                                                DOIT      28
C  SCALAR OPERATIONS -- GO TO 76                                        DOIT      29
      IF (IOPMS.GE.112.AND.IOPMS.LE.122) GO TO 76                       DOIT      30
      IF (IOPMS.LT.0) GO TO 200                                         DOIT      31
      NSTAT=N(IOPMS)                                                    DOIT      32
```

```
      IF(NSTAT.EQ.IINDEF) GO TO 201

C      GO TO (END,=,CRD,BML,DRF,MAG,MAGV,EQU,INV,SHF,
C             REF,**,MMM,CYA,CYB,CYC,SUM,TRKB,VEC,MXV,
C             SHF7,MOVE,TAB,CALC)

      GO TO (5,11,200,200,14,15,16,200,18,19,
     1 20,21,22,201,201,23,51,75,200,104,
     2      47,78,810,77)  NSTAT

 5    RETURN


C  ERROR ENCOUNTERED. OPERATION CALLED IS NOT ALLOWED.
 201  WRITE(3,1) OPNAME
      ERROR = .TRUE.
 1    FORMAT (13H ***ERROR*** A5,57H IS AN OPERATION THAT CANNOT BE CALL
     1ED BY SUBROUTINE DOIT)
      RETURN

C  DRIFT ENTRY
 14   KB = INFF(5,MS)
      IF (KB.EQ.0)  GO TO 1106
      FLTOT = 0.
      DO 1104 K=1,KB
      MM = MDAT(MS,K)
 1104 FLTOT = FLTOT + FDAT(MM,1)
      FLSEP = FDAT(MS,2)
      EL = FLSEP - FLTOT
      CALL REPFLT(MS,1,EL)
      GO TO 1107
 1106 EL = FDAT(MS,1)
 1107 CALL DRIFT(MS,EL)
      GO TO 200

C  MAGNET ENTRY
 15   KA = INFF(4,MS)
      IF (KA.GT.0) GO TO 130
      CALL DATA (MS,1,1,6,PAR)
      CALL MAGNET (MS,PAR)
      GO TO 200
 130  CALL MAGMATS(MS,KA)
      GO TO 200

C   MAGV ENTRY
 16   CALL DATA (MS,1,1,6,PAR)
      CALL MAGV(MS,PAR)
      GO TO 200
C
C  MMM ENTRY
 22   NDAT = INFF(17,MS)
      CALL MIFILL (MS,1,NDAT,NELM,MIM)
      KB = INFF(5,MS)
      IF (KB.NE.0.AND.KB.NE.1)  GO TO 142
      CALL MMM(MS,NELM,MIM)
      GO TO 200
 142  CALL EXECMM(MS,NELM,KB,MIM)
```

```
            GO TO 200                                                   DOIT      90
                                                                        DOIT      91
      C   MOVE ENTRY                                                    DOIT      92
       78   DO 781 I=1,7                                                DOIT      93
       781  V0(I) = 0.                                                  DOIT      94
            CALL MOVE(MS,MS,V0)                                         DOIT      95
            TOG = .TRUE.                                                MY3DOIT    2
            GO TO 200                                                   DOIT      96
      C                                                                 DOIT      97
      C  MXV ENTRY                                                      DOIT      98
       104  M1 = MDAT(MS,1)                                             DOIT      99
            M2 = MDAT(MS,2)                                             DOIT     100
            LOC = 1                                                     DOIT     101
            IF (.NOT.VCSW)  GO TO 105                                   DOIT     102
            DECODE (10,106,M2)  NUM                                     DOIT     103
       106  FORMAT (1X,I1,8X)                                           DOIT     104
            LOC = (NUM-1)*7 + 1                                         DOIT     105
       105  CALL DATA (M2,5,LOC,7,PAR(8))                               DOIT     106
      C          = ENTRY                                                DOIT     107
            CALL RTRV7(M1,T,RW)                                         DOIT     108
            CALL MXV7(T,PAR(8),PAR(1))                                  DOIT     109
            CALL STDAT(MS,5,1,7,PAR)                                    DOIT     110
            GO TO 200                                                   DOIT     111
      C  REF ENTRY                                                      DOIT     112
       20    NJ = MDAT(MS,1)                                            DOIT     113
             CALL REF(MS,NJ)                                            DOIT     114
            GO TO 200                                                   DOIT     115
      C  SHF ENTRY                                                      DOIT     116
       19   CALL DATA(MS,1,1,4,PAR)                                     DOIT     117
            CALL SHF(MS,PAR)                                            DOIT     118
            GO TO 200                                                   DOIT     119
      C   SHF7 ENTRY                                                    DOIT     120
       47   CALL DATA (MS,1,1,6,PAR)                                    DOIT     121
            CALL SHF7 (MS,PAR)                                          DOIT     122
            GO TO 200                                                   DOIT     123
      C                                                                 DOIT     124
      C  INV ENTRY                                                      DOIT     125
       18    NJ = MDAT(MS,1)                                            DOIT     126
             CALL INV(MS,NJ)                                            DOIT     127
            GO TO 200                                                   DOIT     128
      C  STAR ENTRY                                                     DOIT     129
       21   KS = INFF(4,MS)                                             DOIT     130
            CALL STAR(MS,KS)                                            DOIT     131
            GO TO 200                                                   DOIT     132
      C  SUM ENTRY                                                      DOIT     133
       51   KS = INFF(4,MS)                                             DOIT     134
            CALL DATA(MS,1,1,KS,PAR)                                    DOIT     135
            CALL SUM(KS,PAR,RES)                                        DOIT     136
            CALL STDAT(MS,5,1,1,RES)                                    DOIT     137
            GO TO 200                                                   DOIT     138
                                                                        DOIT     139
      C  = ENTRY                                                        DOIT     140
       11   KN = INFF(20,MS)                                            DOIT     141
            IF (KN.NE.4HSCAL) GO TO 266                                 DOIT     142
            IOP = BDAT(MS,1)                                            DOIT     143
            IF (IOP.EQ.1H ) GO TO 200                                   DOIT     144
            IQ = 2                                                      DOIT     145
```

```
      ITP = INFF(21,MS)
      IF (ITP.NE.4HSNGL) IQ = 4
      CALL DATA(MS,1,1,IQ,PAR)
      CALL AROP(MS,PAR,IOP)
      GO TO 200
C  SCALAR OPERATIONS
 76   XX = FDAT(MS,1)
      NOPMS = IOPMS - 111
      GO TO (86,87,88,89,90,91,92,93,94,95,96) NOPMS
 86   YY = SIN(XX)      $ GO TO 99
 87   YY = COS(XX)      $ GO TO 99
 88   YY = EXP(XX)      $ GO TO 99
 89   YY = ALOG(XX)     $ GO TO 99
 90   YY = XX*XX        $ GO TO 99
 91   YY = SQRT(XX)     $ GO TO 99
 92   YY = TAN(XX)      $ GO TO 99
 93   YY = ASIN(XX)     $ GO TO 99
 94   YY = ACOS(XX)     $ GO TO 99
 95   YY = ATAN(XX)     $ GO TO 99
 96   YY = ABS(XX)      $ GO TO 99
 99   CALL STDAT(MS,5,1,1,YY)
      GO TO 200

 266  KREM(1)=BDAT(MS,1)
      KREM(2)=BDAT(MS,3)
      IOP = BDAT(MS,2)
      CALL AROP(MS,KREM,IOP)
      GO TO 200

*    CYA AND CYC ENTRIES
 23   CONTINUE
 25   NDAT = INFF(17,MS)
      KA = INFF(4,MS)
      IF (MINFLG.NE.4.OR.MIFLG)   CALL MIFILL(MS,1,NDAT,NELS1,MI1)
      CALL CYX(MS)
C     GO TO 280
      GO TO 200
*   CYB ENTRY
C 24  CALL SETCYB(MS)
C     CALL CYX(MS)
C 280 JMT = JMT + 1
*   IF TRSW IS ON, ANGLE FIT WILL BE REPLACED BY TRACE FIT
C 122 IF (TRSW) RETURN
C     GO TO 200

C  TRKB ENTRY
 75   CALL TRKB(MS)
      GO TO 200

C  TAB ENTRY
 810  TABSW = .TRUE.
      CALL TABULAT(MS,TABSW)
      GO TO 200

C  CALC ENTRY
 77   CALL CALC(MS)
      GO TO 200
```

```
  200 NXTM = INFF(8,MS)
      MS = NXTM
      GO TO 100
      END
```

```
      SUBROUTINE DRFMATS(M,KA)                                          DRFMATS   2
                                                                        DRFMATS   3
*        FORM KA NUMBER OF DRIFT MATRICES USING ELEMENTS OF A VECTOR AS DRFMATS   4
*   LENGTHS.  NAME OF VECTOR REPLACED LENGTH ON DRF INPUT CARD.         DRFMATS   5
                                                                        BLANK     2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK     3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ  1
      DIMENSION INFF(24,2000)                                           86MARSIZ  2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ  3
                                                                        BLANK     5
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL    2
     1          LDFLG,FIN                                               CONTRL    3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL    4
                                                                        CONTRL    5
      EQUIVALENCE(IVAR,FVAR)                                            DRFMATS   8
      INTEGER ELNUM                                                     DRFMATS   9
      DATA MASKL/7777777777B/                                           DRFMATS  10
      DATA MASK1/77777777770000000000B/                                 DRFMATS  11
                                                                        DRFMATS  12
*         RETRIEVE NAME OF VECTOR FROM DRF INPUT.  CHECK THAT IT IS A   DRFMATS  13
*   VECTOR WITH AT LEAST KA ELEMENTS.                                   DRFMATS  14
                                                                        DRFMATS  15
*         VECTOR NAME IS PRESUMED TO BE IN SPECIAL INTERNAL FL. PT.     DRFMATS  16
*  SYMBOLIC FORM --    00000NAMEV                                       DRFMATS  17
      LF = INFF(14,M)                                                   DRFMATS  18
      FVAR=STORE(LF)                                                    DRFMATS  19
      NV=IVAR                                                           DRFMATS  20
      NR = NV.AND.MASKL                                                 DRFMATS  21
      NL = NV.AND..NOT.MASKL                                            DRFMATS  22
      IF (NL.EQ.0.AND.NR.NE.0) GO TO 20                                 DRFMATS  23
                                                                        DRFMATS  24
*                    ERROR.  NOT SYMBOLIC FORM                          DRFMATS  25
      WRITE (3,1) NV                                                    DRFMATS  26
      GO TO 50                                                          DRFMATS  27
    1 FORMAT (5X,3H***,1X,*THE FOLOOWING OCTAL NUMBER IS NOT ACCEPTABLE DRFMATS  28
     1AS A VECTOR,*/9X,*WHOSE ELEMENTS ARE NEEDED FOR DEFINING MULTIPLE DRFMATS  29
     2 DRIFTS.*2X,O20)                                                  DRFMATS  30
                                                                        DRFMATS  31
*       GET VECTOR NAME, INDEX AND OPNAME.                              DRFMATS  32
   20 NEW= SHIFT(NV,30)                                                 DRFMATS  33
      NEW= NEW.AND.MASK1                                                DRFMATS  34
      NMVEC = NEW.OR.5555555555B                                        DRFMATS  35
      MV = ELNUM(NMVEC)                                                 DRFMATS  36
      NMOP = INFF(1,MV)                                                 DRFMATS  37
      IF (NMOP.NE.3HVEC) GO TO 40                                       DRFMATS  38
      NK = INFF(4,MV)                                                   DRFMATS  39
      IF (KA.GT.NK) GO TO 45                                            DRFMATS  40
      M1 = M - 1                                                        DRFMATS  41
                                                                        DRFMATS  42
      DO 25 I=1,KA                                                      DRFMATS  43
      EL = FDAT(MV,I)                                                   DRFMATS  44
*       STORE LENGTH IN FL. PT. INPUT                                   DRFMATS  45
      CALL REPFLT(M1,1,EL)                                              DRFMATS  46
      CALL DRIFT(M1,EL)                                                 DRFMATS  47
      M1 = M1 - 1                                                       DRFMATS  48
   25 CONTINUE                                                          DRFMATS  49
                                                                        DRFMATS  50
```

```
      RETURN                                                            DRFMATS   51
   40 WRITE (3,3) NV                                                    DRFMATS   52
    3 FORMAT (5X,3H***,1X,A5,1X,* IS NOT A VECTOR.  CANNOT FORM MULTIPLE DRFMATS   53
     1 DRIFT MATRICES.*)                                                DRFMATS   54
      GO TO 50                                                          DRFMATS   55
   45 WRITE(3,2) NV,NK,KA                                               DRFMATS   56
    2 FORMAT (5X,3H***,1X,A5,1X,*HAS ONLY*1X,I5,1X,*ELEMENTS.*/9X,I5,1X, DRFMATS   57
     1*ARE NEEDED TO DEFINE CURRENT DRIFT MATRICES.*)                   DRFMATS   58
   50 ERROR = .TRUE.                                                    DRFMATS   59
      RETURN                                                            DRFMATS   60
      END                                                               DRFMATS   61
```

```
              SUBROUTINE DRFRSV( M0,NAME0,KA0,KB0,NAME1 )                      DRFRSV      2
      *       SETS UP INFO AND STORAGE FOR A DRIFT                             DRFRSV      3
      *       RESERVES OFR INPUT OF 1 PARAMETER, OUTPUT OF 3X3 MATRIX          DRFRSV      4
      *       M0 = M INDEX,KA0 = KA       KB0 = KB                             DRFRSV      5
      C     NAME0 = NAME       OPNAME IS DRF                                   DRFRSV      6
      *       IF M0 IS  NEGATIVE, THE FIRST LETTER OF NAME1 WILL PRECEDE NAME0 DRFRSV      7
      *     TO FORM NAME.                                                      DRFRSV      8
              COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER  SWTCH       2
              LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                         SWTCH       3
                                                                               SWTCH       4
                                                                               DRFRSV     10
              DATA MXPMY/1/                                                    DRFRSV     11
              CALL STINFO( M0,NAME0,5HDRF   ,KA0,KB0,NAME1 )                   DRFRSV     12
              IF (KB0.EQ.0)  CALL RESRV(M0,MXPMY,MINZER,1,1,0,0,0,0,0,0)       DRFRSV     13
              IF (KB0.GT.0)  CALL RESRV(M0,MXPMY,MINZER,2,1,KB0,2,0,0,0,0)     DRFRSV     14
              RETURN                                                           DRFRSV     15
              END                                                              DRFRSV     16
```

```
      SUBROUTINE DRIFT(M,EL)                                            DRIFT      2
      DIMENSION RX(2,3),RW(3)                                           DRIFT      3
      DATA RW(2),RW(3)/2*0.0/                                           DRIFT      4
C SETS UP DRIFT SPACE MATRICES RX AND RY,DIMENSIONED (2,3)  RY=RX       DRIFT      5
C THERE ARE 3 WAYS TO DEFINE DRIFTS --                                  DRIFT      6
C                                                                       DRIFT      7
C -STANDARD METHOD, EXAMPLES --                                         DRIFT      8
C 2-- 8-- -15 -19 21---*---31---*---41---*---51---*---61---*---71       DRIFT      9
C D4   DRF         3.85                                                 DRIFT     10
C---------------------------------------------------------------------- DRIFT     11
C LD4  =           3.85                                                 DRIFT     12
C D4   DRF         LD4                                                  DRIFT     13
C---------------------------------------------------------------------- DRIFT     14
C  D4 IS THE NAME OF A DRIFT OF LENGTH 3.85                             DRIFT     15
C                                                                       DRIFT     16
C -METHOD 2, CREATE A SET OF DRIFTS --                                  DRIFT     17
C 2-- 8-- -15 -19 21---*---31---*---41---*---51---*---61---*---71       DRIFT     18
C L    VEC         .54       2.14      3.85      22.67     25.          DRIFT     19
C D    DRF         L                                                    DRIFT     20
C---------------------------------------------------------------------- DRIFT     21
C  5 DRIFTS ARE CREATED. D1,D2,D3,D4,D5 OF LENGTHS = .54, 2.14, --25.   DRIFT     22
C                                                                       DRIFT     23
C -METHOD 3 IS USED TO ADJUST THE LENGTH OF THE DRIFT TO MAINTAIN THE   DRIFT     24
C  SUM OF A SET OF DRIFTS AND/OR MAGNETS AT CONSTANT LENGTH.            DRIFT     25
C 2-- 8-- -15 -19 21---*---31---*---41---*---51---*---61---*---71       DRIFT     26
C DDD  DRF      KB LDDD      LTOT      D1   D2   --   DKB               DRIFT     27
C---------------------------------------------------------------------- DRIFT     28
C  IF LTOT .NE. 0, LDDD IS ADJUSTED TO MAKE LDDD+LD1+...+LDKB = LTOT    DRIFT     29
C  IF LTOT .EQ. 0, LTOT IS STORED AS LTOT = LDDD+LD1+...+LDKB           DRIFT     30
C  ON SUBSEQUENT CALLS, LDDD WILL BE ADJUSTED TO MAINTIAN CONSTANT LTOT. DRIFT     31
C                                                                       DRIFT     32
      DO 1 I=1,2                                                        DRIFT     33
      DO 1 J=1,3                                                        DRIFT     34
      RX(I,J)=0.0                                                       DRIFT     35
    1 CONTINUE                                                          DRIFT     36
      RX(1,1)=1.                                                        DRIFT     37
      RX(1,2)=EL                                                        DRIFT     38
      RX(2,2)=1.0                                                       DRIFT     39
      RW(1) = EL                                                        DRIFT     40
      CALL STXY(M,RX,RX,RW)                                             DRIFT     41
      RETURN                                                            DRIFT     42
      END                                                               DRIFT     43
```

```
      FUNCTION DT(A)                                                    DT         2
      DT= RINDEF                                                        DT         3
      RETURN                                                            DT         4
      END                                                               DT         5
```

```
      SUBROUTINE EDRF(NBSY,V,NBP)                                       EDRF      2
      RETURN                                                            EDRF      3
      END                                                               EDRF      4
```

```
      SUBROUTINE EIGSIX(R7,LMBD,UE,TRC,BLMBD)                          EIGSIX     2
      RETURN                                                           EIGSIX     3
      END                                                              EIGSIX     4
```

```
      INTEGER FUNCTION ELNUM(NAME)                                      ELNUM      2
C SEARCHES FOR INDEX OF ELEMENT NAMED NAME                              ELNUM      3
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/STORE/LMAX,LINF,LFILE,MAX,MIN,ICARD(11)                    STORE      2
*  LMAX AND MAX ARE DIMENSIONS OF STORE AND INFF                        STORE      3
                                                                        STORE      4
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1          LDFLG,FIN                                               CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER   SWTCH      2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                          SWTCH      3
                                                                        SWTCH      4
      COMMON/FLTN/IFL(15)                                               FLTN       2
                                                                        FLTN       3
      COMMON /MATCH/ KAT,KBT,MPAR(30),IPAR(30),MSR,MTR,LTRTAB,FNORM,    MATCH      2
     1       ICON(30),ITYPE(30),NPOS(30),IPOS(7,30),BET0(30),WT(30),    MATCH      3
     2       SIG(30),NAMBFC(42),ERPR,IVSAV,LVAR(30),WTVAR(30),VAL(30),  MATCH      4
     3       VARWTS,IPRPOS(6,30)                                        MATCH      5
      LOGICAL ERPR,LVAR,VARWTS                                          MATCH      6
      DIMENSION INB(10)                                                 ELNUM     10
      LOGICAL MINV                                                      ELNUM     11
      DATA IBLNK/1H /                                                   ELNUM     12
      DATA(NAMBFC(I),I=1,42)/5HNUX  ,5HBX   ,5HAX   ,5HGX   ,5HX    ,   MATCHD     2
     1       5HDX   ,5HNUY  ,5HBY   ,5HAY   ,5HGY   ,5HY    ,5HDY   ,   MATCHD     3
     2       5HS    ,5HTHET ,                                           MATCHD     4
     3       5HBXMX ,5HBYMX ,5HXMAX ,5HQX   ,5HQY   ,5H     ,5HGAMT ,   MATCHD     5
     4       5HCHRX ,5HCHRY ,5HALPH ,5HYMAX ,5HBXMN ,5HBYMN ,5HXMIN ,   MATCHD     6
     5       5HYMIN ,5HCIRC ,5HTHTX ,5HTHTY ,5H     ,5HMTRX ,5HRES  ,   MATCHD     7
     6       5HRESN ,5HBXBY ,5HRND2 ,5HRND  ,5HWST  ,5HXDX  ,5HAXAY /   MATCHD     8
      DATA MASK/77000 00000 00000 00000B/,                              ELNUM     14
     .(INB(I),I=1,10)/33B,34B,35B,36B,37B,40B,41B,42B,43B,44B/          ELNUM     15
                                                                        ELNUM     16
      MINV=.FALSE.                                                      ELNUM     17
      VCSW=.FALSE.                                                      ELNUM     18
      MSK=MASK.AND.NAME                                                 ELNUM     19
      MSK1=MASK.AND.IBLNK                                               ELNUM     20
      IF(MSK.EQ.MSK1) GO TO 30                                          ELNUM     21
      IF (MSK.NE.IFL(15)) GO TO 15                                      ELNUM     22
      MINV=.TRUE.                                                       ELNUM     23
      LAG= SHIFT(NAME,6)                                                ELNUM     24
      DECODE (10,3,LAG) NAM                                             ELNUM     25
    3 FORMAT (A5,5X)                                                    ELNUM     26
      GO TO 16                                                          ELNUM     27
   15 NAM = NAME                                                        ELNUM     28
   16 NM = NAM                                                          ELNUM     29
   17 DO 20 I=MIN,MAX                                                   ELNUM     30
      IF (NM.EQ.INFF(2,I)) GO TO 25                                     ELNUM     31
   20 CONTINUE                                                          ELNUM     32
C  IF MODE IS 2, ELEMENT NOT YET DEFINED. SET ELEMENT NO. TO 0,RETURN   ELNUM     33
      IF (MODE.NE.2) GO TO 22                                           ELNUM     34
      ELNUM = 0                                                         ELNUM     35
```

```
      RETURN
   22 IF(VCSW) GO TO 23
C IF SECOND LETTER OF NAME IS NOT A NUMBER, STOP SEARCHING.

      NNAM= SHIFT(NAM,6)
      DECODE(10,4,NNAM) INAM
    4 FORMAT(R1,9X)
      DO 21 I=1,10
      IF(INAM.EQ.INB(I)) GO TO 24
   21 CONTINUE
      GO TO 23
   24 CONTINUE
C  SEARCH FOR ELEMENT WHOSE WHOLE NAME IS FIRST LETTER OF THIS ONE.
      VCSW = .TRUE.
      DECODE (10,2,NAM) NM
    2 FORMAT (A1,9X)
      GO TO 17
C  FAILED TO MATCH NAMES
   23 ERROR = .TRUE.
      WRITE(3,11) NAME
      ELNUM = -0
      RETURN
   11 FORMAT(15H *** ERROR *** ,A5,30H IS AN UNDEFINED ELEMENT NAME.)
   25 II=I
      IF (.NOT.VCSW) GO TO 26
      NOP=INFF(1,II)
      IF ((NOP.EQ.3HCYC).OR.(NOP.EQ.3HCYA))  GO TO 26
      IF ( (NOP.NE.4HPVEC).AND.(NOP.NE.4HCVEC) ) GO TO 23
   26 ELNUM = II
      IF (MODE.EQ.2) RETURN
      IF(.NOT.MINV) RETURN
      ELNUM = - ELNUM
      RETURN
   30 ELNUM=-0
      IF (ERPR) WRITE(3,31) NAME
   31 FORMAT (15H *** ERROR *** ,A5,* IS NOT A LEFT ADJUSTED ELEMENT NAM
     1E*)
      ERPR=.TRUE.
      RETURN
      END
```

```
      SUBROUTINE ELQ(LOC,MM,V0,L,W,E)                                  ELQ        2
                                                                       BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                        BLANK      3
      COMMON STORE(48000),IWORK(10)                                    86MARSIZ   1
      DIMENSION INFF(24,2000)                                          86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                         86MARSIZ   3
                                                                       BLANK      5
C     COMMON PI,R,TH,ER,ET                                             ELQ        4
      REAL L                                                           ELQ        5
      DIMENSION AL(1),BI(4)                                            ELQ        6
      DIMENSION W(3), E(4)                                             ELQ        7
      DATA PI/3.1415926536/                                            ELQ        8
C                                                                      ELQ        9
      X=W(1) $  Y=W(2) $  Z=W(3)                                       ELQ       10
      R=SQRT(X*X+Y*Y)                                                  ELQ       11
      TH=ATAN2(Y,X)                                                    ELQ       12
      V0L=PI*V0/L                                                      ELQ       13
      V0R=2.*V0/R                                                      ELQ       14
      RL=PI*R/L                                                        ELQ       15
       ZL=PI*Z/L                                                       ELQ       16
       V=0.$ ER=0.$ ET=0.$ EZ=0.                                       ELQ       17
      TH2=2.*TH                                                        ELQ       18
      CT=COS(TH)$   ST=SIN(TH)                                         ELQ       19
      CT2=COS(TH2)$ ST2=SIN(TH2)                                       ELQ       20
C                                                                      ELQ       21
      DO 1 M=1,MM                                                      ELQ       22
      ALM = STORE(LOC+M)                                               ELQ       23
      FM=2*M-1                                                         ELQ       24
      BLM=ALM*FM                                                       ELQ       25
      RM=FM*RL                                                         ELQ       26
      ZM=FM*ZL                                                         ELQ       27
      SZ=SIN(ZM) $  CZ=COS(ZM)                                         ELQ       28
      CALL BESIS(RM,BI,3)                                              ELQ       29
       V = ALM*BI(3)*SZ + V                                            ELQ       30
      ER = BLM* (BI(2)+BI(4)) *SZ + ER                                 ELQ       31
    1 EZ = BLM*BI(3)*CZ + EZ                                           ELQ       32
C                                                                      ELQ       33
      ER = -(V0L/2.)*CT2*ER                                            ELQ       34
      ET =   V0R*ST2*V                                                 ELQ       35
      EZ = -V0L*CT2*EZ                                                 ELQ       36
       V = V0*CT2*V                                                    ELQ       37
C                                                                      ELQ       38
      EX = ER*CT - ET*ST                                               ELQ       39
      EY = ER*ST + ET*CT                                               ELQ       40
C                                                                      ELQ       41
      E(1)=EX $  E(2)=EY $  E(3)=EZ $  E(4)=V                          ELQ       42
      RETURN                                                           ELQ       43
      END                                                              ELQ       44
```

```
                SUBROUTINE EMIS                                                 EMIS       2
          C   END MISALIGNMENT                                                  EMIS       3
                                                                                EMIS       4
                COMMON/INSTR/OPNAME,NAME,OP,KA,KB,OBJA,OBJB,NXTM,TRA,LQ,NQ,LIN, INSTR      2
               1             NIN,LFL,NFL,LBC,NBC,NTOT,IX,KIND,TYPE,ROWS,COLS,EXTR, INSTR   3
               2             M,NF,NB,NI,MSUBR,ISIGN                             INSTR      4
                INTEGER OPNAME,OP,OBJA,OBJB,TRA,TYPE,ROWS,COLS,EXTR             INSTR      5
                                                                                INSTR      6
                COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER SWTCH      2
                LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                        SWTCH      3
                                                                                SWTCH      4
                COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,  CONTRL     2
               1          LDFLG,FIN                                             CONTRL     3
                LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN    CONTRL     4
                                                                                CONTRL     5
          C                                                                     EMIS       8
                IF (MODE.EQ.3) GO TO 10                                         EMIS       9
                ISGN = 1H                                                       EMIS      10
                IF (OP.LT.0) ISGN = 1H-                                         EMIS      11
                WRITE (3,21) ISGN                                               EMIS      12
             21 FORMAT (6H   *** ,A1,7X,4HEMIS)                                 EMIS      13
                IF (OP.LT.0) RETURN                                             EMIS      14
                IF (MODE.EQ.2) RETURN                                           EMIS      15
             10 BEND =1.                                                        EMIS      16
                RETURN                                                          EMIS      17
                END                                                             EMIS      18
```

```
      SUBROUTINE ENVTRAC                                          ENVTRAC     2
                                                                  BLANK       2
      LEVEL 2, STORE,INFF,IWORK                                   BLANK       3
      COMMON STORE(48000),IWORK(10)                               86MARSIZ    1
      DIMENSION INFF(24,2000)                                     86MARSIZ    2
      EQUIVALENCE (INFF,STORE)                                    86MARSIZ    3
                                                                  BLANK       5
      COMMON/TRKINT/G,EX,EY,OMSQ,SEND,VMX,VMN,DINT,BEG,DPR,JPR,   TRKINT      2
     1   THET,PLT,NPT,NZ,PMAX,NSIZ,ITITLE(6)                      TRKINT      3
      LOGICAL BEG,DPR,PLT                                         TRKINT      4
      COMMON/BMI/MI(5000)                                         BMIL        1
      COMMON/NELS/NELS                                            BMIL        2
      LEVEL 2,MI                                                  BMIL        3
                                                                  BMIL        4
                                                                  BMI         3
      RETURN                                                      ENVTRAC     6
      END                                                         ENVTRAC     7
```

```
      SUBROUTINE EQIL(M)                                                EQIL       2
                                                                        EQIL       3
C     CALCULATE 3 CLOSED ORBITS.SYNCHRONOUS,DP/P=0,DP/P=1.              EQIL       4
C  M IS A 7X7 MATRIX PARTITIONED AS FOLLOWS.                            EQIL       5
                                                                        EQIL       6
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1          LDFLG,FIN                                               CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      DIMENSION T(49),RW(3),TN(6,7),TT(7,7),V(7,3),U(7,3)               EQIL       8
      EQUIVALENCE (T,TT)                                                EQIL       9
      INTEGER BDAT                                                      EQIL      10
      M1 = MDAT(M,1)                                                    EQIL      11
      IOP = BDAT(M,2)                                                   EQIL      12
      CALL RTRV7(M1,T,RW)                                               EQIL      13
      IF (ERROR) RETURN                                                 EQIL      14
      N = 6                                                             EQIL      15
      DO 25 K=1,3                                                       EQIL      16
      JJ = 7                                                            EQIL      17
      IF (K.EQ.3) JJ= 6                                                 EQIL      18
      DO 10 J = 1,N                                                     EQIL      19
      DO 10 I =1,N                                                      EQIL      20
      TN(I,J) = TT(I,J)                                                 EQIL      21
   10 CONTINUE                                                          EQIL      22
      DO 15 I = 1,N                                                     EQIL      23
      TN (I,I) = TN(I,I) -1.                                            EQIL      24
      TN(I,N+1) = -TT(I,JJ)                                             EQIL      25
   15 CONTINUE                                                          EQIL      26
      CALL MXDIV(TN,N,1,D,6)                                            EQIL      27
      DO 20 I=1,N                                                       EQIL      28
      V(I,K) = TN(I,N+1)                                                EQIL      29
   20 CONTINUE                                                          EQIL      30
      N = 4                                                             EQIL      31
   25 CONTINUE                                                          EQIL      32
      V(7,1) = 1.                                                       EQIL      33
      MN = MNAME(M)                                                     EQIL      34
      MT = MNAME(M1)                                                    EQIL      35
      V(7,2) = 1.                                                       EQIL      36
      V(6,3) = 1.                                                       EQIL      37
      V(5,2) = 0.                                                       EQIL      38
      V(6,2)= 0.                                                        EQIL      39
      V(5,3) = 0.                                                       EQIL      40
      V(7,3) = 0.                                                       EQIL      41
      IF (IOP.EQ.4HNOPR) GO TO 27                                       EQIL      42
      CALL DASH                                                         EQIL      43
C  PRINT VECTORS BEFORE MULTIPLYING BY MATRIX                           EQIL      44
      WRITE (3,100) MN,MT                                               EQIL      45
      WRITE (3,105) ((V(I,J),I=1,7),J=1,3)                              EQIL      46
C  MULTIPLY BY MATRIX                                                   EQIL      47
   27 DO 30 J=1,3                                                       EQIL      48
      CALL MXV7(TT,V(1,J),U(1,J))                                       EQIL      49
   30 CONTINUE                                                          EQIL      50
      IF (IOP.EQ.4HNOPR) GO TO 32                                       EQIL      51
C  PRINT VECTORS AFTER MULTIPLICATION                                   EQIL      52
      WRITE (3,110) ((U(I,J),I=1,7),J=1,2)                              EQIL      53
      WRITE (3,115) MT,(U(I,3),I=1,7)                                   EQIL      54
   32 IF (IOP.EQ.4HNOSV) RETURN                                         EQIL      55
```

```
          CALL DASH                                                     EQIL
          CALL STDAT(M,5,1,21,V)                                        EQIL
          RETURN                                                        EQIL
  100     FORMAT (/20X,15H CLOSED ORBITS A6, 12H DEFINED BY A6/29X,1HX,10X, EQIL
         1 5HDX/DS,8X,1HY,10X,5HDY/DS,7X,4HS-S0,8X,4HDP/P,7X,6HMISVAR//)    EQIL
  105     FORMAT (25H  SYNCHRONOUS C.O.  VS    ,7(F10.5,1X) /25H  DP/P = 0.  EQIL
         1 C.O.  VO    ,7(F10.5,1X)/25H  DP/P = 1.   C.O.  VP    ,7(F10.5,1X)/ EQIL
         2/)                                                            EQIL
  110     FORMAT (25H    VALUES AFTER    VS    ,7(F10.5,1X)/25H    MULTIPLYIN EQIL
         1G      VO    ,7(F10.5,1X))                                    EQIL
  115     FORMAT (5X,3H BY,1X,A4,7X,2HVP,3X,7(F10.5,1X))                EQIL
          END                                                           EQIL
```

```
      SUBROUTINE EQU(M,N)

C EQUATE COMPUTED DATA STORAGE OF M TO THAT OF N.
C   1H* IS A FLAG TO LEAVE THE PRESENT VALUES OF M UNTOUCHED,
C I.E.  KEEP THE ORIGINALS SET FOR M.  OTHERS WILL BE CHANGED TO
C THOSE READ IN FROM N OR RESET.


      DIMENSION INF(24)
      CALL RDINF(INF,N)
      DO 1 I=1,9
    1 INF(I) = 1H*
C USE LQ AND NQ OF N.  COMPUTE NEW NTOT
       DO 2 I=12,17
    2 INF(I) = 1H*
      INF(18) = INF(11) + INF(17)
C  USE KIND, ROWS, ETC OF N
C  REWRITE INFO OF M.
      CALL WRTINF(INF,M)
      RETURN
      END
```

```
      SUBROUTINE ERRSR(M)
C  ERROR HAS BEEN ENCOUNTERED. EXECUTION OF RUN HALTED.
C  READS CARDS TO FIND END OF RUN
C RETURNS TO MAIN FOR NEXT RUN
      INTEGER OPNAME
      DIMENSION ICARD(8)
      CALL WFLSR
    1 READ (2,5) (ICARD(IW),IW=1,8)
    5 FORMAT(8A10)
      DECODE (12,6,ICARD) OPNAME
    6 FORMAT (7X,A5)
      IF (OPNAME.NE.3HFIN) GO TO 1
      WRITE (3,20)
   20 FORMAT (30H *** ERROR *** RUN TERMINATED./1X,119(1H.))
      RETURN
      END
```

```
      SUBROUTINE EVEC(M)                                              EVEC     2
      RETURN                                                          EVEC     3
      END                                                             EVEC     4
```

```
      SUBROUTINE EXCHM(M,N)                                    EXCHM     2
C  RX(N) IS SET EQUAL TO RY(M),AND RY(N) IS SET EQUAL TO RX(M) EXCHM     3
      DIMENSION RX(2,3),RY(2,3),RW(3)                          EXCHM     4
      CALL RXY(M,RX,RY,RW)                                     EXCHM     5
      CALL STXY(N,RY,RX,RW)                                    EXCHM     6
      RETURN                                                   EXCHM     7
      END                                                      EXCHM     8
```

```
1            SUBROUTINE EXECMM(M,NELS,KB,MI)                                       EXECMM     2
2                                                                                  EXECMM     3
3       C  EXECUTES MMM (MATRIX MULTIPLICATION) INSTRUCTION                        EXECMM     4
4       C   MATRIX MULTIPLICATION.  R(M)=R(MK)*...R(M2)*R(M1)                      EXECMM     5
5       C  IF MI(I) IS NEGATIVE, R(M)=R(MK)...(1/R(MI))..R(M2).R(M1) ETC.          EXECMM     6
6                                                                                  EXECMM     7
7       C   SYNCH INSTRUCT MMM                                                     EXECMM     8
8       C   X   MMM    KB // A   B   C   .   .   .                                 EXECMM     9
9       C   A,B,C,... ARE PREVIOUSLY DEFINED MATRICES, OR BML?S, OR NUMBERED       EXECMM    10
10      C   PARENTHESES.  THIS INPUT LIST IS EXPANDED BY MIFILL AND MIEXP TO       EXECMM    11
11      C   FORM A LIST OF MATRIC INDICES, STORED IN MI(I).  THESE ARE             EXECMM    12
12      C   MULTIPLIED TOGETHER TO FORM MATRIX NAMED X.                            EXECMM    13
13                                                                                 EXECMM    14
14      C    IF KB IS NEGATIVE, A MATRIX CORRESPONDING TO THE REFLECTION OF THE    EXECMM    15
15      C    INPUT BEAM LINE IS ADDED TO THE BEAM LINE.  THUS ONLY HALF OF THE     EXECMM    16
16      C    SYMMETRIC SUPERPERIOD NEED BE SPECIFIED.                              EXECMM    17
17                                                                                 EXECMM    18
18      C    THE ABS(KB) IS THE NUMBER OF SUPER PERIODS. IF ABS(KB) .GT. 1, THE    EXECMM    19
19      C    MATRIX IS RAISED TO THAT POWER.                                       EXECMM    20
20                                                                                 EXECMM    21
21           COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL       DIM        2
22                                                                                 DIM        3
23           DIMENSION MI(1)                                                       EXECMM    23
24           LEVEL 2,MI                                                            EXMMML     1
25                                                                                 EXMMML     2
26                                                                                 EXECMM    24
27           CALL MMM(M,NELS,MI)                                                   EXECMM    25
28           IF (KB.GT.1)  GO TO 5                                                 EXECMM    26
29           CALL REF(M7END,M)                                                     EXECMM    27
30           MI(1) = M                                                             EXECMM    28
31           MI(2) = M7END                                                         EXECMM    29
32           CALL MMM(M,2,MI)                                                      EXECMM    30
33         5 KK = IABS(KB)                                                         EXECMM    31
34           IF (KK.LE.1)  RETURN                                                  EXECMM    32
35           DO 10 I=1,KK                                                          EXECMM    33
36        10 MI(I) = M                                                             EXECMM    34
37           CALL MMM(M,KK,MI)                                                     EXECMM    35
38           RETURN                                                                EXECMM    36
39           END                                                                   EXECMM    37
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
31	I	MI	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

```
            SUBROUTINE FCN(NPAR,G,F,X,IFLAG)                                  FCN        2
                                                                              FCN        3
                                                                              BLANK      2
            LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
            COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
            DIMENSION INFF(24,2000)                                           86MARSIZ   2
            EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                              BLANK      5
            COMMON/PAREXT/ U(30),NAM(30),DDD(32)                              FCN        5
            COMMON /MATCH/ KAT,KBT,MPAR(30),IPAR(30),MSR,MTR,LTRTAB,FNORM,    MATCH      2
           1       ICON(30),ITYPE(30),NPOS(30),IPOS(7,30),BET0(30),WT(30),    MATCH      3
           2       SIG(30),NAMBFC(42),ERPR,IVSAV,LVAR(30),WTVAR(30),VAL(30),  MATCH      4
           3       VARWTS,IPRPOS(6,30)                                        MATCH      5
            LOGICAL ERPR,LVAR,VARWTS                                          MATCH      6
            COMMON /MATCH3/NOPR,NONU,FLAG,IPR,IPP,MINFLG,MIFLG,MATFLG,GLOBAL  MATCH3     2
            LOGICAL FLAG,NOPR,NONU,MIFLG,MATFLG,GLOBAL                        MATCH3     3
            LOGICAL BETFLG,BXBYFLG                                            FCN        8
            DIMENSION X(30),BETX(9),BETY(9),T(7,7),RW(3),NPS(6)               FCN        9
            DATA NN,IBLNK/6,1H /                                              FCN       10
                                                                              FCN       11
      C  EVALUATION OF F, IE CHISQUARE, FOR MINUIT                            FCN       12
                                                                              FCN       13
            MINFLG=IFLAG                                                      FCN       14
            NGO=2                                                             FCN       15
            IF (IFLAG.EQ.4) GO TO 4                                           FCN       16
            NGO=1                                                             FCN       17
            NOPR=.FALSE.                                                      FCN       18
            IF (IFLAG.EQ.1.AND.IPP.NE.0)  NOPR = .TRUE.                       FCN       19
            IF (IFLAG.EQ.3.AND.IPP.GT.1)  NOPR=.TRUE.                         FCN       20
            NONU=.FALSE.                                                      FCN       21
            N=0                                                               FCN       22
            RTPI = 1./(4.*ACOS(0.))                                           FCN       23
        4   CONTINUE                                                          FCN       24
      C  PUT PARAMETER VALUES IN DATA ARRAYS                                  FCN       25
            BETFLG = .FALSE.                                                  FCN       26
            NVARS=NPAR                                                        FCN       27
            DO 1 J=1,NVARS                                                    FCN       28
            MEL=MPAR(J)                                                       FCN       29
            INTEL=IPAR(J)                                                     FCN       30
            CALL REPFLT(MEL,INTEL,X(J))                                       FCN       31
        1   CONTINUE                                                          FCN       32
      C  CALL SYNCH SUBROUTINE                                                FCN       33
            CALL DOIT(MSR)                                                    FCN       34
      C  CALCULATE CHISQUARE                                                  FCN       35
            GO TO (8,9) NGO                                                   FCN       36
        8   WRITE(3,100)                                                      FCN       37
        9   F = 0.0                                                           FCN       38
                                                                              FCN       39
            DO 2 K=1,KAT                                                      FCN       40
            ITYPK=ITYPE(K)                                                    FCN       41
            IF(ITYPK.EQ.-1) GO TO 2                                           FCN       42
            NPOSK=NPOS(K)                                                     FCN       43
            ICONK=ICON(K)                                                     FCN       44
            BXBYFLG = .FALSE.                                                 FCN       45
            IF (ICONK.EQ.37)  BXBYFLG = .TRUE.                                FCN       46
            WTK=WT(K)                                                         FCN       47
            BFITK=BET0(K)                                                     FCN       48
```

```
      IF ((ITYPK.EQ.1).OR.(ITYPK.EQ.2)) GO TO 10
      IF (ICONK.EQ.34)  GO TO 22
      IF (MATFLG.OR.ICONK.EQ.35)  GO TO 3
      IF (BXBYFLG)  GO TO 21
      LTRTBK = LTRTAB - 1 + ICONK
      GO TO 28
   21 LBXK = LTRTAB + 1
      LBYK = LTRTAB + 7
   28 CONTINUE

      IF(NPOSK.EQ.1) GO TO 17
      FIK = 0.0

      DO 6 L=1,NPOSK
      IPLK = 14*IPOS(L,K)
      IF (.NOT.BXBYFLG)  GO TO 25
      NIKL1 = LBXK + IPLK
      NIKL2 = LBYK + IPLK
      BX = STORE(NIKL1)
      BY = STORE(NIKL2)
      BETKL = BX/BY
      GO TO 26
   25 NIKL = LTRTBK + IPLK
      BETKL = STORE(NIKL)
   26 DEL = BETKL - BFITK
      FIK = FIK + DEL*DEL
      GO TO (11,6) NGO
   11 NTYP=5H
      N=N+1
      WRITE(3,101) N,NAMBFC(ICONK),NTYP,IPRPOS(L,K),BETKL,BFITK,DEL,
     1             SIG(K)
    6 CONTINUE

      GO TO 5

   17 CONTINUE
      IPLK = 14*IPOS(1,K)
      IF (.NOT.BXBYFLG)  GO TO 27
      NIKL1 = LBXK + IPLK
      NIKL2 = LBYK + IPLK
      BX = STORE(NIKL1)
      BY = STORE(NIKL2)
      BET1K = BX/BY
      GO TO 23
   27 NIK1 = LTRTBK + IPLK
      BET1K=STORE(NIK1)

   23 CONTINUE
      DEL=BET1K-BFITK
      FIK = DEL*DEL
      GO TO (12,5) NGO
   12 NTYP=5H
      N=N+1
      IF (NPOSK.EQ.1)  GO TO 36
      WRITE(3,104) N,NAMBFC(ICONK),NTYP,(IPRPOS(L,K),L=1,NN),BET1K,
     1             BFITK,DEL,SIG(K)
      GO TO 5
```

```
   36 WRITE(3,101) N,NAMBFC(ICONK),NTYP,IPRPOS(1,K),BET1K,BFITK,DEL,
     1              SIG(K)
      GO TO 5

   10 CONTINUE
      FIK=0.0
      DO 14 L=1,NPOSK
      NIKL=LTRTAB+14*IPOS(L,K)-1+ICONK
      BETKL=STORE(NIKL)
      IF (L.EQ.1)  GO TO 140
      DBKJ = BETKL - BETKJ
      DEL = DBKJ - BFITK
      FIK = FIK + DEL*DEL
  140 IF (ITYPK.EQ.2)  BETKL = -BETKL
      BETKJ = BETKL
      IF (L.EQ.1)  GO TO 14
      GO TO (13,14) NGO
   13 NTYP=5HDIF
      IF (ITYPK.EQ.2)  NTYP=5HSUM
      N=N+1
      WRITE(3,101) N,NAMBFC(ICONK),NTYP,IPRPOS(L,K),DBKJ,BFITK,
     1               DEL,SIG(K)
   14 CONTINUE

      GO TO 5

    3 CONTINUE
      IF (BETFLG)  GO TO 33
      CALL BET(MTR,BETX,BETY,0)
      BETFLG = .TRUE.
   33 IF (ICONK.EQ.35)  GO TO 24
      IF (BXBYFLG)  GO TO 29
      IF (ICONK.GT.6)  GO TO 31
      BET1K = BETX(ICONK)
      GO TO 32
   31 BET1K = BETY(ICONK-6)
   32 IF (ICONK.EQ.1.OR.ICONK.EQ.7)  BET1K = RTPI * BET1K
      GO TO 23

   29 BX = BETX(2)
      BY = BETY(2)
      BET1K = BX/BY
      GO TO 23

   24 CONTINUE
      APQN = 0.5*( IPOS(1,K)*BETX(1) + IPOS(2,K)*BETY(1) ) * IPOS(3,K)
     1        / IPOS(4,K)
      BET1K = 1./ (ABS(SIN(APQN)) + .0000000001)
      GO TO 23

   22 CONTINUE
      CALL RTRV7(MTR,T,RW)
      II = IPOS(1,K)
      JJ = IPOS(2,K)
      BET1K = T(II,JJ)
      GO TO 23
```

```
                5    F = F + FIK * WTK*WTK                                          FCN     163
                                                                                    FCN     164
                2    CONTINUE                                                       FCN     165
  175                                                                               FCN     166
                     IF (.NOT.VARWTS) GO TO 20                                      FCN     167
                     DO 18 J=1,NVARS                                                FCN     168
                     IF(.NOT.LVAR(J)) GO TO 18                                      FCN     169
                     DIF=(X(J)-VAL(J))                                              FCN     170
  180                DIFWT=DIF*WTVAR(J)                                             FCN     171
                     F = F + DIFWT*DIFWT                                            FCN     172
                     GO TO (19,18) NGO                                              FCN     173
                19   N=N+1                                                          FCN     174
                     SIGV=1./WTVAR(J)                                               FCN     175
  185                WRITE(3,103) N,NAM(J),X(J),VAL(J),DIF,SIGV                     FCN     176
                18   CONTINUE                                                       FCN     177
                                                                                    FCN     178
                20   F = F*FNORM                                                    FCN     179
                                                                                    FCN     180
  190                GO TO (15,16) NGO                                              FCN     181
                15   WRITE(3,102) N,F                                               FCN     182
                16   IF(IFLAG.NE.1) RETURN                                          FCN     183
                     NOPR=.TRUE.                                                    FCN     184
                     NONU=.TRUE.                                                    FCN     185
  195                DO 7 K=1,KAT                                                   FCN     186
                     IF (ICON(K).EQ.1) NONU=.FALSE.                                 FCN     187
                     IF (ICON(K).EQ.7) NONU=.FALSE.                                 FCN     188
                7    CONTINUE                                                       FCN     189
                                                                                    FCN     190
  200                RETURN                                                         FCN     191
                100  FORMAT(/3X,*CONSTRAINT*,3X,*FUNCTION*,13X,*POSITIONS OR INDICES*  FCN  192
                    1  13X,*PRESENT VALUE*,4X,*DESIRED VALUE*,4X,*ERROR*,11X,       FCN     193
                    2  *TOLERANCE*/)                                                FCN     194
                101  FORMAT(/3X,I5,8X,A5,2X,A5,2X,A5,30X,4F16.7)                    FCN     195
  205           104  FORMAT(/3X,I5,8X,A5,2X,A5,2X,6A5,5X,4F16.7)                    FCN     196
                103  FORMAT(3X,I5,8X,A10,38X,4F16.7)                                FCN     197
                102  FORMAT (/5X,27HFCN = SUM[(ERROR/TOL.)**2]/,I2,2H =,E15.7)      FCN     198
                     END                                                            FCN     199
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
45	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
84	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
108	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
131	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
182	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
190	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

```
      FUNCTION FDAT(M,J)                                                FDAT        2
C                                                                       FDAT        3
C   RETRIEVES ONE PIECE OF FL. PT. DATA FROM INDEX J OF FL PT           FDAT        4
C STORAGE OF ELEMENT M.  IF DATA IS SYMBOLIC, THE VALUE (IN FL PT) IS   FDAT        5
C RETRIEVED.  SYMBOLICS MAY BE NESTED 10 DEEP.                          FDAT        6
C   INTERNAL STORAGE OF SYMBOLIC IS  00NNNINAME                         FDAT        7
C                                                                       FDAT        8
C WHERE    00 ARE PURE ZEROS                                            FDAT        9
C          NNN IS AN INTEGER OR BCD BLANK INDICATING WHICH VARIABLE OF  FDAT       10
C          ELEMENT NAMED INAME IS TO BE RETRIEVED.                      FDAT       11
C                                                                       FDAT       12
C                                                                       FDAT       13
                                                                        BLANK       2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK       3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ    1
      DIMENSION INFF(24,2000)                                           86MARSIZ    2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ    3
                                                                        BLANK       5
C                                                                       FDAT       15
      LOC = INFF(14,M) + J - 1                                          FDAT       16
      FDAT = REALNUM(LOC)                                               FDAT       17
C                                                                       FDAT       18
      RETURN                                                            FDAT       19
      END                                                               FDAT       20
```

```
      SUBROUTINE FITT(MSR,MC,MV1,IV1,PHI,KA,KB)                          FIT       2
*       FITTING ROUTINE CALLED BY FITQ, FITB, FITR, FITV                 FIT       3
                                                                         FIT       4
*   FITS   FITQ  --PHASE ADVANCE OVER 2PI                                FIT       5
*          FITB --BETATRON FUNCTIONS WHOSE INDECES ARE KA AND KB.        FIT       6
*          FITR --MATRIX ELEMENTS INDEXED BY KA AND KB.                  FIT       7
*          FITV --VECTOR ELEMENTS INDEXED BY KA AND KB.                  FIT       8
                                                                         FIT       9
*  INPUT --  NAME   FITX    KA  KB  SUB  C QF  QD  I1  I2  A1  A2        FIT      10
*                    WHERE FITX MAY BE FITQ, FITB,FITR,FITV              FIT      11
*                    SUB IS NAME OF SUBROUTINE TO EXECUTE.               FIT      12
*     VARY THE I1-TH PARAMETER OF QF AND I2-TH OF QD, COMPUTE C,         FIT      13
*  COMPARE THE RESULTS ACCORDING TO X OF FIT INSTRUCTION (PHASE ADVANCE  FIT      14
*  OR BETATRON FUNCTIONS IF FITQ OR FITB.  MATRIX ELEMENTS OR VECTORS    FIT      15
*  NAMED IN KA AND KB IF FITR OR FITV.) WITH A1 AND A2.                  FIT      16
*       ITERATE UNTIL THEY MATCH, BUT NOT MORE THAN 100 TIMES.           FIT      17
                                                                         FIT      18
                                                                         BLANK     2
      LEVEL 2, STORE,INFF,IWORK                                          BLANK     3
      COMMON STORE(48000),IWORK(10)                                      86MARSIZ  1
      DIMENSION INFF(24,2000)                                            86MARSIZ  2
      EQUIVALENCE (INFF,STORE)                                           86MARSIZ  3
                                                                         BLANK     5
      COMMON/GRR/IERR,BSW,VSW,RSW,QSW,XSW                                GRR       2
      LOGICAL IERR,BSW,VSW,RSW,QSW,XSW                                   GRR       3
                                                                         GRR       4
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,     CONTRL    2
     1          LDFLG,FIN                                                CONTRL    3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN       CONTRL    4
                                                                         CONTRL    5
      DIMENSION H(2),D(2),A(2,2),B(2,2),RX(2,3),RY(2,3),S(2),PHI(5)      FIT      22
      DIMENSION MV1(1),IV1(1),V(2),U(2),W(2),VAR(2),FC(2),F(2),G(2)      FIT      23
      DIMENSION RXRY(12),VC(7),BXBY(20),RW(3),Q(2)                       FIT      24
      LOGICAL IND,INF                                                    FIT      25
      LOGICAL INDEFF,INFINN                                              FIT      26
      INTEGER P1,P2,ELNUM                                                FIT      27
                                                                         FIT      28
      LOGICAL SINGL                                                      FIT      29
      SINGL=.FALSE.                                                      FIT      30
      IF(QSW) GO TO 99                                                   FIT      31
      IF(RSW.AND.KB.NE.3H   ) GO TO 99                                   FIT      32
      IF(KA.GE.100.OR.KB.NE.0) GO TO 99                                  FIT      33
      SINGL=.TRUE.                                                       FIT      34
      KB=KA                                                              FIT      35
      MV1(2)=MV1(1)                                                      FIT      36
      IV1(2)=IV1(1)                                                      FIT      37
      PHI(2)=PHI(1)                                                      FIT      38
   99 CONTINUE                                                           FIT      39
      Q(1) = PHI(1)                                                      FIT      40
      Q(2) = PHI(2)                                                      FIT      41
      DEL = 1.E-6                                                        FIT      42
      IF(.NOT.QSW) DEL=1.0E-05                                           FIT      43
      HSTEP = .01                                                        FIT      44
      XLAM = .9                                                          FIT      45
C  RESET PARAMETERS IF INPUT                                             FIT      46
      IF (.NOT.RSW) GO TO 60                                             FIT      47
      DECODE(3,21,KA)I1,I2,I3                                            FIT      48
```

```
          DECODE (3,21,KB) J1,J2,J3
       21 FORMAT (3I1)
          K1 = I1*6 + (I3-1) *2 +I2
          K2 = J1*6 +(J3-1)*2 + J2
          GO TO 16
       60 K1 = KA
          K2 = KB
       16 M1 = ELNUM(MV1(1))
          M2 = ELNUM(MV1(2))
          P1 = IV1(1)
          P2 = IV1(2)
      C  LOOK UP INPUT GUESS FOR VARIABLES
          VAR(1) = FDAT(M1,P1)
          VAR(2) = FDAT(M2,P2)
      C  SAVE ORIGINAL INPUT VARIABLES
          VR1=VAR(1)
          VR2=VAR(2)
          IF (.NOT.QSW) GO TO 50
      C  FIT Q CASE
          PHI(1) = COS(6.283185307 * PHI(1))
          PHI(2) = COS(6.283185307 * PHI(2))
          GO TO 55
      *   OTHER FITS
       50 PHI(1) = PHI(1)/100.
          PHI(2) = PHI(2) /100.
       55 K=0
      C  SET MAXIMUM ITERATIONS
          KI = 100
       75 DO 100 I=1,2
          U(I)=VAR(I)
          V(I)=VAR(I)
      100 W(I)=VAR(I)
          V(1) = V(1) + HSTEP
          IF(SINGL) V(2)=V(1)
          W(2) = W(2) + HSTEP
          DO 115 I=1,3
          GO TO (101,102,103),I
      101 VC1 = V(1)
          VC2 = V(2)
          GO TO 104
      102 VC1 = W(1)
          VC2 = W(2)
           GO TO 104
      103 VC1 = U(1)
          VC2 = U(2)
      104 CALL REPFLT(M1,P1,VC1)
          CALL REPFLT(M2,P2,VC2)
          CALL DOIT(MSR)
          IF (ERROR) RETURN
          IF(BSW) GO TO 105
          IF (VSW) GO TO 110
          IF (RSW) GO TO 112
      C  FITQ
           CALL RXY(MC,RX,RY,RW)
          FC(1) = (RX(1,1) + RX(2,2)) * .5
          FC(2) = (RY(1,1) + RY(2,2)) * .5
          GO TO 106
```

```
  105 CALL BET(MC,BXBY(1),BXBY(11),0)
      FC(1)=BXBY(K1)/100.
      FC(2) = BXBY(K2)/100.
      GO TO 106
  110 CALL DATA(MC,5,1,7,VC)
      FC(1) = VC(K1)/100.
      FC(2)=VC(K2)/100.
      GO TO 106
  112 CALL RXY(MC,RXRY(1),RXRY(7),RW)
      FC(1)=RXRY(K1)/100.
      FC(2)=RXRY(K2)/100.
  106 GO TO (107,108,109),I
  107 G(1) = FC(1)
      G(2) = FC(2)
      GO TO 115
  108 H(1) = FC(1)
      H(2) = FC(2)
      GO TO 115
  109 F(1) = FC(1)
      F(2)= FC(2)
      IND= INDEFF(F(1))
      INF= INFINN(F(1))
      IF(IND.OR.INF) GO TO 130
      IND = INDEFF(F(2))
      INF= INFINN(F(2))
      IF(IND.OR.INF) GO TO 130
  115 CONTINUE
      D(1) = PHI(1) - F(1)
      D(2) = PHI(2) - F(2)
      IF (ABS(D(1)).LE.DEL.AND.ABS(D(2)).LE.DEL) GO TO 120
C  FIT NOT CLOSE ENOUGH.  ITERATE AGAIN UNLES LIMIT HAS BEEN REACHED
      K = K + 1
      IF (K.EQ.KI) GO TO 125
C  SET UP NEXT GUESS
      HD = 1./HSTEP
      A(1,1) = (G(1) - F(1))*HD
      A(1,2) = (H(1) - F(1))*HD
      A(2,1) = (G(2) - F(2))*HD
      A(2,2) = (H(2) - F(2))*HD
      IF(SINGL) GO TO 999
      DET = (A(1,1)*A(2,2)) - (A(1,2) * A(2,1))
      B(1,1) = A(2,2)/DET
      B(1,2) = -A(1,2)/DET
      B(2,1) = -A(2,1) /DET
      B(2,2) = A(1,1) /DET
      S(1) = B(1,1)*D(1) + B(1,2)*D(2)
      S(2)=B(2,1)*D(1) + B(2,2) *D(2)
      VAR(1) = VAR(1) + (S(1)*XLAM)
      VAR(2) = VAR(2) + (S(2)*XLAM)
      GO TO 75
  999 VAR(1)=VAR(1)+XLAM*D(1)/A(1,1)
      VAR(2)=VAR(1)
      GO TO 75

C  FIT SATISFIED.  LEAVE NEW VARIABLES IN STORAGE
  120 WRITE (3,1) P1,MV1(1),VAR(1),P2,MV1(2),VAR(2)
      RETURN
```

```
                1     FORMAT (//39H PARAMETER REPLACEMENTS MADE BY FITTING/5X,I5,1X,     FIT       163
                   1  4H OF ,A5,3H = ,E16.9,7X,I5,1X,4H OF ,A5,3H = ,E16.9)              FIT       164
                                                                                         FIT       165
  175         *   ITERATION LIMIT REACHED.  FIT FAILED.                                  FIT       166
               125  WRITE (3,7) K,VAR                                                    FIT       167
                    GO TO 135                                                            FIT       168
                7     FORMAT (39H NUMBER OF ITERATIONS REACHED LIMIT OF I5/31H  CURRENT  FIT       169
                   1 VARIABLES ARE VAR1 = F14.8,8H VAR2 = F14.8)                         FIT       170
  180          130  CONTINUE                                                             FIT       171
                    WRITE (3,10)                                                         FIT       172
               10   FORMAT (32H ROOT IS INDEFINITE OR INFINITE.//)                       FIT       173
              C REPLACE ORIGINAL INPUT VALUES                                            FIT       174
               135  CALL REPFLT(M1,P1,VR1)                                               FIT       175
  185               CALL REPFLT(M2,P2,VR2)                                               FIT       176
              *   IF FITQ, TRY OTHER FIT ROUTINE                                         FIT       177
                    IF (QSW) CALL FITE(MSR,MC,MV1,IV1,Q,KA,KB)                           FIT       178
                    RETURN                                                               FIT       179
                    END                                                                  FIT       180
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
45	I	MV1	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
46	I	IV1	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
66	I	MV1	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
68	I	IV1	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
94	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
126	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
170	I	MV1	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

```
      SUBROUTINE FITE(MSR,MC,MV1,IV1,Q,KA,KB)                           FITE        2
                                                                        BLANK       2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK       3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ    1
      DIMENSION INFF(24,2000)                                           86MARSIZ    2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ    3
                                                                        BLANK       5
      COMMON/QUE/V1,V2,G,CQXZ,CQYZ,CQX,CQY,M1,M2,P1,P2,MS,ME,VV1,VV2    FITE        4
     1,K1,K2                                                            FITE        5
      INTEGER P1,P2,ELNUM                                               FITE        6
      DIMENSION MV1(1),IV1(1),Q(2)                                      FITE        7
      COMMON/GRR/IERR,BSW,VSW,RSW,QSW,XSW                               GRR         2
      LOGICAL IERR,BSW,VSW,RSW,QSW,XSW                                  GRR         3
                                                                        GRR         4
      EXTERNAL QUEX                                                     FITE        9
                                                                        FITE       10
      IF (.NOT.RSW) GO TO 2                                             FITE       11
      DECODE (3,10,KA) I1,I2,I3                                         FITE       12
      DECODE (3,10,KB) J1,J2,J3                                         FITE       13
   10 FORMAT (3I1)                                                      FITE       14
      K1 = I1*6 +(I3-1)*2 + I2                                          FITE       15
      K2= J1*6 +(J3-1)*2 + J2                                           FITE       16
      GO TO 4                                                           FITE       17
    2 K1=KA                                                             FITE       18
      K2=KB                                                             FITE       19
    4 MS = MSR                                                          FITE       20
      ME = MC                                                           FITE       21
      M1 = ELNUM(MV1(1))                                                FITE       22
      M2 = ELNUM(MV1(2))                                                FITE       23
      P1 = IV1(1)                                                       FITE       24
      P2 = IV1(2)                                                       FITE       25
C     MN = INFF(1,M1)                                                   FITE       26
      V1 = FDAT(M1,P1)                                                  FITE       27
      V2 = FDAT(M2,P2)                                                  FITE       28
      VV1 = V1                                                          FITE       29
      VV2 = V2                                                          FITE       30
      IF (.NOT.QSW) GO TO 3                                             FITE       31
      CQXZ = COS(6.283185307 * Q(1))                                    FITE       32
      CQYZ = COS(6.283185307 * Q(2))                                    FITE       33
      GO TO 5                                                           FITE       34
    3 CQXZ = Q(1)                                                       FITE       35
      CQYZ = Q(2)                                                       FITE       36
      CQXZ = CQXZ/100.                                                  FITE       37
      CQYZ = CQYZ/100.                                                  FITE       38
    5 CALL GRT(1,V1,7,QUEX)                                             FITE       39
      IF(IERR) GO TO 7                                                  FITE       40
    6 WRITE (3,1) P1,MV1(1),V1,P2,MV1(2),V2                             FITE       41
    1 FORMAT (//39H PARAMETER REPLACEMENTS MADE BY FITTING/5X,I5,1X,    FITE       42
     1  4H OF ,A5,3H = ,E16.9,7X,I5,1X,4H OF ,A5,3H = ,E16.9)           FITE       43
      RETURN                                                            FITE       44
    7 V1 = VV1                                                          FITE       45
      V2 = VV2                                                          FITE       46
      CALL REPFLT(M1,P1,V1)                                             FITE       47
      CALL REPFLT(M2,P2,V2)                                             FITE       48
      GO TO 6                                                           FITE       49
      END                                                               FITE       50
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
29	I	MV1	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
31	I	IV1	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
47	I	MV1	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

```
          FUNCTION FLDAT(M,J)                                            FLDAT       2
C  DATA RETRIEVED MUST BE FLOATING POINT.  SYMBOLIC NOT ALLOWED.         FLDAT       3
C  RETRIEVES ONE FLOATING POINT NUMBER FROM INDEX J OF LQ OF ELEMENT M   FLDAT       4
C                                                                        FLDAT       5
C    IF RETRIEVING FROM A VAR SYNCH INSTRUCTION, FETCH VALUE             FLDAT       6
C  IN FL. PT. STORAGE FROM POSITION DEFINED BY KA.                       FLDAT       7
                                                                         BLANK       2
      LEVEL 2, STORE,INFF,IWORK                                          BLANK       3
      COMMON STORE(48000),IWORK(10)                                      86MARSIZ    1
      DIMENSION INFF(24,2000)                                            86MARSIZ    2
      EQUIVALENCE (INFF,STORE)                                           86MARSIZ    3
                                                                         BLANK       5
      INTEGER HVAR                                                       FLDAT       9
      DATA HVAR/3HVAR/                                                   FLDAT      10
C                                                                        FLDAT      11
      LOC = INFF(10,M) + J - 1                                           FLDAT      12
      NAMOP = INFF(1,M)                                                  FLDAT      13
      IF (NAMOP.NE.HVAR)  GO TO 25                                       FLDAT      14
      N = MDAT(M,1)                                                      FLDAT      15
      KA = INFF(4,M)                                                     FLDAT      16
      IF (KA.EQ.0)  KA = 1                                               FLDAT      17
      LOC = INFF(14,N) + KA - 1                                          FLDAT      18
   25 FLDAT = STORE(LOC)                                                 FLDAT      19
      RETURN                                                             FLDAT      20
      END                                                                FLDAT      21
```

```
      SUBROUTINE FLQ(M,L,J,F)                                           FLQ       2
C   FETCH F FROM INDEX OF LQ STORAGE OF ELEMENT M                       FLQ       3
C    L=1 FOR LQ      L=2 FOR LQ2       L=3 FOR LQ3                      FLQ       4
                                                                        BLANK     2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK     3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ  1
      DIMENSION INFF(24,2000)                                           86MARSIZ  2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ  3
                                                                        BLANK     5
      LM = 10                                                           FLQ       6
      IF (L.EQ.2) LM = 19                                               FLQ       7
      IF (L.EQ.3) LM = 24                                               FLQ       8
   18 LOC = INFF(LM,M) + J - 1                                          FLQ       9
C  IF LQ3 OR LQ2 HAVE NOT BEEN SET, USE LQ2 OR LQ                       FLQ      10
       IF (LOC.NE.0) GO TO 15                                           FLQ      11
      IF (LM.EQ.10) GO TO 15                                            FLQ      12
      IF (LM.EQ.24) LM = 19                                             FLQ      13
      IF (LM.EQ.19) LM = 10                                             FLQ      14
      GO TO 18                                                          FLQ      15
   15 F = REALNUM(LOC)                                                  FLQ      16
      RETURN                                                            FLQ      17
      END                                                               FLQ      18
```

```
      SUBROUTINE FXPT(M)                                                FXPT       2
                                                                        FXPT       3
C EQUILIBRIUM ORBITS UNDER LINEAR AND/OR ARBITRARY EXTERNAL TRANSFOR-    FXPT       4
C MATIONS, LINEARIZED 4X4 MATRIX NEAR FIXED POINT (M4), EIGENVALUES     FXPT       5
C AND EIGENVECTORS OF M4, ETC.                                          FXPT       6
C                                                                       FXPT       7
C S    FXPT  KA  KB//P   BLIN/NSTR ISAV/ITERS IFLAG/ITAPE               FXPT683    1
C                   /E1       E2        D1        D2        D3          FXPT       9
                                                                        FXPT      10
C     KA = OUTPUT OPTION ---                                            FXPT      11
C     KA = 0   MATRIX,EIGENVALUES,CLOSED ORBIT,ETA-ORBIT AT POSITION 0  FXPT      12
C     KA = 1   OPTION 0 AND EIGENVECTORS AT POSITION 0                  FXPT      13
C     KA = 2   OPTION 1 AND TRACK VCO, ETA                              FXPT      14
C     KA = 3   OPTION 1 AND TRACK EIGENVECTORS                          FXPT      15
C     KA = 4   OPTIONS 2 AND 3 TOGETHER                                 FXPT      16
C      IF KA  IS NEGATIVE, PRINT OF ORBITS IS SUPPRESSED                FXPT      17
C      AT ALL POSITIONS EXCEPT THOSE OF ELEMENTS WHOSE                  FXPT      18
C      FIRST CHARACTER IS  " ( 4-8 )                                    FXPT      19
                                                                        FXPT      20
C     KB = NUMBER OF SUPERPERIODS                                       FXPT      21
C      P  = PREV. DEF. PART. WHOSE INPUT CONTAINS INITIAL REF. RAY      FXPT      22
C             X   DX   Y   DY   DS   DP/P   1.                          FXPT      23
C     BLIN = BEAM LINE DEFINED BY BML INSTRUCTION                       FXPT      24
C     IF KB IS NEGATIVE, THEN                                           FXPT      25
C      THE REFLECTION OF BLIN IS ADDED TO BLIN TO FORM COMPLETE SP.     FXPT      26
C      THIS FEATURE WILL NOT WORK IF  NSTR .GT. 1                       FXPT      27
                                                                        FXPT      28
C   NSTR = NUMBER OF SUPERPERIODS FOR FIXED POINT ORBIT TO CLOSE        FXPT      29
C     ISAV = 0,1 --- DONT SAVE, SAVE CALCULATED EQ.ORBIT IN P           FXPT      30
C ITERS = MAXIMUM NUMBER OF ITERATIONS                                  FXPT      31
C IFLAG = 0     CLOSED ORBIT RAY ONLY                                   FXPT      32
C IFLAG = 1    PLUS 9 NEIGHBORING RAYS.                                 FXPT      33
C IFLAG = 2   PLUS 4 RAYS OF LINEARIZED EQUATIONS.                      FXPT      34
                                                                        FXPT      35
C IFLAG = 3   PLUS 4 RAYS OF LINEARIZED EQUATIONS,                      FXPT      36
C             FIRST FOR THE ENVELOPE AND SECOND FOR THE IONS OF A BEAM. FXPT      37
                                                                        FXPT      38
C ITAPE = 0,1  DO NOT,DO  WRITE ETA AND CLOSED ORBITS ON TAPE 12        FXPT683    2
C     E1, E2 = CONVERGENCE CRITERIA FOR DISPLACEMENTS, SLOPES           FXPT      39
C             D1,D2 =  DISP. FROM REF. RAY OF NEIGHBOR  RAYS FOR        FXPT      40
C                      CALCULATING M4.                                  FXPT      41
C     D3 = DISP IN DP/P FOR CALCULATING LINEARIZED 3XO MATRICES         FXPT      42
C                OUT = OUTPUT OPTION IN BCD                             FXPT      43
                                                                        FXPT      44
C IF D1=0, NEIGHBORING RAYS NOT CALCULATED, LINEARIZATION WITH REVMAT.  FXPT      45
C IF OUT IS BLANK, EIGENVECTORS NOT CALCULATED, NO TRACKING.            FXPT      46
C IF OUT = 5HEVTRK, EIGENVECTOR ORBITS ARE TRACKED.                     FXPT      47
C LINEARIZED 7X7 MATRIX STORED AT LQ+KADD OF FXPT INSTRUCTION           FXPT      48
C LINEARIZED 4X4 MATRIX STORED AT LQ+KADD+53                            FXPT      49
C EIGENVALUES  STORED AT  AT LQ3, EIGENVECTORS AT LQ3+8                 FXPT      50
C LQ2 = LQ++KADD+49+16,  LQ3 = LQ2+7,  KADD = 3.  KIND IS 2HR7          FXPT      51
                                                                        FXPT      52
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
```

```
                                                                     BLANK      5
         COMMON/BMI1/MI1(16000)                                      NV3BMI1    1
         COMMON/NELS1/NELS1                                          BMI1L      2
         LEVEL 2,MI1                                                 BMI1L      3
                                                                     BMI1L      4
C                                                                    BMI1       3
         COMMON/ MH/ML   $      LEVEL 2,ML                           FXPTL      1
         DIMENSION MI(1)                                             FXPT      55
         EQUIVALENCE (MI,MI1),(NELS,NELS1)                           FXPT      56
         COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,  CONTRL     2
        1          LDFLG,FIN                                         CONTRL     3
         LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN    CONTRL     4
                                                                     CONTRL     5
         COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL DIM        2
                                                                     DIM        3
         COMMON/CDERIV/IFLAG,MR,NR,DP,NV(11),DAT(20),UZ,STOT,        CDERIV     2
        1 DAT2(7),MP2FLG,NF,H0,LOCALPH,NUMALPH                       CDERIV     3
         COMMON/CORB/TH(84),TV(84),NMON,NNH,NNV,IIH,IIV,             CORB613    1
        .            NAMONH,NAMONV,NACORH,NACORV                     CORB613    2
         DIMENSION RW(3)                                             FXPT      60
         DIMENSION P(5),VW(112),V(7,16),E(4),F(4),VV(7),             FXPT      61
        1          RM(4,4),RMI(4,5),LAM(4),BLAM(4),C(4),CM(3,3),X(4,4),  FXPT      62
        2 VZ(7,7),RX(2,3),LABELA(4),LABELB(4),MU(4)                  FXPT      63
         DIMENSION LBL(6),Q(4),R7(7,7),ML(2),VP(7),EMOD(4,2),EPHI(4,2)   FXPT      64
         EQUIVALENCE (V,VW),(VV(1),VW(78)),(VZ(1,1),VW(85))          FXPT      65
         COMPLEX CTR,CSQ,LAM,BLAM,CABS,CSQRT,CM,X,EX1,EX3            FXPT      66
         REAL MU                                                     FXPT      67
         INTEGER TRKR,R5J                                            FXPT      68
         DATA (NV(I),I=1,11) /1,2,3,4,5,6*0/                         FXPT      69
         DATA ( LABELA(I), I = 1, 4 )/ 5H0X..., 1H , 5H0Y..., 1H  /,  FXPT      70
        1     ( LABELB(I), I = 1, 4 )/ 6HLMD1  , 6H1/LMD1, 6HLMD3  ,  FXPT      71
        2       6H1/LMD3 /                                           FXPT      72
         DATA (RW(I),I=1,3)/3*0.0/                                   FXPT      73
         DATA(VP(I),I=5,7)/0.,1.,0./                                 FXPT      74
         DATA ( LBL(J),J=1,6 ) / 10H REAL(X1) ,10H IMAG(X1) ,        FXPT      75
        1                        10H REAL(X3) ,10H IMAG(X3) ,10H EQ ORBIT , FXPT      76
        2                        10H ETA ORBIT/                      FXPT      77
         DATA IBLNK/1H /                                             FXPT      78
                                                                     FXPT      79
         CALL RANGET(NRN)                                            FXPT      80
         IIH = 1                                                     FXPT683    4
         IIV = 1                                                     FXPT683    5
         TPI = 4.*ACOS(0.)                                           FXPT      81
         KA = INFF(4,M)                                              FXPT      82
         KA = IABS(KA)                                               FXPT      83
         KB = INFF(5,M)                                              FXPT      84
         SPER = FLOAT(IABS(KB))                                      FXPT      85
         NSTR = IDAT(M,1)                                            FXPT      86
         IF (NSTR.EQ.0) NSTR=1                                       FXPT      87
         IVSV = IDAT(M,2)                                            FXPT      88
         ITERS=IDAT(M,3)                                             FXPT      89
         IFLAG = IDAT(M,4)                                           FXPT      90
         ITAPE = IDAT(M,5)                                           FXPT683    6
         IF (ITERS.EQ.0) ITERS=10                                    FXPT      91
C  RETRIEVE FLOATING POINT INPUT E1,E2,D1,D2                         FXPT      92
         CALL DATA(M,1,1,5,P)                                        FXPT      93
         EPSW = P(1)                                                 FXPT      94
```

```
      EPST = P(2)
      DELW = P(3)
      DELT = P(4)
      DELP = P(5)
      IF (EPSW.EQ.0.) EPSW=1.E-08
      IF (EPST.EQ.0.) EPST=1.E-08
      IF (DELP.EQ.0) DELP = 1.E-08
      NGO = 2
      IF (IFLAG.EQ.0) GO TO 2
      NGO=1
      IF (DELW.EQ.0.)  DELW=1.E-08
      IF (DELT.EQ.0.)  DELT=1.E-08
    2 ML(1)=M7END-3
      ML(2)=M7END-4
C SET UP BEAM SYSTEM ARRAY MI
      MBML = MDAT(M,2)
      NDAT = INFF(17,MBML)
      CALL MIFILL(MBML,1,NDAT,NELS,MI)
      IF (ERROR) RETURN
C OPTIONS ON ITERATIONS AND PRINT
C SET UP ARRAY FOR PREVIOUSLY DEFINED PARTICLE
C SET UP INITIAL REFERENCE RAY
C ZERO VW ARRAY
      DO 3 I=1,112
      VW(I) = 0.0
    3 CONTINUE
      NP = MDAT(M,1)
      CALL DATA (NP,1,1,7,VW)
      NV(1) = 1
   33 MB=1
      ME = NELS
      NITS = 0
      IF( ITERS.EQ.0 )ITERS = 25
C
      E(1) = EPSW
      E(2) = EPST
      E(3) = EPSW
      E(4) = EPST
      F(1) = DELW
      F(2) = DELT
      F(3) = DELW
      F(4) = DELT
      CALL HED
      NPM = INFF(2,NP)
      NAM = INFF(2,M)
      WRITE(3,1018)
      WRITE(3,1000) NAM,NPM,(VW(I),I=1,7)
      MM = 1
      IOP = IFLAG
      GO TO (31,32) NGO
   32 MN = 1
  391 CALL RTRV7(MUNIT,R7,RW)
      CALL STOR7(M,R7,RW)
      GO TO 4
   31 MN=9
      MR = 4
      NR = 1
```

```
      ASSIGN 4 TO NRS
      IF (IFLAG.EQ.1)  GO TO 800
      NR=5
      MN=1
      GO TO 802
   39 IF (IFLAG.NE.1) GO TO 802
      GO TO 800
C  SAVE INITIAL CONDITIONS OF THE REFERENCE RAY.
    4 DO 5  I=1,7
    5 VV(I) = VW(I)
      IF (IFLAG.EQ.4)  CALL DATA(M,7,1,7,V(1,8))
C  TRACK PARTICLES MM TO MN THROUGH BEAMLINE ELEMENTS MB TO ME.
      ASSIGN 510 TO TRKR
      DO 1 ISTR=1,NSTR
      CALL RANSET(NRN)
      GO TO 900
  510 CONTINUE
    1 CONTINUE
      GO TO (51,52) NGO
C CALCULATE THE 4X4 MATRIX, M.
   51 IF (IFLAG.EQ.1) GO TO 511
      DO 512 J=1,4
      DO 512 I=1,4
  512 RM(I,J) = V(I,J+1)
      GO TO 513
  511 DO 55 J = 1 , 4
      N = J + 1
      DN=.5/F(J)
      DO 55 I=1,4
   55 RM(I,J) = ( V(I,N) - V(I,N+4) ) * DN
  513 IF (KB.GE.0)  GO TO 56
      RX(1,1) = RM(2,1)
      RX(2,1) = RM(4,1)
      RX(1,2) = RM(2,3)
      RX(2,2) = RM(4,3)
      RX(1,3) = -V(2,1)
      RX(2,3) = -V(4,1)
      GO TO 56
   52 CALL RTRV7(M,R7,RW)
      IF (KB.GE.0) GO TO 57
      RX(1,1)=R7(2,1)
      RX(2,1)=R7(4,1)
      RX(1,2)=R7(2,3)
      RX(2,2)=R7(4,3)
      RX(1,3)=- V(2,1)
      RX(2,3)= -V(4,1)
   57 DO 53 J=1,4
      DO 53 I=1,4
   53 RM(I,J) = R7(I,J)
   56 CONTINUE
      IF (KB.GE.0) GO TO 58
      CALL MXDIV(RX,2,1,D,2)
      V(1,1)=RX(1,3)+VV(1)
      V(3,1)=RX(2,3)+VV(3)
      DX=V(2,1)
      DY=V(4,1)
      V(2,1)=0.
```

```
      V(4,1)=0.
      NITS = NITS + 1
      IF ( (ABS(DX).LT.E(2)).AND.(ABS(DY).LT.E(2)) ) GO TO 146
      IF (NITS.GT.ITERS) GO TO 146
      GO TO  (39,391) NGO
  146 GO TO (141,142) NGO
   58 CONTINUE
C  SET UP N - I AND REF RAY JUMP (RHS).
      DO 7 I = 1, 4
      DO 6 J = 1,4
    6 RMI(I,J) = RM(I,J)
    7 RMI(I,I) = RMI(I,I) - 1.0
      DO 61 I=1,4
 61   RMI(I,5) = VV(I)-V(I,1)
C  SOLVE FOR EQ.-ORBIT-INITIAL CONDITIONS ( IN TERMS OF REF. RAY ).
      CALL MXDIV( RMI, 4, 1, D, 4 )
      IF( D.NE.0. ) GO TO 75
      WRITE( 3, 1006 )
      WRITE (3,1001) ((RM(I,J),J=1,4),I=1,4)
 1001 FORMAT (/10X,4F16.8)
      CALL HED
      RETURN
C  CALCULATE EQ.-ORBIT-INITIAL-CONDITIONS.
 75   DO 8 I=1,4
 8    V(I,1) = RMI(I,5) + VV(I)
C  CHECK FOR CONVERGENCE.
      NITS = NITS + 1
      IF (KA.LT.2) GO TO 14
      WRITE (3,1013) NITS,(V(I,1),I=1,4)
 14   CONTINUE
      DO 9 I = 1, 4
      IF( ABS( RMI(I,5) ).LT.E(I) ) GO TO 9
      IF (NITS.GT.ITERS) GO TO 9
      GO TO (39,391) NGO
 9    CONTINUE
      GO TO (141,142) NGO
  141  CALL RTRV7(MUNIT,R7,RW)
      IOP = 1
      DO  200 J=1,4
      DO  200 I=1,4
  200 R7(I,J)=RM(I,J)
      MM = 10
      MN = 11
      NR=1
      IF (IFLAG.EQ.1) GO TO 138
      IF (IFLAG.EQ.4) GO TO 142
      MM = 6
      MN = 7
  138 DO 143 I=1,7
      V(I,MM) = VW(I)
  143 V(I,MN) = VW(I)
      V(6,MM) = VW(6) + DELP
      V(6,MN) = VW(6) - DELP
      ASSIGN 144 TO TRKR
      GO TO 900
  144 DO 145 I=1,5
  145 R7(I,6) = 0.5*(V(I,MM)-V(I,MN))/DELP
```

```
      ASSIGN 142 TO R5J
      GO TO 805
  142 IOP = IFLAG
      CALL ROW5(R7)
      IF (KB.GE.0) GO TO 149
      ML1=ML(1) $ ML2=ML(2)
      CALL STOR7(ML1,R7,RW)  $ CALL REF(ML2,ML1)
      CALL MMM(ML1,2,ML)  $   CALL RTRV7(ML1,R7,RW)
      DO 139 J=1,4
      DO 139 I=1,4
  139 RM(I,J) = R7(I,J)
  149 IF(NGO.EQ.2) GO TO 1400
      DO 151 I=1,7
  151 VV(I) = VW(I)
      CALL MXV7(R7,VV,VV)
      DO 152 I=1,5
  152 R7(I,7) = VW(I) - VV(I)
 1400 CONTINUE
      ML1 = ML(1) $  CALL STOR7(ML1,R7,RW)
C  CALCULATE OFF-MOMENTUM ORBIT VP AND STORE AT LQ3
  140 DO 148 I=1,4
      RMI(I,5)=-R7(I,6)
      DO 147 J=1,4
  147 RMI(I,J)=R7(I,J)
  148 RMI(I,I)=RMI(I,I)-1.0
      CALL MXDIV(RMI,4,1,D,4)
      DO 150 I=1,4
  150 VP(I)=RMI(I,5)
      CALL STDAT(M,8,1,7,VP)
C  REPLACE INITIAL CONDITIONS OF PARTICLE WITH COMPUTED VALUES
      IF ((IVSV.NE.0).AND.(IFLAG.NE.4))
     1    CALL STDAT(NP,1,1,4,VW)
C  STORE CLOSED ORBIT VECTOR AT LQ2
      VW(5)=0.
      IF (IFLAG.NE.4)  CALL STDAT(M,7,1,7,VW)
      WRITE( 3, 1003 ) NITS
      WRITE (3,1007) NAM, ( (R7(I,J),J=1,7) , I = 1,7)
C  CALCULATE THE EIGENVALUES OF M.
      TX = RM(1,1) + RM(2,2)
      TY = RM(3,3) + RM(4,4)
      CTR = CMPLX( TX + TY, 0. )
      CSQ = CSQRT( CMPLX( ( TX - TY )**2 + 4.*( ( RM(1,3) + RM(4,2) )
     1      *( RM(2,4) + RM(3,1) ) - ( RM(1,4) - RM(3,2) )
     2      *( RM(2,3) - RM(4,1) ) ), 0.0 ) )
      BLAM(1) = 0.5*( CTR + CSQ )
      BLAM(2) = 0.5*( CTR - CSQ )
      IF( TX.GE.TY ) GO TO 41
C  SWITCH BLAM(1) AND BLAM(2) IF TX.LT.TY .
      LAM(1) = BLAM(1)
      BLAM(1) = BLAM(2)
      BLAM(2) = LAM(1)
   41 LAM(1) = 0.5*( BLAM(1) + CSQRT( BLAM(1)*BLAM(1) - ( 4.0, 0.0 ) ) )
      LAM(2) = 1./LAM(1)
      LAM(3) = 0.5*( BLAM(2) + CSQRT( BLAM(2)*BLAM(2) - ( 4.0, 0.0 ) ) )
      LAM(4) = 1./LAM(3)
C  WRITE OUT THE EIGENVALUES...BOTH IN CARTESIAN AND POLAR FORMS.
      DO 42 I = 1, 4
```

```
      C(I) = REAL(CABS(LAM(I)))
      MU(I) = ATAN2( AIMAG( LAM(I) ), REAL( LAM(I) ) )
      Q(I) = SPER*MU(I)/TPI
      Q(I) = AMOD(Q(I),1.)
      IF (Q(I).LT.0.) Q(I)=Q(I)+1.
   42 CONTINUE
C  STORE EIGENVALUES AT LQ3+7
      CALL STDAT(M,8,8,8,LAM)
      WRITE( 3, 1004 )
   43 WRITE( 3, 1005 ) ( LABELA(I), LABELB(I), LAM(I), I, C(I), I,
     1                  MU(I), I, Q(I), I = 1, 4 )
      IF (KA.EQ.0) GO TO 93

C  COMPUTE EIGENVECTORS.
C  FIRST AND SECOND EIGENVECTORS.
C  FIRST COMPONENTS.
   35 DO 45 L=1,3
      DO 44 J = 1, 3
      RT = RM(L+1,J+1)
   44 CM(L,J) = CMPLX(RT,0.0)
   45 CM(L,L) = CM(L,L) - LAM(1)
      CALL DET( CM, X(1,1) )
C  SECOND THROUGH FOURTH COMPONENTS.
      D = -1.0
      DO 50 L = 2, 4
      I = L - 1
      DO 46 J = 1, 3
      RT = RM(J+1,I)
   46 CM(J,I) = CMPLX(RT,0.0)
      IF( L.EQ.3 ) CM(1,2) = CM(1,2) - LAM(1)
      IF( L.EQ.4 ) CM(2,3) = CM(2,3) - LAM(1)
      CALL DET( CM, X(L,1) )
      X(L,1) = D*X(L,1)
      D = -D
   50 CONTINUE
      C1 = CABS( X(1,1) )
      EX1 = X(1,1)/C1
      DO 536 I=1,4
  536 X(I,1) = X(I,1)/EX1
      RHO1 = AIMAG( CONJG(X(1,1))*X(2,1) + CONJG(X(3,1))*X(4,1) )
      FAC = 1.
      IF (RHO1.NE.0.) FAC = 1./SQRT(ABS(RHO1))
      DO 537 I=1,4
      X(I,1) = FAC * X(I,1)
  537 X(I,2) = CONJG(X(I,1))
      L = 1
      ASSIGN 467 TO IWR
      GO TO 998
  467 L = 2
      ASSIGN 468 TO IWR
      GO TO 998
C  THIRD AND FOURTH EIGENVECTORS.
C  FIRST COMPONENTS.
  468 DO 531 I = 1, 2
      DO 531 J = 2, 4
      RT = RM(I,J)
  531 CM(I,J-1) = CMPLX(RT,0.0)
```

```
      DO 532 J = 2, 4
      RT = RM(4,J)
  532 CM(3,J-1) = CMPLX(RT,0.0)
      CM(2,1) = CM(2,1) - LAM(3)
      CM(3,3) = CM(3,3) - LAM(3)
      CALL DET( CM, X(1,3) )
      X(1,4) = CONJG( X(1,3) )
C  SECOND THROUGH FOURTH COMPONENTS.
      D = -1.0
      DO 533 L = 2, 4
      I = L - 1
      CM(1,I) = CMPLX( RM(1,I), 0.0 )
      CM(2,I) = CMPLX( RM(2,I), 0.0 )
      CM(3,I) = CMPLX( RM(4,I), 0.0 )
      IF(L.EQ.2 ) CM(1,1) = CM(1,1) - LAM(3)
      IF( L.EQ.3 ) CM(2,2) = CM(2,2) - LAM(3)
      CALL DET( CM, X(L,3) )
      X(L,3) = D*X(L,3)
      D = -D
  533 CONTINUE
      C3 = CABS( X(3,3) )
      EX3 = X(3,3)/C3
      DO 546 I=1,4
  546 X(I,3) = X(I,3)/EX3
      RHO3 = AIMAG( CONJG(X(1,3))*X(2,3) + CONJG(X(3,3))*X(4,3) )
      FAC = 1.
      IF (RHO3.NE.0.) FAC = 1./SQRT(ABS(RHO3))
      DO 547 I=1,4
      X(I,3) = FAC * X(I,3)
  547 X(I,4) = CONJG(X(I,3))
      L = 3
      ASSIGN 534 TO IWR
      GO TO 998
  534 L = 4
      ASSIGN 535 TO IWR
      GO TO 998
C  1. TRACE EQUILIBRIUM ORBIT AND EIGENVECTOR PARTS THROUGH SYSTEM,
C  2. LINEARIZE INDIVIDUAL NLT"S.
  535 CONTINUE
C  SET UP EIGENVECTOR INITIAL CONDITIONS.
      DO 91 I = 1, 3, 2
      DO 91 J=1,4
      VZ(J,I) = REAL( X(J,I) )
      VZ(J,I+1) = AIMAG( X(J,I) )
  91  CONTINUE

C  STORE 4 EIGENVECTORS AT LQ2+7, 7 COMPONENTS EACH.
C  ORDER--- REAL X1, IMAG X1, REAL X3, IMAG X3.

      DO 92 I=1,4
      DO 92 J=5,7
  92  VZ(J,I) = 0.
      DO 97 I=5,7
      DO 97 J=1,7
  97  VZ(J,I) = 0.
      IQ=8
      DO 90 I=1,4
```

```
      CALL STDAT(M,7,IQ,7,VZ(1,I) )
      IQ=IQ+7
   90 CONTINUE
C  WRITE CLOSED AND OFF-MOMENTUM ORBITS AT POSITION  0
   93 WRITE(3,1017)
      WRITE(3,1014) LBL(5),(V(I,1),I=1,7)
      WRITE(3,1014) LBL(6),(VP(I),I=1,7)
      IF (KA.EQ.0)  GO TO 94
      WRITE(3, 1002 )
C  WRITE INITIAL CONDITIONS.
      L=0
      MATL = IBLNK
      ASSIGN 600 TO IBET
      GO TO 599
  600 IF (IFLAG.EQ.0) CALL STOR7(M,VZ,RW)
      IF (IFLAG.EQ.4)  CALL DATA(M,7,1,7,V(1,8))
   94 II = 0
      CALL RANSET(NRN)
  100 II = II + 1
      L = L + 1
      MB = II
      ME = II
C  SAVE INITIAL CONDITIONS.
      DO 101  J=1,7
  101 VV(J) = VW(J)
      IF (IFLAG.NE.4) GO TO 103
      DO 104 J=1,7
  104 VP(J) = V(J,8)
  103 ASSIGN 102 TO TRKR
      MM = 1
      MN = 1
      NR=1
      GO TO 900
  102 CONTINUE
      IF (KA.LT.3.OR.IFLAG.NE.0)  GO TO 604
      CALL RTRV7(M,VZ,RW)
      ASSIGN 604 TO IBET
      GO TO 599
  604 CONTINUE
      GO TO (95,40) NGO

   95 IF( NL.EQ.0 )  GO TO 40
C  LINEARIZE NON-LIN. TRANS
C  FORM 3X3 MATRICES
C  RESTORE INITIAL CONDITIONS.
      DO 111  J=1,7
      F(J) = VW(J)
  111 VW(J) = VV(J)
      IF (IFLAG.NE.4)  GO TO 105
      DO 106 J=1,7
  106 V(J,8) = VP(J)
  105 ASSIGN 112 TO NRS
      ASSIGN 116 TO TRKR
      MM=1
      MN=11
      JP=10
      JQ=11
```

```
      IF (IFLAG.EQ.1)  GO TO 800
      MN=1
      JP=6
      JQ=7
      NR=5
      IF (IFLAG.NE.4)  ASSIGN 114 TO TRKR
      GO TO 802
  112 DO 113 I=1,7
      V(I,JP) = VW(I)
  113 V(I,JQ) = VW(I)
      V(6,JP) = VW(6) + DELP
      V(6,JQ) = VW(6) - DELP
      GO TO 900
  114 MM=6
      MN=7
      NR=1
      IOP = 1
      ASSIGN 115 TO TRKR
      GO TO 900
  115 IOP = IFLAG
C  PUT LINEARIZED 3X3 MATRIX REPRESENTING AN EXT IN ITS STORAGE
  116 CALL RTRV7(MUNIT,R7,RW)
      IF (IFLAG.EQ.1)  GO TO 15
      DO 16 J=1,4
      DO 16 I=1,4
   16 R7(I,J) = V(I,J+1)
      GO TO (15,17,17,18) IFLAG
   15 D=DELW
      DO 11 J=1,4
      D=DELW*2.
      IF (J.EQ.2.OR.J.EQ.4)  D=DELT*2.
      DO 11 I=1,4
   11 R7(I,J) = ( V(I,J+1) - V(I,J+5) ) / D
   17 DO 12 I=1,4
   12 R7(I,6) = 0.5 * ( V(I,JP) - V(I,JQ) ) / DELP
      ASSIGN 18 TO R5J
      GO TO 805
   18 CALL MXV7(R7,VV,VV)
      DO 13 I=1,5
   13 R7(I,7) = VW(I) - VV(I)
      DO 19 I=1,7
   19 VW(I) = F(I)
      RW(1)=FDAT(NL,1)
      CALL STOR7(NL,R7,RW)
   40 IF (II.LT.NELS) GO TO 100

  402 CONTINUE

C  TRACK BETA FUNCTIONS, ETA AND CLOSED ORBITS THROUGH BEAM SYSTEM.
C    TRACKING DONE IN CYX, WHICH CALLS TRKCN FOR ORBIT TRACKING.
C    BETA FUNCTIONS FORMED FROM 2X2 SUBMATRICES ARE ONLY APPROXIMATE
C    IF THERE IS COUPLING.

C  TEST TRACE
      TRX = RM(1,1) + RM(2,2)
      TRY = RM(3,3) + RM(4,4)
      IF ( (ABS(TRX).LT.2.) .AND. (ABS(TRY).LT.2.) )  GO TO 405
```

```
      WRITE (3,1016)                                                    FXPT     552
 1016 FORMAT (5X,*X OR Y BETATRON MOTION UNSTABLE.*)                    FXPT     553
  405 CONTINUE                                                          FXPT     554
      IF (KB.GE.0) GO TO 406                                            FXPT     555
      NELS=NELS+1                                                       FXPT     556
      MI(NELS) = M7END - 4                                              FXPT     557
  406 CONTINUE                                                          FXPT     558
      ML1=ML(1) $ CALL RTRV7(ML1,R7,RW)                                 FXPTL      6
                                                                        FXPTL      7
      CALL STOR7(M,R7,RW)                                               FXPT     560
      IF (IFLAG.NE.4)  GO TO 409                                        FXPT     561
      CALL DATA(M,7,1,7,V(1,8))                                         FXPT     562
      CALL STDAT(M,7,1,7,VW)                                            FXPT     563
  409 CALL RANSET(NRN)                                                  FXPT     564
      CALL CYX(M)                                                       FXPT     565
      CALL RANSET(NRN)                                                  FXPT     566
      CALL HED                                                          FXPT     567
      IF (IFLAG.NE.3) RETURN                                            FXPT     568
      IF (KB.LT.0)  NELS=NELS-1                                         FXPT     569
      DO 408 I=1,6                                                      FXPT     570
  408 VW(I) = 0.                                                        FXPT     571
      IFLAG = 4                                                         FXPT     572
      NV(1)=8                                                           FXPT     573
      GO TO 33                                                          FXPT     574
                                                                        FXPT     575
C  SET UP THE EIGHT NEIGHBORING RAYS                                    FXPT     576
  800 DO 801 J=8,63                                                     FXPT     577
  801 VW(J)=VW(J-7)                                                     FXPT     578
      V(1,6)=V(1,6)-DELW                                                FXPT     579
      V(2,7)=V(2,7)-DELT                                                FXPT     580
      V(3,8)=V(3,8)-DELW                                                FXPT     581
      V(4,9)=V(4,9)-DELT                                                FXPT     582
      V(1,2) = V(1,2) + DELW                                            FXPT     583
      V(2,3) = V(2,3) + DELT                                            FXPT     584
      V(3,4) = V(3,4) + DELW                                            FXPT     585
      V(4,5) = V(4,5) + DELT                                            FXPT     586
      GO TO NRS, ( 4, 112 )                                             FXPT     587
  802 DO 803 J=8,35                                                     FXPT     588
  803 VW(J) = 0.                                                        FXPT     589
      DO 804 I=1,4                                                      FXPT     590
  804 V(I,I+1) = 1.                                                     FXPT     591
      CALL STOR7(M,V(1,2),RW)                                           FXPT     592
      GO TO NRS,(4,112)                                                 FXPT     593
  805 DO 806 J=1,4                                                      FXPT     594
  806 R7(5,J) = R7(2,J) * R7(1,6) - R7(1,J) * R7(2,6)                   FXPT     595
     1          + R7(4,J) * R7(3,6) - R7(3,J) * R7(4,6)                 FXPT     596
      GO TO R5J,(142,18)                                                FXPT     597
C  TRACK PARTICLES MM THROUGH MN THROUGH BEAM ELEMENTS MB THROUGH ME.   FXPT     598
  900 NLF = 0                                                           FXPT     599
      DO 950 J = MB, ME                                                 FXPT     600
      NL = MI(J)                                                        FXPT     601
      MATL = INFF(2,NL)                                                 FXPT     602
      DO 950  MT = MM,MN                                                FXPT     603
      CALL TRKCN( M, NL, V(1,MT), NP, IOP )                             FXPT     604
      KIND = INFF(20,NL)                                                FXPT     605
      IF (KIND.EQ.3HPRD.OR.KIND.EQ.3HMAP) NLF = NL                      FXPT     606
  950 CONTINUE                                                          FXPT     607
```

```
          NL = NLF
           GO TO TRKR,(102,114,115,116,144,510)
C  WRITE OUT AN EIGENVECTOR.
  998 IF (IOUT.NE.5H     ) WRITE (3,1008) LAM(L),(X(J,L),J=1,4)
          GO TO IWR, ( 467, 468, 534, 535 )
C
C  PRINT EIGENFUNCTIONS IN POLAR FORM
  599 CONTINUE
      DO 601 K=1,2
      DO 601 I=1,4
      J = 2 * K - 1
      IF ( (VZ(I,J).NE.0.) .OR. (VZ(I,J+1).NE.0.) )  GO TO 602
      EMOD(I,K) = CMPLX(0.,0.)
      EPHI(I,K) = CMPLX(0.,0.)
      GO TO 601
  602 EMOD(I,K) = SQRT ( VZ(I,J)*VZ(I,J) + VZ(I,J+1)*VZ(I,J+1) )
      EPHI(I,K) = ATAN2 ( VZ(I,J+1),VZ(I,J) )
  601 CONTINUE
      WRITE (3,1003) L,MATL, ( EMOD(I,1),EPHI(I,1),I=1,4 ) ,
     1                       ( EMOD(I,2),EPHI(I,2),I=1,4 )
      GO TO IBET,(600,604)

 1000 FORMAT(64H0CALCULATION OF THE EQUILIBRIUM ORBIT AND BETATRON FUNCT
     1IONS OF A5,1H./34H INITIAL REFERENCE RAY DEFINED BY A5/7H    X = ,
     2  F11.8,3X,*DX = *,F11.8,3X,*Y = *,F11.8,3X,*DY = *,F11.8,
     3 5X,*DS = *,F11.8,3X,*DP/P = *,F11.8,3X,F11.8)
 1002 FORMAT ( 1H1,*EIGENVECTORS 1 AND 3 IN POLAR COORDINATES*/
     1  1X*POS*23X*X1 *22X*DX1*22X*Y1 *22X*DY1*/
     2        27X*X3 *22X*DX3*22X*Y3 *22X*DY3*/ )
 1003 FORMAT ( 1X,I4,1X,A5,1X, 4(2F11.6,3X) / 12X, 4(2F11.6,3X) / )
 1004 FORMAT (*0EIGENVALUES OF THE 4X4 SUBMATRIX*)
 1005 FORMAT(2A6,4H = (,2F12.8,5X,6H ), C(I1,4H) = F12.8,5H, MU(I1,
     1  4H) = F12.8,12H RAD,     Q(I1,4H) = F12.8 )
 1006 FORMAT(21H0SINGULAR 4X4 MATRIX.)
 1007 FORMAT (*07X7 MATRIX FOR *,A5/(/10X,7F16.8))
 1008 FORMAT(16H0EIGENVALUE = ( F16.8,2H, F16.8,21H ),  EIGENVECTOR = (
     1F16.8,2H, F16.8,2H )/(69X,2H( F16.8,2H, F16.8,2H )))
 1013 FORMAT(12H ITERATION =,I4,4H XO=,F16.8,5H DXO=,F16.8,4H YO=,F16.8,
     15H DYO=,F16.8)
 1014 FORMAT (/1X,A10,7F15.8)
 1018 FORMAT ( 1H1)
 1017 FORMAT(/15X,7X,* X*,13X,*DX*,13X,* Y*,13X,*DY*,13X,*DS*,13X,*DP/P*
     1 )
C
      END
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
EQV/COMM	I	BMI1	NOT ALL ITEMS IN THIS COMMON BLOCK OCCUR IN LEVEL STATEMENTS.
164	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
190	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
233	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
234	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
262	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
264	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
496	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

```
  1           SUBROUTINE GENNUM(NM,KC,LIST )                                      GENNUM    2
                                                                                  GENNUM    3
        *     GENERATE NUMBERED NAMES, E.G., N1,N2,N3,...NKC IN BCD WHERE         GENNUM    4
        *  N IS FIRST LETER OF NM.  PUT NAMES INTO ARRAY LIST.                    GENNUM    5
  5           DIMENSION LIST(1)                                                   GENNUM    6
                                                                                  GENNUM    7
        *                    STRIP OFF FIRST LETTER OF NM                         GENNUM    8
              DECODE (10,1,NM)  N1                                                GENNUM    9
        1     FORMAT (A1,9X)                                                      GENNUM   10
 10           IJ = 1                                                              GENNUM   11
              K1 = 27                                                             GENNUM   12
              K2 = 28                                                             GENNUM   13
              K3 = 28                                                             GENNUM   14
              DO 100 JA = 1,KC                                                    GENNUM   15
 15           K1 = K1 + 1                                                         GENNUM   16
              GO TO (25,30,35) IJ                                                 GENNUM   17
        25    ENCODE (10,2,LIST(JA)) N1,K1                                        GENNUM   18
        2     FORMAT (A1,R1,8X)                                                   GENNUM   19
              IF (K1.NE.36) GO TO 100                                             GENNUM   20
 20     26    IJ = IJ + 1                                                         GENNUM   21
        27    K1 = 26                                                             GENNUM   22
              GO TO 100                                                           GENNUM   23
        30    ENCODE (10,3,LIST(JA)) N1,K2,K1                                     GENNUM   24
        3     FORMAT (A1,2R1,7X)                                                  GENNUM   25
 25           IF (K1.NE.36) GO TO 100                                             GENNUM   26
              K2 = K2 + 1                                                         GENNUM   27
              IF (K2.NE.37) GO TO 27                                              GENNUM   28
              K2 = 27                                                             GENNUM   29
              GO TO 26                                                            GENNUM   30
 30     35    ENCODE (10,4,LIST(JA)) N1,K3,K2,K1                                  GENNUM   31
        4     FORMAT (A1,3R1,6X)                                                  GENNUM   32
              IF (K1.NE.36) GO TO 100                                             GENNUM   33
              K2 = K2 + 1                                                         GENNUM   34
              IF (K2.NE.37) GO TO 27                                              GENNUM   35
 35           K2 = 27                                                             GENNUM   36
              K3 = K3 + 1                                                         GENNUM   37
              IF (K3.EQ.37) GO TO 105                                             GENNUM   38
              GO TO 27                                                            GENNUM   39
        100   CONTINUE                                                            GENNUM   40
 40           RETURN                                                              GENNUM   41
        105   PRINT 5                                                             GENNUM   42
        5     FORMAT (5X,*CANNOT GENERATE NUMBERS GREATER THAN 999.*)             GENNUM   43
              RETURN                                                              GENNUM   44
              END                                                                 GENNUM   45
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
16	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

```
      SUBROUTINE GRT( N, C, IN, AUX )                                   GRT        2
                                                                        GRT        3
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1          LDFLG,FIN                                               CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      DIMENSION C(5)                                                    GRT        5
      COMMON/GRR/IERR,BSW,VSW,RSW,QSW,XSW                               GRR        2
      LOGICAL IERR,BSW,VSW,RSW,QSW,XSW                                  GRR        3
                                                                        GRR        4
      LOGICAL IND,INF                                                   GRT        7
      LOGICAL INDEFF,INFINN                                             GRT        8
      DATA FC /1.0E-10/                                                 GRT        9
                                                                        GRT       10
      DO 100 L=1,N                                                      GRT       11
      IERR=.FALSE.                                                      GRT       12
      JK=0                                                              GRT       13
      IF (C(L))45,46,45                                                 GRT       14
 45   RT=.9*C(L)                                                        GRT       15
      ASSIGN 1 TO NN                                                    GRT       16
      GO TO 80                                                          GRT       17
 1    X0=FPRT                                                           GRT       18
      RT=1.1*C(L)                                                       GRT       19
      ASSIGN 2 TO NN                                                    GRT       20
      GO TO 80                                                          GRT       21
 2    X1=FPRT                                                           GRT       22
      RT=C(L)                                                           GRT       23
      ASSIGN 3 TO NN                                                    GRT       24
      GO TO 80                                                          GRT       25
 3    X2=FPRT                                                           GRT       26
      H=-.1*C(L)                                                        GRT       27
      GO TO 50                                                          GRT       28
 46   RT=-1.                                                            GRT       29
      ASSIGN 4 TO NN                                                    GRT       30
      GO TO 80                                                          GRT       31
 4    X0=FPRT                                                           GRT       32
      RT=1.                                                             GRT       33
      ASSIGN 5 TO NN                                                    GRT       34
      GO TO 80                                                          GRT       35
 5    X1=FPRT                                                           GRT       36
      RT=0.                                                             GRT       37
      ASSIGN 6 TO NN                                                    GRT       38
      GO TO 80                                                          GRT       39
 6    X2=FPRT                                                           GRT       40
      H=-1.                                                             GRT       41
   50 D=-.5                                                             GRT       42
 49   DD=1.+D                                                           GRT       43
      BI=(X0*D*D)-(X1*DD*DD)+(X2*(DD+D))                                GRT       44
      DEN=BI*BI-(4.*X2*D*DD)*(X0*D-(X1*DD)+X2)                          GRT       45
      IF (DEN)52,52,51                                                  GRT       46
 52   IF (IN)36,35,36                                                   GRT       47
   35 WRITE (3,25) L,DEN                                                GRT       48
 36   DEN=0.                                                            GRT       49
   51 DEN=SQRT(DEN)                                                     GRT       50
 53   DN=BI+DEN                                                         GRT       51
      DM=BI-DEN                                                         GRT       52
      IF (ABS(DN)-ABS(DM)) 57,57,56                                     GRT       53
```

```
   56   DEN=DN
        GO TO 58
   57   DEN=DM
   58   IF (DEN)55,54,55
   54   DEN=1.
   55   DI=(-2.*X2*DD)/DEN
        H=DI*H
        RT=RT+H
        IF (ABS(H/RT)-1.0E-8) 75,75,60
   60   ASSIGN 7 TO NN
        GO TO 80
      7 IF (ABS(FPRT)-ABS(X2*10.)) 62,61,61
   61   DI=DI*.5
        H=H*.5
        RT=RT-H
        GO TO 80
   62   X0=X1
        X1=X2
        X2=FPRT
        D=DI
        GO TO 49
   75   CALL AUX (RT,FRT)
        IF (ERROR) RETURN
        IF(JK.LT.150) GO TO 76
        IERR=.TRUE.
   74   WRITE (3,28) RT
   28   FORMAT (55H GRT UNABLE TO FIND ROOT IN 150 ITERATIONS.LAST ROOT IS
       1, E20.8)
   76   IND= INDEFF(RT)
        INF= INFINN(RT)
        IF(IND.OR.INF) IERR=.TRUE.
   78   IF(IERR) WRITE (3,29)
   29   FORMAT (32H ROOT IS INDEFINITE OR INFINITE.//)
        C(L)=RT
        IF (IN)100,77,100
     77 WRITE (3,26) L,RT,FRT
  100   CONTINUE
   33   RETURN
   80   JK=JK+1
        IF(150-JK) 75,75,86
   86   CALL AUX (RT,FRT)
        IF (ERROR) RETURN
        IND= INDEFF(RT)
        INF= INFINN(RT)
        IF(IND.OR.INF) IERR=.TRUE.
        IF(IERR) GO TO 78
        FPRT=FRT
        IF (L-1)81,91,81
   81   DO 82 I=2,L
        TEM=RT-C(I-1)
        IF (ABS(TEM)-FC) 85,82,82
   82   FPRT=FPRT/TEM
   91   IF (IN) 89,83,89
     83 WRITE (3,25) L,RT,FRT,FPRT
     89 IF (ABS(FRT)-FC) 90,84,84
     90 IF (ABS(FPRT)-FC) 76,84,84
   84   GO TO NN,(1,2,3,4,5,6,7)
```

```
      85   RT=RT+.001                                              GRT   111
           IF (IN) 88,87,88                                        GRT   112
        87 WRITE (3,25) L                                          GRT   113
      88   GO TO 80                                                GRT   114
      25   FORMAT (I3,3E20.8)                                      GRT   115
      26   FORMAT (I3,2E20.8/1H )                                  GRT   116
      27   FORMAT (//)                                             GRT   117
           END                                                     GRT   118
```

```
      SUBROUTINE GRTB(N,C,IN,AUX)                                       GRTB       2
      RETURN                                                            GRTB       3
      END                                                               GRTB       4
```

```
      SUBROUTINE HED                                                    HED       2
C  PRINTS A LINE OF DOTS.                                               HED       3
      WRITE (3,1000)                                                    HED       4
      RETURN                                                            HED       5
 1000 FORMAT (1X,130(1H.))                                              HED       6
      END                                                               HED       7
```

```
                  FUNCTION IDAT(M,J)                                       IDAT         2
            C  RETRIEVES ONE PIECE OF INTEGER DATA FROM INDEX J OF ELEMENT M    IDAT         3
                                                                           BLANK        2
                  LEVEL 2, STORE,INFF,IWORK                                BLANK        3
                  COMMON STORE(48000),IWORK(10)                            86MARSIZ     1
                  DIMENSION INFF(24,2000)                                  86MARSIZ     2
                  EQUIVALENCE (INFF,STORE)                                 86MARSIZ     3
                                                                           BLANK        5
                  COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,  CONTRL       2
                 1          LDFLG,FIN                                      CONTRL       3
                  LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN    CONTRL       4
                                                                           CONTRL       5
                  EQUIVALENCE(IDATA,FDATA)                                 IDAT         6
                                                                           IDAT         7
                  LOC=INFF(12,M)+J-1                                       IDAT         8
                  IF (LOC.GT.0)  GO TO 1                                   IDAT         9
                  ERROR = .TRUE.                                           IDAT        10
                  RETURN                                                   IDAT        11
              1   FDATA = STORE(LOC)                                       IDAT        12
                  IDAT=IDATA                                               IDAT        13
                  RETURN                                                   IDAT        14
                  END                                                      IDAT        15
```

```
      SUBROUTINE INCR(M,KA)                                               INCR       2
*  CARD IMAGE     INCR    K     // Q    TYPE    DELTA                     INCR       3
*  INCREMENT THE KTH PARAMETER OF Q BY DELTA                              INCR       4
*  TYPE MAY BE F, I, KA OR KB.                                            INCR       5
*  DELTA WILL BE FL. PT. IF TYPE IS F OR OMITTED.                         INCR       6
*   IT WILL BE INTEGER IF I, KA OR KB.                                    INCR       7
C    IF MODE=1,THE NEW VALUE REMAINS IN DATA STORAGE                      INCR       8
C    IF MODE=3,THE ORIGINAL VALUE IS REPLACED AFTER REQUIRED ITERATIONS   INCR       9
C     HAVE BEEN COMPLETED.                                                INCR      10
                                                                          BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                           BLANK      3
      COMMON STORE(48000),IWORK(10)                                       86MARSIZ   1
      DIMENSION INFF(24,2000)                                             86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                            86MARSIZ   3
                                                                          BLANK      5
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,      CONTRL     2
     1          LDFLG,FIN                                                 CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN        CONTRL     4
                                                                          CONTRL     5
      INTEGER BDAT                                                        INCR      13
      INTEGER ELNUM                                                       INCR      14
C                                                                         INCR      15
*   FIND NAME AND INDEX OF ELEMENT WHOSE DATA IS TO BE INCREMENTED.       INCR      16
*   DETERMINE TYPE OF DATA TO BE CHANGED -- FL. PT., INTEGER, KA OR KB.   INCR      17
      NM = BDAT(M,1)                                                      INCR      18
      N = ELNUM(NM)                                                       INCR      19
      ITP = BDAT(M,2)                                                     INCR      20
      IF ((ITP.EQ.2HF ).OR.(ITP.EQ.2H  ))IGO = 1                          INCR      21
      IF (ITP.EQ.2HI ) IGO = 2                                            INCR      22
      IF (ITP.EQ.2HKA) IGO = 3                                            INCR      23
      IF (ITP.EQ.2HKB) IGO = 4                                            INCR      24
*        IPP NOT EQUAL TO 0, SUPPRESSES PRINT                             INCR      25
      IPP = INFF(6,N)                                                     INCR      26
      IF (MODE.EQ.1) GO TO 30                                             INCR      27
*   MS IS M OF THE SUBROUTINE THAT INCR IS IN.                            INCR      28
      MS = INFF(19,M)                                                     INCR      29
*  DETERMINE IF THIS IS FIRST AND/OR LAST ITERATION THROUGH SUBROUTINE.   INCR      30
      IF (IDAT(MS,1).EQ.1) GO TO 50                                       INCR      31
 20   IF (IDAT(MS,2).EQ.1) GO TO 60                                       INCR      32
*                  INCREMENT PARAMETER, RETURN NEW VALUE TO DATA STORE    INCR      33
 30   CONTINUE                                                            INCR      34
      GO TO (31,32,33,34) IGO                                             INCR      35
*  FLOATING POINT                                                         INCR      36
 31   DF = FDAT(M,1)                                                      INCR      37
      CALL AFD(N,KA,DF)                                                   INCR      38
      F = FDAT(N,KA)                                                      INCR      39
      GO TO 40                                                            INCR      40
*                  INTEGER                                                INCR      41
 32   IDF = IDAT(M,1)                                                     INCR      42
      INDAT=IDAT(N,KA)                                                    INCR      43
      INEW = IDF + INDAT                                                  INCR      44
      CALL REPINT(N,KA,INEW)                                              INCR      45
      GO TO 40                                                            INCR      46
*        INCREMENT KA OR KB                                               INCR      47
 33   JI = 4                                                              INCR      48
      GO TO 35                                                            INCR      49
 34   JI = 5                                                              INCR      50
```

```
   35  IDF = IDAT(M,1)
       INDAT = INFF(JI,N)
       INEW = IDF + INDAT
       INFF(JI,N) = INEW
   40  KO = KA
       IF (IPP.GT.0) RETURN
       GO TO (41,42,43,44) IGO
   41  WRITE (3,1) KO,NM,DF,F
       RETURN
   42  WRITE (3,2) KO,NM,IDF,INEW
       RETURN
   43  WRITE (3,3) NM,IDF,INEW
       RETURN
   44  WRITE (3,4) NM,IDF,INEW
       RETURN
  *        SAVE ORIGINAL PARAMETER
   50  GO TO (51,52,53,54) IGO
   51  LOC = INFF(14,N) + KA - 1
       SF = STORE(LOC)
       CALL REPFLT(M,2,SF)
       GO TO 20
   52  ISF = IDAT(N,KA)
       GO TO 55
   53  ISF = INFF(4,N)
       GO TO 55
   54  ISF = INFF(5,N)
   55  CALL REPINT(M,2,ISF)
       GO TO 20
  *        ITERATIONS COMPLETED.  RESTORE ORIGINAL PARAMETER.
   60  GO TO (61,62,62,62) IGO
   61  LOC = INFF(14,M) + 2 - 1
       SF = STORE(LOC)
       CALL REPFLT(N,KA,SF)
       F = FDAT(M,2)
       DF = 0.
       GO TO 40
   62  ISF = IDAT(M,2)
       GO TO (63,63,64,65) IGO
  * SHOULD NOT GET HERE
   63  WRITE (3,5) IGO
       ERROR = .TRUE.
       RETURN
   64  JI = 4
       GO TO 66
   65  JI = 5
   66  INFF(JI,N) = ISF
       INEW = ISF
       IDF = 0
       GO TO 40
  ***********************************FORMATS******************************
   1   FORMAT (6H  *** ,8X,5HINCR ,1X,I3,4X,5H //  ,A5,5X,F10.6,5X,
      1 9H VALUE = ,F14.6)
   2   FORMAT (6H  *** ,8X,5HINCR ,1X,I3,4X,5H //  ,A5,2X,I10,5X,
      1 9H VALUE = ,I10)
   3   FORMAT (6H  *** ,8X,5HINCR ,2X,2HKA,4X,5H //  ,A5,5X,I10,5X,
      1  9H VALUE = ,I10)
   4   FORMAT (6H  *** ,8X,5HINCR ,2X,2HKB,4X,5H //  ,A5,5X,I10,5X,
```

```
          1  9H VALUE = ,I10)                                         INCR     108
    5      FORMAT (6H  *** ,*ERROR IN INCR ROUTINE.*, I10,*IS INVALID COMPUTE INCR     109
          1ED GO TO.*)                                                INCR     110
           END                                                        INCR     111
```

STORAGE ALLOCATION.

```
          ADDRESS   LENGTH               BINARY CONTROL CARDS.

                0        5               IDENT INDEFF
                5                        END


                              ENTRY POINTS.

                              INDEFF           1+




                                         IDENT INDEFF
                               *         FUNCTION INDEFF(X)
                                         ENTRY INDEFF
     0  11160405060600000001             VFD  42/0LINDEFF,18/1
     1                          INDEFF   BSS 0
     1                      1   ENT      BSSZ 1
     2  63110                            SB1 X1
             56110                       SA1 B1
                  43600                  MX6 0
     3  0361000001 +                     DF X1,ENT
                  43674                  MX6 60
     4  0400000001 +                     EQ B0,B0,ENT
     5                                   END

        16100B LCM  31300B SCM STORAGE USED         13 STATEMENTS        2 SYMBOLS
                     7600-TYPE CPU ASSEMBLY       0.010 SECONDS          5 REFERENCES
```

SYMBOLIC REFERENCE TABLE.

```
ENT              1    PROGRAM*      157/19 L   157/23    157/25
INDEFF           1    PROGRAM*      157/16 E   157/18 L
```

```
STORAGE ALLOCATION.

          ADDRESS   LENGTH            BINARY CONTROL CARDS.

                0        5            IDENT INFINN
                5                     END


                                ENTRY POINTS.

                                INFINN          1+




                                      IDENT INFINN                                       I
                               *      FUNCTION INFINN(X)                                 I
                                      ENTRY INFINN                                       I
     0  11160611161600000001          VFD 42/0LINFINN,18/1                               I
     1                          INFINN  BSS 0                                            I
     1                      1   ENT   BSSZ 1                                             I
     2  63110                         SB1 X1                                             I
            56110                     SA1 B1                                             I
                 43600                MX6 0                                              I
     3  0341000001 +                  IR X1,ENT                                          I
                 43674                MX6 60                                             I
     4  0400000001 +                  EQ B0,B0,ENT                                       I
     5                                END                                                I

        16100B LCM  31300B SCM STORAGE USED          13 STATEMENTS        2 SYMBOLS
                    7600-TYPE CPU ASSEMBLY         0.010 SECONDS         5 REFERENCES


SYMBOLIC REFERENCE TABLE.

ENT              1   PROGRAM*      158/19 L  158/23    158/25
INFINN           1   PROGRAM*      158/16 E  158/18 L
```

```
      SUBROUTINE INFW(IA,J,M)                                    INFW        2
C  WRITES ONE INTEGER WORD FROM I INTO INFF(J,M)                 INFW        3
                                                                 BLANK       2
      LEVEL 2, STORE,INFF,IWORK                                  BLANK       3
      COMMON STORE(48000),IWORK(10)                              86MARSIZ    1
      DIMENSION INFF(24,2000)                                    86MARSIZ    2
      EQUIVALENCE (INFF,STORE)                                   86MARSIZ    3
                                                                 BLANK       5
      INFF(J,M) = IA                                             INFW        5
      RETURN                                                     INFW        6
      END                                                        INFW        7
```

```
      SUBROUTINE INP(M,KIND,NQ,                                          INP       2
     . N1,I1,N2,I2,N3,I3,N4,I4                                           INP       3
     .)                                                                  INP       4
C  RESERVES AND STORES INPUT                                             INP       5
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,     CONTRL    2
     1          LDFLG,FIN                                                CONTRL    3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN       CONTRL    4
                                                                         CONTRL    5
      IF(RSRV) CALL RESRV(M,KIND,NQ,                                     INP       7
     . N1,I1,N2,I2,N3,I3,N4,I4                                           INP       8
     .)                                                                  INP       9
      IF(ERROR) RETURN                                                   INP      10
      IF(STOR) CALL LOAD(M,                                              INP      11
     . N1,I1,N2,I2,N3,I3,N4,I4                                           INP      12
     .)                                                                  INP      13
      RETURN                                                             INP      14
      END                                                                INP      15
```

```
      SUBROUTINE INTCON(N,NA)

*  NA IS AN ARRAY OF LENGTH N.  IT CONTAINS NO MORE THAN 4 BCD
*  CHARACTERS, REPRESENTING AN INTEGER WITHIN PARENTHESES.
*    ROUTINE EXTRACTS THE INTEGER AND RETURNS IT IN THE ARRAY IN
*    INTEGER FORM.

      DIMENSION NA(1)

      DO 100 I=1,N
         DECODE (10,1,NA(I)) NB
 1       FORMAT (3X,A1,6X)
         IF (NB.EQ.1H)) GO TO 50
*                  ASSUMES 1 DIGIT INTEGER
         DECODE (10,2,NA(I)) NB
 2       FORMAT (1X,I1,8X)
         GO TO 70
*                  ASSUMES 2 DIGIT INTEGER
 50      CONTINUE
         DECODE (10,3,NA(I)) NB
 3       FORMAT (1X,I2,7X)
 70      NA(I) = NB
 100  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INV(M,N)                                               INV        2
C RETRIEVES RX AND RY MATRICES FROM LQ(N), INVERTS THEM AND STORES THEM INV        3
C  IN LQ(M).  DIMENSIONS ARE (2,3) FOR EACH                             INV        4
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER   SWTCH      2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                          SWTCH      3
                                                                        SWTCH      4
      DIMENSION T(49),RW(3),RX(2,3),RY(2,3),RPQ(4)                      INV        7
      K1 = INFF(20,M)                                                   INV        8
C  REVERSE SIGN FOR INVERSE RETRIEVE                                    INV        9
      NN = - N                                                          INV       10
*        INVERT MATRIX AS 7X7, THEN STORE ACCORDING TO KIND             INV       11
 10   CALL RTRV7(NN,T,RW)                                               INV       12
      IF(K1.EQ.2HR7.OR.K1.EQ.3HROT) GO TO 15                            INV       13
      KN = 3                                                            INV       14
      CALL CONVMAT(KN,T,RX,RY,RW,RPQ)                                   INV       15
      CALL STXY(M,RX,RY,RW)                                             INV       16
      CALL STDAT(M,5,4,4,RPQ)                                           INV       17
      RETURN                                                            INV       18
 15   CALL STOR7(M,T,RW)                                                INV       19
      RETURN                                                            INV       20
      END                                                               INV       21
```

```
      SUBROUTINE INV2(M,MB)                                             INV2      2
                                                                        INV2      3
*  ROTATE MATRIX MB BY 180 DEGREES, REFLECT IT AND STORE IN M.          INV2      4
*  M AND MB ARE INDECES OF MATRICES                                     INV2      5
                                                                        INV2      6
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL   DIM       2
                                                                        DIM       3
      DATA PI/3.1415926536/                                             INV2      8
                                                                        INV2      9
      CALL ROTM(M7END,MB,PI)                                            INV2     10
      NN = M7END                                                        INV2     11
      CALL REF(M,NN)                                                    INV2     12
      RETURN                                                            INV2     13
      END                                                               INV2     14
```

```
      SUBROUTINE KICK(M)                                                KICK       2
C H    KICK   K  // C          0.        BRHO      BK                   KICK       3
C  BK = KICKER FIELD                                                    KICK       4
C  K = 1,2 FOR FIELD IN 1,2 (X,Y) DIRECTION                             KICK       5
C  C = PREVIOSLY DEFINED DRF OR MAG                                     KICK       6
C  IF C IS A DRF OF ZERO LENGTH, BK = KICKER LENGTH X FIELD             KICK       7
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      DIMENSION T(7,7),RW(3)                                            KICK       9
      INTEGER OPNAME                                                    KICK      10
      N=MDAT(M,1)                                                       KICK      11
      CALL RTRV7(N,T,RW)                                                KICK      12
      BK=FDAT(M,3)                                                      KICK      13
       BKK=BK                                                           KICK      14
C IF(KB.GE.1) ERROR IS RANDOM WITH GIVEN RANGE                          KICK      15
      KB=INFF(5,M)                                                      KICK      16
      IF(KB.GE.1) BK=BKK*(RANF(D)-.5)                                   KICK      17
      IF (BK.EQ.0.) GO TO 4                                             KICK      18
      K=INFF(4,M)                                                       KICK      19
      OPNAME=INFF(1,N)                                                  KICK      20
      IF (OPNAME.NE.3HDRF) GO TO 3                                      KICK      21
 2    S=FDAT(N,1)                                                       KICK      22
      BRHO=FDAT(M,2)                                                    KICK      23
      IF (S.NE.0.) GO TO 5                                              KICK      24
      RHO=0.                                                            KICK      25
      THETA=BK/BRHO                                                     KICK      26
      XK=0.                                                             KICK      27
      GO TO 6                                                           KICK      28
 5    CONTINUE                                                          KICK      29
      RHO=BRHO/BK                                                       KICK      30
      THETA=S/RHO                                                       KICK      31
      XK = -RHO*THETA*THETA/2.                                          KICK      32
 6    DXK = -THETA                                                      KICK      33
      T(5,7)=XK*THETA/3.                                                KICK      34
      T(5,6)=-2.*T(5,7)                                                 KICK      35
      RW(3)=T(5,6)                                                      KICK      36
      IF (K.NE.1) GO TO 1                                               KICK      37
      T(3,6)=XK                                                         KICK      38
      T(3,7)=-XK                                                        KICK      39
      T(4,6)=DXK                                                        KICK      40
      T(4,7)=-DXK                                                       KICK      41
      T(5,3)=-DXK                                                       KICK      42
      T(5,4)=-XK                                                        KICK      43
      GO TO 4                                                           KICK      44
 1    T(1,7)=XK                                                         KICK      45
      T(1,6)=-XK                                                        KICK      46
      T(2,6)=-DXK                                                       KICK      47
      T(2,7)=DXK                                                        KICK      48
      T(5,1)=DXK                                                        KICK      49
      T(5,2)=XK                                                         KICK      50
      RW(2)=THETA                                                       KICK      51
      GO TO 4                                                           KICK      52
 3    B0=FDAT(N,4)                                                      KICK      53
```

```
      IF ( (B0.EQ.0.).OR.(K.EQ.1) ) GO TO 2
      DBB=BK/B0
      T(1,7)=-T(1,6)*DBB
      T(2,7)=-T(2,6)*DBB
    4 CALL STOR7(M,T,RW)
      BK=BKK
      RETURN
      END
```

```
 1              SUBROUTINE KIN(PAR)                                                  KIN       2
                COMMON/KINET/KNFLAG,TK,P,GAM,BETT,BETGAM,BRHO                        KINET     2
                LOGICAL KNFLAG                                                       KINET     3
                LOGICAL KNFLAG                                                       KIN       4
 5              DIMENSION PAR(3)                                                     KIN       5
                KNFLAG = .TRUE.                                                      KIN       6
                EK = PAR(1)                                                          KIN       7
                PK = PAR(2)                                                          KIN       8
                BETRO = PAR(3)                                                       KIN       9
10              IF (EK.EQ.0.)  GO TO 5                                               KIN      10
                GAMA = 1. + (EK/E0P)                                                 KIN      11
                BGAMA = SQRT (GAMA*GAMA -1.)                                         KIN      12
                PK = E0P*BGAMA                                                       KIN      13
                GO TO 10                                                             KIN      14
15        C                                                                          KIN      15
            5   IF (PK.EQ.0.)  PK = BETRO/33.356405                                  KIN      16
                BGAMA = PK/E0P                                                       KIN      17
                GAMA = SQRT (BGAMA*BGAMA + 1.)                                       KIN      18
           10   BETK = BGAMA/GAMA                                                    KIN      19
20        C                                                                          KIN      20
                PRINT 20,E0P,PK,GAMA,BGAMA,EK,BETRO,BETK                             KIN      21
           20   FORMAT (/5X,*EO =*,F12.8,2X,*P   =*,F12.8,2X,*GAMMA=*,F12.8,          KIN      22
              1  2X,*BETGAM=*,F12.8/5X,*EK =*,F12.8,2X,*BRHO=*,F12.8,2X,             KIN      23
              2  *BETA=*,F12.8//)                                                    KIN      24
25              RETURN                                                               KIN      25
                END                                                                  KIN      26
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
4	I	KNFLAG	A TYPE WAS DECLARED PREVIOUSLY FOR THIS VARIABLE OR FUNCTION. THIS DECLARATION IGN

```
      SUBROUTINE KINEM(KP,T,P)                                          KINEM      2
C  KP IS TYPE OF PARTICLE, T IS KE, P IS BETGAM                         KINEM      3
      DIMENSION ER(2)                                                   KINEM      4
      DATA ER(1),ER(2)/.511006,938.256/                                 KINEM      5
      E = ER(KP) + T                                                    KINEM      6
      GAM = E/ER(KP)                                                    KINEM      7
      P = SQRT(GAM**2-1.)                                               KINEM      8
      BET = P/GAM                                                       KINEM      9
      RETURN                                                            KINEM     10
      END                                                               KINEM     11
```

```
      SUBROUTINE LINMIN(ITEST,X,F,MAXFUN,ABSACC,RELACC,XSTEP)           LINMIN     2
      RETURN                                                            LINMIN     3
      END                                                               LINMIN     4
```

```
      SUBROUTINE LOAD(M0,                                               LOAD       2
     . N1,I1,N2,I2,N3,I3,N4,I4                                          LOAD       3
     .)                                                                 LOAD       4
C                                                                       LOAD       5
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/INSTR/OPNAME,NAME,OP,KA,KB,OBJA,OBJB,NXTM,TRA,LQ,NQ,LIN,   INSTR      2
     1             NIN,LFL,NFL,LBC,NBC,NTOT,IX,KIND,TYPE,ROWS,COLS,EXTR, INSTR      3
     2             M,NF,NB,NI,MSUBR,ISIGN                               INSTR      4
      INTEGER OPNAME,OP,OBJA,OBJB,TRA,TYPE,ROWS,COLS,EXTR               INSTR      5
                                                                        INSTR      6
      COMMON/STORE/LMAX,LINF,LFILE,MAX,MIN,ICARD(11)                    STORE      2
*  LMAX AND MAX ARE DIMENSIONS OF STORE AND INFF                        STORE      3
                                                                        STORE      4
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1          LDFLG,FIN                                               CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      COMMON/FLTN/IFL(15)                                               FLTN       2
                                                                        FLTN       3
      COMMON/COPY/CPYSW                                                 COPY       2
      LOGICAL CPYSW                                                     COPY       3
                                                                        COPY       4
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL   DIM        2
                                                                        DIM        3
                                                                        LOAD      13
      DIMENSION IFCD(10)                                                LOAD      14
      INTEGER BDAT                                                      LOAD      15
      DIMENSION  IF(6),IB(12),FCD( 6 ),BCD( 12),ICD(12 ),ND(10),IT(10), LOAD      16
     1 II(12)                                                           LOAD      17
      EQUIVALENCE( IF,ICARD(3) )                                        LOAD      18
      INTEGER BCD,OPN,RET                                               LOAD      19
      EQUIVALENCE (FCD,IFCD)                                            LOAD      20
      INTEGER HCOPY                                                     LOAD      21
      DATA HCOPY/4HCOPY/                                                LOAD      22
      DATA MASKF/77000000000000000000B/,                                LOAD      23
     .     MASKL/00007777777777777777B/                                 LOAD      24
      DATA IBLANK/1H /                                                  LOAD      25
C                                                                       LOAD      26
      M = M0                                                            LOAD      27
      IF( .NOT.STOR ) RETURN                                            LOAD      28
      IF( MODE.EQ.3 ) RETURN                                            LOAD      29
      CALL ARGINP(ND,IT,NSETS,                                          LOAD      30
     . N1,I1,N2,I2,N3,I3,N4,I4                                          LOAD      31
     .)                                                                 LOAD      32
      ASSIGN 2 TO RET                                                   LOAD      33
      IF (.NOT.EMPTY) GO TO 1                                           LOAD      34
      NI = 0                                                            LOAD      35
      IF(.NOT.INDEF) GO TO 500                                          LOAD      36
      ND(NSETS)=24                                                      LOAD      37
      IT(NSETS)=-IT(NSETS)                                              LOAD      38
      NS=0                                                              LOAD      39
  500 IPRNT=0                                                           LOAD      40
```

```
      NB = 0
      NF = 0
      EMPTY = .FALSE.
      LDFLG=.TRUE.
      GO TO 3
C
    1 READ(2,100) ( ICARD(IW),IW=1,8 )
  100 FORMAT(8A10)
      IF(ICARD(1).EQ.IBLANK) GO TO 201
    3 DECODE (20,101,ICARD(1)) ISGN,NM,OPN,KAM,KBM
  101 FORMAT (A1,A5,1X,A5,A3,1X,A3,1X)
      IF (.NOT.CPYSW)  GO TO 501
      IF (IPRNT.NE.0)  GO TO 501
      WRITE (3,106) ISGN,NM,OPN,KAM,KBM,(ICARD(IW),IW=3,8)
  106 FORMAT (6H  *** ,A1,A5,2X,A5,1X,A3,1X,A3,5H //  ,6A10)
      IPRNT=IPRNT+1
  501 IF (.NOT.(NS.EQ.NSETS.AND.INDEF)) GO TO 202
      GO TO 74
  201 IF (.NOT.CPYSW)  GO TO 202
      WRITE (3,107) (ICARD(IW),IW=3,8)
  107 FORMAT (6H    * ,21X,5H //  ,6A10)
  202 IF (NSETS.EQ.0) RETURN
      DECODE (60,102,IF(1)) (IB(J),J=1,12)
  102 FORMAT (12A5)
      DECODE( 60,105,IF(1) ) ( II(J),J=1,12 )
  105 FORMAT (12R5)
      NI0 = 0
      NB0 = 0
      NF0 = 0
      I = 1
      J = 1
      GO TO RET (2,5,6,7,8,9)
C.............................................................
    2 DO 4 NS=1,NSETS
      ASSIGN 5  TO RET
      IF ( J.GT.12) GO TO 1
    5 K = 1
      ITNS=IT(NS)
      GO TO ( 6,7,8,9 ) ITNS
C
C FLOATING POINT
C  TEST FOR SYMBOLIC FLOATING POINT
    6 MSK=MASKF.AND.IF(I)
C  TEST FOR - AND +
      IF (MSK.NE.IFL(1).AND.MSK.NE.IFL(12)) GO TO 12
      MSK= SHIFT(IF(I),6)
      MSK = MASKF.AND.MSK
   12 DO 10 L=2,14
      IF (MSK.EQ.IFL(L)) GO TO 120
   10 CONTINUE
C  SYMBOLIC.  CIRCULAR LEFT SHIFT 5 CHARACTERS, FLAG BY FILLING FIRST
C  2 CHARACTERS WITH ZEROES.
      IFCD(NF0+1)= SHIFT(IF(I),30)
      IFCD(NF0+1)=IFCD(NF0+1).AND.MASKL
      GO TO 110
C  FLOATING POINT NUMBER
  120 DECODE (10,103,IF(I)) FCD(NF0+1)
```

```
      110 K = K + 1
      103 FORMAT(F10.5)
          I=I+1
          J=J+2
          NF0=NF0+1
          IF( K.GT.ND(NS) ) GO TO 61
          IF( I.LE.6 ) GO TO 6
          CALL STDAT( M,1,NF + 1,NF0,FCD )
          NF = NF + NF0
          ASSIGN 6 TO RET
          GO TO 1
   61     CALL STDAT(M,1,NF+1,NF0,FCD)
          NF=NF+NF0
          NF0=0
          GO TO 4
C
C   BCD
    7     BCD(NB0+1) = IB(J)
          K = K + 1
   72     J = J + 1
          I = J/2 + 1
          NB0 = NB0 + 1
          IF(K.GT.ND(NS)) GO TO 73
          IF( J.LE.12) GO TO 7
          CALL STDAT( M,2,NB + 1,NB0,BCD )
          NB = NB + NB0
          ASSIGN 7 TO RET
          IF (NS.NE.NSETS) GO TO 1
          IF (INDEF) ND(NS)=ND(NS)+12
          GO TO 1
   73     J = 2*(J/2) + 1
          CALL STDAT(M,2,NB+1,NB0,BCD)
          NB=NB+NB0
          NB0=0
          GO TO 4

C  LAST SET WAS BCD OF INDEFINITE LENGTH
C  WARNING--AS PRESENTLY CODED(7/1975)  , NO INTEGER SETS MAY BE
C  INCLUDED.  TO BE SAFE, THERE SHOULD ONLY BE A SINGLE SET LOADED.
C  E.G.  CALL INP(M,MXPMY,-0,1,-2,0)

   74     EMPTY=.TRUE.
          INDEF=.FALSE.
          NBC=NB
          LFMX=LBC+NBC
          NTOT=LFMX-LFILE
          INFF(17,M)=NBC
          INFF(18,M)=NTOT
          LFILE=LFMX
          IF(LFILE.LE.LMAX) RETURN
          WRITE(3,75)
          NSTO = LMAX - LFILE
          NINFF = MAX - MIN
          WRITE (3,76) LMAX,LFILE,NSTO,MAX,NINFF,M
   75     FORMAT (6H0*****,*STORE OVERFLOWED.*
         1 *  INCREASE STORE ARRAY AND LMAX.*)
   76     FORMAT(/1X*CORE USE SUMMARY*,27X,*MAXIMUM*,17X,*USED*,8X,*UNUSED*/
```

```
     1 34X,*STORE  (ELEMENT STORAGE)*,14X,I10,1X,*(LMAX)*,4X,I10,5X,I10/
     2 34X,*INFF   (ELEMENT DEFINITIONS)*,10X,I10,1X,*(MAX)*,5X,I10,5X,
     3 I10)
      ERROR=.TRUE.
      RETURN

C
C INTEGER
 8    DECODE (10,104,II(J)) ICD(NI0+1)
 104  FORMAT (5X,I5)
      K = K + 1
 82   J = J + 1
      I = J/2 + 1
      NI0 = NI0  + 1
      IF(K.GT.ND(NS)) GO TO 83
      IF ( J .LE.12) GO TO 8
      CALL STDAT( M,3,NI + 1,NI0,ICD )
      NI = NI + NI0
      ASSIGN 8 TO RET
      GO TO 1
 83   J = 2*(J/2) + 1
      CALL STDAT(M,3,NI+1,NI0,ICD)
      NI=NI+NI0
      NI0=0
      GO TO 4
C
C  SKIP ND(NS) WORDS
9     K = K + 1
      I = I + 1
      J = J + 2
      IF( K.GT.ND(NS))  GO TO 4
      IF  ( I.LE.6 ) GO TO 9
      ASSIGN 9 TO RET
      GO TO 1
 4    CONTINUE
C.......................................................................
      IF (OPNAME.EQ.4HEVEC.OR.OPNAME.EQ.4HEQIL) GO TO 310
      IF (KIND.NE.4H    ) RETURN
C     KIND HAS TO BE DETERMINED BY BCD INPUT.
      IJ = 1
      NCD=IABS(MDAT(M,1))
      IF (NCD.NE.0) KND=INFF(20,NCD)
      IF (OPNAME.EQ.1H=) IJ = 2
      IF (OPNAME.EQ.1H=.AND.KND.EQ.4HSCAL) GO TO 315
      DO 305 J=1,NBC,IJ
      NCD=IABS(MDAT(M,J))
C  IF ELEMENT NO.=0,IT HAS NOT YET BEEN DEFINED
      IF (NCD.EQ.0) GO TO 320
 307  KND = INFF(20,NCD)
      IF (KND.EQ.2HR7.OR.KND.EQ.3HROT) GO TO 320
 305  CONTINUE
C MAKE ELEMENT A MXY KIND
      INFF(20,M) = 3HMXY
      ITE = 1
      ITYP = INFF(21,NCD)
      INFF(21,M)=ITYP
      IF(ITYP.NE.4HSNGL) ITE=2
```

```
      IF (OPNAME.NE.3HCYB) NQ = 12*ITE + KADDR
      INFF(22,M) = 2
      GO TO 325
  310 ICD(2) = BDAT(M,2)
      IF (ICD(2).NE.4HNOSV) RETURN
      ICD(1) = BDAT(M,1)
      LBC = LQ
      INFF(16,M) = LBC
      CALL STDAT(M,2,1,2,ICD)
      NQ = 0
      LFILE = LBC + NBC - 1
      GO TO 330
C  KIND IS SCALAR
  315 INFF(20,M) = KND
      NQ = 1
      TYPE = INFF(21,NCD)
      INFF(21,M) = TYPE
      IF (TYPE.NE.4HSNGL) NQ = 2
      GO TO 326
C  SET KIND TO A 7X7
  320 INFF(20,M) = 2HR7
      ITE = 1
C  IF ELEMENT UNDEFINED,LEAVE IT TYPE SINGLE
      IF(NCD.EQ.0) GO TO 324
      ITYP = INFF(21,NCD)
      INFF(21,M) = ITYP
      IF (ITYP.NE.4HSNGL) ITE=2
  324  IF(OPNAME.NE.3HCYB) NQ=49*ITE+KADD
      INFF(22,M)=7
  325 INFF(23,M)=7
  326 LQ = LFILE + 1
      LFILE = LQ + NQ - 1
      INFF(10,M) = LQ
  330 INFF( 11,M) = NQ
      NTOT = NQ + NFL + NIN + NBC
      INFF(18,M) = NTOT
      RETURN
      END
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
266	I	91	NON-INNER LOOP BEGINNING AT THIS CARD IS ENTERED FROM OUTSIDE ITS RANGE.

```
      INTEGER FUNCTION LOC(NAME,NUM,NL,LIST,LOCA,NLOC)                  LOC       2
C                                                                       LOC       3
C  SET LOC EQUAL TO REPETITION NUM OF NAME IN LIST, WHERE LIST          LOC       4
C  IS AN ARRAY OF M-NUMBERS OF SYNCH INSTRUCTIONS.                      LOC       5
C  MORE PRECISELY, NAME=MNAME(LIST(LOC)).                               LOC       6
C  LIST CONTAINS NL M-NUMBERS.                                          LOC       7
C  ALL OCCURRENCES OF NAME ARE PUT IN ARRAY LOCA,                       LOC       8
C  THERE ARE NLOC OF THEM.                                              LOC       9
C  EXAMPLE ---                                                          LOC      10
C  BL    BML          A    B    C    A    D    E    A                   LOC      11
C  THEN MI IS THE MI LIST SET UP BY BML THAT CONTAINS THE M?S           LOC      12
C  CORRESPONDING TO THE BEAMLINE A,B,..., WHICH HAS NELS=7 VALUES, AND  LOC      13
C        LOC(A,2,NELS,MI,LOCA,NLOC) = 4                                 LOC      14
C        LOCA = 1,4,7                                                   LOC      15
C        NLOC = 3                                                       LOC      16
C                                                                       LOC      17
      DIMENSION LIST(1),LOCA(1)                                         LOC      18
      LEVEL 2, LIST                                                     LOCLL     1
      DATA MAX/100/                                                     LOC      19
C                                                                       LOC      20
      DO 1 K=1,MAX                                                      LOC      21
    1 LOCA(K) = 0                                                       LOC      22
C                                                                       LOC      23
      J = 0                                                             LOC      24
      DO 2 I=1,NL                                                       LOC      25
      LL=LIST(I)                                                        LOCLL     2
      NAMEI = MNAME(LL)                                                 LOCLL     3
      IF (NAME.NE.NAMEI)  GO TO 2                                       LOC      27
      J = J + 1                                                         LOC      28
      LOCA(J) = I                                                       LOC      29
      IF (J.EQ.MAX)  GO TO 3                                            LOC      30
    2 CONTINUE                                                          LOC      31
C                                                                       LOC      32
    3 NLOC = J                                                          LOC      33
      LOC = LOCA(NUM)                                                   LOC      34
      RETURN                                                            LOC      35
      END                                                               LOC      36
```

```
      INTEGER FUNCTION LOCC(NAME,NUM,NL,LIST,LOCA,NLOC)               LOC2      1
      LEVEL 2, LOCA                                                   LOC2      2
      DIMENSION LIST(1),LOCA(1)                                       LOC2      3
      LEVEL 2, LIST                                                   LOCLL     4
      DATA MAX/100/                                                   LOC2      4
C                                                                     LOC2      5
      DO 1 K=1,MAX                                                    LOC2      6
    1 LOCA(K) = 0                                                     LOC2      7
C                                                                     LOC2      8
      J = 0                                                           LOC2      9
      DO 2 I=1,NL                                                     LOC2     10
      LL=LIST(I)                                                      LOCLL     5
      NAMEI = MNAME(LL)                                               LOCLL     6
      IF (NAME.NE.NAMEI)  GO TO 2                                     LOC2     12
      J = J + 1                                                       LOC2     13
      LOCA(J) = I                                                     LOC2     14
      IF (J.EQ.MAX)  GO TO 3                                          LOC2     15
    2 CONTINUE                                                        LOC2     16
C                                                                     LOC2     17
    3 NLOC = J                                                        LOC2     18
      LOCC = LOCA(NUM)                                                LOC2     19
      RETURN                                                          LOC2     20
      END                                                             LOC2     21
```

```
                SUBROUTINE LOCS(NAMES,NUMT,NL,LIST,ILIST)                     LOCS       2
          C                                                                   LOCS       3
                DIMENSION NAMES(1),LIST(1),ILIST(1),LOCA(100)                 LOCS       4
                LEVEL 2, LIST                                                 LOCLL      7
                                                                              LOCLL      8
                DATA MASK1/77000 00000 00000 00000B/                          LOCS       5
                DATA IBLNK1/55000 00000 00000 00000B/                         LOCS       6
          C                                                                   LOCS       7
                DO 20 I=1,NUMT                                                LOCS       8
          C TO MAKE INPUT COMPATIBLE WITH VAX VERSION                         86LOCS     1
                DECODE(5,2000,NAMES(I))LL                                     86LOCS     2
           2000 FORMAT(4XA1)                                                  86LOCS     3
                NPS1 = MASK1.AND.NAMES(I)                                     LOCS       9
                IF (NPS1.EQ.IBLNK1)  GO TO 10                                 LOCS      10
                DECODE (5,103,NAMES(I)) NAMEI,NUMI                            LOCS      11
            103 FORMAT(A3,I2)                                                 LOCS      12
                IF(LL.EQ.1H ) NUMI=NUMI/10                                    86LOCS     4
                IF (NUMI.EQ.0)  NUMI = 1                                      LOCS      13
                ILIST(I) = LOC(NAMEI,NUMI,NL,LIST,LOCA,NLOC)                  LOCS      14
                GO TO 20                                                      LOCS      15
             10 DECODE (5,100,NAMES(I)) ILST                                  LOCS      16
                IF(LL.EQ.1H ) ILST=ILST/10                                    86LOCS     5
                ILIST(I) = ILST                                               LOCS      17
            100 FORMAT(I5)                                                    LOCS      18
             20 CONTINUE                                                      LOCS      19
                RETURN                                                        LOCS      20
                END                                                           LOCS      21
```

```
      SUBROUTINE LPAR(THETA,ERL)                                        LPAR       2
C  FUNCTION EVALUATION FOR LONG STRAIGHT SECTION DESIGN                 LPAR       3
C                                                                       LPAR       4
      COMMON/AGS/ALPHA,BETA,RHO,BZ,AVAC,BMAX                            AGS        2
C                                                                       AGS        3
      COMMON/STR/GAM,A,LK,BK,F,LM,BL,LL,BQ,LTO,SMUD,CMUD                STR        2
      REAL LK,LM,LL,LTO                                                 STR        3
C                                                                       STR        4
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1          LDFLG,FIN                                               CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      REAL LDRF,LT                                                      LPAR       8
      INTEGER GAM                                                       LPAR       9
C                                                                       LPAR      10
      ITS = 0                                                           LPAR      11
      C = COS(THETA)                                                    LPAR      12
      S = SIN(THETA)                                                    LPAR      13
      E = EXP(THETA)                                                    LPAR      14
      HTAN = TANH(THETA)                                                LPAR      15
      WOE = 1./E                                                        LPAR      16
      CH = 0.5*(E + WOE)                                                LPAR      17
      SH = 0.5*(E - WOE)                                                LPAR      18
      TANTH = S/C                                                       LPAR      19
      TANDIF = TANTH - HTAN                                             LPAR      20
  100 BL = (1.-(BK*LSEP/2.)*TANDIF)/(BK*TANDIF+BK**2*LSEP*TANTH*HTAN)   LPAR      21
      LT = 4.*BL+2.*LSEP+4.*LM                                          LPAR      22
      IF (GAM.EQ.7)  GO TO 150                                          LPAR      23
  140 F = BK*LSEP*C*SH                                                  LPAR      24
      GO TO 160                                                         LPAR      25
  150 F = BK*LSEP*CH*S                                                  LPAR      26
  160 PWW = C*CH + S*SH + F                                             LPAR      27
      PWT = (C*SH + S*CH)/BK + (BL+LSEP)*C*CH + BL*S*SH + BL*F          LPAR      28
      ASN = A*SQRT ((BETA/BETMAX)*((PWW - ALPHA*PWT/BETA)**2 +          LPAR      29
     1        (PWT/BETA)**2))                                           LPAR      30
      AD = ASN - AS                                                     LPAR      31
      AB = ABS(AD)/A                                                    LPAR      32
      IF (AB - .00001) 120,110,110                                      LPAR      33
  110 ITS = ITS + 1                                                     LPAR      34
      IF (ITS.GT.50)  GO TO 130                                         LPAR      35
      AS = ASN                                                          LPAR      36
      LK = BQ/(AS*B0)                                                   LPAR      37
      BK = SQRT(LK/RHO)                                                 LPAR      38
      LM = THETA/BK                                                     LPAR      39
      GO TO 100                                                         LPAR      40
  120 ERL = LT - LT0                                                    LPAR      41
      RETURN                                                            LPAR      42
  130 WRITE (3,1000)                                                    LPAR      43
      ERROR = .TRUE.                                                    LPAR      44
      RETURN                                                            LPAR      45
 1000 FORMAT (*0DLSS COMPUTATIONS DO NOT CONVERGE.*)                    LPAR      46
      END                                                               LPAR      47
```

```
      SUBROUTINE MAGMATS(M,KA)                                          MAGMATS  2
                                                                        MAGMATS  3
*                      COMPUTE KA NUMBER OF MAGNET MATRICES.            MAGMATS  4
*  INPUT PARAMETERS MAY BE FL. PT. NUMBERS, SYMBOLIC FL. PT. OR         MAGMATS  5
*  VECTOR NAME FROM WHICH ELEMENT  VALUES ARE USED.                     MAGMATS  6
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL   2
     1          LDFLG,FIN                                               CONTRL   3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL   4
                                                                        CONTRL   5
      DIMENSION ITYP(6),IP(6),PAR(6),IPAR(6)                            MAGMATS  8
      EQUIVALENCE (PAR,IPAR)                                            MAGMATS  9
                                                                        MAGMATS 10
      CALL DATA (M,6,1,6,IP)                                            MAGMATS 11
                                                                        MAGMATS 12
*                      DETERMINE TYPE OF PARAMETER.                     MAGMATS 13
      DO 10 I=1,6                                                       MAGMATS 14
      CALL PARTYP(IP(I),ITYP(I))                                        MAGMATS 15
      IF (ERROR) RETURN                                                 MAGMATS 16
   10 CONTINUE                                                          MAGMATS 17
                                                                        MAGMATS 18
*                      STORE MAGNET INPUT                               MAGMATS 19
*                      COMPUTE AND STORE MAGNET MATRICES                MAGMATS 20
      M1 = M - 1                                                        MAGMATS 21
      DO 25 K=1,KA                                                      MAGMATS 22
      DO 20 I=1,6                                                       MAGMATS 23
      IF (ITYP(I).EQ.3) GO TO 15                                        MAGMATS 24
*       IP(I) IS FL. PT. NUMBER                                         MAGMATS 25
      IPAR(I) = IP(I)                                                   MAGMATS 26
      GO TO 20                                                          MAGMATS 27
*       FETCH VECTOR ELEMENT.  IP(I) CONTAINES M OF VECTOR.             MAGMATS 28
   15 MV = IP(I)                                                        MAGMATS 29
      PAR(I) = FDAT(MV,K)                                               MAGMATS 30
   20 CONTINUE                                                          MAGMATS 31
*     STORE PARAMETERS FOR M1 AND COMPUTE MAGNET MATRICES               MAGMATS 32
      CALL STDAT(M1,1,1,6,PAR)                                          MAGMATS 33
      CALL MAGNET(M1,PAR)                                               MAGMATS 34
      M1 = M1 - 1                                                       MAGMATS 35
   25 CONTINUE                                                          MAGMATS 36
      RETURN                                                            MAGMATS 37
      END                                                               MAGMATS 38
```

```
      SUBROUTINE MAGNET(M,PAR)                                          MAGNET     2
C  SET UP MAGNET MATRICES.                                              MAGNET     3
                                                                        MAGNET     4
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER   SWTCH      2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                          SWTCH      3
                                                                        SWTCH      4
      DIMENSION RX(2,3),RY(2,3),F(2,3),D(2,3),A(2,3),PAR(1),RW(3)       MAGNET     6
      REAL L,K,KP                                                       MAGNET     7
      EQUIVALENCE (C,CH),(S,SH),(F,D,A),(RW(1),L),(RW(2),THETA),        MAGNET     8
     1               (RW(3),PL)                                         MAGNET     9
      EQUIVALENCE (ALPHA,E1),(BETA,E2),(ANG1,IANG1),(ANG2,IANG2)        MAGNET    10
      DATA IVERT/0/                                                     MAGNET    11
      DATA RADEG/.0174532925/                                           MAGNET    12
                                                                        MAGNET    13
C IN MISALIGNMENT MODE IF BEND = 0.                                     MAGNET    14
      IF (PAR(1).GE.0)  GO TO 8                                         MAGNET    15
      IVERT = 1                                                         MAGNET    16
      PAR(1) = -PAR(1)                                                  MAGNET    17
 8    L = PAR(1)                                                        MAGNET    18
      PROF = PAR(2)                                                     MAGNET    19
      RHO = PAR(3)                                                      MAGNET    20
      OMEGA = PAR(4)                                                    MAGNET    21
      RHOI = 1./RHO                                                     MAGNET    22
C  CURV = OMEGA/RHO = CURVATURE OF REFERENCE ORBIT                      MAGNET    23
C  OMEGA IS POSITIVE IF CENTER OF CURVATURE HAS A NEGATIVE X VALUE      MAGNET    24
C  RELATIVE TO REFERENCE ORBIT.                                         MAGNET    25
C                                                                       MAGNET    26
      CURV = OMEGA*RHOI                                                 MAGNET    27
C  GAMMA, THE MAGNET WEDGE ANGLE , IS POSITIVE IF THE MAGNET FACES      MAGNET    28
C  INTERSECT ON THE SIDE OF THE REFERENCE ORBIT OPPOSITE TO THE CENTER MAGNET    29
C  OF CURVATURE.                                                        MAGNET    30
C                                                                       MAGNET    31
C  THETA = L*OMEGA/RHO                                                  MAGNET    32
      THETA = L*CURV                                                    MAGNET    33
      IF (CURV.EQ.0.) GO TO 100                                         MAGNET    34
*  PAR(5) AND (6) ARE ENTRANCE AND EXIT ANGLES IN DEGREES.              MAGNET    35
*    IF EITHER OR BOTH ARE $ SIGN, MAGNETS ARE RECTANGULAR.             MAGNET    36
      ANG1 = PAR(5)                                                     MAGNET    37
      ANG2 = PAR(6)                                                     MAGNET    38
      IF ((IANG1.EQ.1H$).OR.(IANG2.EQ.1H$)) GO TO 4                     MAGNET    39
      E1 = PAR(5) * RADEG                                               MAGNET    40
      E2 = PAR(6)*RADEG                                                 MAGNET    41
      GO TO 40                                                          MAGNET    42
*  E1,E2 (EQUIVALENCED TO ALPHA AND BETA) ARE ANGLES IN RADIANS         MAGNET    43
*  RECTANGULAR MAGNET                                                   MAGNET    44
 4    IF (PROF.EQ.0..AND.L.NE.0.)  GO TO 200                            MAGNET    45
      E1 = E2 = ABS(THETA)/2.                                           MAGNET    46
 40   CONTINUE                                                          MAGNET    47
      EFIN = SIN(ALPHA)*ABS(CURV)/COS(ALPHA)                            MAGNET    48
      EFOUT = SIN(BETA)*ABS(CURV)/COS(BETA)                             MAGNET    49
      ID = 1                                                            MAGNET    50
      FC = 1.                                                           MAGNET    51
C  KP = K + OMEGA/RHO                                                   MAGNET    52
 1    KP = FC * PROF + CURV * OMEGA                                     MAGNET    53
      IF (KP.EQ.0.) GO TO 7                                             MAGNET    54
      K = SQRT(ABS(KP*RHOI))                                            MAGNET    55
      PHI = K*L                                                         MAGNET    56
```

```
      IF (KP.LT.0.) GO TO 3
C  FOCUSSING MATRIX
 2    S = SIN(PHI)
      C = COS(PHI)
      F(1,1) = C
      F(1,2) = S/K
      F(1,3) = (CURV*(1.-C)/(K*K))*BEND
      F(2,1) = -K*S
      F(2,2) = C
      F(2,3) = (CURV*S/K)*BEND
      IF (ID.EQ.1)  PL = -(PHI-S)*CURV*CURV/(K*K*K)
      GO TO 10
C  DEFOCUSSING MATRIX
 3    E = EXP(PHI)
      EI = 1./E
      CH = 0.5*(E+EI)
      SH = 0.5*(E-EI)
      D(1,1) = CH
      D(1,2) = SH/K
      D(2,1) = SH*K
      D(1,3) = (CURV*(CH-1.)/(K*K))*BEND
      D(2,2) = CH
      D(2,3) = (CURV*SH/K)*BEND
      IF (ID.EQ.1)  PL = -(SH-PHI)*CURV*CURV/(K*K*K)
      GO TO 10
C  KP = 0 CASE
 7    A(1,1) =1.
      A(1,2) = L
      A(1,3) = (0.5*CURV*L*L)*BEND
      A(2,1) = 0.
      A(2,2) = 1.
      A(2,3) = (CURV*L)*BEND
      IF (ID.EQ.1)  PL = -L*L*L*CURV*CURV/6.
C  CORRECT FOR EDGE FOCUSSING.
 10   A(1,1)=A(1,1)+A(1,2)*EFIN
      A(2,1) = A(1,1)*EFOUT + A(2,1) + A(2,2)*EFIN
      A(2,2) = A(1,2)*EFOUT + A(2,2)
      A(2,3) = A(1,3)*EFOUT + A(2,3)
      IF (ID.EQ.2) GO TO 20
C  SET UP RADIAL TRANSFER MATRIX.
      DO 15 I=1,2
      DO 15 J=1,3
      RX(I,J)=A(I,J)
 15   CONTINUE
      IF (IVERT.EQ.0)  GO TO 16
      IVERT = 0
      GO TO 26
 16   CURV = 0.
      ID = 2
      EFIN=-EFIN
      EFOUT=-EFOUT
      FC=-1.
      GO TO 1
C  SET UP VERTICAL TRANSFER MATRIX.
 20   DO  25 I=1,2
      DO  25 J=1,3
      RY(I,J)=A(I,J)
```

```
       25  CONTINUE                                        MAGNET   114
       26  CALL STXY(M,RX,RY,RW)                           MAGNET   115
           RETURN                                          MAGNET   116
      C QUADRUPOLE                                         MAGNET   117
      100  PHSQ=PROF*L**2/RHO                              MAGNET   118
           PHI = SQRT(ABS(PHSQ))                           MAGNET   119
           IF(PHSQ.NE.0.) GO TO 101                        MAGNET   120
           CALL DRIFT(M,L)                                 MAGNET   121
           RETURN                                          MAGNET   122
      101  C= COS(PHI)                                     MAGNET   123
           S = SIN(PHI)                                    MAGNET   124
           PL=0.                                           MAGNET   125
           RX(1,1)=RX(2,2)=C                               MAGNET   126
           RX(1,2)=L*S/PHI $ RX(2,1) = -S*PHI/L            MAGNET   127
           RX(1,3)=RX(2,3)=0.                              MAGNET   128
           E = EXP(PHI) $ CH=(E +1./E)/2.  $ SH= E - CH    MAGNET   129
           RY(1,1)=RY(2,2) = CH                            MAGNET   130
           RY(1,2)=L*SH/PHI $ RY(2,1)=SH*PHI/L             MAGNET   131
           RY(1,3)=RY(2,3)=0.                              MAGNET   132
           IF(PHSQ.GT.0.) CALL STXY(M,RX,RY,RW)            MAGNET   133
           IF(PHSQ.LT.0.) CALL STXY(M,RY,RX,RW)            MAGNET   134
            RETURN                                         MAGNET   135
      C RECTANGULAR ZERO-GRADIENT DIPOLE                   MAGNET   136
      200  C = COS(THETA)                                  MAGNET   137
           S = SIN(THETA)/THETA $ T= THETA*TAN(THETA/2.)   MAGNET   138
           RX(1,1)=RX(2,2)=1.                              MAGNET   139
           RX(1,2)=L*S                                     MAGNET   140
           RX(2,1) = 0.                                    MAGNET   141
           RX(1,3)=L*(1.-C)*BEND/THETA                     MAGNET   142
           RX(2,3)=2.*T*BEND/THETA                         MAGNET   143
           PL=RX(1,2)-L                                    MAGNET   144
           RY(1,1)=RY(2,2)=1.-T                            MAGNET   145
           RY(1,2)=L $ RY(2,1)=T*(T-2.)/L                  MAGNET   146
           RY(1,3)=RY(2,3)=0.                              MAGNET   147
           GO TO 26                                        MAGNET   148
           END                                             MAGNET   149
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
20	I	PAR	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
21	I	PAR	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
22	I	PAR	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
38	I	PAR	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
39	I	PAR	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
41	I	PAR	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
42	I	PAR	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

```
            SUBROUTINE MAGRSV( M0,NAME0,KA0,KB0,NAME1 )                   MAGRSV    2
*           SETS UP INFO AND STORAGE FOR A MAGNET.                        MAGRSV    3
*           RESERVES FOR INPUT OF 6 PARAMETERS, OUTPUT OF 3X3 MATRIX.     MAGRSV    4
                                                                          MAGRSV    5
*           M0 = M INDEX,KA0  = KA        KB0 = KB                        MAGRSV    6
*           NAME0 = NAME,OPNAME WILL BE MAG                               MAGRSV    7
*           IF M0 IS NEGATIVE, THE FIRST LETTER OF NAME1 WILL PRECEDE NAME0  MAGRSV    8
*     TO FORM NAME.                                                       MAGRSV    9
            COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER  SWTCH     2
            LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                      SWTCH     3
                                                                          SWTCH     4
                                                                          MAGRSV   11
            DATA MXPMY/1/                                                 MAGRSV   12
            CALL STINFO( M0,NAME0,5HMAG  ,KA0,KB0,NAME1 )                 MAGRSV   13
            CALL RESRV(M0,MXPMY,MINZER,                                   MAGRSV   14
           . 6,1,0,0,0,0,0,0                                              MAGRSV   15
           .)                                                             MAGRSV   16
            RETURN                                                        MAGRSV   17
            END                                                           MAGRSV   18
```

```
      SUBROUTINE MAGS(M,N,Z)                                            MAGS       2
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER   SWTCH      2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                          SWTCH      3
                                                                        SWTCH      4
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL   DIM        2
                                                                        DIM        3
      COMMON/MI /MLL(3)  $ LEVEL 2,MLL                                  MAGSLL     1
      LOGICAL BSW                                                       MAGS       6
      DIMENSION RX(2,3),RY(2,3),RW(3),Z(5),Y(6),X(4),ML(3)              MAGS       7
      DIMENSION ZZ(5)                                                   MAGS       8
      EQUIVALENCE (Y(2),X,ML)                                           MAGS       9
      DATA RW/3*0.0/                                                    MAGS      10
                                                                        MAGS      11
C  COMPUTE MATRIX WITH MISALIGNMENTS INCLUDED                           MAGS      12
C  RETRIEVE LENGTH                                                      MAGS      13
C  RETRIEVE MATRIX ELEMENTS OF INDEX N AND STORE IN M.                  MAGS      14
C  EQUATES M TO N INITIALLY. N WILL BE PRESERVED.A NEW MATRIX WILL BE   MAGS      15
C  COMPUTED FOR M                                                       MAGS      16
C  MS AND MT ARE WORKING STORAGE LOCATIONS                              MAGS      17
                                                                        MAGS      18
      MS=MEND-1                                                         MAGS      19
      MT=MEND-2                                                         MAGS      20
      KA = INFF(4,M)                                                    MAGS      21
      IF (KA.EQ.0) KA = 1                                               MAGS      22
C     IF(KB.GE.1)  ERRORS ARE RANDOM WITH GIVEN RANGE                   MAGS      23
      KB=INFF(5,M)                                                      MAGS      24
      DO21 I=1,5                                                        MAGS      25
   21 ZZ(I)=Z(I)                                                        MAGS      26
      IF(KB.LE.0) GO TO 20                                              MAGS      27
      DO22I=1,5                                                         MAGS      28
   22 Z(I)=ZZ(I)*(RANF(D)-.5)                                           MAGS      29
   20 CONTINUE                                                          MAGS      30
      DBB = Z(5)                                                        MAGS      31
      BSW = .FALSE.                                                     MAGS      32
      IF(Z(5).EQ.0) GO TO 10                                            MAGS      33
C  GET MAGNET PARAMETERS                                                MAGS      34
    2 CALL DATA(N,1,1,6,Y)                                              MAGS      35
      IF (KA.EQ.2) GO TO 3                                              MAGS      36
*       INPUT DB/B.  ALTER RHO                                          MAGS      37
      Y(3) = Y(3)/(1.0+Z(5))                                            MAGS      38
      GO TO 7                                                           MAGS      39
*                    INPUT DB.   TEST FOR B=0.    INCREMENT B           MAGS      40
    3 IF (Y(4).EQ.0.) BSW = .TRUE.                                      MAGS      41
      IF (.NOT.BSW) DBB = Z(5)/Y(4)                                     MAGS      42
      Y(4) = Y(4) + Z(5)                                                MAGS      43
    7 SMS = BEND                                                        MAGS      44
      BEND=1.                                                           MAGS      45
C  COMPUTE MAGNET MATRIX WITH FIELD ERROR INCLUDED                      MAGS      46
      CALL MAGNET(M,Y)                                                  MAGS      47
      CALL RXY(M,RX,RY,RW)                                              MAGS      48
      IF (BSW) GO TO 8                                                  MAGS      49
```

```
*        WHERE B NOT 0.                                                 MAGS      50
         RX(1,3) = - RX(1,3)*DBB                                        MAGS      51
         RX(2,3) = -RX(2,3)*DBB                                         MAGS      52
         GO TO 9                                                        MAGS      53
*                     WHERE B = 0.                                      MAGS      54
 8       CURV = Y(4)/Y(3)                                               MAGS      55
         EL = Y(1)                                                      MAGS      56
         XX = -(EL*EL/2.)*CURV                                          MAGS      57
         XP = -EL*CURV                                                  MAGS      58
         RX(1,3) = XX                                                   MAGS      59
         RX(2,3) = XP                                                   MAGS      60
 9       CALL STXY(M,RX,RY,RW)                                          MAGS      61
C    RESTORE MISALIGNMENT MODE SWITCH                                   MAGS      62
         BEND=SMS                                                       MAGS      63
 4       Y(6)=0.                                                        MAGS      64
         DO 5 I=1,4                                                     MAGS      65
 5       Y(6) = Y(6) + ABS(Z(I))                                        MAGS      66
         IF (Y(6).EQ.0.0) GO TO 6                                       MAGS      67
C    COMPUTE POST SHIFT MATRIX                                          MAGS      68
         X(1)=Z(2)                                                      MAGS      69
         X(2)=(Z(2)-Z(1))/Y(1)                                          MAGS      70
         X(3)=Z(4)                                                      MAGS      71
         X(4)=(Z(4)-Z(3))/Y(1)                                          MAGS      72
         CALL SHF(MS,X)                                                 MAGS      73
C    COMPUTE PRE SHIFT MATRIX                                           MAGS      74
         X(1)=-Z(1)                                                     MAGS      75
         X(2)=-X(2)                                                     MAGS      76
         X(3)=-Z(3)                                                     MAGS      77
         X(4)=-X(4)                                                     MAGS      78
         CALL SHF(MT,X)                                                 MAGS      79
         ML(1)=MT                                                       MAGS      80
         ML(2)=M                                                        MAGS      81
         ML(3)=MS                                                       MAGS      82
         MLL(1)=ML(1) $ MLL(2)=ML(2) $ MLL(3)=ML(3) $CALL MMM(M,3,MLL)  MAGSLL     2
                                                                        MAGSLL     3
 6       CONTINUE                                                       MAGS      84
         DO23 I=1,5                                                     MAGS      85
 23      Z(I)=ZZ(I)                                                     MAGS      86
         RETURN                                                         MAGS      87
 10      Y(1)=FDAT(N,1)                                                 MAGS      88
         CALL RXY(N,RX,RY,RW)                                           MAGS      89
         CALL STXY(M,RX,RY,RW)                                          MAGS      90
         GO TO 4                                                        MAGS      91
         END                                                            MAGS      92
```

```
      SUBROUTINE  MAGSRV( M0,NAME0,KA0,KB0,NAME1 )                    MAGSRV     2
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER SWTCH      2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                        SWTCH      3
                                                                      SWTCH      4
      DATA MXPMY/1/                                                   MAGSRV     4
      CALL STINFO( M0,NAME0,5HMAGS ,KA0,KB0,NAME1)                    MAGSRV     5
      CALL INP(M0,MXPMY,MINZER,                                       MAGSRV     6
     .  1,2,5,1,0,0,0,0                                               MAGSRV     7
     .)                                                               MAGSRV     8
      RETURN                                                          MAGSRV     9
      END                                                             MAGSRV    10
```

```
      SUBROUTINE MAGV(M,PAR)                                          MAGV       2
                                                                      MAGV       3
*                  SET UP MATRIX WITH VERTICAL BEND                   MAGV       4
*     IF B IS +,  THEN DO  +PI/2 M -PI/2                              MAGV       5
*     IF B IS -, THEN DO -PI/2 M +PI/2                                MAGV       6
                                                                      MAGV       7
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL DIM        2
                                                                      DIM        3
      DIMENSION T(49),TR(49),RW(3),PAR(1)                             MAGV       9
      DATA PI2/1.5707963268/                                          MAGV      10
                                                                      MAGV      11
*   COMPUTE MATRIX FOR BENDING MAGNET                                 MAGV      12
      CALL MAGNET(MEND,PAR)                                           MAGV      13
      CALL RTRV7(MEND,T,RW)                                           MAGV      14
      THETA = PI2                                                     MAGV      15
      CALL ROTATE (THETA,T,TR,RW)                                     MAGV      16
      CALL STOR7(M,TR,RW)                                             MAGV      17
      RETURN                                                          MAGV      18
      END                                                             MAGV      19
```

```
      SUBROUTINE MAP(V,PAR)                                    MAP      2
      RETURN                                                   MAP      3
      END                                                      MAP      4
```

```
      SUBROUTINE MAP1(M,V,PAT)                                          MAP1     2
      RETURN                                                            MAP1     3
      END                                                               MAP1     4
```

```
                                                                        MAP1      5
      SUBROUTINE MAP3(V,PAR)                                            MAP3      2
                                                                        MAP3      3
C     SIMULATE BEAM-BEAM FORCE WITH ROUND BEAM                          MAP3      4
C     PAR(1)= -DNUY  PAR(2)= RADIUS  PAR(3)= BETAY-Y                    MAP3      5
                                                                        MAP3      6
      DIMENSION V(4),PAR(3)                                             MAP3      7
      DATA TWOPI/6.2831853/                                             MAP3      8
      RS = V(3)**2/PAR(2)**2                                            MAP3      9
      IF (V(3).EQ.0.) GO TO 2                                           MAP3     10
       F = 2.*TWOPI*PAR(1)*(1.-EXP(-RS))/(RS*PAR(3))                    MAP3     11
    3  V(4) = V(4) + V(3)*F                                             MAP3     12
    2 RETURN                                                            MAP3     13
      END                                                               MAP3     14
```

```
      SUBROUTINE MAP8(V,PAR)                                            MAP8       2
      RETURN                                                            MAP8       3
      END                                                               MAP8       4
```

```
      SUBROUTINE MAP9(V,PAR)                                    MAP9       2
      RETURN                                                    MAP9       3
      END                                                       MAP9       4
```

```
      SUBROUTINE  MATRSV( M0,NAME0,KA0,KB0,NAME1 )
      RETURN
      END
```

```
  1              SUBROUTINE MAXMIN(BMAX,BMIN,BET,MB,IP)                  MAXMIN   2
                                                                         MAXMIN   3
           C  DETERMINES ALGEBRAIC MAX OF BMAX,BET AND MINIMUM OF BMIN,BET   MAXMIN   4
           *     IF TWO ARE EQUAL, THE FIRST REMAINS AS MIN OR MAX       MAXMIN   5
  5        *      MB(1),MB(2)  ARE INDICES OF MAX. AND MIN. VALUES        MAXMIN   6
           *      IP = INDEX OF BET                                       MAXMIN   7
                                                                         MAXMIN   8
                 DIMENSION MB(1)                                          MAXMIN   9
                 IF(BMAX.GT.BET) GO TO 10                                 MAXMIN  10
 10              BMAX = BET                                               MAXMIN  11
                 MB(1) = IP                                               MAXMIN  12
            10   IF(BMIN.LT.BET) RETURN                                   MAXMIN  13
                 BMIN = BET                                               MAXMIN  14
                 MB(2) = IP                                               MAXMIN  15
 15              RETURN                                                   MAXMIN  16
                 END                                                      MAXMIN  17
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
14	I	MB	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

```
      FUNCTION MDAT(M,J)                                                  MDAT        2
C  IF THE JTH BCD STORAGE LOCATION OF M CONTAINS AN ELEMENT NAME,         MDAT        3
C  THEN MDAT WILL CONTAIN THE M-NUMBER( OR INDEX ) OF THAT ELEMENT.       MDAT        4
                                                                          BLANK       2
      LEVEL 2, STORE,INFF,IWORK                                           BLANK       3
      COMMON STORE(48000),IWORK(10)                                       86MARSIZ    1
      DIMENSION INFF(24,2000)                                             86MARSIZ    2
      EQUIVALENCE (INFF,STORE)                                            86MARSIZ    3
                                                                          BLANK       5
      EQUIVALENCE(IVAR,FVAR)                                              MDAT        6
      INTEGER ELNUM                                                       MDAT        7
      LOC = INFF(16,M) + J-1                                              MDAT        8
      FVAR=STORE(LOC)                                                     MDAT        9
      MDAT=ELNUM(IVAR)                                                    MDAT       10
      RETURN                                                              MDAT       11
      END                                                                 MDAT       12
```

```
      SUBROUTINE MESH(M,KA,MOP)                                         MESH       2
      DIMENSION IND(10,2),NM(10,2),FMIN(10,2),FDEL(10,2),VSAV(10,2),    MESH       3
     1  LP(10,2),LL(10,2),JN(10,2),FPAR(10,2)                           MESH       4
      DIMENSION FMAX(10),KEJ(2),F(30)                                   MESH       5
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER   SWTCH      2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                          SWTCH      3
                                                                        SWTCH      4
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1            LDFLG,FIN                                             CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      INTEGER ELNUM                                                     MESH       9
      JX = 1                                                            MESH      10
      IF (MOP.EQ.4HVPAR) JX = 2                                         MESH      11
      IF (.NOT.MSSW) GO TO 5                                            MESH      12
      IF (JX.GT.1) GO TO 4                                              MESH      13
      IF (MHS) GO TO 46                                                 MESH      14
      GO TO 5                                                           MESH      15
    4 IF (VPR) GO TO 48                                                 MESH      16
C  INITIAL CALL                                                         MESH      17
C  RETRIEVE VARIABLE NAMES AND INDEX NUMBER                             MESH      18
    5 CONTINUE                                                          MESH      19
      CALL DATA(M,2,2,KA,NM(1,JX) )                                     MESH      20
      CALL DATA(M,3,1,KA,IND(1,JX) )                                    MESH      21
C  RETRIEVE MESH LIMITS THAT WERE INPUT                                 MESH      22
      KM = KA*3                                                         MESH      23
      CALL DATA(M,1,1,KM,F)                                             MESH      24
      I1=1                                                              MESH      25
      I2=2                                                              MESH      26
      I3=3                                                              MESH      27
C                                                                       MESH      28
      DO 10 I=1,KA                                                      MESH      29
      FMIN(I,JX) = F(I1)                                                MESH      30
      FMAX(I)=F(I2)                                                     MESH      31
      FDEL(I,JX) = F(I3)                                                MESH      32
      I1=I1+3                                                           MESH      33
      I2 = I2 + 3                                                       MESH      34
      I3 = I3+3                                                         MESH      35
   10 CONTINUE                                                          MESH      36
C  COMPUTE LOOP LIMITS,SAVE ORIGINAL VALUES,SET BEGINNING VALUES OF MESH MESH     37
      KEJ(JX) = KA                                                      MESH      38
      KE = KA                                                           MESH      39
      DO 15 I=1,KE                                                      MESH      40
      LL(I,JX) = ((FMAX(I)-FMIN(I,JX))/FDEL(I,JX)) + 1.0000001          MESH      41
      JN(I,JX) = ELNUM(NM(I,JX))                                        MESH      42
      F(I) = FDAT(JN(I,JX),IND(I,JX))                                   MESH      43
      VSAV(I,JX) = F(I)                                                 MESH      44
      FPAR(I,JX) = FMIN(I,JX)                                           MESH      45
      CALL STDAT(JN(I,JX),1,IND(I,JX),1,FPAR(I,JX) )                    MESH      46
   15 CONTINUE                                                          MESH      47
C  INITIALIZE LOOP INDICES.  RUNNING INDEX OF LOOP IS LP.  LIMIT IS LL  MESH      48
```

```
      DO 20 I=1,KE
      LP(I,JX) = 1
   20 CONTINUE
      IF (JX.EQ.2) GO TO 30
      MHS=.TRUE.
      GO TO 60
   30 LMX = LL(1,JX)
      DO 31 I=2,KE
      IF (LL(I,JX).GT.LMX) LMX = LL(I,JX)
   31 CONTINUE
      LMM = 1
      VPR = .TRUE.
      GO TO 60
C  VPAR CALCULATION.  STRAIGHT LINE
   48 CONTINUE
C   TEST TO SEE IF FINISHED
      KE = KEJ(JX)
      IF(LMM.EQ.LMX) GO TO 63
      LMM = LMM + 1
      DO 36 I=1,KE
      FPAR(I,JX) = FPAR(I,JX) + FDEL(I,JX)
      CALL STDAT(JN(I,JX),1,IND(I,JX),1,FPAR(I,JX))
   36 CONTINUE
      GO TO 60
C  FULL MESH
   46 CONTINUE
      KE = KEJ(JX)
      KV = KE
      I = 1
   49 IF (LP(KV,JX).NE.LL(KV,JX))GO TO 55
C  LOOP COMPLETED.  RESET
      LP(KV,JX) = 1
      FPAR(KV,JX) = FMIN(KV,JX)
      CALL STDAT(JN(KV,JX),1,IND(KV,JX),1,FPAR(KV,JX))
      KV = KV - 1
      IF (I.EQ.KE) GO TO 64
      I = I + 1
      GO TO 49
   55 LP(KV,JX) = LP(KV,JX) + 1
      FPAR(KV,JX) = FPAR(KV,JX) + FDEL(KV,JX)
      CALL STDAT(JN(KV,JX),1,IND(KV,JX),1,FPAR(KV,JX) )
   60 KA = 1
      WRITE (3,100)
      DO 62 I=1,KE
      WRITE (3,101) IND(I,JX),NM(I,JX),FPAR(I,JX)
   62 CONTINUE
      CALL SUBR(3)
      MSSW = .TRUE.
      MODS=.TRUE.
      RETURN
C  FINISHED.  REPLACE ORIGINAL VALUES OF VARIABLES
   63 VPR=.FALSE.
      MODS=.FALSE.
      GO TO 65
   64 MHS=.FALSE.
      MODS=.FALSE.
   65 DO 70 I=1,KE
```

```
      F(I) = VSAV(I,JX)
      CALL STDAT(JN(I,JX),1,IND(I,JX),1,F(I))
   70 CONTINUE
      IF(MHS) RETURN
      IF (VPR) RETURN
      MSSW = .FALSE.
      MODE = INFF(19,M)
      RETURN
  100 FORMAT (/15H CURRENT VALUES/)
  101 FORMAT (5X,I5,4H OF ,A5,3H = ,F10.5)
      END
```

```
      SUBROUTINE MFD(M,J,FACTOR)                                        MFD        2
C  MULTIPLIES BY FACTOR THE J-TH FL PT PARAMETER OF ELEMENT M AND       MFD        3
C  RETURNS NEW VALUE TO STORAGE.                                        MFD        4
      F=FDAT(M,J)                                                       MFD        5
      F=F * FACTOR                                                      MFD        6
      CALL REPFLT(M,J,F)                                                MFD        7
      RETURN                                                            MFD        8
      END                                                               MFD        9
```

```
      SUBROUTINE MIEXP(M,J,NDAT,NELS,ML)

      DIMENSION ML(1),KBEG(10),NREP(10),NR(10)
       LEVEL 2, ML

      INTEGER BK,BDAT
      DATA IBLANK/10H          /
      DATA MASK/00000077000000000000B/
      DATA IPL/00000051000000000000B/
      DATA IPR/00000052000000000000B/

C  EXPAND PARENTHESISES AND STORE IN ML.  KI INPUT IS NUMBER OF
C  5-CHARACTER BCD DATA TO PROCESS, ON OUTPUT IS NUMBER OF M?S STORED.

      I=0
      LOOP=0
      K=1
    1 IF (K.GT.NDAT) GO TO 5
      BK=BDAT(M,K+J-1)
      IF (BK.EQ.IBLANK) GO TO 6
      LBK=BK.AND.MASK
      IF (LBK.EQ.IPL) GO TO 2
      IF (LBK.EQ.IPR) GO TO 3
      I=I+1
      ML(I)=MDAT(M,K+J-1)
    6 K=K+1
      GO TO 1
    2 DECODE(10,1000,BK) NUM
 1000 FORMAT (I3,7X)
      IF(NUM.EQ.0) GO TO 3
      LOOP=LOOP+1
      KBEG(LOOP)=K+1
      NREP(LOOP)=NUM
      NR(LOOP)=0
      K=K+1
      GO TO 1
    3 NR(LOOP)=NR(LOOP)+1
      IF(NR(LOOP).EQ.NREP(LOOP)) GO TO 4
      K=KBEG(LOOP)
      GO TO 1
    4 K=K+1
      LOOP=LOOP-1
      GO TO 1
    5 NELS = I
      RETURN
      END
```

```
      SUBROUTINE MIFILL(M,J,NDAT,NELS,MI)                          MIFILL      2
                                                                   BLANK       2
      LEVEL 2, STORE,INFF,IWORK                                    BLANK       3
      COMMON STORE(48000),IWORK(10)                                86MARSIZ    1
      DIMENSION INFF(24,2000)                                      86MARSIZ    2
      EQUIVALENCE (INFF,STORE)                                     86MARSIZ    3
                                                                   BLANK       5
      DIMENSION MI(1),ME(16000)                                    NOV3MFLL    1
      COMMON/ME/ME                                                 MIFILLL     2
      LEVEL 2,MI,ME                                                MIFILLL     3
                                                                   MIFILLL     4
      INTEGER OPL                                                  MIFILL      5
                                                                   MIFILL      6
      CALL MIEXP(M,J,NDAT,NELS,ME)                                 MIFILL      7
      KREF = 1                                                     MIFILL      8
      OPL = INFF(1,M)                                              MIFILL      9
      IF (OPL.NE.3HBML)  GO TO 25                                  MIFILL     10
      KAL = INFF(4,M)                                              MIFILL     11
      IF (KAL.GE.0)  GO TO 25                                      MIFILL     12
                                                                   MIFILL     13
      DO 20 N=1,NELS                                               MIFILL     14
   20 MI(N) = ME(NELS-N+1)                                         MIFILL     15
      KREF = -KREF                                                 MIFILL     16
      GO TO 27                                                     MIFILL     17
                                                                   MIFILL     18
   25 DO 26 N=1,NELS                                               MIFILL     19
   26 MI(N) = ME(N)                                                MIFILL     20
                                                                   MIFILL     21
   27 KE = NELS                                                    MIFILL     22
      L=0                                                          MIFILL     23
                                                                   MIFILL     24
    1 L=L+1                                                        MIFILL     25
      IF(L.GT.KE) GO TO 2                                          MIFILL     26
       ML=MI(L)                                                    MIFILL     27
                                                                   MIFILL     28
      IF (ML.GT.999990) GO TO 3                                    MIFILL     29
       OPL=INFF(1,ML)                                              MIFILL     30
       KAL=INFF(4,ML)                                              MIFILL     31
                                                                   MIFILL     32
      IF (OPL.NE.3HBML.AND.OPL.NE.4HLIST)  GO TO 1                 MIFILL     33
       KL=INFF(17,ML)                                              MIFILL     34
       CALL MIEXP(ML,1,KL,NL,ME)                                   MIFILL     35
                                                                   MIFILL     36
      IF (OPL.EQ.3HBML) GO TO 11                                   MIFILL     37
       KBL=INFF(5,ML)                                              MIFILL     38
                                                                   MIFILL     39
      IF (KREF.LT.0.AND.KAL.NE.1)  GO TO 12                        MIFILL     40
       MI(L)=ME(KBL)                                               MIFILL     41
       GO TO 13                                                    MIFILL     42
                                                                   MIFILL     43
   12  MI(L)=ME(NL-KBL+1)                                          MIFILL     44
                                                                   MIFILL     45
   13  KBL=KBL+1                                                   MIFILL     46
                                                                   MIFILL     47
      IF (KBL.LE.NL)  GO TO 14                                     MIFILL     48
       KBL=1                                                       MIFILL     49
   14  INFF(5,ML)=KBL                                              MIFILL     50
```

```
      GO TO 1

   11 KEL=KE-L
      N0=1

      IF(KAL.GE.0)  GO TO 4
      N0=0
      KE=KE+1
      KREF=-KREF

    4 DO 5 N=1,KEL
    5 MI(KE+NL-N)=MI(KE-N+N0)

      IF(KREF.LT.0) GO TO 7
      DO 6 N=1,NL
    6 MI(L+N-1)=ME(N)
      GO TO 9

    7 DO 8 N=1,NL
    8 MI(L+N-1)=ME(NL-N+1)


    9 KE=KE+NL-1
      IF (KAL.LT.0) MI(L+NL)=999999
      L=L-1
      GO TO 1

    3 KEL=KE-L
      DO 10 N=1,KEL
   10 MI(L+N-1)=MI(L+N)
      KE=KE-1
      L=L-1
      KREF=-KREF
      GO TO 1

    2 NELS=KE

      RETURN
      END
```

```
                                                               MIFILL     90
      SUBROUTINE MLIST (M,J,K,MI)                              MLIST       2
C  READS K ELEMENT NAMES FROM THE JTH BCD LOCATION OF  M  AND STORES   MLIST  3
C  THEIR INDICES IN  MI.                                       MLIST       4
      DIMENSION MI(1)                                          MLIST       5
      LEVEL 2,MI                                               MLISTL      1
      DIMENSION ML(200)                                        MLISTL      2
      INTEGER ELNUM                                            MLIST       6
      CALL DATA(M,2,J,K,ML)                                    MLISTL      3
      DO 1 I=1,K                                               MLIST       8
    1 MI(I)=ELNUM(ML(I))                                       MLISTL      4
                                                               MLISTL      5
      RETURN                                                   MLIST      10
      END                                                      MLIST      11
```

```
      SUBROUTINE MMM(M,KA,MI)                                          MMM        2
C MATRIX MULTIPLICATION.  R(M)=R(MK)*...R(M2)*R(M1)                    MMM        3
C  IF MI(I) IS NEGATIVE, R(M)=R(MK)...(1/R(MI))..R(M2).R(M1)  ETC.     MMM        4
                                                                       MMM        5
* SYNCH INSTRUCTION MMM:                                               MMM        6
* X   MMM          A    B    C    .    .    .    .                     MMM        7
* A,B,C,...ARE PREVIOUSLY DEFINED MATRICES, OR BML?S, OR NUMBERED      MMM        8
* PARENTHESES.  THIS INPUT LIST IS EXPANDED BY MIFILL AND MIEXP TO     MMM        9
* FORM A LIST OF MATRIX INDICES, STORED IN MI(I).  THESE ARE           MMM       10
* MULTIPLIED TOGETHER TO FORM MATRIX NAMED X.                          MMM       11
C                                                                      MMM       12
C SEE EXECMM ROUTINE FOR OPTIONS OF MMM INSTRUCTION                    MMM       13
                                                                       BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                        BLANK      3
      COMMON STORE(48000),IWORK(10)                                    86MARSIZ   1
      DIMENSION INFF(24,2000)                                          86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                         86MARSIZ   3
                                                                       BLANK      5
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL  DIM        2
                                                                       DIM        3
      DIMENSION MI(1)                                                  MMM       16
      LEVEL 2,MI                                                       MMML       1
                                                                       MMML       2
                                                                       MMM       17
C NT IS WORKING STORAGE FOR MATRIX                                     MMM       18
      KN = INFF(20,M)                                                  MMM       19
      KK = 3                                                           MMM       20
      IF (KN.EQ.2HR7.OR.KN.EQ.3HROT.OR.KN.EQ.3HMAP)  KK = 7            MMM       21
 3    NT = M7END                                                       MMM       22
      K=3 $ NT = MEND                                                  MMMSPEED   1
      DO 100 I=1,KA                                                    MMMSPEED   2
      IF(INFF(20,IABS(MI(I))).EQ.3HMXY) GO TO 100                      MMMSPEED   3
      K=7  $ NT=M7END                                                  MMMSPEED   4
      GO TO 5                                                          MMMSPEED   5
 100  CONTINUE                                                         MMMSPEED   6
 5    M1= MI(1)                                                        MMM       24
C  IF FIRST INDEX IN MILIST IS -, SET IT + AND INVERT                  MMM       25
      IF(M1)1,2,2                                                      MMM       26
 1    M1=-M1                                                           MMM       27
      CALL INV(NT,M1)                                                  MMM       28
      K1=2                                                             MMM       29
      GO TO 4                                                          MMM       30
 2    CONTINUE                                                         MMM       31
      IF (KA.GE.2) GO TO 7                                             MMM       32
C ONLY ONE ELEMENT.  MULTIPLY BY UNIT MATRIX.                          MMM       33
      M2 = MUNIT                                                       MMM       34
      GO TO 8                                                          MMM       35
 7    M2 = MI(2)                                                       MMM       36
      IF (KA.GT.2) GO TO 12                                            MMM       37
 8    CONTINUE                                                         MMM       38
C  IF THERE ARE ONLY 2 MATRIX MULT.,RESULT SHOULD BE STORED FOR M.     MMM       39
      CALL MXD(M,M1,M2,KK)                                             MMM       40
      GO TO 11                                                         MMM       41
C  ACCUM. RESULTS OF MULT. ARE STORED IN WORKING STORAGE OF NT         MMM       42
 12   CALL MXD(NT,M1,M2,K)                                             MMM       43
      K1=3                                                             MMM       44
 4    DO 10 I=K1,KA                                                    MMM       45
```

```
            M1=MI(I)                                          MMM   46
            IF (I.EQ.KA) GO TO 6                              MMM   47
            CALL MXD(NT,NT,M1,K)                              MMM   48
            GO TO 10                                          MMM   49
      C  ON LAST ITERATION, FINAL MATRIX STORED FOR M.        MMM   50
        6   CALL MXD(M,NT,M1,KK)                              MMM   51
       10   CONTINUE                                          MMM   52
       11   RETURN                                            MMM   53
            END                                               MMM   54
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
48	I	MI	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

```
      SUBROUTINE  MMMRSV( M0,NAME0,KA0,KB0,NAME1 )                     MMMRSV    2
                                                                       MMMRSV    3
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER  SWTCH     2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                         SWTCH     3
                                                                       SWTCH     4
      INTEGER PVECS,R7,ROT,PRD                                         MMMRSV    5
      DATA MXPMY,PVECS,R7,ROT,PRD/1,2,5,6,8/                           MMMRSV    6
      DATA IBLNK/4H    /                                               MMMRSV    7
                                                                       MMMRSV    8
      INM = 3HMMM                                                      MMMRSV    9
      CALL STINFO(M0,NAME0,INM,KA0,KB0,NAME1)                          MMMRSV   10
      KN = IBLNK                                                       MMMRSV   11
      IF (MSIZE.EQ.7) KN=R7                                            MMMRSV   12
      IF (MSIZE.EQ.3) KN= MXPMY                                        MMMRSV   13
C  SET KIND IF INDICATED BY MSIZE                                      MMMRSV   14
      IF(NAME1.NE.0) GO TO 1                                           MMMRSV   15
      IF(KN.EQ.IBLNK) KN=MXPMY                                         MMMRSV   16
      CALL INP(M0,KN,MINZER,                                           MMMRSV   17
     .  1,-2,0,0,0,0,0,0                                               MMMRSV   18
     .)                                                                MMMRSV   19
      RETURN                                                           MMMRSV   20
    1 CALL INP(M0,KN,MINZER,                                           MMMRSV   21
     .  KA0,2,0,0,0,0,0,0                                              MMMRSV   22
     .)                                                                MMMRSV   23
      RETURN                                                           MMMRSV   24
      END                                                              MMMRSV   25
```

```
      FUNCTION MNAME(M)                                    MNAME      2
C  FINDS NAME OF ELEMENT WHOSE NUMBER IS M.                MNAME      3
C  IF M IS NEGATIVE,RETURN/NAME FOR INVERSE                MNAME      4
                                                           BLANK      2
      LEVEL 2, STORE,INFF,IWORK                            BLANK      3
      COMMON STORE(48000),IWORK(10)                        86MARSIZ   1
      DIMENSION INFF(24,2000)                              86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                             86MARSIZ   3
                                                           BLANK      5
      COMMON/FLTN/IFL(15)                                  FLTN       2
                                                           FLTN       3
      MNAME = INFF(2,IABS(M))                              MNAME      7
      IF (M.GT.0) RETURN                                   MNAME      8
      MN= SHIFT(MNAME,-6).AND.00777777777777777777B        MNAME      9
      MNAME = (IFL(15).OR.MN)                              MNAME     10
      RETURN                                               MNAME     11
      END                                                  MNAME     12
```

```
      SUBROUTINE MODQ (M,N,V,W)                                      MODQ       2
C                                                                    MODQ       3
      COMMON/DIM/LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL  MODQ       4
      COMMON /MJ/ ML    $          LEVEL 2, ML                       MODQLL     1
      DIMENSION RX(2,3),RY(2,3),R(2,3),RYW(3),RPQ(4),RW(3)           MODQ       5
      DIMENSION V(1),W(7),PAR(6),T(7,7),D(7,7),ML(3)                 MODQ       6
      EQUIVALENCE (BR,PAR(3))                                        MODQ       7
      DATA RADEG/.0174532925/                                        MODQ       8
C                                                                    MODQ       9
      CALL DATA (M,1,1,6,PAR)                                        MODQ      10
      BR = PAR(3)*(1.+V(6))                                          MODQ      11
      CALL MAGNET (N,PAR)                                            MODQ      12
      CALL RTRV7(N,T,RW)                                             MODQ      13
      CALL MXV7(T,V,W)                                               MODQ      14
      S = PAR(1)                                                     MODQ      15
      RHOIX = (V(2)-W(2))/S                                          MODQ      16
      RHOIY = (V(4)-W(4))/S                                          MODQ      17
      ML(1) = M7END - 8                                              MODQ      18
      ML(2) = N                                                      MODQ      19
      ML(3) = M7END - 9                                              MODQ      20
      CALL RTRV7 (MUNIT,D,RW)                                        MODQ      21
      D(2,1) = -V(4)*RHOIY                                           MODQ      22
      D(4,3) = -V(2)*RHOIX                                           MODQ      23
      CALL STOR7(M7END-8,D,RW)                                       MODQLL     2
      D(2,1) = W(4)*RHOIY                                            MODQ      25
      D(4,3) = W(2)*RHOIX                                            MODQ      26
      CALL STOR7(M7END-9,D,RW)                                       MODQLL     3
                                                                     MODQLL     4
      CALL RXY(N,RX,RY,RW)                                           MODQ      28
      IF (RHOIX.EQ.0.)  GO TO 3                                      MODQ      29
C                                                                    MODQ      30
      SGN = SIGN(1.,RHOIX)/RADEG                                     MODQ      31
      PAR(1) = -PAR(1)                                               MODQ      32
      PAR(4) = BR*RHOIX                                              MODQ      33
      PAR(5) = SGN*V(2)                                              MODQ      34
      PAR(6) = -SGN*W(2)                                             MODQ      35
      CALL MAGNET(N,PAR)                                             MODQ      36
      CALL RXY(N,RX,R,RW)                                            MODQ      37
    3 IF (RHOIY.EQ.0.)  GO TO 4                                      MODQ      38
C                                                                    MODQ      39
      SGN = SIGN(1.,RHOIY)/RADEG                                     MODQ      40
      PAR(1) = -PAR(1)                                               MODQ      41
      PAR(2) = -PAR(2)                                               MODQ      42
      PAR(4) = BR*RHOIY                                              MODQ      43
      PAR(5) = SGN*V(4)                                              MODQ      44
      PAR(6) = -SGN*W(4)                                             MODQ      45
      CALL MAGNET(N,PAR)                                             MODQ      46
      CALL RXY(N,RY,R,RYW)                                           MODQ      47
      RW(3) = RW(3) + RYW(3)                                         MODQ      48
C                                                                    MODQ      49
    4 CALL CONVMAT(7,T,RX,RY,RW,RPQ)                                 MODQ      50
      CALL STOR7(N,T,RW)                                             MODQ      51
      CALL MMM(N,3,ML)                                               MODQ      52
      RETURN                                                         MODQ      53
      END                                                            MODQ      54
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
11	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
16	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
17	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
22	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
23	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
35	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
45	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

```
      SUBROUTINE MOVE(M,N,V)                                            MOVE       2
                                                                        MOVE       3
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL   DIM        2
                                                                        DIM        3
                                                                        MOVE       5
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
C                                                                       MOVTOG     2
      COMMON/MOVTOG/TOG                                                 MOVTOG     3
      LOGICAL TOG                                                       MOVTOG     4
C                                                                       MOVTOG     5
                                                                        MOVE       7
      DIMENSION T(7,7),RW(3),RW1(3),PAR(10),ML(7),V(7)                  MOVE       8
      COMMON /MK /ML  $ LEVEL 2,ML                                      MOVELL     1
      DIMENSION PARR(6)                                                 MOVE       9
      DIMENSION PP(6)                                                   MOV883     1
      REAL L,K                                                          MOVE      10
      INTEGER BDAT                                                      MOVE      11
      EQUIVALENCE (PAR(1),HX),(PAR(2),DHX),(PAR(3),HY),(PAR(4),DHY),    MOVE      12
     1            (PAR(5),HZ),(PAR(6),THETZ)                            MOVE      13
     2           ,(PAR(7),XIN),(PAR(8),YIN),(PAR(9),XOUT),(PAR(10),YOUT) MOVE     14
      DIMENSION W(7),PAT(4)                                             MOVE      15
      LOGICAL QDSW,SXSW                                                 MOVE      16
      INTEGER HSXTP,HMAP,HMAG                                           MOVE      17
      DATA HSXTP,HMAP,HMAG/4HSXTP,3HMAP,3HMAG/                          MOVE      18
      DATA RW1/3*0.0/                                                   MOVE      19
                                                                        MOVE      20
C MISALIGNMENT OF ELEMENTS ---SIX DEGREES OF FREEDOM.                   MOVE      21
C USES 7X7 FORMALISM IN CONJUNCTION WITH FXPT, TRK AND CYEM.            MOVE      22
C CLOSED ORBIT FOUND BY FXPT RATHER THAN CYX WHICH NEEDS MAGS.  EXAMPLE MOVE      23
C...................................................................... MOVE      24
C QF   MAG     //L      /G       /BRHO                                  MY3MOVE    2
C H    VEC  6  //HX     /DHX     /HY      /DHY     /HZ      /THETZ      MY3MOVE    3
C ---FORMAT 1----                                                       MY3MOVE    4
C QF*  MOVE    //QF     /H                                              MY3MOVE    5
C...................................................................... MOVE      29
C ---FORMAT 2.0---                                                      MY3MOVE    6
C QF*  MOVE    //QF     /HX      /HY      /HZ      /THETZ               MY3MOVE    7
C ---FORMAT 2.1---                                                      MY3MOVE    8
C QF*  MOVE    //QF   V /HX     /DHX     /HY      /DHY                  MY3MOVE    9
C...................................................................... MOVE      32
C  M = INDEX OF MOVE COMMAND.  N=INDEX OF ELEMENT AT WHICH MATRIX       MY3MOVE   10
C                                 FOR M TO BE STORED.                   MY3MOVE   11
C  V = ORBIT MAPPED.  MQ=INDEX OF ELEMENT MOVED.  MPAR=INDEX OF VEC.    MY3MOVE   12
C  KB.GE.1 -- PARAMETERS RANDOM.  KB.GE.2--RANDOM SEE RESET SO NEXT     MY3MOVE   13
C HX,HY,HZ ARE TRANSLATIONS ALONG X,Y,Z AXES,                           MOVE      33
C DHX, DHY ARE ROTATIONS ABOUT THE Y,X AXES, I.E. TILTS IN X-Z, Y-Z     MOVE      34
C THETZ IS ROTATION ABOUT Z AXIS.  Z = BEAM DIR., X = HOR., Y = VERT.   MOVE      35
C ORDER OF MISALIGNMENTS----LONGITUDINAL(Z), TRANSVERSE(X,Y), Z-ROTATION MOVE     36
C M IS THE INDEX OF THE MOVE COMMAND.                                   MOVE      37
C N IS THE INDEX OF THE ELEMENT AT WHICH THE MATRIX CORRESPONDING       MOVE      38
C    TO M IS TO BE STORED.                                              MOVE      39
C V IS THE ORBIT TO BE MAPPED BY THE MATRIX T CORRESPONDING TO M.       MOVE      40
```

```
      C  MQ IS THE INDEX OF THE ELEMENT TO BE MOVED.                      MOVE       41
      C  MPAR IS THE INDEX OF THE VEC COMMAND THAT CONTAINS THE MISALIGNMENTS. MOVE  42
      C  KB GE.1 -- PARAMETERS RANDOM                                     MOVE       43
      C  KB GE.2 - RANDOM SEED RESET SO NEXT PASS IS SAME                 MOVE       44
                                                                          MOVE       45
            MQ=MDAT(M,1)                                                  MOVE       46
            KB=INFF(5,M)                                                  MOVE       47
            IF (KB.LT.2)  GO TO 10                                        MY3MOVE    14
            IF (TOG) CALL RANGET(NRN)                                     MY3MOVE    15
         10 QDSW = .FALSE.                                                MY3MOVE    16
            KIND = INFF(20,M)                                             MOVE       50
            IF (KIND.EQ.HMAP)  GO TO 3                                    MOVE       51
            NOPQ=INFF(1,MQ)                                               MOVE       52
                                                                          MOVE       53
            SXSW = .FALSE.                                                MOVE       54
            IF (NOPQ.NE.HMAG)  GO TO 12                                   MOVE       55
            CALL DATA (MQ,1,1,6,PAR)                                      MOVE       56
            IF (PAR(4).EQ.0.) QDSW = .TRUE.                               MOVE       57
            L=PAR(1)                                                      MOVE       58
            K=PAR(2)/PAR(3)/(1.+V(6) )                                    MOVE       59
            GO TO 11                                                      MOVE       60
          3 SXSW = .TRUE.                                                 MOVE       61
            CALL DATA (MQ,1,1,4,PAT)                                      MOVE       62
            L = PAT(1)                                                    MOVE       63
            GO TO 11                                                      MOVE       64
         12 CALL RTRV7 (MQ,T,RW)                                          MOVE       65
            L=RW(1)                                                       MOVE       66
            K = 0.                                                        MOVE       67
         11 IF (M.NE.N)  GO TO 7                                          MOVE       68
            NAVEC = BDAT(M,2)                                             MOVE       69
            IF (NAVEC.EQ.4H    ) GO TO 7                                  MOVE       70
            MVEC = MDAT(M,2)                                              MOVE       71
            CALL DATA (MVEC,5,1,7,V)                                      MOVE       72
          7 NAPAR = BDAT(M,3)                                             MOVE       73
            MPAR = M                                                      MOVE       74
            HZ = 0.                                                       MOVE       75
            THETZ = 0.                                                    MOVE       76
            NVARS = 4                                                     MOVE       77
            IF (NAPAR.EQ.4H    ) GO TO 1                                  MOVE       78
            NVARS = 6                                                     MOVE       79
            MPAR = MDAT(M,3)                                              MOVE       80
          1 CALL DATA(MPAR,1,1,NVARS,PARR)                                MOVE       81
            DO111 I=1,NVARS                                               MOVE       82
            PAR(I)=PARR(I)                                                MOVE       83
            IF(KB.GE.1) PAR(I)=PAR(I)*(RANF(D)-.5)                        MOVE       84
        111 CONTINUE                                                      MOVE       85
            TLX=L*DHX/2.                                                  MOVE       86
            HX1=HX-TLX                                                    MOVE       87
            HX2=HX+TLX                                                    MOVE       88
            TLY=L*DHY/2.                                                  MOVE       89
            HY1=HY-TLY                                                    MOVE       90
            HY2=HY+TLY                                                    MOVE       91
                                                                          MOVE       92
          2 ML(1)=M7END-1                                                 MOVE       93
            ML(2)=M7END-2                                                 MOVE       94
            ML(3)=M7END-6                                                 MOVE       95
                                                                          MOVE       96
```

```
                                                                 MOVE      98
      CALL RTRV7(MUNIT,T,RW)                                     MOVE      99
      T(1,7)=-HX1                                                MOVE     100
      T(2,7)=-DHX                                                MOVE     101
      T(3,7)=-HY1                                                MOVE     102
      T(4,7)=-DHY                                                MOVE     103
      CALL STOR7(M7END-2,T,RW1)                                  MOVELL     3
                                                                 MOVE     105
      IF(HZ.EQ.0..AND.THETZ.EQ.0.) GO TO 201                     MOV883     2
      CALL DRIFT(M7END-1,HZ)                                     MOV883     3
      CALL ROTZ(THETZ,T)                                         MOVE     106
      CALL STOR7(M7END-6,T,RW1)                                  MOVELL     4
                                                                 MOVE     108
      CALL MMM(M7END-2,3,ML)                                     MOVE     109
                                                                 MOVE     110
  201 CONTINUE                                                   MOV883     4
      CALL RTRV7(M7END-2,T,RW)                                   MOVE     111
      CALL MXV7(T,V,V)                                           MOVE     112
      IF (.NOT.QDSW)  GO TO 20                                   MOVE     113
      CALL MODQ (MQ,N,V,W)                                       MOVE     114
      GO TO 8                                                    MOVE     115
   20 IF (.NOT.SXSW)  GO TO 21                                   MOVE     116
      CALL SXTP(N,V,PAT)                                         MOVE     117
      GO TO 14                                                   MOVE     118
   21 CALL DATA(MQ,1,1,6,PP)                                     MOV883     5
      PP(3) = PP(3)*(1.+V(6))                                    MOV883     6
      PP(4) = PP(4)+ PP(2)*(V(1)+V(2)*L/2.)                      MOV883     7
      CALL MAGNET(N,PP)                                          MOV883     8
      CALL RTRV7(N,T,RW)                                         MOV883     9
      CALL MXV7(T,V,W)                                           MOVE     120
C                                                                MOVE     121
    8 XIN=V(1)                                                   MOVE     122
      YIN=V(3)                                                   MOVE     123
      XOUT=W(1)                                                  MOVE     124
      YOUT=W(3)                                                  MOVE     125
      DO 112 I=1,NVARS                                           MOVE     126
  112 PAR(I)=PARR(I)                                             MOVE     127
      CALL STDAT(M,1,7,4,PAR)                                    MOVC       1
      DO 9 I=1,7                                                 MOVE     129
    9 V(I) = W(I)                                                MOVE     130
                                                                 MOVE     131
   14 CALL RTRV7 (MUNIT,T,RW)                                    MOVE     132
      T(1,7)=-HX2                                                MOVE     133
      T(2,7)=-DHX                                                MOVE     134
      T(3,7)=-HY2                                                MOVE     135
      T(4,7)=-DHY                                                MOVE     136
      CALL STOR7(M7END-7,T,RW)                                   MOVE     137
                                                                 MOVE     138
      ML(1)=-(M7END-6)                                           MOVE     139
      ML(2) = -(M7END-7)                                         MOVE     140
      ML(3)=-(M7END-1)                                           MOVE     141
      IF(HZ.EQ.0..AND.THETZ.EQ.0.) ML(1) = ML(3) = MUNIT         MOV883    10
      CALL MMM(M7END-7,3,ML)                                     MOVE     142
                                                                 MOVE     143
      ML(1)=M7END-2                                              MOVE     144
      ML(2)=N                                                    MOVE     145
      ML(3) = M7END - 7                                          MOVE     146
```

```
                                                    MOVE      147
      CALL RTRV7(M7END-7,T,RW)                      MOVELL      5
                                                    MOVELL      6
      CALL MXV7(T,V,V)                              MOVE      149
      IF(N.EQ.0) GO TO 113                          MOVE      150
      CALL MMM(N,3,ML)                              MOVE      151
      IF(M.NE.N) GO TO 113                          MOVE      152
      CALL RTRV7(N,T,RW)                            MOVE      153
      CALL ROW5(T)                                  MOVE      154
      CALL STOR7(N,T,RW)                            MOVE      155
                                                    MOVE      156
  113 IF (KB.LT.2)  RETURN                          MY3MOVE    17
      IF (TOG) GO TO 115                            MY3MOVE    18
      TOG = .TRUE.                                  MY3MOVE    19
      RETURN                                        MY3MOVE    20
  115 CALL RANSET(NRN)                              MY3MOVE    21
      TOG = .FALSE.                                 MY3MOVE    22
      RETURN                                        MOVE      158
      END                                           MOVE      159
```

```
1
            SUBROUTINE MPCONJ(A,B,C,DC)                           MOVE      160
            RETURN                                                MPCONJ      2
            END                                                   MPCONJ      3
                                                                  MPCONJ      4
```

```
      SUBROUTINE MPY2(A,B,C)                                            MPY2       2
      RETURN                                                            MPY2       3
      END                                                               MPY2       4
```

```
      SUBROUTINE MXD(M,M1,M2,KD)                                        MXD       2
C MULTIPLIES PAIRS OF MATRICES OF ELEMENTS M1, AND M2 AND               MXD       3
C     STORES AT LQ(M).  BEAM ORDER IS M1 FOLLOWED BY M2.                MXD       4
C R(M)=R(M1)*R(M2),  MULTIPLY AND DIVIDE                                MXD       5
                                                                        MXD       6
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER   SWTCH     2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                          SWTCH     3
                                                                        SWTCH     4
      DIMENSION RX(2,3),RY(2,3)                                         MXD       8
      DIMENSION RX1(2,3),RX2(2,3),RY1(2,3),RY2(2,3)                     MXD       9
      DIMENSION RW(3),RW1(3),RW2(3),RPQ(4)                              MXD      10
      DIMENSION T(49) ,T1(49),T2(49)                                    MXD      11
                                                                        MXD      12
      DO 5 I=1,3                                                        MXD      13
    5 RW(I) = 0.                                                        MXD      14
      IF(KD.EQ.3) GO TO 100                                             MXD      15
*        MULTIPLY MATRICES AS 7 X 7, THEN STORE ACCORDING TO KIND       MXD      16
  200 CALL RTRV7(M1,T1,RW1)                                             MXD      17
  205 CALL RTRV7(M2,T2,RW2)                                             MXD      18
      CALL MXM7(T2,T1,T)                                                MXD      19
      RW(1) = RW1(1) + RW2(1)                                           MXD      20
      RW(2) = RW1(2) + RW2(2)                                           MXD      21
      RW(3) = T(40)                                                     MXD      22
      IF (KD.EQ.3) GO TO 210                                            MXD      23
      CALL STOR7(M,T,RW)                                                MXD      24
      RETURN                                                            MXD      25
  210 CALL CONVMAT(KD,T,RX,RY,RW,RPQ)                                   MXD      26
      CALL STXY(M,RX,RY,RW)                                             MXD      27
      CALL STDAT(M,5,4,4,RPQ)                                           MXD      28
      RETURN                                                            MXD      29
  100 CONTINUE                                                          MXD      30
      CALL RXY(M1,RX1,RY1,RW1)                                          MXD      31
      CALL RXY(M2,RX2,RY2,RW2)                                          MXD      32
      DO 102 I=1,2                                                      MXD      33
      DO 101 J=1,2                                                      MXD      34
      RX(I,J)=RX2(I,1)*RX1(1,J)+RX2(I,2)*RX1(2,J)                       MXD      35
  101 RY(I,J)=RY2(I,1)*RY1(1,J)+RY2(I,2)*RY1(2,J)                       MXD      36
      RX(I,3)=RX2(I,1)*RX1(1,3)+RX2(I,2)*RX1(2,3)+RX2(I,3)              MXD      37
      RY(I,3)=RY2(I,1)*RY1(1,3)+RY2(I,2)*RY1(2,3)+RY2(I,3)              MXD      38
  102 RW(I)=RW1(I)+RW2(I)                                               MXD      39
      DO 103 I=1,2                                                      MXD      40
      RPQ(I)   = RX(1,3)*RX(2,I)-RX(2,3)*RX(1,I)                        MXD      41
  103 RPQ(I+2)= RY(1,3)*RY(2,I)-RY(2,3)*RY(1,I)                         MXD      42
      RW(3)=RW1(3)+RW2(3)-RX1(1,3)*(RX2(2,3)*RX2(1,1)-RX2(1,3)*RX2(2,1)) MXD      43
     1                    -RY1(1,3)*(RY2(2,3)*RY2(1,1)-RY2(1,3)*RY2(2,1)) MXD      44
     2                    -RY1(2,3)*(RY2(2,3)*RY2(1,2)-RY2(1,3)*RY2(2,2)) MXD      45
     3                    -RX1(2,3)*(RX2(2,3)*RX2(1,2)-RX2(1,3)*RX2(2,2)) MXD      46
      CALL STXY(M,RX,RY,RW)                                             MXD      47
      CALL STDAT(M,5,4,4,RPQ)                                           MXD      48
      RETURN                                                            MXD      49
      END                                                               MXD      50
```

```
      SUBROUTINE  MXDIV(V,IN,IM,DET,IDIM)
C                         MATRIX DIVISION
C     REPLACES B BY (A INVERSE)*B.  DESTROYS A.
C     A IS N BY N, B IS N BY M.  A AND B ARE STORED IN V(N,N+M).
C     N MUST NOT EXCEED 4 .
      INTEGER PERM,P
      DIMENSION PERM(4)
      DIMENSION V(IDIM,1)
      N=IN
      M=MAX0(IM,0)
C
      NM1=N-1
      NP1=N+1
      NPM=N+M
  200 DO 201 I=1,N
      PERM(I)=I
  201 CONTINUE
      DETERM=1.0
      P=1
  202 CONTINUE
      IMIN=P+1
C                              FIND PIVOT
      PIVOT=0.
      DO 209 I=P,N
      DO 208 J=P,N
      SAVE= ABS(V(I,J))
      IF (SAVE .LE. PIVOT) GO TO 208
      PIVOT=SAVE
      IBIG=I
      JBIG=J
  208 CONTINUE
  209 CONTINUE
      IF (PIVOT .GT. 0.0  ) GO TO 210
      DETERM=0.0
      GO TO 290
C                              ROW EXCHANGE AND ROW REDUCTION
  210 PIVOT=V(IBIG,JBIG)
      DETERM=DETERM*PIVOT
      DO 219 J=P,NPM
      SAVE=V(IBIG,J)/PIVOT
      V(IBIG,J)=V(P,J)
      V(P,J)=SAVE
  219 CONTINUE
      IF(IBIG .NE. P) DETERM=-DETERM
  220 CONTINUE
      IF (P .GE. N) GO TO 250
C                              COLUMN EXCHANGE
      IF (JBIG .EQ. P) GO TO 230
      ISAVE=PERM(P)
      PERM(P)=PERM(JBIG)
      PERM(JBIG)=ISAVE
      DO 229 I=1,N
      SAVE=V(I,P)
      V(I,P)=V(I,JBIG)
      V(I,JBIG)=SAVE
  229 CONTINUE
      DETERM=-DETERM
```

```
  230 CONTINUE
      V(P,P)=1.0
C                               BLOCK REDUCTION
      DO 239 I=IMIN,N
      DO 238 J=IMIN,NPM
      V(I,J)=V(I,J)-V(I,P)*V(P,J)
  238 CONTINUE
      V(I,P)=0.0
  239 CONTINUE
      P=P+1
  249 GO TO 202
C                               BACK SUBSTITUTION
  250 CONTINUE
      IF (M .LE. 0) GO TO 290
      DO 259 K=NP1,NPM
      V(N,1)=V(N,K)
      DO 253 P=1,NM1
      I=N-P
      V(I,1)=V(I,K)
      DO 252 J=I,NM1
      V(I,1)=V(I,1)-V(J+1,1)*V(I,J+1)
  252 CONTINUE
  253 CONTINUE
      DO 256 J=1,N
      P=PERM(J)
      V(P,K)=V(J,1)
  256 CONTINUE
  259 CONTINUE
C
  290 CONTINUE
      DET=DETERM
      RETURN
        END
```

```
      SUBROUTINE MXM6(A,B,C)

C  MULTIPLES 6 X 6 MATRICES
      DIMENSION A(6,1),B(6,1),C(6,1),CC(6,6)
      DO 100 I=1,6
      DO 90 J=1,6
      SUM = 0.
      DO 80 II=1,6
   80 SUM = SUM + A(I,II)*B(II,J)
   90 CC(I,J) = SUM
  100 CONTINUE
      DO 115 I=1,6
      DO 115 J=1,6
      C(I,J) = CC(I,J)
  115 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MXM7(A,B,C)                                            MXM7       2
                                                                        MXM7       3
      DIMENSION A(7,1),B(7,1),C(7,1),CC(7,7)                            MXM7       4
C  MULTIPLES 7 X 7 MATRICES                                             MXM7       5
      DO 100 I=1,7                                                      MXM7       6
      DO 90 J=1,7                                                       MXM7       7
      SUM = 0.                                                          MXM7       8
      DO 80 II=1,7                                                      MXM7       9
   80 SUM = SUM + A(I,II)*B(II,J)                                       MXM7      10
   90 CC(I,J)= SUM                                                      MXM7      11
  100 CONTINUE                                                          MXM7      12
      DO 115 I=1,7                                                      MXM7      13
      DO 115 J=1,7                                                      MXM7      14
      C(I,J) = CC(I,J)                                                  MXM7      15
  115 CONTINUE                                                          MXM7      16
      RETURN                                                            MXM7      17
      END                                                               MXM7      18
```

```
      SUBROUTINE MXV6(T,V,VN)                                    MXV6     2
                                                                 MXV6     3
C  MULTIPLES A 6 X 6 MATRIX BY A 6 COMPONENT VECTOR              MXV6     4
      DIMENSION  T(6,6),V(6),VN(6),VV(6)                         MXV6     5
      DO 10 I=1,6                                                MXV6     6
      VV(I) = 0.                                                 MXV6     7
      DO 10 J=1,6                                                MXV6     8
      VV(I) = VV(I) + T(I,J)*V(J)                                MXV6     9
   10 CONTINUE                                                   MXV6    10
      DO 15 J=1,6                                                MXV6    11
   15 VN(J) = VV(J)                                              MXV6    12
      RETURN                                                     MXV6    13
      END                                                        MXV6    14
```

```
      SUBROUTINE MXV7(T,V,VN)

C  MULTIPLIES MATRIX BY A VECTOR
      DIMENSION T(7,7),V(7),VN(7),VV(7)
      DO 10 I=1,7
      VV(I) = 0.
      DO 10 J=1,7
      VV(I) = VV(I) + T(I,J)*V(J)
   10 CONTINUE
      DO 15 J=1,7
   15 VN(J) = VV(J)
      RETURN
      END
```

```
      SUBROUTINE NAMCHK(M)                                            NAMCHK     2
C  LOOKS FOR IDENTICAL NAME PREVIOUSLY DEFINED.                       NAMCHK     3
C   BLANKS OR NO NAMES ARE NOT CONSIDERED IDENTICAL.                  NAMCHK     4
C     INFO OF PREVIOUSLY DEFINED ELEMENTS IS REPLACED BY THAT OF THE  NAMCHK     5
C     LAST DEFINED ONE, EXCEPT FOR INFO(3,I),WHICH CONTAINS NXTM      NAMCHK     6
                                                                      BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                       BLANK      3
      COMMON STORE(48000),IWORK(10)                                   86MARSIZ   1
      DIMENSION INFF(24,2000)                                         86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                        86MARSIZ   3
                                                                      BLANK      5
      COMMON/STORE/LMAX,LINF,LFILE,MAX,MIN,ICARD(11)                  STORE      2
*  LMAX AND MAX ARE DIMENSIONS OF STORE AND INFF                      STORE      3
                                                                      STORE      4
      DIMENSION INF(24)                                               NAMCHK     9
C                                                                     NAMCHK    10
      NAME=INFF(2,M)                                                  NAMCHK    11
      M1=MIN+1                                                        NAMCHK    12
      DO 120 J=M1,MAX                                                 NAMCHK    13
      INAM = INFF(2,J)                                                NAMCHK    14
      IF (INAM.EQ.NAME.AND.NAME.NE.5H     ) GO TO 105                 NAMCHK    15
      GO TO 120                                                       NAMCHK    16
C DUPLICATE NAME,NOT BLANK HAS BEEN FOUND                             NAMCHK    17
C  REPLACE INFO(I,J) WITH THAT OF INFO(I,M), EXCEPT FOR INFO(3,J) WHICH NAMCHK  18
C    CONTAINS NXTM AND TRA                                            NAMCHK    19
  105 CALL RDINF(INF,M)                                               NAMCHK    20
      INF(8) = 1H*                                                    NAMCHK    21
      INF(9) = 1H*                                                    NAMCHK    22
      CALL WRTINF(INF,J)                                              NAMCHK    23
      RETURN                                                          NAMCHK    24
  120 CONTINUE                                                        NAMCHK    25
      RETURN                                                          NAMCHK    26
      END                                                             NAMCHK    27
```

```
      SUBROUTINE NPOL(M,V,PAR)                                           NPOLJUN83   1
C                                                                        NPOM        3
C  Q   NPOL   N   J    L       CM     BRHO                               NPOM        4
C   N = ORDER OF PERTURBATION IN HAMILTONIAN, M=N-1                      NPOM        5
C   J =0,1  FOR NORMAL OR SKEW N-POLE LENS                               NPOM        6
C   L = EEFFECTIVE LENGTH                                                NPOM        7
C   CM = TAYLOR EXPANSION COEFFICIENT OF MEDIAN-PLANE EXPRESSION OF FIEL NPOM        8
C                                                                        NPOM        9
C J=0: BY + I*BX = SUM(CM/M!) Z**M     (Z = X+I*Y)                       NPOLJUN83   2
C J=1: BX - I*BY = SUM(CM/M!) Z**M                                       NPOLJUN83   3
C      PARTICLE  V(I) = X,PX,Y,PY,O,O,DPP                                NPOM       12
C      N =2   QUAD,  N=3  SEXT,   N=4 OCT,   N=5,DEC                     NPOM       13
C                                                                        NPOM       14
      COMMON /ML/ ML    $         LEVEL 2, ML                            NPOLLL      1
                                                                         NPOLLL      2
                                                                         BLANK       2
      LEVEL 2, STORE,INFF,IWORK                                          BLANK       3
      COMMON STORE(48000),IWORK(10)                                      86MARSIZ    1
      DIMENSION INFF(24,2000)                                            86MARSIZ    2
      EQUIVALENCE (INFF,STORE)                                           86MARSIZ    3
                                                                         BLANK       5
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL    DIM         2
                                                                         DIM         3
                                                                         NPOM       18
      DIMENSION V(1),PAR(1),R(7,7),RW(3),ML(3),PR(3),W(6)                NPOM       19
      REAL L,LH                                                          NPOM       20
      EQUIVALENCE (PR(1),L),(PR(2),CM),(PR(3),BRHO),(W(1),X),            NPOM       21
     1            (W(2),PX),(W(3),Y),(W(4),PY),(W(6),DPP)                NPOM       22
      COMPLEX II,Z,DP,Q                                                  NPOL682     1
      DATA II/(0.,1.)/                                                   NPOL682     2
C                                                                        NPOM       23
      NN=M                                                               NPOLJUN83   4
      DO 5 I=1,3                                                         NPOM       24
    5 PR(I) = PAR(I)                                                     NPOM       25
      DO 6 I=1,6                                                         NPOM       26
    6 W(I) = V(I)                                                        NPOM       27
      BR = (1.+DPP)*BRHO                                                 NPOM       28
      UM = CM/BR                                                         NPOM       29
      IF (L.EQ.0.)  GO TO 7                                              NPOM       30
      LH = L/2.                                                          NPOM       31
      X = X + PX*LH                                                      NPOM       32
      Y = Y + PY*LH                                                      NPOM       33
      UM=L*UM                                                            NPOLJUN83   5
    7 NP = INFF(4,NN)                                                    NPOM       35
      JP = INFF(5,NN)                                                    NPOM       36
      MP = NP - 1                                                        NPOM       37
      MPM=NP-2                                                           NPOL682     3
      Z=X+II*Y                                                           NPOL682     4
      CALL RTRV7(MUNIT,R,RW)                                             NPOM       38
C                                                                        NPOM       39
      Q=UM                                                               NPOL682     5
      IF(JP.EQ.1)Q=II*Q                                                  NPOL682     6
      IF(MPM.EQ.0) GO TO 12                                              NPOL682     7
      DO11 I=1,MPM                                                       NPOL682     8
   11 Q=Q*Z/I                                                            NPOL682     9
   12 DP=Q*Z/MP                                                          NPOL682    10
      R(2,1)=-REAL(Q)                                                    NPOL682    11
```

```
58            R(2,3)=AIMAG(Q)                                   NPOLJUN83    6
59            DPX=-REAL(DP)                                     NPOL682     13
60            DPY=AIMAG(DP)                                     NPOL682     14
61      C                                                       NPOM        86
62       40   R(4,3) = -R(2,1)                                  NPOM        87
63            R(4,1) = R(2,3)                                   NPOM        88
64            V(2) = PX + DPX                                   NPOM        89
65            V(4) = PY + DPY                                   NPOM        90
66            IF (L.EQ.0)  GO TO 41                             NPOM        91
67            V(1) = X + V(2) *LH                               NPOM        92
68            V(3)=Y+V(4)*LH                                    NPOLJUN83    7
69      C                                                       NPOM        94
70       41   R(2,6) = -DPX                                     NPOM        95
71            R(2,7) = -DPX                                     NPOM        96
72            R(4,6) = -DPY                                     NPOM        97
73            R(4,7) = -DPY                                     NPOM        98
74            R(5,1) = DPX                                      NPOM        99
75            R(5,3) = DPY                                      NPOM       100
76            RW(2)=-DPX                                        NPOLJUN83    8
77            IF (L.EQ.0)  GO TO 45                             NPOM       101
78            ML(1) = MEND - 3                                  NPOM       102
79            ML(2) = M                                         NPOM       103
80            ML(3) = MEND - 3                                  NPOM       104
81            CALL DRIFT(MEND-3,LH)                             NPOM       105
82            CALL STOR7(M,R,RW)                                NPOM       106
83            CALL MMM(M,3,ML)                                  NPOM       107
84            RETURN                                            NPOM       108
85       45   CALL STOR7(M,R,RW)                                NPOM       109
86            RETURN                                            NPOM       110
87            END                                               NPOM       111
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
64	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
65	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
67	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
68	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
68	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

```
      INTEGER FUNCTION OPNUM(OPNAME)                                   OPNUM      2
                                                                       OPNUM      3
*   CHECKS OPNAME AGAINST LIST OF ACCEPTABLE NAMES                     OPNUM      4
*   ASSIGNS AN OP NUMBER TO OPNAME.                                    OPNUM      5
                                                                       OPNUM      6
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,   CONTRL     2
     1          LDFLG,FIN                                              CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN     CONTRL     4
                                                                       CONTRL     5
      COMMON/GRR/IERR,BSW,VSW,RSW,QSW,XSW                              GRR        2
      LOGICAL IERR,BSW,VSW,RSW,QSW,XSW                                 GRR        3
                                                                       GRR        4
      COMMON/OPLIST/LIST(200)                                          OPLIST     2
                                                                       OPLIST     3
                                                                       OPNUM     10
      INTEGER OPNAME                                                   OPNUM     11
      DECODE(10,8,OPNAME) KOP                                          OPNUM     12
    8 FORMAT (A3,7X)                                                   OPNUM     13
*                        LOOK FOR MAP INSTRUCTION.                     OPNUM     14
*                             MAP1 THRU MAP9 ARE TREATED ALIKE.        OPNUM     15
      IF (KOP.EQ.3HMAP) GO TO 110                                      OPNUM     16
*   RUN THROUGH LIST                                                   OPNUM     17
      DO 100 I=1,200                                                   OPNUM     18
      IF (OPNAME.EQ.LIST(I)) GO TO 105                                 OPNUM     19
  100 CONTINUE                                                         OPNUM     20
*   IF OPNAME IS ILLEGAL, OP IS SET TO 0 AND ERROR SWITCH IS TURNED ON. OPNUM     21
      OPNUM = 0                                                        OPNUM     22
*     IF OPNAME IS BLANK, IT IS ERROR UNLESS IERR IS ON.               OPNUM     23
      IF (OPNAME.NE.4H    ) GO TO 102                                  OPNUM     24
      IF (IERR) RETURN                                                 OPNUM     25
  102 ERROR = .TRUE.                                                   OPNUM     26
      WRITE (3,1) OPNAME                                               OPNUM     27
    1 FORMAT (15H *** ERROR *** ,A5,* IS AN ILLEGAL INSTRUCTION.*)     OPNUM     28
      RETURN                                                           OPNUM     29
*                        ASSIGN INDEX NUMBER                           OPNUM     30
  105 OPNUM = I                                                        OPNUM     31
      IF(I.NE.82) RETURN                                               OPNMPL     1
      OPNUM=68                                                         OPNMPL     2
      OPNAME = 4HSXTP                                                  OPNMPL     3
      RETURN                                                           OPNUM     32
  110 OPNUM = 69                                                       OPNUM     33
      RETURN                                                           OPNUM     34
      END                                                              OPNUM     35
```

```
      SUBROUTINE ORBIT(MM)                                              ORBITC      1
C ORBIT PROGRAM USES "MICADO" TO CALCULATE OPTIMIZED CORRECTORS         ORBITPRNT   1
C  FOR GIVEN ORBIT WITH FIELD ERRORS.                                   ORBITPRNT   2
C  FORMAT IS                                                            ORBITPRNT   3
C                                                                       ORBITPRNT   4
C     ORBC  KA  KB CY   FX   MON  COR  DELM      DELY                   NOV28ORB    1
C  WHERE KA=0 FOR HORIZONTAL, 1 FOR VERTICAL CORRECTION,                ORBITPRNT   6
C  FX = NAME OF A FXPT INSTRUCTION DEFINING THE INITIAL ORBIT           ORBITPRNT   7
C  (FX MUST HAVE A 1 IN COLUMNS 55)                                     ORBITPRNT   8
C    CY IS REDUNDANT                                                    NOV28ORB    2
C  MON IS NAME OF ELEMENTS IN BEAM LINE WHERE DISPLACEMENTS ARE         ORBITPRNT  10
C  ASSUMED TO BE MEASURED                                               ORBITPRNT  11
C  COR IS NAME OF CORRECTION ELEMENTS IN BEAM LINE                      ORBITPRNT  12
C  (THERE SHOULD BE AMNY IDENTICAL MON-S AND COR-S IN BEAM LINE)        ORBITPRNT  13
C  DELM = ASSUMED MEASUREMENT ERROR (+- DELM/2)                         ORBITPRNT  14
C  DELY = TOLERABLE RANGE OF FINAL RESIDUAL ORBIT ERRORS                ORBITPRNT  15
C  (ITERATIONS WILL STOP WHEN THIS LEVEL IS REACHED)                    ORBITPRNT  16
C  ABS(KB) = MAX NO. OF ITERATIONS. 2F KB NEGATIVE, ALL INTER-          ORBITPRNT  17
C  MEDIATE ITERATIONS AS WELL AS THE MATRIX RELATING ORBIT              ORBITPRNT  18
C  DISPLACEMENTS TO CORRECTORS ARE PRINTED OUT,                         ORBITPRNT  19
C  IF KB POSITIVE, ONLY FIRST AND LAST ITERATIONS ARE PRINTED           ORBITPRNT  20
C  AND MATRIX PRINT IS SUPPRESSED.                                      ORBITPRNT  21
      COMMON/CORB/TH(84),TV(84),NMON,NNH,NNV,IIH,IIV,                   CORB613     1
     .            NAMONH,NAMONV,NACORH,NACORV                           CORB613     2
                                                                        BLANK       2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK       3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ    1
      DIMENSION INFF(24,2000)                                           86MARSIZ    2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ    3
                                                                        BLANK       5
      COMMON/BMI1/MI1(16000)                                            NV3BMI1     1
      COMMON/NELS1/NELS1                                                BMI1L       2
      LEVEL 2,MI1                                                       BMI1L       3
                                                                        BMI1L       4
C                                                                       BMI1        3
      LEVEL 2,   BETA,ALPHA,AP,B,NB,R,X,RHO,NX,NV,XA,NA,NC,XX,A,AA,BB   DEBUG       1
      COMMON/AAA/BETA,ALPHA,AP,B,NB,R,X,RHO,NX,NV,XA,NA,NC,XX,A,AA,BB   DEBUG       2
      DIMENSION BETA(2,84),ALPHA(84),AP(84),B(84),NB(1),R(84),X(84),    ORBIT       6
     .RHO(252),NX(84),NV(84),XA(1),NA(1),NC(1),XX(84),A(84,84),DUM(7),  ORBIT       7
     .AA(84,84),BB(84)                                                  ORBIT       8
      REAL MX,MY,MU(2,84),NU,K(84),L(84),M11,M12                        ORBIT       9
      LOGICAL KPR,KPP                                                   ORBITPRNT  22
      INTEGER BDAT                                                      ORBITD      1
C                                                                       ORBIT      12
      PR(A,B,C,D)=A*C+B*D                                               ORBIT      13
      CALL RANGET(NRR)                                                  ORBITC      4
      MFXP = MDAT(MM,2)                                                 ORBITC      6
      KA = INFF(4,MM)                                                   ORBITC      7
      KB = INFF(5,MM)                                                   ORBITC      8
      MBML = MDAT(MFXP,2)                                               NOV28ORB    3
      NDAT=INFF(17,MBML)                                                ORBIT      19
      CALL MIFILL(MBML,1,NDAT,NELS1,MI1)                                ORBIT      20
      MON = BDAT(MM,3)                                                  ORBITD      2
      LM1 = LOCC(MON,1,NELS1,MI1,NB,NMON)                               DEBUG       3
      M=NMON                                                            ORBIT      22
      KPR = .F.                                                         ORBITPRNT  23
      IF(KB.LE.0) KPR = .T.                                             ORBITPRNT  24
```

```
      ITER = IABS(KB) $ IF(KB.EQ.0) ITER = 1 $ KPP=KPR               ORBITPRNT  25
      IFLAG=KA                                                        ORBIT      24
      EPS = FDAT(MM,1)                                                ORBITC     10
      EPSB = FDAT(MM,2)                                               ORBITC     11
      EPSZ=EPS                                                        ORBITPRNT  26
      NACOR = BDAT(MM,4)                                              ORBIT613    1
      LCOR = LOCC(NACOR,1,NELS1,MI1,NX,N)                             ORBIT613    2
      IF (KA.EQ.1) GO TO 9                                            ORBIT613    3
      NAMONH = MON                                                    ORBIT613    4
      NACORH = NACOR                                                  ORBIT613    5
      NNH = N                                                         ORBIT613    6
      GO TO 26                                                        ORBIT613    7
 9    NAMONV = MON                                                    ORBIT613    8
      NACORV = NACOR                                                  ORBIT613    9
      NNV = N                                                         ORBIT613   10
 26   CONTINUE                                                        ORBIT613   11
C                                                                     ORBIT      35
      PRINT 3                                                         ORBIT      36
    3 FORMAT(1H1)                                                     ORBIT      37
      PRINT 106,EPSB,EPS                                              ORBIT      38
  106 FORMAT(1X*DISPERSION DE MESURE *,F10.5/                         ORBIT      39
     .        1X*C.A.C. RESIDUEL 0     *,F10.5/)                      ORBIT      40
      PRINT 100                                                       ORBIT      41
  100 FORMAT(1X*CORRECTEURS*/)                                        ORBIT      42
      PRINT 2,(NX(I),I=1,N)                                           ORBIT      43
 2    FORMAT (2I5)                                                    ORBITC     12
      PRINT 4                                                         ORBIT      45
      PRINT 101                                                       ORBIT      46
  101 FORMAT(1X*STATIONS*/)                                           ORBIT      47
      PRINT 1,(NB(I),I=1,M)                                           ORBIT      48
 1    FORMAT (2I5)                                                    ORBITC     13
      PRINT 4                                                         ORBIT      50
    4 FORMAT(1H0)                                                     ORBIT      51
      IB=1                                                            ORBIT      52
      IC=1                                                            ORBIT      53
      REWIND 5                                                        ORBITC     14
      IF(IFLAG.EQ.0) GO TO 10                                         ORBIT      54
      PX=0.                                                           ORBIT      55
      PY=1.                                                           ORBIT      56
      GO TO 11                                                        ORBIT      57
   10 PX=1.                                                           ORBIT      58
      PY=0.                                                           ORBIT      59
   11 READ(5) IEL                                                     ORBIT      60
      READ (5)                                                        NOV28ORB    4
      PI=4.*ATAN(1.)                                                  ORBIT      61
      ICC=0                                                           ORBIT      62
      DO 12 I=1,IEL                                                   ORBIT      63
      READ(5)IE ,NO,NAME,DUM,BY,BX,AY,AX,MY,MX                        ORBIT      64
      READ (5) EX,EXP,EY,EYP,XP,DXP,Y,DYP                             NOV28ORB    5
      IF(NO.EQ.NB(IB)) GO TO 13                                       ORBIT      66
      IF(NO.EQ.NX(IC)) GO TO 14                                       ORBIT      67
      GO TO 12                                                        ORBIT      68
   13 IA=1                                                            ORBIT      69
      IX=IB                                                           ORBIT      70
      IB=IB+1                                                         ORBIT      71
      B      (IX)=PR(XP, Y,PX,PY)+(RGEN(0)-.5)*EPSB                   ORBIT      72
      AP(IX) = 0.                                                     ORBITC     16
```

```
      GO TO 15                                             ORBIT      74
   14 IF(ICC.NE.0) GO TO 21                                ORBIT      75
      NX(IC)=NX(IC)+1                                      ORBIT      76
      ICC=1                                                ORBIT      77
      IA=2                                                 ORBIT      78
      IX=IC                                                ORBIT      79
      L      (IX)=DUM(1)                                   ORBITD      6
      K      (IX)=PR(-DUM(5),DUM(5),PX,PY)                 ORBITD      7
      ALPHA  (IX)=PR(AX,AY,PX,PY)                          ORBIT      80
   15 BETA(IA,IX)=PR(BX,BY,PX,PY)                          ORBIT      81
      MU  (IA,IX)=PR(MX,MY,PX,PY)                          ORBIT      82
      IF(NO.EQ.NX(IC)) GO TO 14                            ORBIT      83
      GO TO 12                                             ORBIT      84
   21 ICC=0                                                ORBIT      85
      IX=IC                                                ORBIT      86
      IC=IC+1                                              ORBIT      87
   12 CONTINUE                                             ORBIT      90
      CALL RANSET(NRR)                                     ORBITC     17
      NU=PR(MX,MY,PX,PY)*PI                                ORBIT      91
      IMAX=MXMIN(B,M,1)  $ IMIN = MXMIN(B,M,0)             ORBITPRNT  27
      DB = (B(IMAX)-B(IMIN))/1000.                         ORBITPRNT  28
      PRINT 203,DB                                         ORBITPRNT  29
  203 FORMAT(*0BEFORE CORRECTIONS RMAX-RMIN = *E13.4)      ORBITPRNT  30
      PRINT 102                                            ORBIT      92
  102 FORMAT(* ORBITE AVANT CORRECTION(MM)*/)              ORBITC     18
      PRINT 7,( B(I),I=1,M)                                ORBIT      94
    7 FORMAT(7F10.5)                                       ORBIT      95
      SP=2.*SIN(NU)                                        ORBIT      96
      DO 118 I=1,M                                         ORBITD      8
      SBI = SQRT(BETA(1,I))                                ORBITD      9
      DO 118 J=1,N                                         ORBITD     10
      DM=(MU(1,I)-MU(2,J))*2.*PI+NU                        ORBIT      99
      IF(MU(2,J).LT.MU(1,I)) DM=DM-2.*NU                   ORBIT     100
      SB=SQRT(BETA(2,J))                                   ORBIT     101
      CM=COS(DM)                                           ORBIT     102
      IF(I.GT.1) GO TO111                                  ORBITD     11
      NX(J) = NX(J)-1                                      ORBITD     12
  111 CONTINUE                                             ORBITD     13
      IF (L(J).EQ.0.) GO TO 116                            ORBITD     14
      SM=SIN(DM)                                           ORBIT     103
      IF(K(J)) 17,18,19                                    ORBIT     104
   17 AK=ABS(K(J))                                         ORBIT     105
      SK=SQRT(AK)                                          ORBIT     106
      M11=SINH(SK*L(J))/SK                                 ORBIT     107
      M12=(COSH(SK*L(J))-1.)/AK                            ORBIT     108
      GO TO 16                                             ORBIT     109
   18 M11=L(J)                                             ORBIT     110
      M12=L(J)*L(J)/2.                                     ORBIT     111
      GO TO 16                                             ORBIT     112
   19 SK=SQRT(K(J))                                        ORBIT     113
      M11=SIN(SK*L(J))/SK                                  ORBIT     114
      M12=(1.-COS(SK*L(J)))/K(J)                           ORBIT     115
   16 A(I,J)=SQRT(BETA(1,I))*(SB*CM*M11+(SM-ALPHA(J)*CM)*M12/SB)/SP   ORBIT     116
      GO TO 118                                            ORBITD     15
  116 A(I,J) = SBI*SB*CM/SP                                ORBITD     16
  118 CONTINUE                                             ORBITD     17
      IF(.NOT.KPP) GO TO 1181                              ORBITPRNT  31
```

```
      PRINT 4                                                           ORBIT      117
      PRINT 103                                                         ORBIT      118
  103 FORMAT(1X*MATRICE A*/)                                            ORBIT      119
      PRINT 8,((A(I,J),J=1,N),I=1,M)                                    ORBIT      120
    8 FORMAT(6(1X12F10.4/)/)                                            ORBIT      121
 1181 CONTINUE                                                          ORBITPRNT   32
      DO 22 J=1,ITER                                                    ORBIT      122
      PRINT 4                                                           ORBIT      123
      DO 23 I1=1,M                                                      ORBIT      124
      BB(I1)=B(I1)/1000.                                                ORBITC      21
      DO 23 I2=1,N                                                      ORBIT      126
   23 AA(I1,I2)=A(I1,I2)                                                ORBIT      127
      JB=J                                                              ORBIT      128
      JTER=J                                                            ORBIT      129
      CALL MICADO(AA,BB,84,M,N,AP,0.,0 ,0 ,0 ,EPS,JTER,DP,X,NV,R,RHO)   ORBIT      130
      IF(J.EQ.1.OR.J.EQ.ITER.OR.EPS.LT.EPSZ)KPP=.T.                     ORBITPRNT   33
      IF(JTER.LT.JB) RETURN                                             ORBIT      131
      IF(KPP) PRINT 104                                                 ORBITPRNT   34
  104 FORMAT(1X*CORRECTIONS    MRAD   MRAD,MM*/)                        ORBIT      133
      DO 20 I=1,J                                                       ORBIT      134
      X(I)=X(I)*1000.                                                   ORBIT      135
      XX(I)=X(I)                                                        ORBIT      136
      NK=NV(I)                                                          ORBIT      137
      IF(K(NK).NE.0.) XX(I)=X(I)/K(NK)                                  ORBIT      138
      IF(KPP) PRINT 5,NK,NX(NK),X(I),XX(I)                              ORBITPRNT   35
      IF(IFLAG.EQ.0) TH(NV(I))=-X(I)/1000.                              ORBIT      140
      IF(IFLAG.EQ.1) TV(NV(I))= X(I)/1000.                              ORBIT      141
   20 CONTINUE                                                          ORBIT      142
    5 FORMAT(2I5,2F10.5)                                                ORBIT      143
      DO 24 I=1,M                                                       ORBIT      144
   24 R(I)=R(I)*1000.                                                   ORBIT      145
      IF(.NOT.KPP) GO TO 22                                             ORBITPRNT   36
      IMAX = MXMIN(R(1),M,1)                                            ORBITPRNT   37
      IMIN = MXMIN(R(1),M,0)                                            ORBITPRNT   38
      EPS1=R(IMAX)-R(IMIN)                                              ORBITPRNT   39
      EPS1 = EPS1/1000.                                                 ORBITPRNT   40
      PRINT 201,J,EPS1                                                  ORBITPRNT   41
  201 FORMAT(1H0,5HAFTER,I3,22H ITERATIONS,RMAX-RMIN=,E13.4)            ORBITPRNT   42
      PRINT 4                                                           ORBIT      146
      PRINT 105                                                         ORBIT      147
  105 FORMAT(1X*ORBITE CORRIGEE (MM)*/)                                 ORBIT      148
      PRINT 6,(I,R(I),I=1,M)                                            ORBITPRNT   43
 22   KPP=KPR                                                           ORBITPRNT   44
    6 FORMAT(7(I5,F10.5))                                               ORBIT      150
      RETURN                                                            ORBIT      151
      END                                                               ORBIT      152
```

```
      SUBROUTINE MICADO(A,B,NDIM,M,N,AP,XA,NA,NB,NC,EPS,ITER,DP,X,NX,R, MICADO     2
     1RHO)                                                               MICADO     3
C*********************************************************************** MICADO     4
      LEVEL 2,A,B,AP,X,NX,R,RHO                                          MICADO4    1
       DIMENSION A(NDIM,1),B(1),AP(1),XA(1),NA(1),NB(1),NC(1),X(1),NX(1) MICADO     5
     1,R(1),RHO(1)                                                       MICADO     6
C                                                                        MICADO     7
C   REDUCTION DE AP ET DE A                                              MICADO     8
      CALL NBR(NA,N1)                                                    MICADO     9
      CALL NBR(NC,N3)                                                    MICADO    10
      CALL NBR(NB,N2)                                                    MICADO    11
      M1=M-N3                                                            MICADO    12
      M2=N-N2                                                            MICADO    13
C     CALL MATRED(A,NDIM,N,N3,N2,NC,NB)                                  MICADOC    2
C     CALL MATRED(AP,1,M,0,N3,0,NC)                                      MICADOC    3
C                                                                        MICADO    16
C***PRISE EN COMPTE DES CONDITIONS INITIALES                             MICADO    17
                                                                         MICADO    18
C                                                                        MICADO    19
      IF(N1.EQ.0)GO TO 4                                                 MICADO    20
      DO 1 I=1,N1                                                        MICADO    21
      K=0                                                                MICADO    22
      IF(N2.EQ.0)GO TO 1                                                 MICADO    23
      DO 2 J=1,N2                                                        MICADO    24
      IF(NA(I).GT.NB(J))K=K+1                                            MICADO    25
    2 CONTINUE                                                           MICADO    26
    1 NA(I)=NA(I)-K                                                      MICADO    27
      DO 3 I=1,M1                                                        MICADO    28
      DO 3 J=1,N1                                                        MICADO    29
      K=NA(J)                                                            MICADO    30
    3 B(I)=B(I)-A(I,K)*XA(J)                                             MICADO    31
C                                                                        MICADO    32
C   CALCUL DE DP/P                                                       MICADO    33
C                                                                        MICADO    34
    4 S1=0.0                                                             MICADO    35
      S2=0.0                                                             MICADO    36
      DO 5 I=1,M1                                                        MICADO    37
      S1=S1+B(I)*AP(I)                                                   MICADO    38
    5 S2=S2+AP(I)*AP(I)                                                  MICADO    39
      IF(S2.EQ.0.0)GO TO 6                                               MICADO    40
      DP=S1/S2                                                           MICADO    41
       DO 21 I=1,M1                                                      MICADO    42
   21 B(I)=B(I)-DP*AP(I)                                                 MICADO    43
    6 NN=N+N                                                             MICADO    44
        DO 7 I=1,M1                                                      MICADO    45
      RHO(NN+I)=B(I)                                                     MICADO    46
    7 B(I)=-B(I)                                                         MICADO    47
      CALL HTLS(A,B,NDIM,M1,M2,ITER,X,NX,R,RHO,EPS)                      MICADO    48
      DO 8 I=1,M                                                         MICADO    49
      R(I)=-R(I)                                                         MICADO    50
    8 B(I)=RHO(NN+I)                                                     MICADO    51
      IF(N2.EQ.0) RETURN                                                 MICADO    52
      DO 9 J=1,N2                                                        MICADO    53
      DO 9 I=1,M2                                                        MICADO    54
    9 IF(NX(I).GE.NB(J)) NX(I)=NX(I)+1                                   MICADO    55
       RETURN                                                            MICADO    56
      END                                                                MICADO    57
```

```
      SUBROUTINE NBR(L,N)                                    MICADO   58
      DIMENSION L(12)                                        MICADO   59
      N=0                                                    MICADO   60
      DO 1 I=1,12                                            MICADO   61
      IF(L(I).EQ.0) RETURN                                   MICADO   62
      IF(L(I).EQ.10H          )RETURN                        MICADO   63
    1 N=I                                                    MICADO   64
      END                                                    MICADO   65
```

```
      SUBROUTINE HTLS(A,B,M1,M,N,ITER,X,IPI,R,AUX,EPS)                  HTLS       2
C-----------------------------------------------------------------      HTLS       3
      LEVEL 2,A,B,X,IPI,R,AUX                                           DEBUG      7
      DIMENSION A(M1,1),B(1),X(1),IPI(1),R(1),AUX(1)                    HTLS       4
      DOUBLE PRECISION G,H,SIG                                          HTLS       5
C                                                                       HTLS       6
C   CALCUL DU PREMIER PIVOT                                             HTLS       7
C                                                                       HTLS       8
      K2=N + 1                                                          HTLS       9
      PIV=0.0                                                           HTLS      10
      DO 4 K=1,N                                                        HTLS      11
      IPI(K)=K                                                          HTLS      12
      H=0.                                                              HTLS      13
      G = 0.                                                            HTLS      14
      DO 2 I=1,M                                                        HTLS      15
      H=H+DBLE(A(I,K))*DBLE(A(I,K))                                     HTLS      16
      G=G+DBLE(A(I,K))*DBLE(B(I))                                       HTLS      17
    2 CONTINUE                                                          HTLS      18
      AUX(K)=H                                                          HTLS      19
      AUX(K2) = G                                                       HTLS      20
      PIVT = G*G/H                                                      HTLS      21
      IF(PIVT-PIV)4,4,3                                                 HTLS      22
    3 PIV = PIVT                                                        HTLS      23
      KPIV=K                                                            HTLS      24
    4 K2 = K2 + 1                                                       HTLS      25
C   BOUCLE POUR CHAQUE ITERATION                                        HTLS      26
      DO 21 K=1,ITER                                                    HTLS      27
      IF(KPIV.EQ.K)GO TO 8                                              HTLS      28
C   ON ECHANGE LES K ET KPIV SI KPIVPLUS GRAND QUE K                    HTLS      29
      H=AUX(K)                                                          HTLS      30
      AUX(K)=AUX(KPIV)                                                  HTLS      31
      AUX(KPIV)=H                                                       HTLS      32
       K2=N+K                                                           HTLS      33
      K3=N+KPIV                                                         HTLS      34
      G = AUX(K2)                                                       HTLS      35
      AUX(K2) = AUX(K3)                                                 HTLS      36
      AUX(K3) = G                                                       HTLS      37
      DO 7 I=1,M                                                        HTLS      38
      H=A(I,K)                                                          HTLS      39
      A(I,K)=A(I,KPIV)                                                  HTLS      40
      A(I,KPIV)=H                                                       HTLS      41
    7 CONTINUE                                                          HTLS      42
C   CALCUL DE BETA,SIGMA ET UK DANS HTUL                                HTLS      43
C                                                                       HTLS      44
    8 CALL HTUL(A,M,N,K,SIG,BETA,M1)                                    HTLS      45
C                                                                       HTLS      46
C   ON GARDE SIGMA DANS AUX(N+K)                                        HTLS      47
      J=N+K                                                             HTLS      48
      AUX(J)=-SIG                                                       HTLS      49
      IP=IPI(KPIV)                                                      HTLS      50
       IPI(KPIV)=IPI(K)                                                 HTLS      51
      IPI(K)=IP                                                         HTLS      52
      IF(K.EQ.N) GO TO 13                                               HTLS      53
C                                                                       HTLS      54
C   TRANSFORMATION DE A DANS HTAL                                       HTLS      55
C                                                                       HTLS      56
      CALL HTAL(A,M,N,K,BETA,M1)                                        HTLS      57
```

```
C
C   TRANSFORMATION DE B DANS HTBL
C
   13 CALL HTBL(A,B,M,N,K,BETA,M1)
C
C   RECHERCHE DU PIVOT (K+1)
C
      AUX(K)=SQRT(PIV)
      IF(K.EQ.N) GO TO 11
      PIV=0.0
      KPIV = K + 1
      J1 = KPIV
      K2=N + J1
      DO 18 J=J1,N
      H=AUX(J)-DBLE(A(K,J))*DBLE(A(K,J))
      AUX(J)=H
      G=AUX(K2)-DBLE(A(K,J))*DBLE(B(K))
      AUX(K2) = G
      PIVT = G*G/H
      IF(PIVT.LT.PIV)GO TO 18
      KPIV=J
      PIV=PIVT
   18 K2 = K2 + 1
C
C   CALCUL DES X
C
   11  X(K)=B(K)/AUX(N+K)
      IF(K.EQ.1)GO TO 27
      DO 15 I=2,K
      KK=K-I+1
      X(KK)=B(KK)
      KI=KK+1
      DO 25 J=KI,K
   25 X(KK)=X(KK)-A(KK,J)*X(J)
      X(KK)=X(KK)/AUX(N+KK)
   15 CONTINUE
C  27 CALL UCOPY(B,R,M)
 27   CONTINUE
      DO 29 III=1,M
      R(III) = B(III)
 29   CONTINUE
C
C   CALCUL DU VECTEUR RESIDUEL DANS HTRL
      CALL HTRL(A,R,M,N,K,AUX,M1)
C     IMAX=MAXFZE(R(1),M)
C     IMIN=MINFZE(R(1),M)
      IMAX = MXMIN(R(1),M,1)
      IMIN = MXMIN(R(1),M,0)
      EPS1=R(IMAX)-R(IMIN)
      GOTO 1000
C     PRINT OUT OF INTERMEDIATE CORRECTOR STRENGTHS/ORBITS
      PRINT 6000,K,R(IMAX),IMAX,R(IMIN),IMIN
      PRINT 6002, (IX,IPI(IX),X(IX),IX=1,K,1)
      PRINT 6004
      PRINT 6006, (IX,R(IX),IX=1,M,1)
 1000 CONTINUE
      IF(EPS1.GT.EPS)GO TO 21
```

```
          GO TO 202                                                    HTLS     115
       21 CONTINUE                                                     HTLS     116
          RETURN                                                       HTLS     119
      202 EPS=EPS1                                                     HTLS     120
          ITER=K                                                       HTLS     121
          RETURN                                                       HTLS     122
     6000 FORMAT(5X,15HNO. CORRECTORS=,I5/                             HTLS     123
        .  5X,11HMAX. ORBIT=,3PF8.3,13HMM AT MONITOR,I5/               HTLS     124
        .  5X,11HMIN. ORBIT=,3PF8.3,13HMM AT MONITOR,I5//              HTLS     125
        .  5X,10HCORRECTORS/                                           HTLS     126
        .  5X,4H  I ,2X,5H NAME,2X,8HSTRENGTH/                         HTLS     127
        .  5X,13X,11HMICRO. RAD./)                                     HTLS     128
     6002 FORMAT(5X,I4,2X,I5,2X,6PF11.6)                               HTLS     129
     6004 FORMAT(5X,31HORBIT DISPLACEMENTS AT MONITORS/                HTLS     130
        .  5X,4H  I ,2X,10HMONITOR(I)/                                 HTLS     131
        .  5X,6X,3H MM/)                                               HTLS     132
     6006 FORMAT(5X,I4,3PF10.5)                                        HTLS     133
          END                                                          HTLS     134
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
109	I		THERE IS NO PATH TO THIS STATEMENT.

```
      SUBROUTINE HTUL(A,M,N,K,SIG,BETA,M1)                 HTLS   135
      LEVEL 2,A                                            DEBUG    8
      DIMENSION A(M1,1)                                    HTLS   136
      DOUBLE PRECISION SIG                                 HTLS   137
      SIG=0.0                                              HTLS   138
C                                                          HTLS   139
      DO 10 I=K,M                                          HTLS   140
      SIG=SIG+DBLE(A(I,K))*DBLE(A(I,K) )                   HTLS   141
   10 CONTINUE                                             HTLS   142
      SIG=DSQRT(SIG)                                       HTLS   143
C                                                          HTLS   144
C   ON CHOISIT LE SIGNE CORRECT POUR SIG.                  HTLS   145
      H=A(K,K)                                             HTLS   146
      IF(H.LT.0.0)SIG=-SIG                                 HTLS   147
      BETA=H + SIG                                         HTLS   148
      A(K,K)=BETA                                          HTLS   149
      BETA=1./(SIG*BETA)                                   HTLS   150
      RETURN                                               HTLS   151
      END                                                  HTLS   152
```

```
      SUBROUTINE HTAL(A,M,N,K,BETA,M1)                              HTLS     153
      LEVEL 2,A                                                     DEBUG      9
      DIMENSION A(M1,1)                                             HTLS     154
      DOUBLE PRECISION H                                            HTLS     155
      NC=N-K                                                        HTLS     156
      DO 21 J=1,NC                                                  HTLS     157
      H=0.0                                                         HTLS     158
      DO 20 K1=K,M                                                  HTLS     159
   20 H=H+DBLE(A(K1,K))*DBLE(A(K1,K+J))                             HTLS     160
      H=BETA*H                                                      HTLS     161
      DO 21 K1=K,M                                                  HTLS     162
   21 A(K1,K+J)=A(K1,K+J)-A(K1,K)*H                                 HTLS     163
      RETURN                                                        HTLS     164
      END                                                           HTLS     165
```

```
      SUBROUTINE HTBL(A,B,M,N,K,BETA,M1)                         HTLS    166
      LEVEL 2,A,B                                                DEBUG    10
      DIMENSION A(M1,1),B(1)                                     HTLS    167
      DOUBLE PRECISION H                                         HTLS    168
      H=0.                                                       HTLS    169
      DO 20 K1=K,M                                               HTLS    170
   20 H=H+DBLE(A(K1,K))*DBLE(B(K1))                              HTLS    171
      H=BETA*H                                                   HTLS    172
      DO 21 K1=K,M                                               HTLS    173
   21 B(K1)=B(K1)-A(K1,K)*H                                      HTLS    174
      RETURN                                                     HTLS    175
      END                                                        HTLS    176
```

```
                SUBROUTINE HTRL(A,B,M,N,K,AUX,M1)                          HTLS     177
                LEVEL 2,A,B,AUX                                             DEBUG     11
                DIMENSION A(M1,1),B(1),AUX(1)                               HTLS     178
          C     CALL UZERO(B,1,K)                                           HTLS     179
                DO 10 I=1,K                                                 HTLS     180
                B(I)=0.0                                                    HTLS     181
           10   CONTINUE                                                    HTLS     182
                DO 30 KK=1,K                                                HTLS     183
                LV=M-K+KK                                                   HTLS     184
                KN=N+K-KK+1                                                 HTLS     185
                KL=K-KK+1                                                   HTLS     186
          C                                                                 HTLS     187
          C                                                                 HTLS     188
                BETA=-1./(AUX(KN)*A(KL,KL))                                 HTLS     189
                CALL HTBL(A,B,M,N,KL,BETA,M1)                               HTLS     190
             30 CONTINUE                                                    HTLS     191
                RETURN                                                      HTLS     192
                END                                                         HTLS     193
```

```
      FUNCTION MXMIN (A,N,M)                                        HTLS   194
C  IF M=0, MAXMIN=LOWEST INDEX OF MINIMUM ELEMENT IN A              HTLS   195
C  IF M=1, MAXMIN=LOWEST INDEX OF MAXIMUM ELEMENT IN A              HTLS   196
C  IF MZ1, MAXMIN=1                                                 HTLS   197
      LEVEL 2,A                                                     DEBUG   12
      DIMENSION A(N)                                                HTLS   198
      MXMIN = 1                                                     HTLS   199
      IF (N.LT.1) RETURN                                            HTLS   200
      CURENT=A(1)                                                   HTLS   201
      DO 10 I=2,N                                                   HTLS   202
      IF ((M.EQ.0).AND.(A(I).GE.CURENT)) GO TO 10                   HTLS   203
      IF ((M.EQ.1).AND.(A(I).LE.CURENT)) GO TO 10                   HTLS   204
      CURENT=A(I)                                                   HTLS   205
      MXMIN = I                                                     HTLS   206
   10 CONTINUE                                                      HTLS   207
      RETURN                                                        HTLS   208
      END                                                           HTLS   209
```

```
      SUBROUTINE PARSR (THET,ERR)                                      PARSR   2
C  FUNCTION EVALUATION FOR COLLINS STRAIGHT SECTION DESIGN             PARSR   3
      COMMON/AGS/ALPHA,BETA,RHO,BZ,AVAC,BMAX                           AGS     2
C                                                                      AGS     3
      COMMON/STR/GAM,A,LK,BK,F,LM,BL,LL,BQ,LTO,SMUD,CMUD               STR     2
      REAL LK,LM,LL,LTO                                                STR     3
C                                                                      STR     4
      REAL KL                                                          PARSR   6
C                                                                      PARSR   7
      S = SIN(THET)                                                    PARSR   8
      C = COS(THET)                                                    PARSR   9
      SH = (EXP(THET) - EXP(-THET))/2.                                 PARSR  10
      CS = (EXP(THET) + EXP(-THET))/2.                                 PARSR  11
      X = C*SH + S*CH                                                  PARSR  12
      Y = C*SH - S*CH                                                  PARSR  13
      Z = C*CH                                                         PARSR  14
      W = S*SH                                                         PARSR  15
      KL = 2.*(ALPHA*SMUD - W)/X                                       PARSR  16
      RK = (KL*W -Y)/(GAM*SMUD)                                        PARSR  17
      BL = KL*RK                                                       PARSR  18
      BK = 1./RK                                                       PARSR  19
      LL = (KL*Y + 2.*(Z -CMUD))/(2.*GAM*SMUD)                         PARSR  20
      LK = RHO*(BK*BK)                                                 PARSR  21
      LM = THET/BK                                                     PARSR  22
      A = BQ/(LK*BZ)                                                   PARSR  23
      XI = (1. + ALPHA*LL/BETA)**2 + (LL/BETA)**2                      PARSR  24
      XQ = AVAC*SQRT(BETA*XI/BMAX)                                     PARSR  25
      ERR = XQ - A                                                     PARSR  26
      RETURN                                                           PARSR  27
      END                                                              PARSR  28
```

```
            SUBROUTINE PBML(M)                                              PBMLL      2
      C  PRINT NAMED BEAM ELEMENTS WITH LENGTHS AND ACCUMULATED LENGTHS.    PBMLL      3
                                                                            BLANK      2
            LEVEL 2, STORE,INFF,IWORK                                       BLANK      3
            COMMON STORE(48000),IWORK(10)                                   86MARSIZ   1
            DIMENSION INFF(24,2000)                                         86MARSIZ   2
            EQUIVALENCE (INFF,STORE)                                        86MARSIZ   3
                                                                            BLANK      5
            COMMON/BMI1/MI1(16000)                                          NV3BMI1    1
            COMMON/NELS1/NELS1                                              BMI1L      2
            LEVEL 2,MI1                                                     BMI1L      3
                                                                            BMI1L      4
      C                                                                     BMI1       3
      C                                                                     PBMLL      6
      C                                                                     PBMLL      7
            NBCD = 1                                                        PBMLL      8
            DO 100 I=1,NBCD                                                 PBMLL      9
            NN = MDAT(M,I)                                                  PBMLL     10
            MNM = MNAME(NN)                                                 PBMLL     11
            NDAT = INFF(17,NN)                                              PBMLL     12
            CALL MIFILL(NN,1,NDAT,NELS1,MI1)                                PBMLL     13
      C                                                                     PBMLL     14
            WRITE (3,1) MNM                                                 PBMLL     15
            ELT = 0.                                                        PBMLL     16
      C                                                                     PBMLL     17
            DO 110 J=1,NELS1                                                PBMLL     18
            JJ = MI1(J)                                                     PBMLL     19
            MNAM = MNAME(JJ)                                                PBMLL     20
            EL = FDAT(JJ,1)                                                 PBMLL     21
            ELT = ELT + EL                                                  PBMLL     22
            WRITE (3,2) MNAM,EL,ELT                                         PBMLL     23
        110 CONTINUE                                                        PBMLL     24
        100 CONTINUE                                                        PBMLL     25
      C                                                                     PBMLL     26
        1   FORMAT (1H1,5X,*ELEMENTS THROUGH BEAM LINE *,A5//6X,*NAME*,5X,  PBMLL     27
           1     *  LENGTH*,6X,* ACC.LENGTH*)                               PBMLL     28
        2   FORMAT (6X,A5,2X,F12.8,2X,F12.8)                                PBMLL     29
            RETURN                                                          PBMLL     30
            END                                                             PBMLL     31
```

```
      SUBROUTINE PARTYP(IP,ITYP)                                   PARTYP     2
      RETURN                                                       PARTYP     3
      END                                                          PARTYP     4
```

```
      SUBROUTINE PLPAR(PM,FL,IT)

*     ROUND LIMITS FOR PLOT OF BETATRON FUNCTIONS.  SET TICK INTERVALS.

      IF (PM.GT.5.) GO TO 43
      FL = 5.
      IT = 1
      RETURN
   43 IF (PM.GT.10.) GO TO 44
      FL = 10.
      IT = 2
      RETURN
   44 IF (PM.GT.25.) GO TO 45
      FL = 25.
      IT = 5
      RETURN
   45 IF (PM.GT.50.) GO TO 46
      FL = 50.
      IT = 10
      RETURN
   46 IF (PM.GT.100.) GO TO 471
      FL = 100.
      IT = 20
      RETURN
  471 IF (PM.GT.150.) GO TO 47
      FL = 150.
      IT = 30
      RETURN
   47 IF (PM.GT.250.) GO TO 48
      FL = 250.
      IT = 50
      RETURN
   48 IF (PM.GT.500.) GO TO 49
      FL = 500.
      IT = 100
      RETURN
   49 IF (PM.GT.1000.) GO TO 50
      FL = 1000.
      IT = 200
      RETURN
   50 IF (PM.GT.1500.) GO TO 51
      FL = 1500.
      IT = 300
      RETURN
   51 IF (PM.GT.2500.) GO TO 52
      FL = 2500.
      IT = 500
      RETURN
   52 IF (PM.GT.5000.) GO TO 53
      FL = 5000.
      IT = 1000
      RETURN
   53 IP = PM/1000.
      FL = (IP+1)*1000
      IT = 2000
      RETURN
      END
```

```
      SUBROUTINE PLOTBET(M)                                            PLOTBEV    2
                                                                       PLOTBEV    3
*  PLOT BETAX,BETAY,XEQ, WICH WERE COMPUTED AND WRITTEN TO A FILE BY   PLTB682    1
*  A BEST INSTRUCTION.                                                 PLTB682    2
                                                                       PLTB682    3
                                                                       BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                        BLANK      3
      COMMON STORE(48000),IWORK(10)                                    86MARSIZ   1
      DIMENSION INFF(24,2000)                                          86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                         86MARSIZ   3
                                                                       BLANK      5
      LEVEL 2, XX,Y1,Y2,Y3,Y4                                          BETPTS     2
      COMMON XX(4000),Y1(4000),Y2(4000), Y3(4000),Y4(4000)             RENAM83    1
      COMMON/BPLTCOM/MN,KW,BXX,BYX,NPLT                                RENAM83    2
      COMMON/CCPOOL/XMIN,XMAX,YMIN,YMAX,CCXMIN,CCXMAX,CCYMIN,CCYMAX    PL6683     1
      COMMON/CCFACT/FACTOR                                             PL6683     2
                                                                       BETPTS     5
      COMMON/BMI/MI(5000)                                              BMIL       1
      COMMON/NELS/NELS                                                 BMIL       2
      LEVEL 2,MI                                                       BMIL       3
                                                                       BMIL       4
                                                                       BMI        3
      COMMON/SVNAM/NAMRUN                                              SVNAM      2
      DIMENSION IPRS(2),PMX(3),FLT(3),PAR(3),ITIK(3)                   PLTB682    8
      DIMENSION X(6),Y(6),IFLT(3),MOP(8)                               PLTJL29    1
      DIMENSION XPL(4000),YPL(4000)                                    BMINV82    6
                                                                       PLTB682   11
      COMMON/TRKINT/G,EX,EY,OMSQ,SEND,VMX,VMN,DINT,BEG,DPR,JPR,        TRKINT     2
     1   THET,PLT,NPT,NZ,PMAX,NSIZ,ITITLE(6)                           TRKINT     3
      LOGICAL BEG,DPR,PLT                                              TRKINT     4
      INTEGER BDAT                                                     PLTB682   13
      LOGICAL LIMSW                                                    PLTB682   14
      LOGICAL PSW                                                      PLTB682   15
      DATA MOP/3HDRF,3HMAG,4HMAGV,4HMAGS,3HREF,3HINV,3HSHF,3HEQU/      PLTB682   16
                                                                       PLTB682   17
****  INPUT PARAMETERS                                                PLTB682   18
C   KB -- OPTION FOR UNITS OF BETA PLOT.                               PLTB682   19
C   KB = 0 OR BLANK -- PLOT SQ. RT. OF BETAS.                          PLTB682   20
C   KB .NE. 0, PLOT BETAS                                              PLTB682   21
C   KA IS MULTIPLE, NOT TO EXCEED 14, OF 10 INCHES FOR LENGTH OF       PLTB682   22
C  FRAME DESIRED, I.E. X AXIS.                                         PLTB682   23
*     NC IS NAME OF ELEMENT THAT COMPUTED THE BETATRON FUNCTIONS.      PLTB682   24
*****                                                                  PLTB682   25
                                                                       PLTB682   26
*** COMPUTED VARIABLES                                                 PLTB682   27
*       XX = LENGTH S    Y1=BETAX,  Y2=XEQ, Y3=BETAY, Y4 = YEQ         PLTB682   28
*   PMX = MAX OF BETA, MAX OF XEQ,YEQ, AND MIN OF XEQ,YEQ              PLTB682   29
*   FLT = RADJUSTED MAX OF BETAX,BETAY, THE READJUSTED MAX OF XEQ, AND PLTB682   30
*  THE READJUSTED MIN OF XEQ                                           PLTB682   31
****                                                                   PLTB682   32
      LIMSW = .FALSE.                                                  PLOTBEV   35
      BXMX = BXX                                                       PLOTBEV   36
      IF(NPLT.EQ.0) CALL SETPLT                                        PLT      129
      REWIND 98                                                        PLT      130
      NPLT=NPLT+1                                                      PLT      131
      BYMX = BYX                                                       PLOTBEV   37
                                                                       PLOTBEV   38
```

```
      *                GET INPUT DATA                                   PLOTBEV   39
            INS=0                                                       PLDRF      1
            KA = INFF(4,M)                                              PLOTBEV   40
            KB = INFF(5,M)                                              PLOTBEV   41
            IF (KA.EQ.0) KA = 1                                         PLOTBEV   42
            NC = BDAT(M,1)                                              PLOTBEV   43
            NAM = MDAT(M,1)                                             PLOTBEV   44
            NNOP=INFF(1,NAM)                                            PLBAG17    1
            IF(NNOP.EQ.4HFXPT) NAM=MDAT(NAM,2)                          PLBAG17    2
            KO = 2                                                      PLOTBEV   45
            CALL DATA (M,3,1,2,IPRS)                                    PLOTBEV   46
            CALL DATA(M,1,1,3,PAR)                                      PLOTBEV   47
      *                SET UP MI LIST OF BEAM                           PLOTBEV   48
            NDAT=INFF(17,NAM)                                           PLOTBEV   49
            CALL MIFILL(NAM,1,NDAT,NELS,MI)                             PLOTBEV   50
            NK=NELS                                                     PLOTBEV   51
                                                                        PLOTBEV   53
            REWIND 12                                                   NOV279    15
      C SEEK CORRECT FILE OF BETATRON FUNCTIONS AND READ IT              NOV279    16
       30   READ(12) MN,KW                                              NOV279    17
            IF(MN.EQ.NC) GO TO 35                                       NOV279    18
            READ(12)                                                    NOV279    19
            READ(12)                                                    NOV279    20
             GO TO 30                                                   NOV279    21
       35   CONTINUE                                                    NOV279    22
            READ(12) (XX(I),Y1(I),Y2(I),Y3(I),Y4(I),I=1,KW)             NOV279    23
            READ(12) BXMX,BYMX                                          NOV279    24
       36   NPT = KW                                                    PLOTBEV   64
            NZ = KW - 1                                                 PLOTBEV   65
                                                                        PLOTBEV   66
            IF (KB.NE.0) GO TO 33                                       PLOTBEV   67
      *        TAKE SQUARE ROOTS OF BETAX,BETAY                         PLOTBEV   68
            BXMX = SQRT(BXMX)                                           PLOTBEV   69
            BYMX = SQRT(BYMX)                                           PLOTBEV   70
            DO 32 I=1,KW                                                PLOTBEV   71
            Y1(I) = SQRT(Y1(I))                                         PLOTBEV   72
            Y3(I) = SQRT(Y3(I))                                         PLOTBEV   73
       32   CONTINUE                                                    PLOTBEV   74
       33   CONTINUE                                                    PLOTBEV   75
                                                                        PLOTBEV   76
      *        DETERMINE PAPER COORDINATES                              PLOTBEV   77
      *     LENGTH OF PAPER MAY NOT EXCEED 120 INCHES                   PLOTBEV   78
            IF (KA.GT.14)  KA=14                                        PLOTBEV   79
                                                                        PLOTBEV   80
      *        ALLOW SPACE FOR RIGHT AND LEFT MARGINS.                  PLOTBEV   81
            CCXMIN = 70.                                                PLOTBEV   82
            FKA = KA                                                    PLOTBEV   83
            IF(KA.LE.1) FKA=1.4                                         DEC85PLB   1
            CCXMAX = (FKA-1.)*850. + 745.                               PLOTBEV   84
            T1 = 100./8.                                                PLOTBEV   85
            T2 = 100./16.                                               PLOTBEV   86
                                                                        PLOTBEV   87
      ***************************************************************** PLOTBEV   88
      *     FIND MAXIMUMS AND MINIMUMS.  PUT BETAX AND BETAY ON SAME SCALE.  PLOTBEV   89
      *        MINIMUM OF BETAS IS 0. XEQ,YEQ MAY BE NEGATIVE           PLOTBEV   90
                                                                        PLOTBEV   91
      *       COMPUTE MAXIMUM LIMITS OF BETAS AND EQS.  THEN CHECK INPUT.   PLOTBEV   92
```

```
      *  IF ANY ARE INPUT, ALL SHOULD BE THERE                      PLOTBEV   93
         PMX(1) = BXMX                                              PLOTBEV   94
         IF (BYMX.GT.PMX(1))  PMX(1) = BYMX                         PLOTBEV   95
         QMX = Y2(1)                                                PLOTBEV   96
         QMN = Y2(1)                                                PLOTBEV   97
         DO 40 I=1,NPT                                              PLOTBEV   98
         IF (Y4(I).GT.QMX)  QMX = Y4(I)                             PLOTBEV   99
         IF (Y2(I).GT.QMX) QMX = Y2(I)                              PLOTBEV  100
         IF (Y4(I).LT.QMN)  QMN = Y4(I)                             PLOTBEV  101
         IF (Y2(I).LT.QMN) QMN = Y2(I)                              PLOTBEV  102
      40 CONTINUE                                                   PLOTBEV  103
         PMX(2) = QMX                                               PLOTBEV  104
         PMX(3) = QMN                                               PLOTBEV  105
      *     CHECK ON INPUT MAXIMUMS                                 PLOTBEV  106
         IF (PAR(1).NE.0.) GO TO 41                                 PLOTBEV  107
         IF (PAR(3).EQ.0.) GO TO 42                                 PLOTBEV  108
      *     USE INPUT LIMITS, UNLESS MAX IS 0.                      PLOTBEV  109
      41 IF (PAR(1).NE.0.)  PMX(1) = PAR(1)                         PLOTBEV  110
         IF (PAR(3).NE.0.) PMX(2) = PAR(3)                          PLOTBEV  111
         PMX(3) = PAR(2)                                            PLOTBEV  112
         LIMSW = .TRUE.                                             PLTB682   33
         DO 43 I=1,2                                                PLTB682   34
      43 FLT(I) = PMX(I)                                            PLTB682   35
         GO TO 44                                                   PLTB682   36
       42 CONTINUE                                                  PLTB682   37
      *     ROUND LIMITS AND SET TICK INTERVALS                     PLOTBEV  115
         DO 55 I=1,2                                                PLOTBEV  116
         CALL PLPAR(PMX(I),FM,ITK)                                  PLOTBEV  117
         FLT(I) = FM                                                PLOTBEV  118
         ITIK(I) = ITK                                              PLOTBEV  119
      55 CONTINUE                                                   PLOTBEV  120
      44 CONTINUE                                                   PLTB682   38
      *     ROUND FOR NEGATIVE SCALE OF XEQ.                        PLOTBEV  121
      *        IF LESS THAN 10, ROUND TO NEXT INTEGER               PLOTBEV  122
      *        IF GREATER THAN 10, ROUND TO NEXT FACTOR OF 10.      PLTB682   39
         PSW = .FALSE.                                              PLTB682   40
         ABP = ABS(QMN)                                             PLTB682   41
         IF (PMX(3).LT.0.)  GO TO 56                                PLTB682   42
         IF (QMN.GE.0.) GO TO 63                                    PLTB682   43
      56 IF ((PMX(3).EQ.QMN).OR.(PMX(3).EQ.0))  GO TO 59            PLTB682   44
         GO TO 57                                                   PLTB682   45
      C IF ABS (LARGEST NEG) .LT. .05 XMAX PLOT BELOW X AXIS, RETAIN SCALE  DEC85PLB  2
      59 ONEP= .05*FLT(2)                                           DEC85PLB   3
         IF (ABP.GT.ONEP)  GO TO 60                                 PLTB682   49
         PSW = .TRUE.                                               PLTB682   50
         GO TO 63                                                   PLTB682   51
      60 IF (.NOT.LIMSW)  GO TO 57                                  PLTB682   52
      C   IF QMIN SET TO 0 ON INPUT AND ABS OF LARGEST NEGATIVE .GT. 10, SET  PLTB682  53
      C  ALL NEGATIVES TO 0 AND RETAIN MIN SET ON INPUT.            PLTB682   54
      C   IF NEGATIVES ARE WITHIN 10 LIMIT, PLOT BELOW 0 AXIS.      PLTB682   55
         IF (PMX(3).NE.0.)  GO TO 57                                PLTB682   56
         DO 61 I=1,KW                                               PLTB682   57
         IF (Y2(I).LT.0.)  Y2(I) = 0.                               PLTB682   58
         IF (Y4(I).LT.0.)  Y4(I) = 0.                               PLTB682   59
      61 CONTINUE                                                   PLTB682   60
         GO TO 63                                                   PLTB682   61
      57 IF (ABP.GT.10.) GO TO 62                                   PLTB682   62
```

```
      IP = ABP                                                          PLOTBEV   132
      APP = IP                                                          PLOTBEV   133
      IF (ABP.GT.APP)  IP = IP + 1                                      PLOTBEV   134
      FLT(3) = -IP                                                      PLOTBEV   135
      GO TO 65                                                          PLOTBEV   136
   62 IP = ABP/10.                                                      PLOTBEV   137
      ABP = (IP+1)*10                                                   PLOTBEV   138
      FLT(3) = -ABP                                                     PLOTBEV   139
      GO TO 65                                                          PLOTBEV   140
   63 FLT(3) = 0.                                                       PLOTBEV   141
      PMX(3) = 0.                                                       PLOTBEV   142
   65 CONTINUE                                                          PLOTBEV   143
      DO 66 I=1,3                                                       PLOTBEV   144
      IFLT(I) = FLT(I)                                                  PLOTBEV   145
   66 CONTINUE                                                          PLOTBEV   146
                                                                        PLOTBEV   147
*                        DETERMINE LIMITS IN X SCALE                    PLOTBEV   148
      IF (IPRS(1).GT.IPRS(2))  GO TO 205                                PLOTBEV   149
      IF (IPRS(2).NE.0.) GO TO 70                                       PLOTBEV   150
**  DEFAULT -- ENTIRE BEAM LINE                                         PLOTBEV   151
      IPRS(1) = 0                                                       PLOTBEV   152
      IPRS(2) = NZ                                                      PLOTBEV   153
                                                                        PLOTBEV   154
************************************************************************ PLOTBEV   155
*     ORIGINALLY SET UP TO LOOP FROM HERE TO 200 TO PLOT MULTIPLE FRAMES PLOTBEV   156
*     INPUT CHANGED TO REQUIRE CARD INPUT FOR EACH FRAME.               PLOTBEV   157
************************************************************************ PLOTBEV   158
   70 DO 200 KK=1,KO,2                                                  PLOTBEV   159
      N1 = IPRS(KK) + 1                                                 PLOTBEV   160
      NBT = IPRS(KK+1) + 1                                              PLOTBEV   161
      NBE = NBT - 1                                                     PLOTBEV   162
      NP = NBT - N1 + 1                                                 PLOTBEV   163
*                        LIMITS OF X FROM S1 TO SND                     PLOTBEV   164
      S1 = XX(N1)                                                       PLOTBEV   165
      SND = XX(NBT)                                                     PLOTBEV   166
      J1 = N1                                                           PLOTBEV   167
                                                                        PLOTBEV   168
      CCYMAX=950.                                                       PLFORM      3
      CCYMIN = 5.                                                       PLOTBEV   170
                                                                        PLOTBEV   171
*     PRINT TITLE FROM RUN CARD                                         PLOTBEV   172
      CCX = CCXMIN - 75.                                                PLOTBEV   173
      CCY = CCYMIN + 150.                                               PLOTBEV   174
      CALL CCLTR(CCX,CCY,1,2,ITITLE,40)                                 PLDRF       2
C  PRINT ON BOTTOM OF PAGE                                              PLOTBEV   177
      CCX = (CCXMAX + CCXMIN) / 2. - 112.5                              PLOTBEV   178
      CCY=CCYMIN+5.                                                     PLDRF       3
      CALL CCLTR(CCX,CCY,0,2,15HPATH LENGTH (M),15)                     PLT       132
                                                                        PLOTBEV   181
*        DRAW BOUNDARIES                                                PLOTBEV   182
                                                                        PLOTBEV   183
      PTOP=850.                                                         PLFORM      4
      PBOT = 60.                                                        PLOTBEV   185
      YMAX = 10.                                                        PLOTBEV   186
      YMIN = 0.                                                         PLOTBEV   187
      XMAX = SND                                                        PLOTBEV   188
      XMIN = S1                                                         PLOTBEV   189
```

```
      CYB = ((PBOT-CCYMIN)*(YMAX-YMIN))/(CCYMAX-CCYMIN) + YMIN          PLOTBEV  190
      CYT = ((PTOP-CCYMIN)*(YMAX-YMIN))/(CCYMAX-CCYMIN) + YMIN          PLOTBEV  191
      X(1) = S1                                                         PLOTBEV  192
      X(2) = S1                                                         PLOTBEV  193
      Y(1) = CYB                                                        PLOTBEV  194
      Y(2) = CYT                                                        PLOTBEV  195
      X(3)=X(2) $ Y(3)=Y(2)                                             PLFORM     5
      CALL CCPLOT(X,Y,3,4HJOIN,1,1,1HI)                                 PLFORM     6
      Y(1) = CYT                                                        PLOTBEV  199
      X(2) = SND                                                        PLOTBEV  200
      X(1) = SND                                                        PLOTBEV  204
      Y(2) = CYB                                                        PLOTBEV  205
      X(3)=X(2) $ Y(3)=Y(2)                                             PLFORM     7
      CALL CCPLOT(X,Y,3,4HJOIN,1,1,1HI)                                 PLFORM     8
      X(2) = S1                                                         PLOTBEV  209
      Y(1) = CYB                                                        PLOTBEV  210
      X(3)=X(2) $ Y(3)=Y(2)                                             PLFORM     9
      CALL CCPLOT(X,Y,3,4HJOIN,1,1,1H-)                                 PLFORM    10
C WRITE NAME OF ARRAY                                                   MAY1179    8
      CALL CCLTR(CCX,PTOP-90.,0,1,NC,5)                                 PLFORM    11
                                                                        PLOTBEV  214
*********************************************************************** PLOTBEV  215
                                                                        PLOTBEV  216
*                     DRAW BEAM SYSTEM                                  PLOTBEV  217
      S = S1                                                            PLOTBEV  218
      DO 140 K=N1,NBE                                                   PLOTBEV  219
      NN = MI(K)                                                        PLOTBEV  220
C  FIND NAME AND OPNAME OF BEAM ELEMENT                                 PLOTBEV  221
      NAMK = MNAME(NN)                                                  PLOTBEV  222
      NN2=MI(K+2)                                                       PLBNV82    1
      IF(NN2.EQ.NN) NAMK=4H                                             PLBNV82    2
      NAMOP = INFF(1,NN)                                                PLOTBEV  223
      CALL DATA(NN,5,1,1,EL)                                            PLOTBEV  224
      SE = S + EL                                                       PLOTBEV  225
      ASSIGN 110 TO IGO                                                 PLOTBEV  226
  105 DO 106 JK=1,8                                                     PLOTBEV  227
      IF (NAMOP.EQ.MOP(JK)) GO TO 107                                   PLOTBEV  228
  106 CONTINUE                                                          PLOTBEV  229
      IF(NAMOP.EQ.4HMOVE) GO TO 112                                     PLTJL29    3
      GO TO 109                                                         PLOTBEV  230
  107  JN = JK                                                          PLOTBEV  231
      GO TO IGO (110,116)                                               SYFTN    207
  110 GO TO (133,117,117,112,112,112,112,112) JN                        PLOTBEV  233
*   IF MAGS,REF,INV,REF,EQU REFER TO MATRIX DEFINED BY A MAGNET, GET IT. PLOTBEV  234
*  OTHERWISE TREAT IT AS A COMPOSITE.                                   PLOTBEV  235
  112 NN = MDAT(NN,1)                                                   PLOTBEV  236
      NAMOP = INFF(1,NN)                                                PLOTBEV  237
      ASSIGN 116 TO IGO                                                 PLOTBEV  238
      GO TO 105                                                         PLOTBEV  239
  116  GO TO (133,117,117,109,109,109,109,109) JN                       PLOTBEV  240
*        COMPOSITE.  DRAW BOX LIKE BENDING MAGNET, BUT MUCH SMALLER.    PLOTBEV  241
  109 PCT = PTOP + 5.                                                   PLOTBEV  242
      PCB = PTOP - 5.                                                   PLOTBEV  243
      TY = PTOP + 7.                                                    PLOTBEV  244
      GO TO 125                                                         PLOTBEV  245
  117 CONTINUE                                                          PLOTBEV  246
                                                                        PLOTBEV  247
```

```
      *        DRAW MAGNET ABOVE OR BELOW THE LINE FOR FOCCUSSING OR    PLOTBEV  248
      *  DEFOCUSSING GRADIENT.                                          PLOTBEV  249
      *  STRADDLE THE LINE IF ZERO GRADIENT.                            PLOTBEV  250
            G = FDAT(NN,2)                                              PLOTBEV  251
            IF (G) 113,115,120                                          PLOTBEV  252
      *       DEFOCUSSING                                               PLOTBEV  253
        113 PCT = PTOP                                                  PLOTBEV  254
            PCB = PTOP - 50.                                            PLOTBEV  255
            GO TO 121                                                   PLOTBEV  256
      *                  FOCUSSING                                      PLOTBEV  257
        120 PCT = PTOP + 50.                                            PLOTBEV  258
            PCB = PTOP                                                  PLOTBEV  259
        121 TY = PCB + 20.                                              PLOTBEV  260
            GO TO 125                                                   PLOTBEV  261
      *                    ZERO GRADIENT                                PLOTBEV  262
        115 PCT = PTOP + 25.                                            PLOTBEV  263
            PCB = PTOP - 25.                                            PLOTBEV  264
            TY = PTOP + 5.                                              PLOTBEV  265
            GO TO 125                                                   PLOTBEV  266
                                                                        PLOTBEV  267
        125 PT = ((PCT - CCYMIN)*(YMAX-YMIN))/(CCYMAX-CCYMIN) + YMIN    PLOTBEV  268
            PB = ((PCB - CCYMIN)*(YMAX-YMIN))/(CCYMAX-CCYMIN) + YMIN    PLOTBEV  269
      *       DRAW MAGNETS                                              PLOTBEV  270
            X(1) = S                                                    PLOTBEV  271
            X(2) = S                                                    PLOTBEV  272
            Y(1) = PT                                                   PLOTBEV  273
            Y(2) = PB                                                   PLOTBEV  274
            DO 126 I=1,3                                                PLOTBEV  275
            X(3)=X(2) $ Y(3)=Y(2)                                       PLFORM    12
            CALL CCPLOT(X,Y,3,4HJOIN,1,1,1H.)                           PLFORM    13
        126 CONTINUE                                                    PLOTBEV  277
            X(2) = SE                                                   PLOTBEV  278
            Y(1) = PB                                                   PLOTBEV  279
            DO 128 I=1,3                                                PLOTBEV  280
            X(3)=X(2) $ Y(3)=Y(2)                                       PLFORM    14
            CALL CCPLOT(X,Y,3,4HJOIN,1,1,1H.)                           PLFORM    15
        128 CONTINUE                                                    PLOTBEV  282
        127 X(1) = SE                                                   PLOTBEV  283
            Y(2) = PT                                                   PLOTBEV  284
            DO 130 I=1,3                                                PLOTBEV  285
            X(3)=X(2) $ Y(3)=Y(2)                                       PLFORM    16
            CALL CCPLOT(X,Y,3,4HJOIN,1,1,1H.)                           PLFORM    17
        130 CONTINUE                                                    PLOTBEV  287
            X(2) = S                                                    PLOTBEV  288
            Y(1) = PT                                                   PLOTBEV  289
            DO 132 I=1,3                                                PLOTBEV  290
            X(3)=X(2) $ Y(3)=Y(2)                                       PLFORM    18
            CALL CCPLOT(X,Y,3,4HJOIN,1,1,1H.)                           PLFORM    19
        132 CONTINUE                                                    PLOTBEV  292
            GO TO 135                                                   PLOTBEV  293
      *                  SET PAPER POSITION FOR DRIFT                   PLOTBEV  294
        133 TY = PTOP + 5.                                              PLOTBEV  295
      C  IF DRIFT LONGER THAN 1 METER, SUBDIVIDE                        PLDRF      4
            NQ= 0                                                       PLDRF      5
            IF(EL.LE.1.0) GO TO 1339                                    PLDRF      6
            NQ= EL $ NQP=NQ+1                                           PLDRF      7
            KPL=K+INS                                                   PLDRF      8
```

```
          ELP = EL/NQP                                          PLDRF      9
          DXP=(Y2(KPL+1)-Y2(KPL))/NQP                           PLDRF     10
          DYP=(Y4(KPL+1)-Y4(KPL))/NQP                           PLDRF     11
          SS=0.                                                 PLDRF     12
          IF(KB.NE.0) GO TO 1330                                JUN680     1
          DO 1340 JJ=KPL,KW                                     JUN680     2
          Y1(JJ)=Y1(JJ)**2                                      JUN680     3
     1340 Y3(JJ)=Y3(JJ)**2                                      JUN680     4
     1330  CONTINUE                                             JUN680     5
          BXS=SQRT(Y1(KPL)*Y1(KPL+1)-EL**2)                     PLDRF     13
          BYS=SQRT(Y3(KPL)*Y3(KPL+1)-EL**2)                     PLDRF     14
          BX3=Y1(KPL)-BXS  $ BX4=Y1(KPL+1)-BXS                  PLDRF     15
          BY3=Y3(KPL)-BYS $ BY4=Y3(KPL+1)-BYS                   PLDRF     16
          GX=(BX3+BX4)/EL**2 $ GY=(BY3 +BY4)/EL**2              PLDRF     17
          SSX=BX3/GX/EL $ SSY=BY3/GY/EL                         PLDRF     18
          KWW=KW-1                                              PLDRF     19
    * MOVE DATA ARRAYS                                          PLDRF     20
          DO 1331 KJ=KPL,KWW                                    PLDRF     21
          KN=KW+KPL+NQ-KJ                                       PLDRF     22
          XX(KN)=XX(KN-NQ)                                      PLDRF     23
          Y1(KN)=Y1(KN-NQ)                                      PLDRF     24
          Y2(KN)=Y2(KN-NQ)                                      PLDRF     25
          Y3(KN)=Y3(KN-NQ)                                      PLDRF     26
     1331 Y4(KN)=Y4(KN-NQ)                                      PLDRF     27
          INS=INS+NQ                                            PLDRF     28
          KW=KW+NQ                                              PLDRF     29
          NP=NP+NQ                                              PLDRF     30
    * INSERT INTERPOLATED DATA                                  PLDRF     31
          DO 1332 JJ=1,NQ                                       PLDRF     32
          KN=KPL+JJ                                             PLDRF     33
          SS=SS+ELP                                             PLDRF     34
          XX(KN)=XX(KPL)+SS                                     PLDRF     35
          Y2(KN)=Y2(KN-1)+DXP                                   PLDRF     36
          Y4(KN)=Y4(KN-1)+DYP                                   PLDRF     37
          Y1(KN)=(SS-SSX)**2*GX+1./GX                           PLDRF     38
     1332 Y3(KN)=(SS-SSY)**2*GY+1./GY                           PLDRF     39
          IF(KB.NE.0) GO TO 1339                                JUN680     6
          DO 1349 JJ=KPL,KW                                     JUN680     7
          Y1(JJ)=SQRT(Y1(JJ))                                   JUN680     8
     1349 Y3(JJ)=SQRT(Y3(JJ))                                   JUN680     9
     1339 CONTINUE                                              PLDRF     40
          S=SE                                                  PLT      133
          GO TO 140                                             PLT      134
     135  CONTINUE                                              PLOTBEV  297
    C   BY PASS LETTERING OF BEAM SYSTEM                        PLOTBEV  299
     136  CONTINUE                                              PLOTBEV  300
    *   POSITION PAPER COORDINATES TO PRINT NAME AT TOP         PLOTBEV  301
          DECODE(10,3010,NAMK) CHAR                             JUN680    10
          IF(CHAR.NE.1H") GO TO 3020                            JUN680    11
     3010 FORMAT(A1,9X)                                         JUN680    12
          DECODE(10,3011,NAMK) NAMK                             JUN680    13
     3011 FORMAT(1XA4,5X)                                       JUN680    14
     3020 TY=PCB-20.                                            PLTB682   63
          WRITE (98,2) NAMK                                     PLOTBEV  302
    C  CENTER POINT                                             PLOTBEV  303
          CENT = S + EL/2.                                      PLOTBEV  304
    C  FIND PAPER COORDINATE                                    PLOTBEV  305
```

```
            CCX = CCXMIN + (CENT-XMIN)*(CCXMAX-CCXMIN)/(XMAX-XMIN)         PLOTBEV   306
      C  CENTER SECOND LETTER OF ELEMENT                                    PLOTBEV   307
            CALL CCLTR(CCX-10.,TY,0,1)                                      PLTB682    64
            S = SE                                                          PLOTBEV   310
       140  CONTINUE                                                        PLOTBEV   311
                                                                            PLOTBEV   312
                                                                            PLOTBEV   313
      ****************************************************************** PLOTBEV   314
                                                                            PLOTBEV   315
            CCYMAX = PTOP                                                   PLOTBEV   316
            CCYMIN = PBOT                                                   PLOTBEV   317
      *******************  WRITE LIMITS AND LABEL AXES                      PLOTBEV   318
      *             X AXIS                                                  PLTB682    65
            CCY = CCYMIN - 15.                                              PLTB682    66
            CCX = CCXMIN - 20.                                              PLTB682    67
            WRITE (98,6) S1                                                 PLTB682    68
            CALL CCLTR(CCX,CCY-25.,0,1)                                     PLBNV82     3
            WRITE (98,3) SND                                                PLTB682    70
            CCX = CCXMAX- 20.                                               PLTB682    71
            CALL CCLTR(CCX,CCY-25.,0,1)                                     PLBNV82     4
            CCX = CCXMIN - 20.                                              PLTB682    73
            IF (KB.NE.0) GO TO 142                                          PLTB682    74
            CCY = 460.                                                      PLTB682    75
            CALL CCLTR(CCX,CCY,1,2,8HW=(BETA))                              PLTB682    76
            CCY = 571.                                                      PLTB682    77
            CCX = CCX - 18.                                                 PLTB682    78
            CALL CCLTR(CCX,CCY,1,1,5H  1/2)                                 PLBNV82     5
            GO TO 143                                                       PLTB682    80
       142  CONTINUE                                                        PLTB682    81
            CCY = 500.                                                      PLTB682    82
            CALL CCLTR(CCX,CCY,1,2,4HBETA)                                  PLTB682    83
       143  CONTINUE                                                        PLTB682    84
            CCX = CCXMAX + 50.                                              PLTB682    85
            CALL CCLTR(CCX,CCY,1,2,3HXEQ)                                   PLTB682    86
      *             Y AXIS                                                  PLTB682    87
            CCY = CCYMIN                                                    PLTB682    88
            CCX = CCXMIN - 15.                                              PLTB682    89
            CALL CCLTR(CCX,CCY,0,1,1H0)                                     PLTB682    90
            CCY = CCYMIN                                                    PLTB682   100
            CCX = CCXMAX + 2.                                               PLTB682   101
            IF (LIMSW)  WRITE(98,3) FLT(3)                                  PLTB682   102
            IF (.NOT.LIMSW)  WRITE(98,9) IFLT(3)                            PLTB682   103
            CALL CCLTR(CCX,CCY,0,1)                                         PLTB682   104
            YMAX = FLT(2)                                                   PLOTBEV   353
            YMIN = FLT(3)                                                   PLOTBEV   354
            IF (YMIN.EQ.0.) GO TO 141                                       PLOTBEV   355
            IF(LIMSW) GO TO 141                                             PLBNV82     6
            CCY = CCYMIN + (0.-YMIN)*(CCYMAX-CCYMIN)/(YMAX-YMIN)            PLOTBEV   356
            CALL CCLTR(CCX+10.,CCY,0,1,1H0)                                 PLOTBEV   357
       141  CONTINUE                                                        PLOTBEV   358
                                                                            PLOTBEV   363
      ****************************************************************** PLOTBEV   364
                                                                            PLOTBEV   365
      *                 DETERMINE TICK INTERVALS FOR X AXIS AND PLOT THEM.  PLOTBEV   366
            CCYMAX = PTOP                                                   PLOTBEV   367
            CCYMIN = PBOT                                                   PLOTBEV   368
            CCYMIN = CCYMIN - T2                                            PLOTBEV   369
```

```
          YMIN = 0.                                                      PLOTBEV   370
          YMAX = FLT(1)                                                  PLOTBEV   371
*         FOR EACH 10 INCHES OF PAPER LENGTH, ALLOW ABOUT 5 INTERVALS    PLOTBEV   372
          CCY = CCYMIN + T1                                              PLOTBEV   373
          CYTK1 = 0.                                                     PLOTBEV   374
          CYTK2 = (CCY-CCYMIN)*(YMAX-YMIN)/(CCYMAX-CCYMIN) + YMIN        PLOTBEV   375
          Y(1) = CYTK1                                                   PLOTBEV   376
          Y(2 ) = CYTK2                                                  PLOTBEV   377
          DS = (SND-S1)/FKA                                              PLOTBEV   378
          CALL PLPAR(DS,FM,ITK)                                          PLOTBEV   379
          FTK = ITK                                                      PLOTBEV   380
          IS1 = S1/FTK                                                   PLOTBEV   381
          IS1 = (IS1+1)*ITK                                              PLOTBEV   382
          IEND = SND                                                     PLOTBEV   383
          CCY = CCYMIN - 15.                                             PLOTBEV   384
          DO 165 I=IS1,IEND,ITK                                          PLOTBEV   385
          X(1) = I                                                       PLOTBEV   386
          X(2) = I                                                       PLOTBEV   387
          CALL CCPLOT(X,Y,2,4HJOIN)                                      PLOTBEV   388
          CCX = CCXMIN + (X(1) - XMIN)*(CCXMAX-CCXMIN)/(XMAX-XMIN)       PLOTBEV   389
          WRITE (98,9) I                                                 PLOTBEV   390
          CALL CCLTR(CCX-26.,CCY,0,1)                                    PLTB682   105
      165 CONTINUE                                                       PLOTBEV   392
          CCYMIN = PBOT                                                  PLOTBEV   393
                                                                         PLOTBEV   394
****** PLOT TICK MARKS ON LEFT Y AXIS FOR BETAX AND BETAY.               PLOTBEV   395
*          PLOT TICK MARKS ON RIGHT Y AXIS FOR XEQ.                      PLOTBEV   396
          CCXMIN = CCXMIN - T2                                           PLOTBEV   397
          J = 1                                                          PLOTBEV   398
          YMIN = 0.                                                      PLOTBEV   399
      166 YMAX = FLT(J)                                                  PLOTBEV   400
          CCX = CCXMIN + T1                                              PLOTBEV   401
          CXTK1 = XX(J1)                                                 PLOTBEV   402
          CXTK2 = (CCX -CCXMIN)*(XMAX-XMIN)/(CCXMAX-CCXMIN) + XMIN       PLOTBEV   403
          X(1) = CXTK1                                                   PLOTBEV   404
          X(2) = CXTK2                                                   PLTB682   106
          IF (.NOT.LIMSW)  GO TO 164                                     PLTB682   107
          TIK = YMAX/5.                                                  PLTB682   108
          NTK=0 $ IF(YMIN.LT.0.) NTK=-YMIN/TIK                           PLBNV82     7
          TK=-NTK*TIK  $  NTT=NTK+4                                      PLBNV82     8
          DO 167 I=1,NTT                                                 PLBNV82     9
          TK = TK + TIK                                                  PLTB682   111
          Y(1) = TK                                                      PLTB682   112
          Y(2) = TK                                                      PLTB682   113
          CALL CCPLOT(X,Y,2,4HJOIN)                                      PLTB682   114
          IF(.NOT.LIMSW) GO TO 167                                       PLBNV82    10
          FLI=(I-NTK)*FLT(J)/5.0                                         PLBNV82    11
          WRITE(98,6) FLI                                                PLBNV82    12
          CCY=CCYMIN +(Y(1)-YMIN)*(CCYMAX-CCYMIN)/(YMAX-YMIN)            PLBNV82    13
          CALL CCLTR(CCX,CCY,0,1)                                        PLBNV82    14
      167 CONTINUE                                                       PLTB682   115
          GO TO 169                                                      PLTB682   116
      164 IS1 = ITIK(J)                                                  PLTB682   117
          ISM = YMAX                                                     PLTB682   118
          ISM = ISM - IS1                                                PLOTBEV   408
          CCXS=CCX $ CCYS=CCY                                            PLDRF      45
          CCX= CCXS-70.                                                  PLDRF      46
```

```
          IF(J.EQ.2) CCX=CCXS-20.                                         PLDRF      47
          DO 168 I=IS1,ISM,IS1                                            PLOTBEV   409
          Y(1) = I                                                        PLOTBEV   410
          Y(2) = I                                                        PLOTBEV   411
          X(3)=X(2) $ Y(3)=Y(2)                                           PLFORM     20
          CALL CCPLOT(X,Y,3,4HJOIN,1,1,1H-)                               PLFORM     21
          WRITE(98,9) I                                                   PLDRF      48
          CCY=CCYMIN +(Y(1)-YMIN)*(CCYMAX-CCYMIN)/(YMAX-YMIN)             PLDRF      49
          CALL CCLTR(CCX,CCY,0,1)                                         PLDRF      50
      168 CONTINUE                                                        PLOTBEV   413
      169 CONTINUE                                                        PLTB682   119
          YMAX=FLT(2)                                                     PLTB682   120
          IF (J.EQ.2) GO TO 170                                           PLOTBEV   414
          J = 2                                                           PLOTBEV   415
          CCXMIN = CCXMIN + T2                                            PLOTBEV   416
          SAVXMN = CCXMIN                                                 PLOTBEV   417
          CCXMIN = CCXMAX - T2                                            PLOTBEV   418
          CCXMAX = CCXMAX + T2                                            PLOTBEV   419
          YMIN = FLT(3)                                                   PLOTBEV   420
          GO TO 166                                                       PLOTBEV   421
      170 IF(YMIN.EQ.0.) GO TO 175                                        PLOTBEV   422
          IF(LIMSW) GO TO 175                                             PLBNV82    15
*                         PLOT TICK MARKS FOR NEGATIVE XEQ.               PLOTBEV   423
          Y(1) = 0.                                                       PLOTBEV   424
          Y(2) = 0.                                                       PLOTBEV   425
          X(3)=X(2) $ Y(3)=Y(2)                                           PLFORM     22
          CALL CCPLOT(X,Y,3,4HJOIN,1,1,1H-)                               PLFORM     23
          ISM = ABS(YMIN)                                                 PLTB682   121
          DO 171 I=IS1,ISM,IS1                                            PLOTBEV   430
          Y(1) = -I                                                       PLOTBEV   431
          Y(2) = -I                                                       PLOTBEV   432
          X(3)=X(2) $ Y(3)=Y(2)                                           PLFORM     24
          CALL CCPLOT(X,Y,3,4HJOIN,1,1,1H-)                               PLFORM     25
          IM=-I                                                           PLDRF      51
          WRITE(98,9) IM                                                  PLDRF      52
          CCY=CCYMIN +(Y(1)-YMIN)*(CCYMAX-CCYMIN)/(YMAX-YMIN)             MAY1179    10
          CALL CCLTR(CCX,CCY,0,1)                                         PLDRF      53
      171 CONTINUE                                                        PLOTBEV   434
      175 CONTINUE                                                        PLOTBEV   435
          CCX=CCXS $ CCY=CCYS                                             PLDRF      54
          CCXMAX = CCXMAX - T2                                            PLOTBEV   436
          CCXMIN = SAVXMN                                                 PLOTBEV   437
                                                                          PLOTBEV   438
************************************************************************* PLOTBEV   439
C   SET UP PLOT                                                           PLOTBEV   440
                                                                          PLOTBEV   441
          CCYMAX = PTOP                                                   PLOTBEV   442
          CCYMIN = PBOT                                                   PLOTBEV   443
          YMAX = FLT(1)                                                   PLOTBEV   444
          YMIN = 0.                                                       PLOTBEV   445
          NN = NP                                                         PLOTBEV   446
          DO 172 I=1,KW                                                   PLOTBEV   447
      172 XPL(I)=XX(I)                                                    PLOTBEV   448
                                                                          PLOTBEV   449
****                      PLOT CURVES                                     PLOTBEV   450
*          WRITE NAMES OF PLOTS AT MAXIMUM POINTS                         PLOTBEV   451
******                                                                    PLOTBEV   452
```

```
                                                                  PLOTBEV   453
*        PLOT BETAX                                               PLOTBEV   455
      DO 173 I=1,KW                                               PLOTBEV   456
  173 YPL(I)=Y1(I)                                                PLOTBEV   457
      CALL CCPLOT(XPL(J1),YPL(J1),NN,4HJOIN,1,1,1HH)              PLBNV82    16
*                  PLOT BETAY                                     PLOTBEV   461
      DO 174 I=1,KW                                               PLOTBEV   462
  174 YPL(I)=Y3(I)                                                PLOTBEV   463
      CALL CCPLOT(XPL(J1),YPL(J1),NN,4HJOIN,1,1,1HV)              PLB882      2
                                                                  PLOTBEV   467
*  FIND PAPER COORDINATES OF MAXIMUM POINTS                       PLOTBEV   468
                                                                  PLOTBEV   469
      M1 = J1                                                     PLOTBEV   470
      M2 = J1                                                     PLOTBEV   471
      M3 = J1                                                     PLOTBEV   472
      M4 = J1                                                     PLOTBEV   473
      JJ = J1 + 1                                                 PLOTBEV   474
      Y1MX = Y1(J1)                                               PLOTBEV   475
      Y2MX = Y2(J1)                                               PLOTBEV   476
      Y3MX = Y3(J1)                                               PLOTBEV   477
      Y4MX = Y4(J1)                                               PLOTBEV   478
      JN = J1 + NN - 1                                            PLOTBEV   479
      DO 184 J=JJ,JN                                              PLOTBEV   480
      IF (Y1(J).LE.Y1MX)  GO TO 178                               PLOTBEV   481
      Y1MX = Y1(J)                                                PLOTBEV   482
      M1 = J                                                      PLOTBEV   483
  178 IF (Y2(J).LE.Y2MX) GO TO 179                                PLOTBEV   484
      Y2MX = Y2(J)                                                PLOTBEV   485
      M2 = J                                                      PLOTBEV   486
  179 IF (Y3(J).LE.Y3MX) GO TO 182                                PLOTBEV   487
      Y3MX = Y3(J)                                                PLOTBEV   488
      M3 = J                                                      PLOTBEV   489
  182 IF (Y4(J).LE.Y4MX)  GO TO 184                               PLOTBEV   490
      Y4MX = Y4(J)                                                PLOTBEV   491
      M4 = J                                                      PLOTBEV   492
  184 CONTINUE                                                    PLOTBEV   493
                                                                  PLOTBEV   494
      PY = Y1(M1)                                                 PLOTBEV   495
      IF (PY.GT.YMAX)    GO TO 183                                PLOTBEV   496
      CCY = CCYMIN + (PY-YMIN)*(CCYMAX-CCYMIN)/(YMAX-YMIN)        PLOTBEV   497
      PX = XX(M1)                                                 PLOTBEV   498
      CCX = CCXMIN +(PX-XMIN)*(CCXMAX-CCXMIN)/(XMAX-XMIN)         PLOTBEV   499
      IF (M1.EQ.J1)  CCX = CCX + 8.                               PLOTBEV   500
      IF (M1.EQ.JN)  CCX = CCX - 10.                              PLOTBEV   501
      IF (KB.NE.0) GO TO 181                                      PLOTBEV   502
      CALL CCLTR(CCX-5.,CCY+11.,0,1,1HW)                          PLOTBEV   503
      CALL CCLTR(CCX   ,CCY+7.,0,0,2H X)                          PLBNV82    17
      GO TO 183                                                   PLOTBEV   505
  181 CONTINUE                                                    PLOTBEV   506
      CALL CCLTR(CCX-5.,CCY+12.,0,1,5HBETAX)                      PLTB682   134
  183 CONTINUE                                                    PLOTBEV   508
      PY = Y3(M3)                                                 PLOTBEV   509
      IF (PY.GT.YMAX)    GO TO 187                                PLOTBEV   510
      CCY = CCYMIN + (PY-YMIN)*(CCYMAX-CCYMIN)/(YMAX-YMIN)        PLOTBEV   511
      PX = XX(M3)                                                 PLOTBEV   512
      CCX = CCXMIN + (PX-XMIN)*(CCXMAX-CCXMIN)/(XMAX-XMIN)        PLOTBEV   513
      IF (M3.EQ.J1)  CCX = CCX + 8.                               PLOTBEV   514
```

```
          IF (M3.EQ.JN)  CCX = CCX - 10.                          PLOTBEV   515
          IF (KB.NE.0) GO TO 186                                  PLOTBEV   516
          CALL CCLTR(CCX-5.,CCY+11.,0,1,1HW)                      PLOTBEV   517
          CALL CCLTR(CCX   ,CCY+7.,0,0,2H Y)                      PLBNV82    18
          GO TO 187                                               PLOTBEV   519
      186 CONTINUE                                                PLOTBEV   520
          CALL CCLTR(CCX-5.,CCY+12.,0,1,5HBETAY)                  PLTB682   135
      187 CONTINUE                                                PLOTBEV   522
                                                                  PLOTBEV   523
*            PLOT XEQ,YEQ                                         PLOTBEV   524
          YMAX = FLT(2)                                           PLOTBEV   525
          YMIN = FLT(3)                                           PLOTBEV   526
C IF YEQ=0, DO NOT PLOT IT                                        MAY1179    11
          IF(M4.EQ.J1.AND.Y4MX.EQ.0.) GO TO 1891                  MAY1179    12
          DO 189 I=1,KW                                           PLOTBEV   530
      189 YPL(I)=Y4(I)                                            PLOTBEV   531
          CALL CCPLOT(XPL(J1),YPL(J1),NN,4HJOIN,1,1,1HY)          PRCH        8
     1891 CONTINUE                                                MAY1179    13
          IF (.NOT.PSW)  GO TO 191                                PLTB682   136
C  READJUST LOWER LIMITS FOR SMALL NEGATIVE VALUE                 PLTB682   137
          SVCYMIN = CCYMIN                                        PLTB682   138
          CCY = (CCYMAX-CCYMIN)*.01                               PLTB682   139
          CCYMIN = CCYMIN - CCY                                   PLTB682   140
          YMIN = YMIN - ONEP                                      PLTB682   141
      191 DO 188 I = 1,KW                                         PLTB682   142
      188 YPL(I)=Y2(I)                                            PLTB682   143
          CALL CCPLOT(XPL(J1),YPL(J1),NN,4HJOIN,1,1,1HX)          PLTB682   144
*            DRAW X AXIS                                          PLTB682   145
          IF (.NOT.PSW)  GO TO 195                                PLTB682   146
C  RESTORE PREVIOUS LIMITS                                        PLTB682   147
          YMIN = FLT(3)                                           PLTB682   148
          CCYMIN = SVCYMIN                                        PLTB682   149
          GO TO 185                                               PLTB682   150
      195 IF (YMIN.EQ.0.)  GO TO 185                              PLTB682   151
          X(1) = XMIN                                             PLTB682   152
          X(2) = XMAX                                             PLTB682   153
          Y(1) = 0.                                               PLTB682   154
          Y(2) = 0.                                               PLTB682   155
          CALL CCPLOT(X,Y,2,4HJOIN)                               PLTB682   156
      185 CONTINUE                                                PLTB682   157
                                                                  PLTB682   158
          PY = Y2(M2)                                             PLTB682   159
          IF (PY.GT.YMAX)    GO TO 190                            PLTB682   160
          CCY = CCYMIN + (PY-YMIN)*(CCYMAX-CCYMIN)/(YMAX-YMIN)    PLTB682   161
          PX = XX(M2)                                             PLTB682   162
          CCX = CCXMIN + (PX-XMIN)*(CCXMAX-CCXMIN)/(XMAX-XMIN)    PLTB682   163
          IF (M2.EQ.J1)  CCX = CCX + 8.                           PLTB682   164
          IF (M2.EQ.JN) CCX = CCX - 15.                           PLTB682   165
          CALL CCLTR(CCX-5.,CCY+12.,0,1,3HXEQ)                    PLTB682   166
      190 CONTINUE                                                PLOTBEV   551
          PY = Y4(M4)                                             PLOTBEV   552
          IF (PY.GT.YMAX)    GO TO 192                            PLOTBEV   553
          IF(M4.EQ.J1.AND.Y4MX.EQ.0.) GO TO 192                   MAY1179    14
          CCY = CCYMIN + (PY-YMIN)*(CCYMAX-CCYMIN)/(YMAX-YMIN)    PLOTBEV   554
          PX = XX(M4)                                             PLOTBEV   555
          CCX = CCXMIN + (PX-XMIN)*(CCXMAX-CCXMIN)/(XMAX-XMIN)    PLOTBEV   556
          IF (M4.EQ.J1)  CCX = CCX + 8.                           PLOTBEV   557
```

```
            IF (M4.EQ.JN)  CCX = CCX - 15.                             PLOTBEV   558
            CALL CCLTR(CCX-5.,CCY+12.,0,1,3HYEQ)                       PLTB682   167
        192 CONTINUE                                                   PLOTBEV   560
            CALL CCNEXT                                                PLOTBEV   561
        200 CONTINUE                                                   PLOTBEV   562
      *           IF PLOTTED W, RETURN ARRAYS TO BETA                  PLOTBEV   563
            IF (KB.NE.0) RETURN                                        PLOTBEV   564
            DO 202 I=1,KW                                              PLOTBEV   565
            Y1(I) = Y1(I)*Y1(I)                                        PLOTBEV   566
            Y3(I) = Y3(I)*Y3(I)                                        PLOTBEV   567
        202 CONTINUE                                                   PLOTBEV   568
            RETURN                                                     PLOTBEV   569
      *        ERROR IN INPUT PLOT POSITIONS                           PLOTBEV   570
        205 PRINT 10,IPRS                                              PLOTBEV   571
        1   FORMAT (8A10)                                              PLOTBEV   572
        2   FORMAT (A4)                                                PLOTBEV   573
        3   FORMAT (F8.2)                                              PLOTBEV   574
        4   FORMAT (6A10)                                              PLOTBEV   575
         5  FORMAT(F8.1)                                               PLTB682   168
         6  FORMAT(F7.2)                                               PLBNV82    19
        8   FORMAT (I3)                                                PLOTBEV   576
        9   FORMAT (I5)                                                PLOTBEV   577
        10  FORMAT (5X,*ERROR.  CANNOT PLOT FROM POSITION *,I5,* TO POSITION*, PLOTBEV   578
          1  I5)                                                       PLOTBEV   579
            END                                                        PLOTBEV   580
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
402	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
416	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
419	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
423	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
426	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
430	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
433	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
437	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
442	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
448	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
475	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
478	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
501	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
506	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
522	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
550	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
616	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
617	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
620	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
630	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
631	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
634	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
666	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
676	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.
686	I	)	ARGUMENT COUNT INCONSISTENT WITH PRIOR USAGE.

```
      SUBROUTINE PRINTV(M,K)                                            PRINTV     2
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER   SWTCH      2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                          SWTCH      3
                                                                        SWTCH      4
      LOGICAL BEGIN                                                     PRINTV     5
      DIMENSION NAM(7),V(10,7),IH(12),CV(70)                            PRINTV     6
      EQUIVALENCE (CV,V)                                                PRINTV     7
      INTEGER ELNUM,BDAT                                                PRINTV     8
      DATA (IH(I),I=1,12)/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,2H10,     PRINTV     9
     12H11,2H12/                                                        PRINTV    10
      DATA IM,IB/1H-,1H /                                               PRINTV    11
                                                                        PRINTV    12
      BEGIN = .TRUE.                                                    PRINTV    13
      JJ = 0                                                            PRINTV    14
      DO 100 IJ=1,K                                                     PRINTV    15
      IF (PV) GO TO 13                                                  PRINTV    16
      NM = BDAT(M,IJ)                                                   PRINTV    17
      M1 = ELNUM(NM)                                                    PRINTV    18
      GO TO 14                                                          PRINTV    19
   13 NM = MNAME(M)                                                     PRINTV    20
      M1=M                                                              PRINTV    21
   14 NCOL = INFF(23,M1)                                                PRINTV    22
      NROW = INFF(22,M1)                                                PRINTV    23
      NTYPE= INFF(21,M1)                                                PRINTV    24
      LOC = 1                                                           PRINTV    25
      IF (.NOT.VCSW) GO TO 15                                           PRINTV    26
      NCOL = 1                                                          PRINTV    27
      DECODE (10,6,NM) NUM                                              PRINTV    28
      LOC = (NUM-1)*NROW + 1                                            PRINTV    29
   15 IF(NTYPE.NE.4HSNGL.AND.JJ.NE.0) GO TO 21                          PRINTV    30
      IF (NCOL.GT.7) NCOL=7                                             PRINTV    31
      ISUB=JJ+NCOL                                                      PRINTV    32
      IF (ISUB.GT.7) GO TO 21                                           PRINTV    33
      IF (NTYPE.NE.4HSNGL) NCOL=1                                       PRINTV    34
      IF (NROW.NE.NROW1.AND..NOT.BEGIN) GO TO 21                        PRINTV    35
      DO 20 JK =1,NCOL                                                  PRINTV    36
      JJ = JJ + 1                                                       PRINTV    37
      NAM(JJ) = NM                                                      PRINTV    38
      JM = IB                                                           PRINTV    39
      JN = IB                                                           PRINTV    40
      IF(NCOL.LE.1) GO TO 17                                            PRINTV    41
      JM = IM                                                           PRINTV    42
      JN = IH(JK)                                                       PRINTV    43
   17 ENCODE(10,7,NAM(JJ)) NM,JM,JN                                     PRINTV    44
      IF (NTYPE.EQ.4HSNGL) GO TO 16                                     PRINTV    45
C  COMPLEX OR DOUBLE PRECISION VECTOR                                   PRINTV    46
      NC = NROW*2                                                       PRINTV    47
      CALL DATA(M1,5,LOC,NC,CV(1))                                      PRINTV    48
      GO TO 20                                                          PRINTV    49
   16 CALL DATA(M1,5,LOC,NROW,V(1,JJ))                                  PRINTV    50
      LOC = LOC + NROW                                                  PRINTV    51
```

```
   20 CONTINUE
      IF (NTYPE.NE.4HSNGL) GO TO 35
      BEGIN = .FALSE.
      IF (ISUB.EQ.7.OR.IJ.EQ.K) GO TO 22
      NROW1 = NROW
      GO TO 100
   21 ASSIGN 15 TO IRET
      NR = NROW1
      GO TO 25
   22 ASSIGN 100 TO IRET
      NR = NROW
C  TIME TO PRINT
   25 WRITE (3,1)
      CALL DASH
      WRITE (3,4)(NAM(J),J=1,JJ)
      DO 30 L=1,NR
      WRITE (3,5)(V(L,J),J=1,JJ)
   30 CONTINUE
      GO TO 40
C  PRINT COMPLEX OR DOUBLE PRECISION VECTORS
   35 WRITE (3,1)
      CALL DASH
      IF (NTYPE.EQ.4HCPLX) GO TO 37
      WRITE (3,9) NAM(1)
      GO TO 38
   37 WRITE (3,8) NAM(1)
   38 WRITE (3,10) (CV(J),J=1,NC)
      ASSIGN 100 TO IRET
   40 BEGIN = .TRUE.
      JJ = 0
      GO TO IRET (15,100)
  100 CONTINUE
      CALL DASH
      RETURN
    1 FORMAT (//)
    4 FORMAT (4X,7(9X,A7))
    5 FORMAT (4X,7(2X,F14.8))
    6 FORMAT (1X,I1,8X)
    7 FORMAT (A4,A1,A2,3X)
    8 FORMAT (16H COMPLEX VECTOR ,A5)
    9 FORMAT (25H DOUBLE PRECISION VECTOR ,A5)
   10 FORMAT (30X,2(E14.8,5X))
      END
```

```
      SUBROUTINE PRNT(M)                                                  PRNT        2
C  PRINTS THE KA-TH PARAMETER OF TYPE KB FOR SPECIFIED ELEMENTS.          PRNT        3
C  DEFAULTS ARE PARAMETER 1 OF TYPE 1                                     PRNT        4
                                                                          BLANK       2
      LEVEL 2, STORE,INFF,IWORK                                           BLANK       3
      COMMON STORE(48000),IWORK(10)                                       86MARSIZ    1
      DIMENSION INFF(24,2000)                                             86MARSIZ    2
      EQUIVALENCE (INFF,STORE)                                            86MARSIZ    3
                                                                          BLANK       5
      COMMON/BMI/MI(5000)                                                 BMIL        1
      COMMON/NELS/NELS                                                    BMIL        2
      LEVEL 2,MI                                                          BMIL        3
                                                                          BMIL        4
                                                                          BMI         3
C                                                                         PRNT        7
      DIMENSION VAL(100),IVAL(100),NAMTYPE(8),NAM(100)                    PRNT        8
      EQUIVALENCE (VAL,IVAL)                                              PRNT        9
      INTEGER BDAT,ELNUM                                                  PRNT       10
C                                                                         PRNT       11
      DATA LIM,IBLNK/10,1H /                                              PRNT       12
      DATA NAMTYPE/5HFL PT,3HBCD,3HINT,1H ,2HLQ,10HFL STORAGE,            PRNT       13
     1             3HLQ2,3HLQ3/                                           PRNT       14
C                                                                         PRNT       15
C  RETRIEVE ELEMENTS, PARAMETER AND TYPE.  SET DEFAULTS.                  PRNT       16
      NUM = INFF(17,M)                                                    PRNT       17
      KNDX = INFF(4,M)                                                    PRNT       18
      KTYPE = INFF(5,M)                                                   PRNT       19
      IF (KNDX.EQ.0)  KNDX = 1                                            PRNT       20
      IF (KTYPE.EQ.0)  KTYPE = 5                                          PRNT       21
      J = 0                                                               PRNT       22
      DO 10 I=1,NUM                                                       PRNT       23
      NAME = BDAT(M,I)                                                    PRNT       24
      IF (NAME.EQ.IBLNK)  GO TO 10                                        PRNT       25
      J = J + 1                                                           PRNT       26
      NAM(J) = NAME                                                       PRNT       27
      MI(J) = ELNUM(NAME)                                                 PRNT       28
   10 CONTINUE                                                            PRNT       29
      NUM = J                                                             PRNT       30
C  PRINT HEADING                                                          PRNT       31
      WRITE (3,100) KNDX,KTYPE,NAMTYPE(KTYPE)                             PRNT       32
      DO 18 I=1,NUM                                                       PRNT       33
      NK = MI(I)                                                          PRNT       34
       GO TO (15,12,13,14,15,15,15,15) KTYPE                              PRNT       35
   12 IVAL(I) = BDAT(NK,KNDX)                                             PRNT       36
      GO TO 18                                                            PRNT       37
   13 IVAL(I) = IDAT(NK,KNDX)                                             PRNT       38
      GO TO 18                                                            PRNT       39
   14 RETURN                                                              PRNT       40
   15 CALL DATA (NK,KTYPE,KNDX,1,VAL(I))                                  PRNT       41
   18 CONTINUE                                                            PRNT       42
C                                                                         PRNT       43
C WRITE NAMES OF ELEMENTS WITH VALUES UNDERNEATH, 10 TO A LINE.           PRNT       44
      NM = 0                                                              PRNT       45
   20 I1 = NM + 1                                                         PRNT       46
      NM = NM + LIM                                                       PRNT       47
      IF (NM.GT.NUM)  NM = NUM                                            PRNT       48
      WRITE (3,105) (NAM(I),I=I1,NM)                                      PRNT       49
```

```
      GO TO (21,22,23,24,21,21,21,21) KTYPE
   21 WRITE (3,106) (VAL(I),I=I1,NM)
      GO TO 25
   22 WRITE (3,107) (IVAL(I),I=I1,NM)
      GO TO 25
   23 WRITE (3,108) (IVAL(I),I=I1,NM)
      GO TO 25
   24 RETURN
   25 IF(NM.NE.NUM)  GO TO 20
      WRITE (3,110)
      RETURN
C
  100 FORMAT(1X,130(1H-)/2X,*VARIABLE*,I3,1X,*OF TYPE*,I2,1X,*(*,
     1     A5,*)*)
  105 FORMAT(/1X,10(7XA5))
  106 FORMAT(2X,10(F12.6))
  107 FORMAT(10(8X,A4))
  108 FORMAT(10(2X,I10))
  110 FORMAT(1X,130(1H-))
      END
```

```
      SUBROUTINE PRNTAB(M)                                            PRNTAB     2
                                                                      BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                       BLANK      3
      COMMON STORE(48000),IWORK(10)                                   86MARSIZ   1
      DIMENSION INFF(24,2000)                                         86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                        86MARSIZ   3
                                                                      BLANK      5
      EQUIVALENCE(IVAR,FVAR)                                          PRNTAB     4
      DIMENSION NAM(10),NCOL(10),F(10)                                PRNTAB     5
      INTEGER ELNUM                                                   PRNTAB     6
      LOGICAL PRES                                                    PRNTAB     7
                                                                      PRNTAB     8
C  IF M IS INDEX OF TAB INSTRUCTION, PRINT THAT ONE ONLY              PRNTAB     9
      PRES = .FALSE.                                                  PRNTAB    10
      IF (INFF(1,M).NE.3HTAB) GO TO 200                               PRNTAB    11
C  CALLED FROM TABULATE WHEN VECTOR FILLED                            PRNTAB    12
      PRES = .TRUE.                                                   PRNTAB    13
      IK = 1                                                          PRNTAB    14
      NAM(1) = INFF(2,M)                                              PRNTAB    15
      GO TO 205                                                       PRNTAB    16
  200 CONTINUE                                                        PRNTAB    17
      KA = INFF(4,M)                                                  PRNTAB    18
      IK = KA                                                         PRNTAB    19
C  PICK UP TAB NAMES                                                  PRNTAB    20
      CALL DATA(M,2,1,KA,NAM)                                         PRNTAB    21
  205 CONTINUE                                                        PRNTAB    22
      DO 220 I=1,IK                                                   PRNTAB    23
C  GET INDEX OF TABLE NAME AND NUMBER OF TABLE ENTRIES                PRNTAB    24
      INDX = ELNUM(NAM(I))                                            PRNTAB    25
      NUM = INFF(4,INDX)                                              PRNTAB    26
C   GET NAMES FOR COLUMN HEADINGS AND PRINT THEM                      PRNTAB    27
      CALL DATA(INDX,2,1,NUM,NCOL)                                    PRNTAB    28
      WRITE (3,1) (NCOL(J),J=1,NUM)                                   PRNTAB    29
C  GET CURSOR.  COMPUTE NUMBER OF ROWS TO BE PRINTED                  PRNTAB    30
      LQ = INFF(10,INDX)                                              PRNTAB    31
      FVAR = STORE(LQ)                                                PRNTAB    32
      LC=IVAR                                                         PRNTAB    33
      NC = (LC-2)/NUM                                                 PRNTAB    34
      IC = 2                                                          PRNTAB    35
      DO 210 J=1,NC                                                   PRNTAB    36
      CALL DATA(INDX,5,IC,NUM,F)                                      PRNTAB    37
      IAST = 55B                                                      PRNTAB    38
      NK  =0                                                          PRNTAB    39
      DO 209 K=1,NUM                                                  PRNTAB    40
  209 IF (F(K).EQ.0.) NK = NK + 1                                     PRNTAB    41
      IF (NK.EQ.NUM) IAST = 47B                                       PRNTAB    42
      WRITE (3,2) IAST,(F(K),K=1,NUM)                                 PRNTAB    43
      IC = IC + NUM                                                   PRNTAB    44
  210 CONTINUE                                                        PRNTAB    45
C  RESET CURSOR                                                       PRNTAB    46
      IVAR=2                                                          PRNTAB    47
      STORE(LQ)=FVAR                                                  PRNTAB    48
  220 CONTINUE                                                        PRNTAB    49
C  IF CALLED FROM TAB, RESTORE PAGE.                                  PRNTAB    50
      IF (PRES) WRITE (3,100)                                         PRNTAB    51
  100  FORMAT (1H1)                                                   PRNTAB    52
      RETURN                                                          PRNTAB    53
```

```
    1     FORMAT (1H1,6X,10(7X,A5)//)                         PRNTAB    54
    2     FORMAT (4X,A1,2X,10F12.8)                           PRNTAB    55
          END                                                 PRNTAB    56
```

```
      SUBROUTINE PRNTV7(M,K)                                            PRNTV7     2
                                                                        PRNTV7     3
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER   SWTCH      2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                          SWTCH      3
                                                                        SWTCH      4
      LOGICAL PH                                                        PRNTV7     6
      DIMENSION V(7)                                                    PRNTV7     7
      INTEGER ELNUM,BDAT                                                PRNTV7     8
                                                                        PRNTV7     9
C  PV IS LOGICAL VARIABLE THAT TELLS PRINTV IT HAS BEEN CALLED BY PRNTV7 PRNTV7    10
C  AND SO DOES NOT HAVE LIST OF DIFFERENT VECTORS TO PRINT.             PRNTV7    11
C     PH CONTROLS COLUMN HEADING. IF PRINTV WAS CALLED, NEW HEADING     PRNTV7    12
C  IS NEEDED.                                                           PRNTV7    13
      PH = .FALSE.                                                      PRNTV7    14
      CALL DASH                                                         PRNTV7    15
      DO 100 I=1,K                                                      PRNTV7    16
      PV=.FALSE.                                                        PRNTV7    17
      LOC = 1                                                           PRNTV7    18
      NM = BDAT(M,I)                                                    PRNTV7    19
      M1 = ELNUM(NM)                                                    PRNTV7    20
      NROW = INFF(22,M1)                                                PRNTV7    21
      NTYPE=INFF(21,M1)                                                 PRNTV7    22
C IF VECTOR IS NOT 7 ROW OR IF COMPLEX OR DOUBLE,CALL OTHER PRINT ROUT. PRNTV7    23
      IF (NROW.EQ.7.AND.NTYPE.EQ.4HSNGL) GO TO 20                       PRNTV7    24
      PV = .TRUE.                                                       PRNTV7    25
      CALL PRINTV(M1,1)                                                 PRNTV7    26
      PV=.FALSE.                                                        PRNTV7    27
      PH = .FALSE.                                                      PRNTV7    28
      GO TO 100                                                         PRNTV7    29
C  PARTICLE VECTOR WITH 7 ROWS                                          PRNTV7    30
 20   NCOL = INFF(23,M1)                                                PRNTV7    31
      LOC = 1                                                           PRNTV7    32
      IF (.NOT.VCSW) GO TO 25                                           PRNTV7    33
      DECODE (10,2,NM) NUM                                              PRNTV7    34
      LOC = (NUM-1)*7+1                                                 PRNTV7    35
      NCOL = 1                                                          PRNTV7    36
 25   IF (.NOT.PH) WRITE (3,10)                                         PRNTV7    37
      DO 30 J=1,NCOL                                                    PRNTV7    38
      CALL DATA (M1,5,LOC,7,V)                                          PRNTV7    39
 28   JJ = J                                                            PRNTV7    40
      IF (VCSW) JJ=NUM                                                  PRNTV7    41
      WRITE (3,13) NM,JJ,(V(L),L=1,7)                                   PRNTV7    42
 29   LOC = LOC + 7                                                     PRNTV7    43
 30   CONTINUE                                                          PRNTV7    44
      PH = .TRUE.                                                       PRNTV7    45
 100  CONTINUE                                                          PRNTV7    46
      CALL DASH                                                         PRNTV7    47
      WRITE (3,14)                                                      PRNTV7    48
      RETURN                                                            PRNTV7    49
 2    FORMAT (1X,I1,8X)                                                 PRNTV7    50
 10   FORMAT (//17H   VECTOR       X,10X,2HDX,9X,1HY,10X,2HDY,9X,1HS,9X, PRNTV7    51
```

```
     14HDP/P)                                              PRNTV7    52
   13 FORMAT (5X,A4,I2,2X,7(F10.5,1X))                     PRNTV7    53
   14 FORMAT (///1H )                                      PRNTV7    54
      END                                                  PRNTV7    55
```

```
      SUBROUTINE QFUNCT(XK,XKPRIM,Q)                           QFUNCT     2
      XKPRIM=SQRT(1.-XK**2)                                    QFUNCT     3
      A=AMAX1(ABS(XK),XKPRIM)                                  QFUNCT     4
      EPSI=(1.-A)/(2.*A+4.*SQRT(A)+2.)                         QFUNCT     5
      E=EPSI**4                                                QFUNCT     6
      Q=EPSI+EPSI*E*(2.+E*(15.+E*150.))                        QFUNCT     7
      IF(XKPRIM.LT.A) Q=EXP(9.869604401089359/ALOG(Q))         QFUNCT     8
      RETURN                                                   QFUNCT     9
      END                                                      QFUNCT    10
```

```
CARD NR. SEVERITY  DETAILS    DIAGNOSIS OF PROBLEM

     7    I                   CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
```

```
                                                                       QFUNCT   11
      SUBROUTINE QUEX (X,DQX)                                          QUEX      2
      COMMON/QUE/V1,V2,G,CQXZ,CQYZ,CQX,CQY,M1,M2,P1,P2,MSR,MC,VV1,VV2  QUEX      3
     1,K1,K2                                                           QUEX      4
      COMMON/GRR/IERR,BSW,VSW,RSW,QSW,XSW                              GRR       2
      LOGICAL IERR,BSW,VSW,RSW,QSW,XSW                                 GRR       3
                                                                       GRR       4
      EXTERNAL QUEY                                                    QUEX      6
      IF(IERR) RETURN                                                  QUEX      7
      V1 = X                                                           QUEX      8
      CALL GRTB( 1,V2,7,QUEY)                                          QUEX      9
      DQX = (CQX - CQXZ) **2 + G*G                                     QUEX     10
      RETURN                                                           QUEX     11
      END                                                              QUEX     12
```

```
      SUBROUTINE QUEY (Y,DQY)                                             QUEY     2
      COMMON/QUE/V1,V2,G,CQXZ,CQYZ,CQX,CQY,M1,M2,P1,P2,MSR,MC,VV1,VV2     QUEY     3
     1,K1,K2                                                              QUEY     4
      COMMON/GRR/IERR,BSW,VSW,RSW,QSW,XSW                                 GRR      2
      LOGICAL IERR,BSW,VSW,RSW,QSW,XSW                                    GRR      3
                                                                          GRR      4
      INTEGER P1,P2                                                       QUEY     6
      DIMENSION RX(2,3) ,RY(2,3),RW(3)                                    QUEY     7
      DIMENSION RXRY(12),V(7),BXBY(20)                                    QUEY     8
                                                                          QUEY     9
      V2 = Y                                                              QUEY    10
      CALL REPFLT (M1,P1,V1)                                              QUEY    11
      CALL REPFLT (M2,P2,V2)                                              QUEY    12
      CALL DOIT(MSR)                                                      QUEY    13
      IF (BSW) GO TO 3                                                    QUEY    14
      IF(VSW) GO TO 4                                                     QUEY    15
      IF (RSW) GO TO 8                                                    QUEY    16
      CALL RXY(MC,RX,RY,RW)                                               QUEY    17
      CQX = (RX(1,1) + RX(2,2)) * .5                                      QUEY    18
      CQY = (RY(1,1) + RY(2,2)) * .5                                      QUEY    19
      GO TO 5                                                             QUEY    20
    3 CALL BET(MC,BXBY(1),BXBY(11),0)                                     QUEY    21
      CQX = BXBY(K1)                                                      QUEY    22
      CQY = BXBY(K2)                                                      QUEY    23
      GO TO 6                                                             QUEY    24
    4 CALL DATA (MC,5,1,7,V)                                              QUEY    25
      CQX = V(K1)                                                         QUEY    26
      CQY = V(K2)                                                         QUEY    27
      GO TO 6                                                             QUEY    28
    8 CALL RXY(MC,RXRY(1),RXRY(7),RW)                                     QUEY    29
      CQX = RXRY(K1)                                                      QUEY    30
      CQY = RXRY(K2)                                                      QUEY    31
    6 CQX = CQX/100.                                                      QUEY    32
      CQY=CQY/100.                                                        QUEY    33
    5 DQY = CQY - CQYZ                                                    QUEY    34
      G = DQY                                                             QUEY    35
      RETURN                                                              QUEY    36
      END                                                                 QUEY    37
```

```
      SUBROUTINE RDINF(INF,M)                                    RDINF       2
C  READS ONE BLOCK OF INFF(M) INTO INF                           RDINF       3
                                                                 BLANK       2
      LEVEL 2, STORE,INFF,IWORK                                  BLANK       3
      COMMON STORE(48000),IWORK(10)                              86MARSIZ    1
      DIMENSION INFF(24,2000)                                    86MARSIZ    2
      EQUIVALENCE (INFF,STORE)                                   86MARSIZ    3
                                                                 BLANK       5
      DIMENSION INF(1)                                           RDINF       5
      DO 20 I=1,24                                               RDINF       6
   20 INF(I) = INFF(I,M)                                         RDINF       7
      RETURN                                                     RDINF       8
      END                                                        RDINF       9
```

```
      SUBROUTINE RDINST                                                 RDINST     2
C PROCESSES FIRST CARD OF NEW INSTRUCTION WHICH IS CURRENTLY IN CARD (8) RDINST     3
C BY STORING NAME,OPNAME.                                               RDINST     4
      COMMON/INSTR/OPNAME,NAME,OP,KA,KB,OBJA,OBJB,NXTM,TRA,LQ,NQ,LIN,   INSTR      2
     1             NIN,LFL,NFL,LBC,NBC,NTOT,IX,KIND,TYPE,ROWS,COLS,EXTR, INSTR      3
     2             M,NF,NB,NI,MSUBR,ISIGN                               INSTR      4
      INTEGER OPNAME,OP,OBJA,OBJB,TRA,TYPE,ROWS,COLS,EXTR               INSTR      5
                                                                        INSTR      6
      COMMON/STORE/LMAX,LINF,LFILE,MAX,MIN,ICARD(11)                    STORE      2
*  LMAX AND MAX ARE DIMENSIONS OF STORE AND INFF                        STORE      3
                                                                        STORE      4
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1          LDFLG,FIN                                               CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL   DIM        2
                                                                        DIM        3
      INTEGER OPNUM,HPERIOD                                             RDINST     9
      DIMENSION IC(24)                                                  RDINST    10
      EQUIVALENCE( IC(1),OPNAME )                                       RDINST    11
      DATA HPERIOD,NULL/1H.,4HNULL/                                     RDINST    12
                                                                        RDINST    13
      IF(MIN.GE.1) GO TO 7                                              RDINST    14
      WRITE (3,200)                                                     RDINST    15
  200 FORMAT (6H0*****,*ELEMENTS OVERFLOWED. INCREASE INFF ARRAY AND *, RDINST    16
     1 *MAX.*)                                                          RDINST    17
      NSTO = LMAX - LFILE                                               RDINST    18
      NINFF = MAX - MIN                                                 RDINST    19
      WRITE (3,201) LMAX,LFILE,NSTO,MAX,NINFF,M                         RDINST    20
  201 FORMAT(/1X*CORE USE SUMMARY*,27X,*MAXIMUM*,17X,*USED*,8X,*UNUSED*/ RDINST    21
     1 34X,*STORE  (ELEMENT STORAGE)*,14X,I10,1X,*(LMAX)*,4X,I10,5X,I10/ RDINST    22
     2 34X,*INFF   (ELEMENT DEFINITIONS)*,10X,I10,1X,*(MAX)*,5X,I10,5X, RDINST    23
     3 I10)                                                             RDINST    24
      ERROR=.TRUE.                                                      RDINST    25
      RETURN                                                            RDINST    26
    7 CONTINUE                                                          RDINST    27
C                                                                       RDINST    28
C  ZERO COMMON BLOCK/INSTR/,BUT DONT ZERO M                             RDINST    29
      DO 10 I=1,24                                                      RDINST    30
   10 IC(I) = 0                                                         RDINST    31
      IC(21)=4HSNGL                                                     RDINST    32
      NFSW = .FALSE.                                                    RDINST    33
      IF(.NOT.LDFLG) EMPTY=.FALSE.                                      RDINST    34
      LDFLG=.FALSE.                                                     RDINST    35
      IF (EMPTY) GO TO 4                                                RDINST    36
      EMPTY=.TRUE.                                                      RDINST    37
      READ(2,5) (ICARD(IW),IW=1,8)                                      RDINST    38
    5 FORMAT(8A10)                                                      RDINST    39
                                                                        RDINST    40
    4 DECODE (10,3,ICARD) ISIGN                                         RDINST    41
    3 FORMAT (A1,9X)                                                    RDINST    42
*    IF ISIGN = P OR C, IT IS PAGE OR COMMENT AND NEEDS SPECIAL HANDLING RDINST    43
      IF (ISIGN.EQ.1HP) GO TO 100                                       RDINST    44
      IF (ISIGN.EQ.1HC) GO TO 100                                       RDINST    45
      IF (ISIGN.EQ.HPERIOD) GO TO 100                                   RDINST    46
                                                                        RDINST    47
      DECODE(20,2,ICARD)  ISIGN,NAME,OPNAME,KA,KB                       RDINST    48
```

```
 2    FORMAT (A1,A5,1X,A5,I3,1X,I3,1X)
C  KA,KB BECOME -0 IF BLANK.  RESET TO +0.
      IF (KA.EQ.0) KA=0
      IF (KB.EQ.0) KB=0
      OP = OPNUM(OPNAME)
      IF (OPNAME.EQ.3HRUN) M = M7END -ISAV7
      IF (ISIGN.EQ.1H-) OP = -OP
      CALL WRTINF( IC,M )
      RETURN

 100  OPNAME = NULL
      RETURN
      END
```

```
      FUNCTION REALNUM(LOCI)                                            REALNUM     2
C                                                                       REALNUM     3
                                                                        BLANK       2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK       3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ    1
      DIMENSION INFF(24,2000)                                           86MARSIZ    2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ    3
                                                                        BLANK       5
      COMMON/FLTN/IFL(15)                                               FLTN        2
                                                                        FLTN        3
      COMMON/TSW/TRSW,MCY(20),JM,LOCC                                   NOV3TSW     1
      LOGICAL TRSW                                                      TSW         3
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL      2
     1          LDFLG,FIN                                               CONTRL      3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL      4
                                                                        CONTRL      5
C                                                                       REALNUM     7
      INTEGER ELNUM,HRAND,RTBLNK                                        REALNUM     8
      LOGICAL NEG,INEG                                                  REALNUM     9
      DIMENSION Y(2)                                                    REALNUM    10
      EQUIVALENCE (IVAR,FVAR),(NMVAR,FMVAR)                             REALNUM    11
      EQUIVALENCE(LOC,LOCC)                                             NOV3RLNM    2
C                                                                       REALNUM    12
      DATA MASK/00007 77777 77777 77777 B/                              REALNUM    13
      DATA MASK1/77777 77777 00000 00000 B/                             REALNUM    14
      DATA MASKF/77000 00000 00000 00000 B/                             REALNUM    15
      DATA RTBLNK/00000 00000 55555 55555 B/                            REALNUM    16
      DATA HRAND/4HRAND/                                                REALNUM    17
C                                                                       REALNUM    18
      NEG = .FALSE.                                                     REALNUM    19
      ITR = 0                                                           REALNUM    20
       LOC = LOCI                                                       REALNUM    21
C                                                                       REALNUM    22
   20 FVAR = STORE(LOC)                                                 REALNUM    23
   21 REALNUM=FVAR                                                      RLNTIM      1
C    TEST FOR SYMBOLIC                                                  REALNUM    25
      NR = IVAR.AND.MASK                                                REALNUM    26
      NL = IVAR.AND..NOT.MASK                                           REALNUM    27
      IF(NL.NE.0.OR.NR.EQ.0) GO TO 80                                   RLNTIM      2
C    SYMBOLIC.    NESTING CANNOT EXCEED 10                              REALNUM    33
   25 ITR = ITR + 1                                                     REALNUM    34
      IF (ITR.GT.10)  GO TO 90                                          REALNUM    35
C  GET ELEMENT NAME                                                     RLNTIM      3
      DECODE(10,110,IVAR) NMVAR                                         RLNTIM      4
      IF (NMVAR.NE.1H$) GO TO 28                                        REALNUM    40
      REALNUM = FMVAR                                                   REALNUM    41
      RETURN                                                            REALNUM    42
   28 DECODE(5,100,IVAR) NPT                                            RLNTIM      5
  100 FORMAT (2X,I3)                                                    REALNUM    45
      MSK = MASKF.AND.NMVAR                                             REALNUM    46
C  CHECK FOR +, - IN ELEMENT NAME                                       REALNUM    47
      IF (MSK.NE.IFL(1)) GO TO 29                                       REALNUM    48
      NEG = (.NOT.NEG)                                                  RLRAND      1
   29 IF (MSK.NE.IFL(1).AND.MSK.NE.IFL(12))  GO TO 30                   REALNUM    52
      DECODE(6,105,NMVAR) NMVAR                                         RLNTIM      6
  105 FORMAT(1XA5)                                                      RLNTIM      7
   30 MN = ELNUM(NMVAR)                                                 REALNUM    56
```

```
C    LQ IS DEFAULT                                                   REALNUM     57
         IF (NPT.EQ.0) NPT = -1                                      REALNUM     58
         IF (NPT.LT.0)  GO TO 50                                     REALNUM     59
         LFMN = INFF(14,MN)                                          REALNUM     60
         LOC = LFMN - 1 + NPT                                        REALNUM     61
         GO TO 20                                                    REALNUM     62
C              RETRIEVE FROM LQ                                      NOV3RLNM     3
C        GET LOCC TO PASS THRU COMMON BLOCK FOR CHROM CORRECTION     NOV3RLNM     4
  50     LOCC=INFF(10,MN)-NPT-1                                      NOV3RLNM     5
         REALNUM=FLDAT(MN,-NPT)                                      NOV3RLNM     6
         FVAR=REALNUM                                                REALNUM     65
         IF(INFF(1,MN).NE.HRAND) GO TO 21                            RLRAND       2
C            RANDOM VALUE  =V1 <- (V2/2)                             REALNUM     69
         DO 58 I=1,2                                                 REALNUM     70
         FVAR = FLDAT(MN,I)                                          REALNUM     71
         NR = IVAR.AND.MASK                                          REALNUM     72
         NL = IVAR.AND..NOT.MASK                                     REALNUM     73
         IF (NL.EQ.0.AND.NR.NE.0)  GO TO 56                          REALNUM     74
         Y(I) = FVAR                                                 REALNUM     75
         GO TO 58                                                    REALNUM     76
  56     DECODE (10,110,NR) NAM2                                     REALNUM     77
  110    FORMAT (5X,A5)                                              REALNUM     78
         Y(I) = FLDAT(ELNUM(NAM2),1)                                 REALNUM     79
  58     CONTINUE                                                    REALNUM     80
         REALNUM = Y(1) + Y(2) * (RANF(D) - .5)                      REALNUM     81
  80     CONTINUE                                                    REALNUM     82
         IF (NEG) REALNUM=-REALNUM                                   REALNUM     83
         RETURN                                                      REALNUM     84
C    EXCEEDED DEPTH OF SYMBOLIC                                      REALNUM     85
  90     WRITE (3,115)                                               REALNUM     86
  115    FORMAT (5H ****,* SYMBOLIC FLOATING POINT VARIABLE EXCEEDED *  REALNUM     87
       1  *ALLOWABLE NESTING DEPTH OF 10.*)                          REALNUM     88
         ERROR = .TRUE.                                              REALNUM     89
         RETURN                                                      REALNUM     90
         END                                                         REALNUM     91
```

```
      SUBROUTINE REAR(NC)                                               REAR       2
      RETURN                                                            REAR       3
      END                                                               REAR       4
```

```
      SUBROUTINE REF(M,N)                                          REF        2
C  COMPUTES MATRIX REFLECTION                                      REF        3
*        REFLECT MATRIX AS 7 X 7, THEN STORE ACCORDING TO KIND     REF        4
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL DIM     2
                                                                   DIM        3
                                                                   BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                    BLANK      3
      COMMON STORE(48000),IWORK(10)                                86MARSIZ   1
      DIMENSION INFF(24,2000)                                      86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                     86MARSIZ   3
                                                                   BLANK      5
      DIMENSION RW(3),T(49),TK(7,7),RWT(3)                         REF        7
      DIMENSION ML(3)                                              REF        8
      COMMON /MM/ ML    $    LEVEL 2,ML                            REFLL      1
                                                                   REFLL      2
      DATA TK/49*0.0/                                              REF        9
                                                                   REF       10
*        COMPUTE REFLECTION BY MULTIPLYING MATRICES                REF       11
*                    K      M(INV)     K(INV)                      REF       12
*        WHERE K IS MATRIX TK AND K = K(INV)                       REF       13
                                                                   REF       14
*     GET INVERSE OF MATRIX                                        REF       15
      NN = -N                                                      REF       16
      CALL RTRV7(NN,T,RW)                                          REF       17
      CALL STOR7(M7END-2,T,RW)                                     REF       18
*        FORM K MATRIX AND MULTIPLY                                REF       19
      TK(1,1) = 1.                                                 REF       20
      TK(2,2) = -1.                                                REF       21
      TK(3,3) = 1.                                                 REF       22
      TK(4,4) = -1.                                                REF       23
      TK(5,5) = -1.                                                REF       24
      TK(6,6) = 1.                                                 REF       25
      TK(7,7) = 1.                                                 REF       26
      DO 1 I=1,3                                                   REF       27
    1 RWT(I) = 0.                                                  REF       28
      NT = M7END - 1                                               REF       29
      CALL STOR7(NT,TK,RWT)                                        REF       30
      ML(1)=NT                                                     REF       31
      ML(2)=M7END-2                                                REF       32
      ML(3)=NT                                                     REF       33
      CALL MMM(M,3,ML)                                             REF       34
C  LENGTH SHOULD BE SAME AS FIRST MATRIX,NOT THAT OF INVERSE.      REF       35
C    REVERSE SIGN OF LENGTH                                        REF       36
      RW(1) = -RW(1)                                               REF       37
      RW(2) = -RW(2)                                               REF       38
      RW(3) = -RW(3)                                               REF       39
      CALL STDAT(M,5,1,3,RW)                                       REF       40
      RETURN                                                       REF       41
      END                                                          REF       42
```

```
      SUBROUTINE REPBCD(M,J,B)                                          REPBCD     2
C  REPLACES BCD DATA FROM B INTO INDEX J OF ELEMENT M BCD DATA          REPBCD     3
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      LOC=INFF(16,M)+J-1                                                REPBCD     5
      STORE(LOC)=B                                                      REPBCD     6
      RETURN                                                            REPBCD     7
      END                                                               REPBCD     8
```

```
      SUBROUTINE REPFLT(M,J,F)                                          REPFLT     2
C REPLACES FLOATING POINT DATA FROM F INTO INDEX J OF ELEMENT M          REPFLT     3
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      LOC=INFF(14,M)+J-1                                                REPFLT     5
      STORE(LOC)=F                                                      REPFLT     6
      RETURN                                                            REPFLT     7
      END                                                               REPFLT     8
```

```
      SUBROUTINE REPL(M)                                                REPL       2
                                                                        REPL       3
*         REPLACE A VALUE IN A PREVIOUSLY DEFINED INSTRUCTION           REPL       4
*   THE 2ND BCD PARAMETER TELLS WHAT TYPE.                              REPL       5
*   F FOR FL.PT., H FOR HOLLERITH(BCD), I FOR INTEGER, K FOR KA OR KB   REPL       6
                                                                        REPL       7
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1          LDFLG,FIN                                               CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      DIMENSION ISTORE(1)                                               REPL      10
      EQUIVALENCE (ISTORE,STORE)                                        REPL      11
      INTEGER BDAT                                                      REPL      12
                                                                        REPL      13
      NM = BDAT(M,1)                                                    REPL      14
      N = MDAT(M,1)                                                     REPL      15
      KT = BDAT(M,2)                                                    REPL      16
      K = INFF(4,M)                                                     REPL      17
*     IF KT IS BLANK, ASSUME FLOATING POINT.                            REPL      18
      IF (KT.EQ.1H ) GO TO 100                                          REPL      19
      IF (KT.EQ.1HF) GO TO 100                                          REPL      20
      IF(KT.EQ.2HSF)  GO TO 105                                         REPL      21
      IF (KT.EQ.1HH) GO TO 110                                          REPL      22
      IF (KT.EQ.1HI) GO TO 115                                          REPL      23
      IF (KT.EQ.2HKA) GO TO 95                                          REPL      24
      IF (KT.EQ.1HQ) GO TO 117                                          REPL      25
      IF (KT.NE.2HKB) GO TO 120                                         REPL      26
      IK = 5                                                            REPL      27
      GO TO 96                                                          REPL      28
   95 IK = 4                                                            REPL      29
   96 IP = IDAT(M,1)                                                    REPL      30
      INFF(IK,N) = IP                                                   REPL      31
      IF (MODE.EQ.3)  WRITE (3,97) KT,NM,IP                             REPL      32
   97 FORMAT (6H  *** ,8X,*REPLACE *,A3,* OF *,A5,* WITH *,I5)          REPL      33
      RETURN                                                            REPL      34
                                                                        REPL      35
*                   FLOATING POINT                                      REPL      36
*   IF SYMBOLIC FL PT VARIABLE, INSERT AS STORE IN REPL                 REPL      37
  100 LOCM = INFF(14,M)                                                 REPL      38
      LOCN = INFF(14,N) -1 + K                                          REPL      39
      STORE(LOCN) = STORE(LOCM)                                         REPL      40
      IF (MODE.EQ.3)  WRITE (3,101) K,NM,F                              REPL      41
  101 FORMAT (6H  *** ,8X,*REPLACE *,I3,* OF *,A5,* WITH *,F10.6)       REPL      42
      RETURN                                                            REPL      43
                                                                        REPL      44
*                 SYMBOLIC FLOATING POINT -- INSERT AS IS               REPL      45
  105 LOCM = INFF(14,M)                                                 REPL      46
      LOCN = INFF(14,N) - 1 + K                                         REPL      47
      ISTORE(LOCN) = ISTORE(LOCM)                                       REPL      48
      RETURN                                                            REPL      49
                                                                        REPL      50
```

```
      *                 HOLLERITH (BCD)
  110 IP = BDAT(M,3)
      IF (ERROR)  GO TO 125
      CALL REPBCD(N,K,IP)
      IF (MODE.EQ.3)  WRITE (3,111) K,NM,IP
  111 FORMAT (6H  *** ,8X,*REPLACE *,I3,* OF *,A5,* WITH *,A5)
      RETURN

      *                 INTEGER
  115 IP = IDAT(M,1)
      IF (ERROR)  GO TO 125
      CALL REPINT(N,K,IP)
      IF (MODE.EQ.3)  WRITE (3,116) K,NM,IP
  116 FORMAT (6H  *** ,8X,*REPLACE *,I3,* OF *,A5,* WITH *,I5)
      RETURN

      *              FLOATING POINT VARIABLE INSERTED IN LQ STORAGE
  117 F = FDAT(M,1)
      CALL REPLQ(N,1,K,F)
      IF (MODE.EQ.3) WRITE(3,101) K,NM,F
      RETURN

      *                 ERROR
  120 WRITE (3,121) KT
  121 FORMAT (//6H  *** ,*ERROR IN REPL INSTRUCTION.  *,A5,* IS AN UNKNO
     1WN TYPE.*)
  130 ERROR = .TRUE.
      RETURN
  125 WRITE(3,126) KT
  126 FORMAT (//6H  *** ,*ERROR IN REPL INSTRUCTION. NO VALUES OF TYPE*,
     1  A5,*.*)
      RETURN
      END
```

```
      SUBROUTINE REPLQ(M,L,J,F)                                       REPLQ      2
C   PLACES F INTO INDEX J OF LQ STORAGE OF ELEMENT M                  REPLQ      3
C     L=1 FOR LQ        L=2 FOR LQ2     L=3 FOR LQ3                   REPLQ      4
                                                                      BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                       BLANK      3
      COMMON STORE(48000),IWORK(10)                                   86MARSIZ   1
      DIMENSION INFF(24,2000)                                         86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                        86MARSIZ   3
                                                                      BLANK      5
      LM = 10                                                         REPLQ      6
      IF (L.EQ.2) LM = 19                                             REPLQ      7
      IF (L.EQ.3) LM = 24                                             REPLQ      8
   12 LOC = INFF(LM,M) + J - 1                                        REPLQ      9
C  IF LQ3 OR LQ2 HAVE NOT BEEN SET, GET LQ2 OR LQ                     REPLQ     10
       IF (LOC.NE.0) GO TO 15                                         REPLQ     11
      IF (LM.EQ.10) GO TO 15                                          REPLQ     12
      IF (LM.EQ.24) LM = 19                                           REPLQ     13
      IF (LM.EQ.19) LM = 10                                           REPLQ     14
      GO TO 12                                                        REPLQ     15
   15 STORE(LOC) = F                                                  REPLQ     16
      RETURN                                                          REPLQ     17
      END                                                             REPLQ     18
```

```
                                                                         REPINT     2
      SUBROUTINE REPINT(M,J,FI)                                          REPINT     3
                                                                         REPINT     4
C  REPLACES INTEGER DATA FROM I INTO INDEX J OF ELEMENT INTEGER DATA     REPINT     5
                                                                         BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                          BLANK      3
      COMMON STORE(48000),IWORK(10)                                      86MARSIZ   1
      DIMENSION INFF(24,2000)                                            86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                           86MARSIZ   3
                                                                         BLANK      5
      LOC=INFF(12,M)+J-1                                                 REPINT     7
      STORE(LOC)=FI                                                      REPINT     8
      RETURN                                                             REPINT     9
      END                                                                REPINT    10
```

```
      SUBROUTINE RESRV(M0,KKND,IQ,                                      RESRV    2
     . N1,I1,N2,I2,N3,I3,N4,I4                                          RESRV    3
     .)                                                                 RESRV    4
C  MAKES RESERVATIONS FOR INPUT                                         RESRV    5
C    SEE COMMENTS IN LOAD FOR VARIABLE INPUT                            RESRV    6
C                                                     THE NJ INTEGERS  RESRV    7
C      NSETS = NUMBER OF SETS OF INPUT DATA     IF IJ= 0 SPECIFY NJ-S OF RESRV    8
C      NJ    = NUMBER OF DATA OF JTH SET               SUBSEQUENT SETS RESRV    9
C      IJ    = TYPE   OF DATA OF JTH SET.          = 1  FLOATING F10.6 RESRV   10
C                                                  = 2  BCD      A5    RESRV   11
C ONLY THE FIRST SET, J=1, MAY BE OF THE TYPE IJ=0.  = 3  INTEGER  I5    RESRV   12
C  SEE EXAMPLES IN LOAD.                           = 4  SKIP     10X   RESRV   13
*  IF THE LAST SET IS BCD AND THERE ARE NO INTEGER SETS, ONE MAY PUT   RESRV   14
C I(NSETS)=-2  THEN LOAD WILL READ CARDS FILLED WITH BCD DATA          RESRV   15
C  UNTIL IT FINDS A NEW INSTRUCTION CARD, AND SET NBC=12*CARDS READ.   RESRV   16
C                                                                       RESRV   17
                                                                        BLANK    2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK    3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ 1
      DIMENSION INFF(24,2000)                                           86MARSIZ 2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ 3
                                                                        BLANK    5
      COMMON/INSTR/OPNAME,NAME,OP,KA,KB,OBJA,OBJB,NXTM,TRA,LQ,NQ,LIN,   INSTR    2
     1             NIN,LFL,NFL,LBC,NBC,NTOT,IX,KIND,TYPE,ROWS,COLS,EXTR, INSTR    3
     2             M,NF,NB,NI,MSUBR,ISIGN                               INSTR    4
      INTEGER OPNAME,OP,OBJA,OBJB,TRA,TYPE,ROWS,COLS,EXTR               INSTR    5
                                                                        INSTR    6
      COMMON/STORE/LMAX,LINF,LFILE,MAX,MIN,ICARD(11)                    STORE    2
*  LMAX AND MAX ARE DIMENSIONS OF STORE AND INFF                        STORE    3
                                                                        STORE    4
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL   2
     1          LDFLG,FIN                                               CONTRL   3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL   4
                                                                        CONTRL   5
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL   DIM      2
                                                                        DIM      3
      DIMENSION ND(10),I(10),NVAR(10),NKIND(10),INF(24)                 RESRV   23
      LOGICAL ROWSW,COLSW,NQSW,CYSW                                     RESRV   24
      DATA (NKIND(J),J=1,10)/3HMXY,3HPVC,3HMAT,2HM6,2HR7,3HROT,3HMAP,   RESRV   25
     13HPRD,4HSCAL,1H /                                                 RESRV   26
C                                                                       RESRV   27
                                                                        RESRV   28
      IF(MODE.EQ.3) RETURN                                              RESRV   29
      IF (.NOT.RSRV) RETURN                                             RESRV   30
      NQSW=.FALSE.                                                      RESRV   31
      ROWSW=.FALSE.                                                     RESRV   32
      COLSW=.FALSE.                                                     RESRV   33
      IF (ROWS.NE.0) ROWSW=.TRUE.                                       RESRV   34
      IF (COLS.NE.0) COLSW=.TRUE.                                       RESRV   35
C STORE DATA SPECIFICATION NUMBERS IN ARRAYS ND(J) AND I(J)             RESRV   36
      CALL ARGINP(ND,I,NSETS,                                           RESRV   37
     . N1,I1,N2,I2,N3,I3,N4,I4                                          RESRV   38
     .)                                                                 RESRV   39
 8    NIN =0                                                            RESRV   40
      NFL =0                                                            RESRV   41
      NBC =0                                                            RESRV   42
      LQ = LFILE+ 1                                                     RESRV   43
```

```
      LQ2 = LQ
      LQ3 = LQ
C  IF IQ.GE.0, NQ=IQ=NO. LOCATIONS RESERVED STARTING AT LQ
C  IF IQ.EQ.-0, NQ WILL BE CALCULATED ACCORDING TO KIND
C  IF IQ.LE.-1, ABS(IQ) REFERS TO THE IQ-TH SET OF DATA.
C  LQ WILL BE SET = LIN=LBC,OR LFL, DEPENDING ON DATA TYPE
      IF (IQ.EQ.0) GO TO 10
      IF (IQ.GE.0) GO TO 15
C  IQ IS NEGATIVE NUMBER.
C  NQ WILL BE DETERMINED BY THE IQ-TH SET OF INPUT DATA
      NQ = 0
      GO TO 16
C  DETERMINE SIGN OF ZERO
 10   IZ = IQ.AND.1
      IF (IZ.NE.0) NQSW = .TRUE.
      IF (NQSW) GO TO 18
C  IQ IS POSITIVE NUMBER. SET NQ=IQ
 15   NQ = IQ
C  IF KIND NOT SPECIFIED, SET IT TO OPNAME
 16   IF (KKND.EQ.0) KIND = INFF(1,M0)
      IF (KIND.EQ.0) KIND = NKIND(KKND)
      IF (OPNAME.NE.4HFXPT) GO TO 30
C  FXPT.  SET UP 3 LQ LOCATIONS
      LQ2 = LQ + KADD + 49
      LQ3 = LQ2 + 35
      NQ = LQ3 + 15 - LQ
      GO TO 30
C  IQ IS -0. NQ DETERMINED BY KIND
 18   IF (KKND.NE.4H    ) GO TO 181
C  KIND IS UNKNOWN. SET NQ = 0. LOAD WILL SET LQ AND NQ AFTER LOADING.
      KIND = KKND
      NQ = 0
      GO TO 30
C  KIND IS KNOWN. FIND OUT WHICH ONE.
 181  NK = 1
      DO 19 J=1,10
      IF (KKND.EQ.NK) GO TO 20
      NK=NK+1
 19   CONTINUE
      WRITE (3,110)
 110  FORMAT (/5X,25H ILLEGAL KIND ENCOUNTERED//)
      ERROR=.TRUE.
      RETURN
 20   KIND=NKIND(NK)
      GO TO (21,22,23,24,25,26,25,25,29), NK
C  KIND IS MXPMY. MATRIX PAIRS RX AND RY
 21   NQ = 12 + KADDR
      IF (.NOT.ROWSW) ROWS=2
      IF (.NOT.COLSW) COLS=7
      GO TO 30
C  KIND IS PVECS, PARTICLE VECTOR
C  IF NOT OTHERWISE INDICATED, 7 ROWS BY KA COLUMNS
 22   NQ = 7*KA
      IF (.NOT.ROWSW) ROWS = 7
      IF (.NOT.COLSW) COLS= KA
      GO TO 30
C  KIND IS SCALAR
```

```
   29 NQ = 1
      IF (TYPE.NE.4HSNGL) NQ = 2
      GO TO 30
C  KIND IS MAT, MATRIX OF KA ROWS AND KB COLUMNS
   23 GO TO 28
C  KIND IS 6X6 MATRIX
   24 NQ = 22 + KADD
   28 IF(.NOT.ROWSW) ROWS = KA
      IF (.NOT.COLSW) COLS=KB
      GO TO 30
C  KIND IS R7, MAP, OR PRD (7X7 MATRIX REPRESENTATION)
   25 NQ = 49 + KADD
      ROWS = 7
      COLS = 7
      GO TO 30
C  KIND IS ROT,ROTATION
   26 NQ = KADD
      KIND = NKIND(NK)
C SET LFMX FOR NSETS=0 CASE. IF NSETS NOT 0, LFMX COMPUTED BELOW.
   30 LFMX=LQ+NQ-1
      IF(NSETS.EQ.0) GO TO 60
      IF (I(NSETS).GE.0) GO TO 31
C NUMBER OF DATA IN LAST SET IS INDEFINITE.
      INDEF=.TRUE.
      I(NSETS) = -I(NSETS)
   31 IF(I(1).NE.0) GO TO 32
C STORE VARIABLE-INPUT-LENGTH NUMBERS IN NVAR(I).
      DECODE (50,100,ICARD(3)) (NVAR(K),K=1,N1)
  100 FORMAT (10I5)
C CALCULATE NUMBER OF DATA OF EACH TYPE.
   32 DO  44  J=1,NSETS
      NAB = IABS(ND(J))
      IF (ND(J).LT.0) ND(J) = NVAR(NAB)
      IJ = I(J) + 1
      GO TO (44,41,42,43,44),IJ
   41 NFL=NFL + ND(J)
      GO TO 44
   42 NBC=NBC + ND(J)
      GO TO 44
   43 NIN=NIN + ND(J)
   44 CONTINUE
C  IF A CYX INSTRUCTION, SET UP LQ2
      CYSW =.FALSE.
      IF ((OPNAME.EQ.3HCYB).OR.(OPNAME.EQ.3HCYC)) CYSW = .TRUE.
      IF ((OPNAME.EQ.3HCYA).OR.(OPNAME.EQ.4HBEST)) CYSW=.TRUE.
      IF (.NOT.CYSW) GO TO 45
      LQ2 = LQ + NQ
      NQ = NQ + 10
   45 CONTINUE
      IF (OPNAME.NE.3HTAB) GO TO 46
C  SETS UP EXTRA LQ FOR TABLE USE
      LQ2 = LQ + NQ
      LQ3 = LQ2 + 10
      NQ = NQ + 20
   46 CONTINUE
C CALCULATE FILE LOCATIONS.
C  ORDER SETS WITH FLOATING POINT AFTER LQ
```

```
      NIWS = NIN
      NBWS = NBC
   53 LFL = LQ + NQ
      LBC = LFL + NFL
      LIN = LBC + NBWS
      LFMX = LIN + NIWS - 1
   60 NTOT = LFMX - LFILE
      IF (IQ.GE.0) GO TO 65
C  RESET NQ. RESET LQ ACCORDING TO TYPE OF DATA IN IQ-TH SET
C IQ REFERS TO THE ABS IQ-TH SET OF INPUT DATA
      IQ=-IQ
      NQ=ND(IQ)
      IN=I(IQ)
      GO TO (61,62,63),IN
   61 LQ=LFL
      NQ = NFL
      GO TO 65
   62 LQ=LBC
      GO TO 65
   63 LQ=LIN
C PACK  LOCATIONS AND LENGTHS INTO INFO(I,N).
   65 IF(NIN.EQ.0) LIN=0
      IF(NFL.EQ.0) LFL=0
      IF(NBC.EQ.0) LBC=0
      DO 70  J=1,9
   70 INF(J) = 1H*
      INF(10) = LQ
      INF(11) = NQ
      INF(12) = LIN
      INF(13) = NIN
      INF(14) = LFL
      INF(15) = NFL
      INF(16) = LBC
      INF(17) = NBC
      INF(18) = NTOT
      INF(19)=LQ2
      INF(20) = KIND
      INF(21)=TYPE
      INF(22) = ROWS
      INF(23) = COLS
      INF(24) = LQ3
      IF (M0.GT.0)  CALL WRTINF(INF,M0)
C RESET BOTTOM OF  FILE,LFILE,TO BOTTOM  OF MTH SECTION OF THE FILE.
      LFILE= LFMX
C     CHECK FOR STORAGE OVERFLOW
      NSTO = LMAX - LFILE
      NINFF = MAX - MIN
      IF (LFILE.LE.LMAX)  GO TO 80
      WRITE (3,201)
      WRITE (3,202) LMAX,LFILE,NSTO,MAX,NINFF,M0
      ERROR = .TRUE.
   80 IF (M0.GT.0)  RETURN
      WRITE (3,200)
      IF (.NOT.ERROR)  WRITE (3,202) LMAX,LFILE,NSTO,MAX,NINFF,M0
      ERROR = .TRUE.
  200 FORMAT (6H0*****,*ELEMENTS OVERFLOWED. INCREASE INFF ARRAY AND *
     1  *MAX.*)
```

```
      201  FORMAT (6H0*****,*STORE OVERFLOWED. INCREASE STORE ARRAY AND LMAX* RESRV    215
         1  )                                                                 RESRV    216
      202  FORMAT(/1X*CORE USE SUMMARY*,27X,*MAXIMUM*,17X,*USED*,8X,*UNUSED*/ RESRV    217
         1 34X,*STORE   (ELEMENT STORAGE)*,14X,I10,1X,*(LMAX)*,4X,I10,5X,I10/ RESRV    218
         2 34X,*INFF    (ELEMENT DEFINTIONS)*,10X,I10,1X,*(MAX)*,5X,I10,5X,   RESRV    219
         3  I10)                                                              RESRV    220
           RETURN                                                             RESRV    221
           END                                                                RESRV    222
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
185	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

```
                                                                          REVMAT     2
          SUBROUTINE REVMAT(N,M,V)                                        REVMAT     3
                                                                          REVMAT     4
C MAPS PARTICLE VECTOR V BY ELEMENT WITH INDEX M.                          REVMAT     5
C ELEMENT MAY BE ANY LINEAR ELEMENT OR SEXTUPOLE.                         REVMAT     6
C IF THE ELEMENT IS NOT COMPOSITE (E.G. AN MMM), LINEARIZED MATRICES      REVMAT     7
C THAT GIVE THE TRANSFORMATION RELATIVE TO V ARE CALCULATED AND           REVMAT     8
C STORED AT INDEX N.                                                      REVMAT     9
C IF M IS A COMPOSITE ELEMENT, A NEW MATRIX IS NOT CALCULATED,            REVMAT    10
C THE OLD ONE IS STORED AT N.                                             REVMAT    11
                                                                          REVMAT    12
C ELEMENTS DEFINED BY MAG AND SXTP WILL BE RECALCULATED BY REVMAT,        REVMAT    13
C INCLUDING BR=BR*(1+DP/P).                                               REVMAT    14
                                                                          REVMAT    15
C **NOTE**  THIS ROUTINE USES MEND-3, MEND-4, AND MEND-5 (3X3 LOCS)       REVMAT    16
                                                                          REVMAT    17
                                                                          BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                           BLANK      3
      COMMON STORE(48000),IWORK(10)                                       86MARSIZ   1
      DIMENSION INFF(24,2000)                                             86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                            86MARSIZ   3
                                                                          BLANK      5
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL     DIM        2
                                                                          DIM        3
      COMMON /MN/ ML    $         LEVEL 2, ML                             REVMATL    1
      COMMON/CORB/TH(84),TV(84),NMON,NNH,NNV,IIH,IIV,                     CORB613    1
     .             NAMONH,NAMONV,NACORH,NACORV                            CORB613    2
      DIMENSION PAR(6),T(7,7),RW(3),ML(3),V(1),V0(7),W(7),U(7)            REVMAT    20
      EQUIVALENCE (ANG1,IANG1),(ANG2,IANG2),(BR,PAR(3))                   REVMAT    21
      DATA RADEG/.0174532925/                                             REVMAT    22
                                                                          REVMAT    23
      NOP = INFF(1,M)                                                     REVMAT    24
                                                                          REVMAT    25
C MOVE                                                                    REVMAT    26
      IF (NOP.NE.4HMOVE) GO TO 100                                        REVMAT    27
      CALL MOVE(M,N,V)                                                    REVMAT    28
      RETURN                                                              REVMAT    29
                                                                          REVMAT    30
  100 DO 2 I=1,7                                                          REVMAT    31
      U(I)=V(I)                                                           REVMAT    32
    2 V0(I)=V(I)                                                          REVMAT    33
                                                                          REVMAT    34
C SEXTUPOLE                                                               REVMAT    35
      IF (NOP.NE.4HSXTP) GO TO 1                                          REVMAT    36
      CALL DATA(M,1,1,4,PAR)                                              REVMAT    37
      INFF(4,N)=INFF(4,M)                                                 REVMLTP    1
      INFF(5,N)=INFF(5,M)                                                 REVMLTP    2
      CALL SXTP(N,V,PAR)                                                  REVMAT    38
      RETURN                                                              REVMAT    39
                                                                          REVMAT    40
C MAGNET DEFINED BY MAG INSTRUCTION                                       REVMAT    41
    1 IF (NOP.NE.5HMAG  ) GO TO 7                                         REVMAT    42
      CALL DATA(M,1,1,6,PAR)                                              REVMAT    43
      GRAD=PAR(2)                                                         REVMAT    44
      BZ=PAR(4)                                                           REVMAT    45
      IRECT=0                                                             REVMAT    46
      ANG1=PAR(5)                                                         REVMAT    47
```

```
      ANG2=PAR(6)                                                          REVMAT    48
      IF ( (IANG1.EQ.1H$).OR.(IANG2.EQ.1H$) ) IRECT=1                      REVMAT    49
      IF ( (GRAD.NE.0.).OR.(IRECT.EQ.0).OR.(BZ.EQ.0.) ) GO TO 11           REVMAT    50
                                                                           REVMAT    51
C  RECTANGULAR,ZERO-GRADIENT BENDING MAGNET                                REVMAT    52
      CALL BEND(N,V,PAR,2)                                                 REVMAT    53
      GO TO 9                                                              REVMAT    54
   11 CONTINUE                                                             REVMAT    55
      IF (BZ.NE.0.)  GO TO 111                                             REVMAT    56
C   QUADRUPOLE                                                             REVMAT    57
      CALL MODQ (M,N,V,W)                                                  REVMAT    58
      GO TO 45                                                             REVMAT    59
  111 BR = BR*(1.+V(6))                                                    REVMAT    60
      CALL MAGNET(N,PAR)                                                   REVMAT    61
      CALL RTRV7(N,T,RW)                                                   REVMAT    62
      CALL MXV7(T,V,W)                                                     REVMAT    63
                                                                           REVMAT    64
C  BENDING MAGNET                                                          REVMAT    65
      THETA = RW(2)                                                        REVMAT    66
      IF (IRECT.EQ.1) GO TO 3                                              REVMAT    67
      E1=PAR(5)*RADEG                                                      REVMAT    68
      E2=PAR(6)*RADEG                                                      REVMAT    69
      GO TO 4                                                              REVMAT    70
    3 E1=.5*THETA                                                          REVMAT    71
      E2=E1                                                                REVMAT    72
    4 E1T=E1+V(2)                                                          REVMAT    73
      E2T=E2-W(2)                                                          REVMAT    74
      X1=V(1)                                                              REVMAT    75
      X2=W(1)                                                              REVMAT    76
      IF (BZ.GE.0.) GO TO 40                                               REVMAT    77
       E1T=E1-V(2)                                                         REVMAT    78
       E2T=E2+W(2)                                                         REVMAT    79
       X1=-X1                                                              REVMAT    80
       X2=-X2                                                              REVMAT    81
   40 PAR(5)=E1T/RADEG                                                     REVMAT    82
      PAR(6)=E2T/RADEG                                                     REVMAT    83
      D1=X1*SIN(E1)/COS(E1T)                                               REVMAT    84
      D2=X2*SIN(E2)/COS(E2T)                                               REVMAT    85
      DL=V(5)-W(5)                                                         REVMAT    86
      PAR(1) = PAR(1) + DL - D1 - D2                                       REVMAT    87
      THETA = THETA + V(2) - W(2)                                          REVMAT    88
      RHO = PAR(1)/THETA                                                   REVMAT    89
      PAR(4) = BR/RHO                                                      REVMAT    90
      DO 5 I=1,3                                                           REVMAT    91
    5 ML(I) = MEND - 2 - I                                                 REVMAT    92
      CALL DRIFT(MEND-3,D1)                                                REVMATL    2
      CALL MAGNET(MEND-4,PAR)   $  CALL DRIFT(MEND-5,D2)                   REVMATL    3
                                                                           REVMATL    4
      CALL MMM(N,3,ML)                                                     REVMAT    96
   45 DO 50 I=1,5                                                          REVMAT    97
   50 V(I)=W(I)                                                            REVMAT    98
      GO TO 9                                                              REVMAT    99
                                                                           REVMAT   100
                                                                           REVMAT   101
C  DRIFT                                                                   REVMAT   102
    7 IF(NOP.NE.3HDRF) GO TO 8                                             REVMAT   103
      EL = FDAT(M,1)                                                       REVMAT   104
```

```
           EL=EL*SQRT(1.+V(2)*V(2))                                 REVMAT      105
           CALL DRIFT(N,EL)                                         REVMAT      106
           GO TO 10                                                 REVMAT      107
                                                                    REVMAT      108
C  KICK                                                             RVMT683       2
   8       IF (NOP.NE.4HKICK)  GO TO 82                             RVMT683       3
           NAKICK = INFF(2,M)                                       RVMT683       4
           IF (NAKICK.NE.NACORH) GO TO 81                           REVMAT613     1
           TKH = TH(IIH)                                            RVMT683       6
           CALL STDAT(M,1,3,1,TKH)                                  RVMT683       7
           CALL KICK(M)                                             RVMT683       8
           IIH = IIH + 1                                            RVMT683       9
           IF (IIH.GT.NNH) IIH=1                                    RVMT683      10
           GO TO 10                                                 RVMT683      11
   81      IF (NAKICK.NE.NACORV) GO TO 10                           REVMAT613     2
           TKV = TV(IIV)                                            REVMAT613     3
           CALL STDAT(M,1,3,1,TKV)                                  REVMAT613     4
           CALL KICK(M)                                             RVMT683      15
           IIV = IIV + 1                                            RVMT683      16
           IF (IIV.GT.NNV) IIV=1                                    RVMT683      17
           GO TO 10                                                 RVMT683      18
C  OTHER LINEAR ELEMENTS---MMM, KICK, ETC.                          REVMAT      109
   82      CONTINUE                                                 RVMT683      19
   10      CALL RTRV7(M,T,RW)                                       REVMAT      111
           CALL MXV7(T,V,V)                                         REVMAT      112
           CALL STOR7(N,T,RW)                                       REVMAT      113
           RETURN                                                   REVMAT      114
                                                                    REVMAT      115
   9       CALL RTRV7(N,T,RW)                                       REVMAT      116
   91      CALL MXV7(T,V0,W)                                        REVMAT      117
           DO 12 I=1,5                                              REVMAT      118
   12      T(I,7)=V(I)-W(I)                                         REVMAT      119
           CALL STOR7(N,T,RW)                                       REVMAT      120
                                                                    REVMAT      121
           RETURN                                                   REVMAT      122
           END                                                      REVMAT      123
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
70	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
83	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
88	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
96	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
98	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
115	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
115	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

```
      FUNCTION RGEN(D)                                      RGEN     2
      RGEN=RANF(D)                                          RGEN     3
      END                                                   RGEN     4
```

```
               SUBROUTINE ROTATE(THETA,T,TR,RW)                             ROTATE    2
                                                                            ROTATE    3
        *         ROTATE MATRIX T BY THETA RADIANS AND FORM NEW MATRIX TR.  ROTATE    4
                                                                            ROTATE    5
              COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL DIM     2
                                                                            DIM       3
              COMMON /MO/ ML(3) $        LEVEL 2, ML                        ROTLL     1
              EQUIVALENCE(ML,MI)                                            ROTLL     2
                                                                            ROTLL     3
              DIMENSION MI(3)                                               ROTATE    7
              DIMENSION T(1),TR(1),RW(1),RW1(3)                             ROTATE    8
              DATA RW1/3*0.0/                                               ROTATE    9
                                                                            ROTATE   10
        *                  COMPUTE ROTATION MATRIX R+ AND R-                ROTATE   11
              THETM = -THETA                                                ROTATE   12
              CALL ROTZ(THETA,TR)                                           ROTATE   13
              M1 = M7END - 1                                                ROTATE   14
              CALL STOR7(M1,TR,RW1)                                         ROTATE   15
              MI(1) = M1                                                    ROTATE   16
              M1 = M1 - 1                                                   ROTATE   17
              CALL STOR7(M1,T,RW)                                           ROTATE   18
              MI(2) = M1                                                    ROTATE   19
              M1 = M1 - 1                                                   ROTATE   20
              CALL ROTZ(THETM,TR)                                           ROTATE   21
              CALL STOR7(M1,TR,RW1)                                         ROTATE   22
              MI(3) = M1                                                    ROTATE   23
              M1 = M7END                                                    ROTATE   24
              CALL MMM(M1,3,MI)                                             ROTATE   25
              CALL RTRV7(M1,TR,RW)                                          ROTATE   26
              RETURN                                                        ROTATE   27
              END                                                           ROTATE   28
```

```
CARD NR. SEVERITY  DETAILS   DIAGNOSIS OF PROBLEM

EQV/COMM    I       MO       NOT ALL ITEMS IN THIS COMMON BLOCK OCCUR IN LEVEL STATEMENTS.
```

```
      SUBROUTINE ROTM(M,MB,THETA)

*  ROTATE MATRIX MB BY THETA RADIANS AND STORE IN MATRIX M.
* MB AND M ARE INDECES OF MATRICES

      DIMENSION T(49),TR(49),RW(3)

      CALL RTRV7(MB,T,RW)
      CALL ROTATE (THETA,T,TR,RW)
      CALL STOR7(M,TR,RW)
       RETURN
      END
```

```
      SUBROUTINE ROTZ(THET,T)
*    DEFINE A MATRIX REPRESENTING A ROTATION ABOUT THE Z-AXIS
*     THET IS IN RADIANS

      DIMENSION T(7,7)
      DO 10 I=1,7
      DO 10 J=1,7
      T(I,J) = 0.
   10 CONTINUE
      SN = SIN(THET)
      CS = COS(THET)
      T(1,1) = CS
      T(2,2) = CS
      T(3,3) = CS
      T(4,4) = CS
      T(1,3) = SN
      T(2,4) = SN
      T(3,1) = -SN
      T(4,2) = - SN
      T(5,5) = 1.
      T(6,6) = 1.
      T(7,7) = 1.
      RETURN
      END
```

```
1                 SUBROUTINE ROW5(T)                                          ROW5      2
            C   COMPUTES ROW 5 OF 7X7 MATRIX T                               ROW5      3
                                                                              ROW5      4
                  DIMENSION T(7,1)                                            ROW5      5
5                                                                             ROW5      6
                  DO 10 I=1,4                                                 ROW5      7
             10   T(5,I)=T(2,I)*T(1,6)-T(1,I)*T(2,6)+T(4,I)*T(3,6)-T(3,I)*T(4,6)  ROW5      8
                  RETURN                                                      ROW5      9
                  END                                                         ROW5     10
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
7	I	T	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
7	I	T	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
7	I	T	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
7	I	T	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

```
 1          SUBROUTINE RSVMAT                                                      RSVMAT    2
                                                                                   RSVMAT    3
      *       RESERVES SPACE FOR KA MATRICES, BEGINNING WITH INDEX M - 1           RSVMAT    4
            COMMON/INSTR/OPNAME,NAME,OP,KA,KB,OBJA,OBJB,NXTM,TRA,LQ,NQ,LIN,         INSTR     2
 5         1             NIN,LFL,NFL,LBC,NBC,NTOT,IX,KIND,TYPE,ROWS,COLS,EXTR,      INSTR     3
           2             M,NF,NB,NI,MSUBR,ISIGN                                    INSTR     4
            INTEGER OPNAME,OP,OBJA,OBJB,TRA,TYPE,ROWS,COLS,EXTR                     INSTR     5
                                                                                   INSTR     6
            COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,          CONTRL    2
10         1           LDFLG,FIN                                                   CONTRL    3
            LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN            CONTRL    4
                                                                                   CONTRL    5
            DIMENSION LIST(25)                                                     RSVMAT    7
                                                                                   RSVMAT    8
15          IG = 0                                                                 RSVMAT    9
            IF (OPNAME.EQ.3HDRF) IG = 1                                            RSVMAT   10
            IF (OPNAME.EQ.3HMAG) IG = 2                                            RSVMAT   11
            IF (IG.LT.1) GO TO 25                                                  RSVMAT   12
      *                   GENERATE NUMBERED NAMES                                  RSVMAT   13
20          CALL GENNUM(NAME,KA,LIST)                                              RSVMAT   14
      *           RESERVE ACCORDING TO OPNAME                                      RSVMAT   15
            M1 = M - 1                                                             RSVMAT   16
            K1 = 0                                                                 RSVMAT   17
            K2 = 0                                                                 RSVMAT   18
25          DO 20 I=1,KA                                                           RSVMAT   19
            GO TO (10,15) IG                                                       RSVMAT   20
         10 CALL DRFRSV(M1,LIST(I),K1,K2,0)                                        RSVMAT   21
            GO TO 18                                                               RSVMAT   22
         15 CALL MAGRSV(M1,LIST(I),K1,K2,0)                                        RSVMAT   23
30       18 M1 = M1 - 1                                                            RSVMAT   24
         20 CONTINUE                                                               RSVMAT   25
            RETURN                                                                 RSVMAT   26
                                                                                   RSVMAT   27
         25 ERROR = .TRUE.                                                         RSVMAT   28
35          WRITE (3,1) OPNAME                                                     RSVMAT   29
          1 FORMAT (5X,3H***,1X,*ERROR*,1X,A5,1X,*IS NOT A DRF OR MAG.*/           RSVMAT   30
           1   9X,*ROUTINE RSVMAT CANNOT RESERVE MATRIX SPACE FOR IT.*)            RSVMAT   31
            RETURN                                                                 RSVMAT   32
            END                                                                    RSVMAT   33
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
26	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

```
            SUBROUTINE RTRV7(M,TT,RW)                                       RTRV7      2
      C  RETRIEVES 7X7 MATRIX ELEMENTS                                      RTRV7      3
      C  IF ELEMENT NUMBER IS -, RETURN INVERSE                             RTRV7      4
                                                                            BLANK      2
            LEVEL 2, STORE,INFF,IWORK                                       BLANK      3
            COMMON STORE(48000),IWORK(10)                                   86MARSIZ   1
            DIMENSION INFF(24,2000)                                         86MARSIZ   2
            EQUIVALENCE (INFF,STORE)                                        86MARSIZ   3
                                                                            BLANK      5
            COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,  CONTRL     2
           1          LDFLG,FIN                                             CONTRL     3
            LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN    CONTRL     4
                                                                            CONTRL     5
            DIMENSION TT(1),TM(49),RW(3),RX(2,3),RY(2,3),T(7,7),RPQ(4)      RTRV7      7
            COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL DIM        2
                                                                            DIM        3
            EQUIVALENCE (T,TM)                                              RTRV7      9
            LOGICAL INV                                                     RTRV7     10
            DATA RADEG/.0174532925/                                         RTRV7     11
      C                                                                     RTRV7     12
            INV = .FALSE.                                                   RTRV7     13
            DO 4 I=1,3                                                      RTRV7     14
          4 RW(I) = 0.                                                      RTRV7     15
            IF (M)5,6,6                                                     RTRV7     16
          5 M = - M                                                         RTRV7     17
            INV = .TRUE.                                                    RTRV7     18
          6 KIND = INFF(20,M)                                               RTRV7     19
            CALL DATA(M,5,1,3,RW)                                           RTRV7     20
      C  IS KIND R7 OR ROT                                                  RTRV7     21
            IF (KIND.EQ.3HROT) GO TO 40                                     RTRV7     22
            IF (KIND.NE.3HMXY) GO TO 28                                     RTRV7     23
      C  KIND IS MXY.  RETURN 2X3 MATRIX                                    RTRV7     24
            CALL RXY(M,RX,RY,RW)                                            RTRV7     25
            CALL DATA(M,5,4,4,RPQ)                                          RTRV7     26
            KD = 7                                                          RTRV7     27
            CALL CONVMAT(KD,T,RX,RY,RW,RPQ)                                 RTRV7     28
            GO TO 50                                                        RTRV7     29
      C  KIND IS R7                                                         RTRV7     30
         28 LQ = INFF(10,M)                                                 RTRV7     31
            NQ = INFF(11,M)                                                 RTRV7     32
      C  CHECK STORAGE                                                      RTRV7     33
            NNQ = 49 + KADD                                                 RTRV7     34
            IF (NNQ.GT.NQ) GO TO 45                                         RTRV7     35
            NJ = LQ + KADD - 1                                              RTRV7     36
            DO35 I =1 , 49                                                  RTRV7     37
            TM(I) = STORE(NJ + I)                                           RTRV7     38
         35 CONTINUE                                                        RTRV7     39
            GO TO 50                                                        RTRV7     40
         45 MN = INFF(2,M)                                                  RTRV7     41
            WRITE (3,1) MN                                                  RTRV7     42
          1 FORMAT (61H *** ERROR *** RETRIEVAL REQUEST EXCEEDS STORAGE RESERV RTRV7   43
           1E FOR A6)                                                       RTRV7     44
            ERROR = .TRUE.                                                  RTRV7     45
            RETURN                                                          RTRV7     46
      C  KIND IS ROT                                                        RTRV7     47
         40 CALL DATA (M,1,1,1,THETA)                                       RTRV7     48
            THETA = THETA*RADEG                                             RTRV7     49
```

```
                     DO 41 I=1,49                                        RTRV7       50
                     TM(I) = 0.                                          RTRV7       51
  60           41    CONTINUE                                            RTRV7       52
                     CALL ROTZ(THETA,T)                                  RTRV7       53
               50    IF (.NOT.INV) GO TO 52                              RTRV7       54
                     CALL TINV7(TT,TM)                                   RTRV7       55
                     RW(1) = - RW(1)                                     RTRV7       56
  65                 RW(2) = -RW(2)                                      RTRV7       57
                     GO TO 56                                            RTRV7       58
               52    DO 55 I=1,49                                        RTRV7       59
                     TT(I) = TM(I)                                       RTRV7       60
               55    CONTINUE                                            RTRV7       61
  70           56    RW(3) = TT(40)                                      RTRV7       62
                     RETURN                                              RTRV7       63
                     END                                                 RTRV7       64
```

```
CARD NR. SEVERITY  DETAILS   DIAGNOSIS OF PROBLEM

  70       I       TT        ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
```

```
      SUBROUTINE R7TOU(R7,U)                                            R7TOU      2
      RETURN                                                            R7TOU      3
      END                                                               R7TOU      4
```

```
      SUBROUTINE RXRY(M,DAT)                                            RXRY       2
C  READS MATRIX RX AND RY,DIMENSIONED (2,3) FROM CARDS.                 RXRY       3
C   STORES RX AND RY IN LQ(M)                                           RXRY       4
      DIMENSION RX(2,3),RY(2,3),DAT(12),RW(3)                           RXRY       5
      DATA (RW(I),I=1,3)/3*0.0/                                         RXRY       6
C  SETS UP MATRICES WHOSE ELEMENTS ARE READ FROM CARDS                  RXRY       7
      IX=1                                                              RXRY       8
      IY=7                                                              RXRY       9
      DO 1 I=1,2                                                        RXRY      10
      DO 1 J=1,3                                                        RXRY      11
      RX(I,J)=DAT(IX)                                                   RXRY      12
      RY(I,J)=DAT(IY)                                                   RXRY      13
      IX=IX+1                                                           RXRY      14
      IY=IY+1                                                           RXRY      15
    1 CONTINUE                                                          RXRY      16
      CALL STXY(M,RX,RY,RW)                                             RXRY      17
      RETURN                                                            RXRY      18
      END                                                               RXRY      19
```

```
      SUBROUTINE RXY(M,RX,RY,RW)                                        RXY        2
C                                                                       RXY        3
C  RETRIEVES X AND Y MATRICES FROM LQ(M). DIMENSIONS ARE (2,3) FOR EACH. RXY       4
C  RETURN RX AND RY                                                     RXY        5
C  IF ELEMENT NUMBER IS -, RETURN INVERSE                               RXY        6
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1          LDFLG,FIN                                               CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER   SWTCH      2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                          SWTCH      3
                                                                        SWTCH      4
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL   DIM        2
                                                                        DIM        3
      LOGICAL INV                                                       RXY       11
      DIMENSION RX(2,3),RY(2,3),AX(2,3),AY(2,3),RW(3)                   RXY       12
      DIMENSION TT(49),T(7,7),RPQ(4)                                    RXY       13
      EQUIVALENCE (T,TT)                                                RXY       14
C                                                                       RXY       15
      INV = .FALSE.                                                     RXY       16
      IF (M.GE.0) GO TO 5                                               RXY       17
    4 M = - M                                                           RXY       18
      INV = .TRUE.                                                      RXY       19
    5 LQ = INFF(10,M)                                                   RXY       20
      NQ = INFF(11,M)                                                   RXY       21
      DO 7 J=1,3                                                        RXY       22
      DO 3 I=1,2                                                        RXY       23
      RX(I,J) = 0.                                                      RXY       24
      RY(I,J) = 0.                                                      RXY       25
    3 CONTINUE                                                          RXY       26
      RW(J) = 0.                                                        RXY       27
    7 CONTINUE                                                          RXY       28
      KIND = INFF(20,M)                                                 RXY       29
      IF (KIND.EQ.3HROT) GO TO 12                                       RXY       30
      IF(KIND.EQ.3HMXY) GO TO 6                                         RXY       31
C  KIND IS R7.  RETRIEVE AND CONVERT                                    RXY       32
      NN = M                                                            RXY       33
      IF (INV) NN = -NN                                                 RXY       34
      CALL RTRV7(NN,T,RW)                                               RXY       35
      KD = 3                                                            RXY       36
      CALL CONVMAT(KD,T,RX,RY,RW,RPQ)                                   RXY       37
      RETURN                                                            RXY       38
C  KIND IS ROT                                                          RXY       39
   12 THETA = FDAT(M,1)                                                 RXY       40
      CS = COS(THETA)                                                   RXY       41
      DO 14 I=1,2                                                       RXY       42
      RX(I,I) = CS                                                      RXY       43
      RY(I,I)= CS                                                       RXY       44
   14 CONTINUE                                                          RXY       45
      GO TO 24                                                          RXY       46
C  CHECK STORAGE                                                        RXY       47
```

```
    6 CONTINUE
      NNQ = 12 + KADDR
      IF (NNQ.GT.NQ) GO TO 20
      NJ = LQ + KADDR
      NI = NJ + 6
      DO 10 IJ=1,3
      DO 10 II=1,2
      RX(II,IJ)=STORE(NJ)
      RY(II,IJ)=STORE(NI)
      NJ=NJ+1
      NI=NI+1
   10 CONTINUE
      CALL DATA(M,5,1,3,RW)
   24 IF (.NOT.INV) RETURN
      INV = .FALSE.
      DO 25 J=1,3
      DO 25 I=1,2
      AX(I,J) = RX(I,J)
   25 AY(I,J) = RY(I,J)
      RX(1,1) = AX(2,2)
      RX(2,2) = AX(1,1)
      RX(1,2) = -AX(1,2)
      RX(2,1) = -AX(2,1)
      RX(1,3) = AX(1,2)*AX(2,3) - AX(2,2)*AX(1,3)
      RX(2,3) = AX(2,1)*AX(1,3) -  AX(1,1)*AX(2,3)
      RY(1,1) = AY(2,2)
      RY(2,2) = AY(1,1)
      RY(1,2) = -AY(1,2)
      RY(2,1) =-AY(2,1)
      RY(1,3) = AY(1,2)*AY(2,3) - AY(2,2)*AY(1,3)
      RY(2,3) = AY(2,1)*AY(1,3) - AY(1,1)*AY(2,3)
      RW(1) = -RW(1)
      RW(2) = -RW(2)
      IF (BEND.NE.0.) RW(3)= - RW(3)
      RETURN
   20 WRITE (3,1) M
    1 FORMAT (66H *** ERROR *** RETRIEVAL REQUEST EXCEEDS STORAGE RESERV
     1E AT INDEX I5)
      ERROR=.TRUE.
      RETURN
      END
```

```
      SUBROUTINE SCOPCN                                          SCOPCN     2
      RETURN                                                     SCOPCN     3
      END                                                        SCOPCN     4
```

```
      SUBROUTINE SETALPH (M)                                      SETALPH      2
                                                                  BLANK        2
      LEVEL 2, STORE,INFF,IWORK                                   BLANK        3
      COMMON STORE(48000),IWORK(10)                               86MARSIZ     1
      DIMENSION INFF(24,2000)                                     86MARSIZ     2
      EQUIVALENCE (INFF,STORE)                                    86MARSIZ     3
                                                                  BLANK        5
      COMMON/CDERIV/IFLAG,MR,NR,DP,NV(11),DAT(20),UZ,STOT,        CDERIV       2
     1  DAT2(7),MP2FLG,NF,H0,LOCALPH,NUMALPH                      CDERIV       3
C                                                                 SETALPH      5
      MALPH = MDAT(M,1)                                           SETALPH      6
      LOCALPH = INFF(10,MALPH) - 1                                SETALPH      7
      NUMALPH = INFF(11,MALPH)                                    SETALPH      8
      H0 = DAT(5)                                                 SETALPH      9
      RETURN                                                      SETALPH     10
      END                                                         SETALPH     11
```

```
      SUBROUTINE SETCYB(M,KA)                                         SETCYB     2
                                                                      SETCYB     3
                                                                      SETCYB     4
                                                                      SETCYB     5
*        SETS UP MILIST AND MATRIX FOR CYB INSTRUCTION                SETCYB     6
                                                                      SETCYB     7
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,  CONTRL     2
     1           LDFLG,FIN                                            CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN    CONTRL     4
                                                                      CONTRL     5
      COMMON/BMI/MI(5000)                                             BMIL       1
      COMMON/NELS/NELS                                                BMIL       2
      LEVEL 2,MI                                                      BMIL       3
                                                                      BMIL       4
                                                                      BMI        3
                                                                      SETCYB    10
      DIMENSION RX(2,3),RY(2,3),RW(3),T(49)                           SETCYB    11
                                                                      BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                       BLANK      3
      COMMON STORE(48000),IWORK(10)                                   86MARSIZ   1
      DIMENSION INFF(24,2000)                                         86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                        86MARSIZ   3
                                                                      BLANK      5
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL DIM        2
                                                                      DIM        3
                                                                      SETCYB    14
      KN = INFF(20,M)                                                 SETCYB    15
      NT1 = MEND - 1                                                  SETCYB    16
*        GET M OF INPUT ELEMENT AND ITS SYNCH OPNAME                  SETCYB    17
      MPR = MDAT(M,1)                                                 SETCYB    18
      NB = INFF(1,MPR)                                                SETCYB    19
      KNB = INFF(4,MPR)                                               SETCYB    20
                                                                      SETCYB    21
      ASSIGN 10 TO NGO                                                SETCYB    22
      GO TO 55                                                        SETCYB    23
   10 CONTINUE                                                        SETCYB    24
      GO TO (11,15,20) ITP                                            SETCYB    25
                                                                      SETCYB    26
*   IF IT IS A CELL TYPE, MATRIX IS IN STORAGE AND MI LIST IS AT LIN  SETCYB    27
   11 IF (KA.GT.12) GO TO 65                                          SETCYB    28
      CALL DATA (MPR,3,1,12,MI)                                       SETCYB    29
      GO TO 30                                                        SETCYB    30
                                                                      SETCYB    31
*        MAKE MI LIST FROM BEAM LINE.                                 SETCYB    32
   15 CONTINUE                                                        SETCYB    33
      NDAT=INFF(17,MPR)                                               SETCYB    34
      CALL MIFILL(MPR,1,NDAT,KNB,MI)                                  SETCYB    35
      IF(KA.EQ.0) KA=KNB                                              SETCYB    36
      INFF(4,M)=KA                                                    SETCYB    37
      IF (KA.GT.KNB) GO TO 65                                         SETCYB    38
      GO TO (21,30) IGO                                               SETCYB    39
*                    FORM MATRIX FROM BEAM LINE                       SETCYB    40
   21 CALL MMM(NT1,KNB,MI)                                            SETCYB    41
      RETURN                                                          SETCYB    42
                                                                      SETCYB    43
   20 CONTINUE                                                        SETCYB    44
C  ** SYNCH INSTR.  GET M, DETERMINE TYPE AND FORM MI LIST.           SETCYB    45
```

```
      MM = MDAT(MPR,1)
      NB = INFF(1,MM)
      ASSIGN 25 TO NGO
      GO TO 55
   25 CONTINUE
      IF (ITP.GT.2) GO TO 61
      GO TO (26,27) ITP
   26 IF (KA.GT.12) GO TO 65
      CALL DATA (MM,3,1,12,MI)
      GO TO 30
   27 CONTINUE
      KNB = INFF(4,MM)
      IF (KA.GT.KNB) GO TO 65
      CALL MLIST(MM,1,KNB,MI)
   30 CONTINUE
*  IF KN IS 5, IT IS AN R7 MATRIX.
      IF (KN.EQ.5) GO TO 50

*  TRANSFER MATRIX FROM INDEX MPR TO WORKING STORAGE WITH INDEX MEND-1.
      CALL RXY(MPR,RX,RY,RW)
      CALL STXY (NT1,RX,RY,RW)
      RETURN

*   7 X 7 MATRIX
   50 NT1 = M7END - 1
      CALL RTRV7(MPR,T,RW)
      CALL STOR7 (NT1,T,RW)
      RETURN

***********************************************************************

*        ANALYZE TYPE OF ELEMENT TO BE USED
   55 IF ((NB.EQ.4HCELL).OR.(NB.EQ.4HALTC)) GO TO 51
      IF ((NB.EQ.4HCFD ).OR.(NB.EQ.4HDCFD)) GO TO 51
      IF (NB.NE.4HCELC) GO TO 52
   51 ITP = 1
      GO TO NGO,(10,25)
   52 IGO = 2
      IF ((NB.EQ.4HCYA ).OR.(NB.EQ.4HCYC )) GO TO 53
      IF (NB.EQ.4HMMM ) GO TO 53
      IF (NB.NE.4HBML ) GO TO 54
      IGO = 1
   53 ITP = 2
      GO TO NGO,(10,25)
   54 IF (NB.NE.4H**  ) GO TO 60
      ITP = 3
      GO TO NGO (10,25)
***********************************************************************

***********************************************************************
*                    ERROR EXITS
   60 WRITE (3,2) NB
    2 FORMAT (5X,5H***  ,*CYB CANNOT USE A *,A5,* SYNCH INSTRUCTION.*)
      GO TO 70
   61 WRITE (3,3)
    3 FORMAT (5X,5H***  ,* CANNOT GET ELEMENTS FOR CYB.  STAR INSTR.
     1 REFERS TO A BML OR  ANOTHER STAR INSTR.*)
```

```
      GO TO 70
   65 WRITE (3,1) KA,NB,KNB
    1 FORMAT (5X,5H***  ,*ERROR*,1H*, *CYB WANTS *,I5,* ELEMENTS FOR
     1CYCLING.*,A5,* HAS ONLY *,I5)
   70 ERROR = .TRUE.
      RETURN
      END
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
37	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
51	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
64	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

```
      SUBROUTINE SETOPS                                          SETOPS     2
                                                                 SETOPS     3
*    SETS UP LIST OF SYNCH INSTRUCTIONS.                         SETOPS     4
*           OPNUM ROUTINE WILL USE INDEX TO SET OP NUMBERS       SETOPS     5
                                                                 SETOPS     6
      COMMON/OPLIST/LIST(200)                                    OPLIST     2
                                                                 OPLIST     3
      DATA LIST/200(1H )/                                        SETOPS     8
                                                                 SETOPS     9
      LIST(1) = 4HSTOP                                           SETOPS    10
      LIST(2) = 4HFIN                                            SETOPS    11
      LIST(3) = 4HRUN                                            SETOPS    12
      LIST(4) = 4HSUB                                            SETOPS    13
      LIST(5) = 4HEND                                            SETOPS    14
      LIST(6) = 4HCALL                                           SETOPS    15
      LIST(7) = 4HINCR                                           SETOPS    16
      LIST(8) = 4HREPL                                           SETOPS    17
      LIST(9) = 4HMESH                                           SETOPS    18
      LIST(10) = 4HVPAR                                          SETOPS    19
      LIST(11) = 4H=                                             SETOPS    20
      LIST(12) = 4HCRD                                           SETOPS    21
      LIST(13) = 4HBML                                           SETOPS    22
      LIST(14) = 4HDRF                                           SETOPS    23
      LIST(15) = 4HMAG                                           SETOPS    24
      LIST(16) = 4HMAGV                                          SETOPS    25
      LIST(17) = 4HEQU                                           SETOPS    26
      LIST(18) = 4HINV                                           SETOPS    27
      LIST(19) = 4HSHF                                           SETOPS    28
      LIST(20) = 4HREF                                           SETOPS    29
      LIST(21) = 4H**                                            SETOPS    30
      LIST(22) = 4HMMM                                           SETOPS    31
      LIST(23) = 4HCYA                                           SETOPS    32
      LIST(24) = 4HCYB                                           SETOPS    33
      LIST(25) = 4HCYC                                           SETOPS    34
      LIST(26) = 4HCELL                                          SETOPS    35
      LIST(27) = 4HALTC                                          SETOPS    36
      LIST(28) = 4HTAB                                           SETOPS    37
      LIST (29) = 4HPTAB                                         SETOPS    38
      LIST(30) = 4HFITQ                                          SETOPS    39
       LIST(31) = 4HFITB                                         SETOPS    40
       LIST(32) = 4HFITR                                         SETOPS    41
       LIST(33) = 4HFITV                                         SETOPS    42
      LIST(34) = 4HPAGE                                          SETOPS    43
      LIST(35) = 4HREM                                           SETOPS    44
      LIST(36) = 4HWMA                                           SETOPS    45
      LIST(37) = 4HSIZE                                          SETOPS    46
      LIST(38) = 4HWBE                                           SETOPS    47
       LIST(39) = 4HFIT                                          SETOPS    48
      LIST(40) = 4HCOPY                                          SETOPS    49
      LIST(41) = 4HNCPY                                          SETOPS    50
      LIST(42) = 4HBEST                                          SETOPS    51
      LIST(43) = 3HBEP                                           SETOPS    52
      LIST(44)=4HBETA                                            SETOPS    53
      LIST(45)=4HKICK                                            SETOPS    54
      LIST(46) = 4HRAND                                          SETOPS    55
      LIST(47) = 4HSHF7                                          SETOPS    56
      LIST(48) = 3HSOL                                           SETOPS    57
```

```
      LIST(49) = 3HVAR                                              SETOPS    58
      LIST(50) = 4HPRNT                                             SETOPS    59
                                                                    SETOPS    60
      LIST(51) = 4HSUM                                              SETOPS    61
      LIST(52) = 4HACT                                              SETOPS    62
      LIST(53) = 4HDELE                                             SETOPS    63
      LIST(54) = 4HSCOP                                             SETOPS    64
      LIST(55) = 4HTEST                                             SETOPS    65
      LIST(56) = 4HWFL                                              SETOPS    66
      LIST(57) = 4HMAGS                                             SETOPS    67
      LIST(58) = 4HCELC                                             SETOPS    68
      LIST(59) = 4HCFD                                              SETOPS    69
      LIST(60) = 4HDCFD                                             SETOPS    70
      LIST(61) = 4HSTR2                                             SETOPS    71
      LIST(62) = 4HSTR4                                             SETOPS    72
      LIST(63) = 4HSTRP                                             SETOPS    73
      LIST(64) = 4HSTRN                                             SETOPS    74
                                                                    SETOPS    75
      LIST(65) = 4HFXPT                                             SETOPS    76
      LIST(66) = 4HTRK                                              SETOPS    77
      LIST(67) = 4HPRD                                              SETOPS    78
      LIST(68) = 4HSXTP                                             SETOPS    79
      LIST(69) = 3HMAP                                              SETOPS    80
      LIST(70) = 4HBMIS                                             SETOPS    81
      LIST(71) = 4HEMIS                                             SETOPS    82
      LIST(72) = 4HROTZ                                             SETOPS    83
      LIST(73) = 4HROT                                              SETOPS    84
      LIST(74) = 4HINV2                                             SETOPS    85
      LIST(75) = 4HELQ                                              SETOPS    86
      LIST(78) = 4HMOVE                                             SETOPS    87
      LIST(79) = 4HMOD                                              SETOPS    88
      LIST(80) = 4HCON                                              SETOPS    89
      LIST(81) = 4HVAR                                              SETOPS    90
      LIST(82) = 4HNPOL                                             SETOPS    91
      LIST(83) = 4HORBC                                             STPS683    1
                                                                    SETOPS    92
      LIST(100) = 4HPBML                                            MY3STOP    1
      LIST(101) = 4HMAT                                             SETOPS    93
      LIST(102) = 4HVEC                                             SETOPS    94
      LIST(103) = 4HPVEC                                            SETOPS    95
      LIST(104) = 4HMXV                                             SETOPS    96
      LIST(105) = 4HCPLX                                            SETOPS    97
      LIST(106) = 4HEQIL                                            SETOPS    98
      LIST(107) = 4HEVEC                                            SETOPS    99
      LIST(108) = 4HPRTV                                            SETOPS   100
      LIST(109) = 4HPRV7                                            SETOPS   101
      LIST(110) = 4HCVEC                                            SETOPS   102
      LIST(111) = 4HLIST                                            SETOPS   103
      LIST(112) = 4HSIN                                             SETOPS   104
      LIST(113) = 4HCOS                                             SETOPS   105
      LIST(114) = 4HEXP                                             SETOPS   106
      LIST(115) = 4HALOG                                            SETOPS   107
      LIST(116) = 4HSQ                                              SETOPS   108
      LIST(117) = 4HSQRT                                            SETOPS   109
      LIST(118) = 4HTAN                                             SETOPS   110
      LIST(119) = 4HASIN                                            SETOPS   111
      LIST(120) = 4HACOS                                            SETOPS   112
```

```
                  LIST(121) = 4HATAN                                        SETOPS     113
                  LIST(122) = 4HABS                                         SETOPS     114
                  LIST(123) = 4HCALC                                        SETOPS     115
                                                                            SETOPS     116
                  LIST(151) = 4HBEAM                                        SETOPS     117
                  LIST(152) = 4HDKE                                         SETOPS     118
                  LIST(153) = 4HEDRF                                        SETOPS     119
                  LIST(155) = 4HCYEM                                        SETOPS     120
                  LIST(156) = 4HBVAL                                        SETOPS     121
                  LIST(157) = 4HCYAE                                        SETOPS     122
            C                                                               SETOPS     123
                  LIST(161) = 4HTRKB                                        SETOPS     124
                  LIST(162) = 4HSOLV                                        SETOPS     125
                  LIST(164) = 4HTRKE                                        86SETOPS     1
                  LIST(165) = 4HTRKM                                        86SETOPS     2
                  LIST(166) = 4HIBET                                        86SETOPS     3
                  LIST(163) = 4HSMIN                                        SETOPS     126
                  RETURN                                                    SETOPS     127
                  END                                                       SETOPS     128
```

```
      SUBROUTINE SETFIT                                                 SETFIT     2
                                                                        SETFIT     3
*   SETS UP FIT INPUT DATA FOR FIT ROUTINE.                             SETFIT     4
* INPUT -- NAME    FITX    KA  KB  SUB  C QF  QD  I1  I2  A1  A2        SETFIT     5
                                                                        SETFIT     6
                                                                        SETFIT     7
      DIMENSION MV(2),IV(2),Q(2)                                        SETFIT     8
      COMMON/INSTR/OPNAME,NAME,OP,KA,KB,OBJA,OBJB,NXTM,TRA,LQ,NQ,LIN,   INSTR      2
     1             NIN,LFL,NFL,LBC,NBC,NTOT,IX,KIND,TYPE,ROWS,COLS,EXTR, INSTR     3
     2             M,NF,NB,NI,MSUBR,ISIGN                               INSTR      4
      INTEGER OPNAME,OP,OBJA,OBJB,TRA,TYPE,ROWS,COLS,EXTR               INSTR      5
                                                                        INSTR      6
      COMMON/GRR/IERR,BSW,VSW,RSW,QSW,XSW                               GRR        2
      LOGICAL IERR,BSW,VSW,RSW,QSW,XSW                                  GRR        3
                                                                        GRR        4
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1          LDFLG,FIN                                               CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      COMMON/STORE/LMAX,LINF,LFILE,MAX,MIN,ICARD(11)                    STORE      2
* LMAX AND MAX ARE DIMENSIONS OF STORE AND INFF                         STORE      3
                                                                        STORE      4
      LOGICAL FSW                                                       SETFIT    13
      INTEGER BDAT                                                      SETFIT    14
                                                                        SETFIT    15
      QSW = .FALSE.                                                     SETFIT    16
      BSW = .FALSE.                                                     SETFIT    17
      VSW = .FALSE.                                                     SETFIT    18
      RSW = .FALSE.                                                     SETFIT    19
      FSW = .FALSE.                                                     SETFIT    20
                                                                        SETFIT    21
*                 TURN ON APPROPRIATE SWITCH                            SETFIT    22
      IF (OPNAME.EQ.4HFITB)  BSW = .TRUE.                               SETFIT    23
      IF (OPNAME.EQ.4HFITR)  RSW = .TRUE.                               SETFIT    24
      IF (OPNAME.EQ.4HFITV)  VSW = .TRUE.                               SETFIT    25
      IF (OPNAME.NE.4HFITQ) GO TO 100                                   SETFIT    26
      QSW = .TRUE.                                                      SETFIT    27
                                                                        SETFIT    28
*     IF KA NOT 0, TURN ON FSW TO TRIGGER CALL TO FITE (FOR FITQ ONLY)  SETFIT    29
      IF (KA.NE.0) FSW = .TRUE.                                         SETFIT    30
  100 IF (MODE.EQ.3) GO TO 116                                          SETFIT    31
                                                                        SETFIT    32
      IQ = 4                                                            SETFIT    33
      IF (RSW) IQ = 6                                                   SETFIT    34
  105 CALL RESRV(M,0,0,                                                 SETFIT    35
     .  IQ,2,2,1,2,3,0,0                                                SETFIT    36
     .)                                                                 SETFIT    37
      CALL LOAD(M,                                                      SETFIT    38
     .  4,2,2,3,2,1,0,0                                                 SETFIT    39
     .)                                                                 SETFIT    40
      IF (.NOT.RSW) GO TO 110                                           SETFIT    41
                                                                        SETFIT    42
* IN FITR CASE, KA AND KB EACH CONTAIN 2 ELEMENT NUMBERS, WHICH ARE     SETFIT    43
* STORED IN BCD INPUT TO BE DECODED LATER INTO INTEGERS.                SETFIT    44
      DECODE (10,1,ICARD(2))  K1,K2                                     SETFIT    45
    1 FORMAT (2X,A3,1X,A3,1X)                                           SETFIT    46
      CALL REPBCD(M,5,K1)                                               SETFIT    47
```

```
      CALL REPBCD(M,6,K2)

  110 IF (MODE.EQ.2) RETURN
*                    XSW SUPPRESSES EXECUTION
  115 IF (XSW) RETURN

  116 CONTINUE
      CALL DATA (M,2,3,2,MV)
      CALL DATA(M,3,1,2,IV)
      CALL DATA(M,1,1,2,Q)
      MSR = MDAT(M,1)
      MHZ = MDAT(M,2)
      K1 = KA
      K2 = KB
      IF (.NOT.RSW) GO TO 117
      K1 = BDAT(M,5)
      K2 = BDAT(M,6)
  117 IF (FSW) GO TO 120

*  CALL FITT EXCEPT ON OPTION IN CASE OF FITQ.
      CALL FITT(MSR,MHZ,MV,IV,Q,K1,K2)
      RETURN
  120 CALL FITE(MSR,MHZ,MV,IV,Q,K1,K2)
      RETURN
      END
```

```
      SUBROUTINE SETMAT(MT)
C  ROUTINE SETS UP AND STORES UNIT AND TRANSPOSE MATRICES
      DIMENSION T(7,7),RW(3)
      ASSIGN 6 TO NEXT
      DO 3 I=1,3
    3 RW(I) = 0.
    4 DO 5 J=1,7
      DO 5 I=1,7
    5 T(I,J) = 0.
      GO TO NEXT,(6,10)
C  COMPUTE UNIT MATRIX NAMED (1)
    6 DO 7 I=1,7
    7 T(I,I) = 1.
      RW(3) = T(5,6)
      CALL STOR7(MT,T,RW)
      MT = MT - 1
      ASSIGN 10 TO NEXT
      GO TO 4
C  COMPUTE TRANSPOSE NAMED (S)
   10 DO 12 I=1,5,2
      T(I,I + 1) = -1.
   12 T(I+1,I) = + 1.
      T(7,7) = +1.
      RW(3) = T(5,6)
      CALL STOR7(MT,T,RW)
      RETURN
      END
```

```
      SUBROUTINE SHF(M,PAR)                                             SHF        2
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      DIMENSION RX(2,3),RY(2,3),PAR(4),RW(3)                            SHF        4
      DATA (RW(I),I=1,3)/3*0.0/                                         SHF        5
C  SETS UP A SHIFT MATRIX                                               SHF        6
      RX(1,1)=1.                                                        SHF        7
      RX(1,2)=0.                                                        SHF        8
      RX(1,3)=PAR(1)                                                    SHF        9
      RX(2,1)=0.                                                        SHF       10
      RX(2,2)=1.                                                        SHF       11
      RX(2,3)=PAR(2)                                                    SHF       12
      RY(1,1)=1.                                                        SHF       13
      RY(1,2)=0.                                                        SHF       14
      RY(1,3)=PAR(3)                                                    SHF       15
      RY(2,1)=0.                                                        SHF       16
      RY(2,2)=1.                                                        SHF       17
      RY(2,3)=PAR(4)                                                    SHF       18
      KB=INFF(5,M)                                                      SHF       19
      IF(KB.LE.0) GO TO 2                                               SHF       20
C     IF(KB.GE.1)  ERRORS ARE RANDOM WITH GIVEN RANGE                   SHF       21
      DO1 K=1,2                                                         SHF       22
      RX(K,3)=RX(K,3)*(RANF(D)-.5)                                      SHF       23
    1 RY(K,3)=RY(K,3)*(RANF(D)-.5)                                      SHF       24
    2 CONTINUE                                                          SHF       25
      CALL STXY(M,RX,RY,RW)                                             SHF       26
      RETURN                                                            SHF       27
      END                                                               SHF       28
```

```
      SUBROUTINE SKIP1
C  SKIPS ONE PRINT LINE
      WRITE (3,1)
 1    FORMAT (1H )
      RETURN
      END
```

```
      SUBROUTINE SHF7(M,PAR)                                          SHF7      2
C .................................................................  SHF7      3
C   SQ     SHF7          R17     R27    R37    R47    R57    R67      SHF7      4
C .................................................................  SHF7      5
C   DEFINES A 7X7 SHIFT MATRIX, CONSISTING OF THE UNIT MATRIX WITH    SHF7      6
C   THE 7TH COLUMN REPLACED BY THE INPUTED SIX NUMBERS.              SHF7      7
                                                                      SHF7      8
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL DIM       2
                                                                      DIM       3
      DIMENSION T(7,7),RW(3),PAR(6)                                   SHF7     10
                                                                      SHF7     11
      CALL RTRV7(MUNIT,T,RW)                                          SHF7     12
      DO 10 I=1,6                                                     SHF7     13
   10 T(I,7) = PAR(I)                                                 SHF7     14
      CALL STOR7(M,T,RW)                                              SHF7     15
      RETURN                                                          SHF7     16
      END                                                             SHF7     17
```

```
      SUBROUTINE SMIN(M)                                                SMIN       2
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1          LDFLG,FIN                                               CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      COMMON /MATCH2/ AA(30),BB(30),KBTS,SMINCAL,                       MATCH2     2
     1        DWORD(10),DWORD2(10),DWORD3(10),DWORD7(7,10),MINCOM       MATCH2     3
      LOGICAL SMINCAL                                                   MATCH2     4
      INTEGER BDAT,DWORD,DWORD2,DWORD3                                  SMIN       6
                                                                        SMIN       7
C  INSTRUCTION SMIN ------ SET COMMANDS FOR MINUIT                       SMIN       8
                                                                        SMIN       9
C  INPUT HAS SAME STRUCTURE AS MINUIT COMMANDS, BUT SHIFTED TO START    SMIN      10
C  IN COLUMN 21, WITH FIVE WORDS OF FLOATING POINT DATA PER COMMAND.    SMIN      11
C  EXAMPLE                                                              SMIN      12
                                                                        SMIN      13
C FIT  SMIN   4      PRINTOUT   2.                                      SMIN      14
C                    SEEK       500.                                    SMIN      15
C                    SIMPLEX    1500.     0.1                           SMIN      16
C                    END RETURN                                         SMIN      17
                                                                        SMIN      18
      SMINCAL = .TRUE.                                                  SMIN      19
      KA=INFF(4,M)                                                      SMIN      20
                                                                        SMIN      21
      DO 1 I=1,KA                                                       SMIN      22
      NCOM1 = BDAT(M,1+2*(I-1))                                         SMIN      23
      NCOM2 = BDAT(M,2+2*(I-1))                                         SMIN      24
      ENCODE(10,100,NCOM) NCOM1,NCOM2                                   SMIN      25
  100 FORMAT(2A5)                                                       SMIN      26
      DECODE(10,101,NCOM) DWORD(I),DWORD2(I),DWORD3(I)                  SMIN      27
  101 FORMAT(2A4,A2)                                                    SMIN      28
                                                                        SMIN      29
      LFPI=5*(I-1)+1                                                    SMIN      30
      CALL DATA(M,1,LFPI,5,DWORD7(1,I))                                 SMIN      31
    1 CONTINUE                                                          SMIN      32
                                                                        SMIN      33
      RETURN                                                            SMIN      34
      END                                                               SMIN      35
```

```
      SUBROUTINE SOL(M,PAR)
C   FORMS A MATRIX  REPRESENTING A SOLENOID.
C  ..............................................................
C    S     SOL      L      (BLANK)     BRHO      B0
C ...............................................................
C
C     L = LENGTH OF SOLENOID
C     BRHO = B*RHO  (THE RIGIDITY OF THE PARTICLE)
C     B0    = LONGITUDINAL FIELD IN SOLENOID
C
      REAL K,L
      DIMENSION PAR(1),T(7,7),RW(3)
C
      DO 11 J=1,7
      DO 10 I=1,7
   10 T(I,J) = 0.
   11 CONTINUE
      DO 12 I=1,3
   12 RW(I) = 0.
C
      L = PAR(1)
      BRHO = PAR(3)
      B0 = PAR(4)
      K = B0/(2.*BRHO)
      C = COS(K*L)
      S = SIN(K*L)
      SC = S*C
      SSQ = S*S
      CSQ = C*C
      T(1,1) = CSQ
      T(2,1) = -K*SC
      T(3,1) = -SC
      T(4,1) = K*SSQ
      T(1,2) = SC/K
      T(2,2) = CSQ
      T(3,2) = -SSQ/K
      T(4,2) = -SC
      T(1,3) = SC
      T(2,3) = -K*SSQ
      T(3,3) = CSQ
      T(4,3) = -K*SC
      T(1,4) = SSQ/K
      T(2,4) = SC
      T(3,4) = SC/K
      T(4,4) = CSQ
      T(5,5) = 1.
      T(6,6) = 1.
      T(7,7) = 1.
C
      RW(1) = L
      CALL STOR7(M,T,RW)
      RETURN
      END
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
22	I	PAR	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
23	I	PAR	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

```
      SUBROUTINE SOLV(M)                                                SOLV        2
                                                                        BLANK       2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK       3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ    1
      DIMENSION INFF(24,2000)                                           86MARSIZ    2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ    3
                                                                        BLANK       5
      COMMON/BMI1/MI1(16000)                                            NV3BMI1     1
      COMMON/NELS1/NELS1                                                BMI1L       2
      LEVEL 2,MI1                                                       BMI1L       3
                                                                        BMI1L       4
C                                                                       BMI1        3
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL      2
     1          LDFLG,FIN                                               CONTRL      3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL      4
                                                                        CONTRL      5
       COMMON                                                           COMMONT     2
     1/MINERR/ ERP(30)     ,ERN(30)                                     COMMONT     4
     2/PARINT/ X(15)       ,XT(15)      ,DIRIN(15)  ,MAXINT     ,NPAR   COMMONT     5
     3/PAREXT/ U(30)       ,NAM(30)     ,WERR(30)   ,MAXEXT     ,NU     COMMONT     6
     4/LIMITS/ ALIM(30)    ,BLIM(30)    ,LCODE(30)  ,LCORSP(30) ,LIMSET COMMONT     7
     5/VARIAN/ V(15,15)                                                 COMMONT     8
     7/FIX   / IPFIX(15)   ,XS(15)      ,XTS(15)    ,WTS(15)   ,NPFIX   COMMONT     9
     C/CASC/ JH, JL, Y(16)                                              COMMONT    10
     F/DERIVA/ G(30)       ,G2(30)                                      COMMONT    11
     G/SIMVEC/ P(15,16)    ,PSTAR(15)   ,PSTST(15)  ,PBAR(15)   ,PRHO(15) COMMONT  12
     J/VARIAT/ VT(15,15)                                                COMMONT    13
     6/UNIT  / ISYSRD      ,ISYSWR      ,ISYSPU                         COMMONT    27
     8/TITLE / TITLE(13)   ,DATE(2)     ,ISW(7)     ,NBLOCK             COMMONT    28
     9/CONVER/ EPSI  ,APSI ,VTEST       ,NSTEPQ     ,NFCN       ,NFCNMX COMMONT    29
     A/CARD  / CWORD       ,CWORD2      ,CWORD3     ,WORD7(7)           COMMONT    30
     B/MINIMA/ AMIN        ,UP          ,NEWMIN     ,ITAUR      ,SIGMA  COMMONT    31
     K/COMMND/ NNAME       ,CNAME(25)   ,CNAM2(25)  ,CNAM3(25)          COMMONT    32
      COMMON /MATCH/ KAT,KBT,MPAR(30),IPAR(30),MSR,MTR,LTRTAB,FNORM,    MATCH       2
     1       ICON(30),ITYPE(30),NPOS(30),IPOS(7,30),BET0(30),WT(30),    MATCH       3
     2       SIG(30),NAMBFC(42),ERPR,IVSAV,LVAR(30),WTVAR(30),VAL(30),  MATCH       4
     3       VARWTS,IPRPOS(6,30)                                        MATCH       5
      LOGICAL ERPR,LVAR,VARWTS                                          MATCH       6
      LOGICAL JWT                                                       SOLV        8
C     DATA(NAMBFC(I),I=1,42)/5HNUX  ,5HBX   ,5HAX   ,5HGX   ,5HX    ,   SOLV        9
C    1       5HDX   ,5HNUY  ,5HBY   ,5HAY   ,5HGY   ,5HY    ,5HDY   ,   SOLV       10
C    2       5HS    ,5HTHET ,                                           SOLV       11
C    3       5HBXMX ,5HBYMX ,5HXMAX ,5HQX   ,5HQY   ,5H     ,5HGAMT ,   SOLV       12
C    4       5HCHRX ,5HCHRY ,5HALPH ,5HYMAX ,5HBXMN ,5HBYMN ,5HXMIN ,   SOLV       13
C    5       5HYMIN ,5HCIRC ,5HTHTX ,5HTHTY ,5H     ,5HMTRX ,5HRES  ,   SOLV       14
C    6       5HRESN ,5HBXBY ,5HRND2 ,5HRND  ,5HWST  ,5HXDX  ,5HAXAY /   SOLV       15
C** DATA STATEMENT IS IN ELNUM **                                       SOLV       16
      COMMON /MATCH2/ AA(30),BB(30),KBTS,SMINCAL,                       MATCH2      2
     1        DWORD(10),DWORD2(10),DWORD3(10),DWORD7(7,10),MINCOM       MATCH2      3
      LOGICAL SMINCAL                                                   MATCH2      4
      COMMON /MATCH3/NOPR,NONU,FLAG,IPR,IPP,MINFLG,MIFLG,MATFLG,GLOBAL  MATCH3      2
      LOGICAL FLAG,NOPR,NONU,MIFLG,MATFLG,GLOBAL                        MATCH3      3
      DIMENSION U0(30),    NAIND(5),IND(5),NAMI(5),NAMPAR(6),           SOLV       19
     1          BXI(9),BYI(9)                                           SOLV       20
      DIMENSION NAIPOS(6),NIP(2),IP(2)                                  SOLV       21
      INTEGER BDAT,ELNUM,OPTR,HCYC,HTRKB                                SOLV       22
      DATA HTRKB/4HTRKB/,HCYC/3HCYC/                                    SOLV       23
```

```
      DATA M0/0/
      DATA LIM1,LIM2/7HINITIAL,7H  FINAL/
      DATA MINZER/77777 77777 77777 77777B/
      DATA MASK1/77000 00000 00000 00000B/
      DATA IBLNK1/55000 00000 00000 00000B/
      DATA NPR/6/
      DATA IBLNK/1H /
C-----------------------------------------------------------------------

C  INSTRUCTION SOLV -- BETATRON FUNCTION FITTING USING MINUIT

C.......................................................................
C XX    SOLV  KA  KB//SR    TR    IP1  IP2  ITS   TOL  PRN  SAV  PRI  LAMDA
C (CONSTR CD   1)   BNAM TYPE POS1 POS2 POS3 POS4 POS5 POS6 BFIT    SIG
C (CONSTR CD   2)   BNAM TYPE POS1 POS2 POS3 POS4 POS5 POS6 BFIT    SIG

C (CONSTR CD  KA)   BNAM TYPE POS1 POS2 POS3 POS4 POS5 POS6 BFIT    SIG
C (VARIABLE CD  1)  PAR     I1   I2   I3   I4   I5 LOWER     UPPER  STP
C (VARIABLE CD  2)  PAR     I1   I2   I3   I4   I5 LOWER     UPPER  STP

C (VARIABLE CD KB)  PAR     I1   I2   I3   I4   I5 LOWER     UPPER  STP
C.......................................................................
C   2ND FORMAT FOR VARIABLE CARDS ---
C .......................................................................
C                   PAR1 PAR1 PAR3 PAR4 PAR5    I LOWER     UPPER    STO
C.......................................................................
C  PARK IS NAME OF KTH ELEMENT WHOSE ITH INPUT NUMBER IS TO BE VARIED..
C-----------------------------------------------------------------------

C  FIRST CARD ---
C  KA = NUMBER OF CONSTRAINT CARDS
C  KB = NUMBER OF VARIABLE CARDS
C  SR = NAME OF SYNCH SUBROUTINE WITH VARIABLES AND A TRKB INSTRUCTION
C  TR = NAME OF TRKB INSTRUCTION IN SR.
C       TR MAY ALSO BE A CYC INSTRUCTION, WITH ITS KA SET TO 21,
C       OR AN MMM OR EQUIVALENT INSTRUCTION THAT DEFINES A SINGLE
C       TRANSFER MATRIX FOR THE LATTICE PERIOD.
C  IP1 = START POSITION OF TRKB RUN
C  IP2 = END   POSITION OF TRKB RUN.  IF ZERO, USE VALUES ON TRKB CARD.
C  ITS = MAX NO. OF CALLS TO FCN BY MINUIT
C  TOL = DESIRED TOLERANCE ON FCN ( FCN .LE. 10**TOL )--AN INTEGER
C  PRN = PRINTOUT LEVEL FROM MINUIT ( 0 GIVES LEAST OUTPUT )

C  SAV----INITIALIZED TO 0 BY SRUN.  IF SAV DIFFERENT FROM PREVIOUS
C         VALUE, CURRENT VARIABLE VALUES ARE SAVED IN ARRAY V.
C         SOLV ALWAYS TAKES INITIAL VARIABLE VALUES FROM V.
C         NORMALLY SET SAV=0, THEN VALUES FROM PREVIOUS FIT ARE USED
C         FOR STARTING VALUES.  SET SAV=1 TO ALWAYS START FROM
C         SAME INITIAL VALUES.
C  PRI = 0 (OR BLANK) --- PRINT BETAS BEFORE AND AFTER FITTING
C        1            --- PRINT BETAS AFTER FITTING ONLY
C        2            --- DO NOT PRINT BETAS
C        10,11,12     --- SAME AS 0,1,2 AND FORCE DOIT TO ALWAYS CALL MI

C  LAMDA --- SEE BELOW UNDER VARIABLES
C-----------------------------------------------------------------------
C  CONSTRAINT CARDS ---
```

```
C                    BNAM TYPE IP1  IP2  IP3  IP4  IP5  IP6   BFIT    SIG
C  EACH CONSTRAINT CARD MAY INVOLVE UP TO 6 SEPARATE POSITIONS
C BNAM = NAMES OF BETA FUNCTION CONSTRAINTS-----
C        NUX  BX   AX   GX   X    DX   NUY  BY   AY   GY   Y    DY
C        S     THET ----- WST  MATRX
C
C   S = PATH LENGTH, THET = HORIZONTAL BEND ANGLE,
C   WST COMBINES THREE CONSTRAINTS AX, AY, AND X
C   MTRX ---  CONSTRAIN MATRIX ELEMENT I1,I2 OF MATRIX
C             CORRESPONDING TO TR.
C   GAMT --- TRANSITION GAMMA ( TR MUST BE A CYC   21 INSTR.)
C
C TYPE = STD OR BLANK -----BETAS SHOULD FIT SPECIFIED VALUES
C  TYPE = DIF ---- DIFFERENCE OF BETAS BETWEEN SUCCESSIVE POSITIONS
C        SUM ---- SUM        (DITTO)
C                SHOULD FIT SPECIFIED VALUE BFIT.
C       WHEN BFIT = S, DIF AFFORDS A WAY TO OBTAIN GIVEN SEPARATIONS
C       BETWEEN SPECIFIED POSITIONS.
C TYPE = BFIT ---- BNAM IS THE NAME OF A PREVIOUSLY DEFINED MATRIX;
C                THE BETA FUNCTIONS OF THAT MATRIX ARE CALCULATED;
C                THEIR VALUES ARE THOSE THAT SOLV SEEKS TO FIT.
C TYPE = NO ------ PASS OVER THIS CONSTRAINT
C
C  I1,I2, ...I5 ARE POSITIONS AT WHICH THE CONSTRAINT APPLIES.
C THE POSITIONS CAN BE ENTERED SYMBOLICALLY IN THE FORM AAANN,
C WHERE AAA IS THE NAME OF A BEAMLINE ELEMENT, AND NN DENOTES
C WHICH OCCURRENCE OF AAA IN THE BEAMLINE IS MEANT.
C IF NN IS BLANK, THE FIRST OCCURENCE OF THE NAME IS UNDERSTOOD.
C
C
C  SIGMA = GOODNESS OF FIT DESIRED (INVERSE WEIGHT).
C
C ---------------------------------------------------------------------
C IF BNAM IS NOT A RECOGNIZED NAME, CONSTRAINT IGNORED, BUT POSITIONS
C CAN BE USED TO MARK FIT POINTS FOR SUBSEQUENT CONSTRAINT CARDS.
C IF A CONSTRAINT CARD HAS NO POSITIONS, THE POSITIONS ON THE NEAREST
C PREVIOUS CARD WITH POSITIONS PUNCHED WILL APPLY.

C FCN = FUNCTION TO BE MINIMIZED BY SUBROUTINE FCN IS
C FCN = SUM[( (VALUE-FITVALUE)/TOLERANCE)**2)]/(NUMBER OF CONSTRAINTS)
C ---------------------------------------------------------------------
C VARIABLE CARDS ---
C KB = NUMBER OF VARIABLE CARDS, EACH MAY HAVE 1 PARAMETER NAME
C      AND UP TO 5 PARAMETER INDICES.
C   SECOND FORMAT ----UP TO 5 PARAMETER NAMES FOLLOWED BY 1 INDEX.
C IF A VARIABLE CARD HAS NO POSITIONS, THE POSITIONS ON THE NEAREST
C PREVIOUS CARD WITH POSITIONS PRESENT WILL APPLY.
C
C IF STEPSIZ NOT ENTERED, DEFAULT  IS (UPRBOUND-LOWBOUND)/2.
C IF STEPSIZ IS NEGATIVE, VARIABLE ENTERS INTO FCN WITH
C FIT VALUE = (UPPER+LOWER)/2,  TOLERANCE = LAMDA*(UPPER-LOWER)/2
C ---------------------------------------------------------------------
C PASS ITS, TOL, PRN TO MINUIT
      IF (SMINCAL) GO TO 5
C  SAVE DEFAULT VALUES
       ISV1 = DWORD7(1,1)
       ISV2 = DWORD7(1,2)
```

```
       ISV3 = DWORD7(2,2)
      ITS = IDAT(M,1)
      IF(ITS.EQ.0) GO TO 3
      DWORD7(1,2)=ITS
    3 ITOL = IDAT(M,2)
      DWORD7(2,2)=10.**ITOL
    4 IPR = IDAT(M,3)
      IF(IPR.EQ.0) GO TO 5
      DWORD7(1,1)=IPR
    5 CONTINUE
      IVSAV0=IVSAV
      IVSAV = IDAT(M,4)
      IPP = IDAT(M,5)
      MIFLG = .FALSE.
      MATFLG = .FALSE.
      GLOBAL = .FALSE.
      IF (IPP.LT.10) GO TO 52
      MIFLG = .TRUE.
      IPP = IPP - 10
   52 KAT=INFF(4,M)
      KAR=KAT
      KBT=INFF(5,M)
      KBR=KBT
      KBTS=KBT
      MSR=MDAT(M,1)
      MTR=MDAT(M,2)
      KATR = 0
      OPTR = INFF(1,MTR)
      IF ((OPTR.NE.HTRKB).AND.(OPTR.NE.HCYC))   GO TO 55
      MBML = MDAT(MTR,1)
      NDAT = INFF(17,MBML)
      CALL MIFILL(MBML,1,NDAT,NELS1,MI1)
      NELS2 = NELS1
      KBTR = INFF(5,MTR)
      IF (OPTR.NE.HTRKB)  GO TO 53
      KATR = INFF(4,MTR)
      CALL DATA (M,2,3,2,NIP)
      CALL LOCS(NIP,2,NELS1,MI1,IP)
      IP1 = IP(1)
      IP2 = IP(2)
      IF (IP1.EQ.0.AND.IP2.EQ.0)  GO TO 54
      CALL REPINT(MTR,1,IP1)
      CALL REPINT(MTR,2,IP2)
      GO TO 54
   53 IF (KBTR.LT.0)  NELS2 = NELS1 + 1
   54 LQTR = INFF(24,MTR)
      LTRTAB = LQTR - 14*KATR
      GO TO 51
   55 MATFLG = .TRUE.
C 51 NASR = INFF(2,MSR)
   51 CONTINUE
      XLAMDA=FDAT(M,1)


C  GET CONSTRAINT DATA AND STORE IN ARRAYS FOR FCN

      DO 61 KK=1,30
```

```
   61 IPOS(7,KK)=0

      NCONSTR = 0
      K=0

      DO 2 N=1,KAR
      K=K+1
      ICON(K)=0
      NTYPE = BDAT(M,8*N-2)
      ITYPE(K)=0
      IF(NTYPE.EQ.5HNO    ) ITYPE(K)=-1
      IF(NTYPE.EQ.5HSTD   ) ITYPE(K)=0
      IF(NTYPE.EQ.5HDIF   ) ITYPE(K)=1
      IF (NTYPE.EQ.5HSUM   ) ITYPE(K)=2
      NCONK = BDAT(M,8*N-3)
      IF (NTYPE.NE.5HBFIT )  GO TO 62
      ITYPE(K)=3
      GO TO 63

   62 DO 6 I=1,42
    6 IF (NCONK.EQ.NAMBFC(I)) ICON(K)=I

      ICONK = ICON(K)
      IF (ICON(K).GE.34.AND.ICON(K).LE.37)   MIFLG = .TRUE.
      IF(ICON(K).EQ.0) ITYPE(K)=-1
   63 LPOSK = 7 + 8*(N-1)
      IF (ICON(K).LE.14.OR.ICON(K).GE.34)  GO TO 65
      IPOS(1,K) = NELS2
      NPOS(K) = 1
      GLOBAL = .TRUE.
      GO TO 8
   65 CONTINUE
      CALL DATA (M,2,LPOSK,6,NAIPOS)
      CALL LOCS(NAIPOS,6,NELS1,MI1,IPOS(1,K))
      NPOS(K)=0

      I=0
    7 I=I+1
      IF (NAIPOS(I).NE.IBLNK)  GO TO 72
      NP1=0
      I1=I

      DO 71 II=I1,6
      IPOS(II,K) = IPOS(II+1,K)
   71 IF (NAIPOS(I).NE.IBLNK)  NP1 = NP1 + 1

      I=I-1
      IF(NP1.EQ.0) GO TO 73
      GO TO 7
   72 NPOS(K)=NPOS(K)+1
      IF(I.LT.6) GO TO 7

   73 CONTINUE
      IF(NPOS(K).NE.0) GO TO 8
      IF (ICON(K).NE.0) GO TO 74
      K=K-1
      GO TO 2
```

```
   74 CONTINUE
      IF (K.GT.1)  GO TO 10
      NUMCON=1
      GO TO 32
   10 NPOS(K)=NPOS(K-1)
      DO 9 I=1,6
    9 IPOS(I,K)=IPOS(I,K-1)

    8 CONTINUE
      NUMCON=NPOS(K)
   32 CONTINUE
      IF (ITYPE(K).EQ.-1) NUMCON=0
      IF ( (ITYPE(K).EQ.1).OR.(ITYPE(K).EQ.2) )  NUMCON=NUMCON-1

      LFPK=2*(N-1)+1
      BET0(K) = FDAT(M,LFPK+1)
       SIG(K) = FDAT(M,LFPK+2)
      IF(SIG(K).EQ.0.) SIG(K)=1.E20
        WT(K)=1./SIG(K)

      IF (ICON(K).LT.38)  GO TO 13
      LL = 1
      IF (ICONK.NE.40.AND.ICONK.NE.39)  GO TO 28
      LL = 2
      ICON(K)  =3
      ICON(K+1)=6
      ICON(K+2)=9

      IF (ICONK.EQ.40)  GO TO 27
      LL = 3
      ICON(K+3) = 37
      BET0F = BET0(K)
      BET0(K) = 0.
      GO TO 27

   28 IF (ICONK.NE.41)  GO TO 29
      ICON(K) = 5
      ICON(K+1) = 6
      GO TO 27

   29 IF (ICONK.NE.42.AND.ICONK.NE.38)  GO TO 27
      ICON(K) = 3
      ICON(K+1) = 9

      IF (ICONK.EQ.42)  GO TO 27
      LL = 2
      ICON(K+2) = 37
      BET0F = BET0(K)
      BET0(K) = 0.


   27 CONTINUE
      KL=K
      DO 11 L=1,LL
      KL=KL+1
      ITYPE(KL)=ITYPE(K)
      NPOS(KL)=NPOS(K)
```

```
      BET0(KL)=BET0(K)
      SIG(KL)=SIG(K)
      WT(KL)=WT(K)
      DO 12 I=1,6
   12 IPOS(I,KL)=IPOS(I,K)
   11 CONTINUE
      IF (ICONK.EQ.38.OR.ICONK.EQ.39)  BET0(KL)=BET0F
      NUMCON = (LL+1)*NUMCON
      K = KL
      KAT = KAT + LL
      GO TO 131
   13 CONTINUE
      IF ((ICON(K).NE.35).AND.(ICON(K).NE.36))  GO TO 14
      NPOS(K) = 4
      NUMCON = 1
      IF (IPOS(3,K).EQ.0)  IPOS(3,K)=1
      IF (IPOS(4,K).EQ.0)  IPOS(4,K)=1
   14 CONTINUE
      IF (ICON(K).NE.36)  GO TO 24
      NN = IPOS(1,K)/2
      ISGN = 1
      IF (IPOS(2,K).LT.0)  ISGN = -1
      IPOS(2,K) = 0
      ICON(K) = 35
      KK = K
      DO 25 II=1,NN
      KK = KK + 1
      DO 26 JJ=3,4
   26 IPOS(JJ,KK) = IPOS(JJ,K)
      ICON(KK) = 35
      NPOS(KK) = 4
      SIG(KK) = SIG(K)
      BET0(KK) = BET0(K)
      WT(KK) = WT(K)
      IPOS(1,KK) = IPOS(1,KK-1) -2
      IPOS(2,KK) = IPOS(2,KK-1) + 2*ISGN
   25 CONTINUE
      K = K + NN
      KAT = KAT + NN
      NUMCON = NN + 1
   24 CONTINUE
      IF (ITYPE(K).NE.3) GO TO 131
      MBET=ELNUM(NCONK)
      CALL BET(MBET,BXI,BYI,0)
      ICON(K   )=2
      ICON(K+1)=3
      ICON(K+2)=5
      ICON(K+3)=6
      ICON(K+4)=8
      ICON(K+5)=9
      BET0(K   )=BXI(2)
      BET0(K+1)=BXI(3)
      BET0(K+2)=BXI(5)
```

```
      BET0(K+3)=BXI(6)
      BET0(K+4)=BYI(2)
      BET0(K+5)=BYI(3)

      DO 132 L=1,5
      ITYPE(K+L)=ITYPE(K)
      NPOS(K+L)=NPOS(K)
      SIG(K+L)=SIG(K)
      WT(K+L)=WT(K)

      DO 133 I=1,6
  133 IPOS(I,K+L)=IPOS(I,K)

  132 CONTINUE

      NUMCON=6*NUMCON
      K=K+5
      KAT=KAT+5
  131 CONTINUE

      NCONSTR = NCONSTR + NUMCON
    2 CONTINUE
      DO 82 K=1,30
      DO 82 I=1,6
   82 IPRPOS(I,K) = IBLNK
      DO 90 K=1,NCONSTR
      NP=NPOS(K)
      IF (ICON(K).GT.14)  GO TO 84
      IF ((OPTR.NE.HTRKB).AND.(OPTR.NE.HCYC))  GO TO 90
      GO TO 86
   84 IF (ICON(K).LE.32)  GO TO 90
      IF (ICON(K).LT.38)  GO TO 86
      IF ((OPTR.NE.HTRKB).AND.(OPTR.NE.HCYC))  GO TO 90
   86 DO 88 I=1,NP
   88 ENCODE(5,106,IPRPOS(I,K)) IPOS(I,K)
  106 FORMAT(I5)
   90 CONTINUE


C  GET DATA ON VARIABLES AND LOAD COMMON BLOCKS FOR FCN AND MIDATA
C  MPAR(J) = M-NO OF JTH VARIABLE

      VARWTS=.FALSE.
      NVARWT=0
      IF(XLAMDA.EQ.0.) XLAMDA=1.0
      L=0

      DO 1 J=1,KBR
      L2=1
      LOCJ=2*KAR+3*J-2
      AAJ=FDAT(M,LOCJ+1)
      BBJ=FDAT(M,LOCJ+2)
      WRJ=FDAT(M,LOCJ+3)
      VALJ=(BBJ+AAJ)/2.
      DEL=ABS(BBJ-AAJ)/2.
      IF (DEL.EQ.0.) DEL=1.
      SIGV=XLAMDA*DEL
```

```
      WTVARJ=1./SIGV
      JWT=.FALSE.
      IF(WRJ.EQ.0.) WRJ=DEL
      IF(WRJ.GE.0.) GO TO 23
      VARWTS=.TRUE.
      JWT=.TRUE.
      WRJ=-WRJ
   23 CONTINUE
      LVARJ = 8*KAR + 6*J -1
      MPARJ=MDAT(M,LVARJ)
      MPAR(L+1)=MPARJ
      NAMPAR(1)=BDAT(M,LVARJ)

C  FIND OUT FORMAT TYPE
      IFORM=1
      LI=L+1

      DO 17 I=1,5
      ERPR=.FALSE.
      MPARI=MDAT(M,LVARJ+I)
      IF (MPARI.EQ.MINZER)  GO TO 16
      IFORM=2
      L2=L2+1
      LI=LI+1
      MPAR(LI)=MPARI
      NAMPAR(L2)=BDAT(M,LVARJ+I)
      GO TO 17
   16 NAIND(I)=BDAT(M,LVARJ+I)
      DECODE(5,100,NAIND(I)) IND(I)
  100 FORMAT(I5)
C TO MAKE INPUT COMPATIBLE WITH VAX VERSION
      DECODE(5,2000, NAIND(I))LL
 2000 FORMAT(4XA1)
      IF(LL.EQ.1H ) IND(I)=IND(I)/10
      IF(IND(I).EQ.0) GO TO 17
      NAIND2=NAIND(I)
      IND2=IND(I)
   17 CONTINUE

      L0=L+1
      IF(IFORM.EQ.2) GO TO 18
C  FORMAT 1

      DO 15 I=1,5
      IF(IND(I).EQ.0) GO TO 15
      ENCODE(10,101,NAMI(I)) NAMPAR(1),NAIND(I)
  101 FORMAT(2A5)
      L=L+1
      NAM(L)=NAMI(I)
      MPAR(L)=MPARJ
      IPAR(L)=IND(I)
   15 CONTINUE
C   IF NO INDEX WAS PRESENT, USE PREVIOUSLY DEFINED ONE.
      IF (L.GE.L0)  GO TO 20
      L = L + 1
      ENCODE (10,101,NAM(L))  NAMPAR(1),NAIND2
      MPAR(L) = MPARJ
```

```
            IPAR(L) = IND2
            GO TO 20

      C  FORMAT 2
   18       LL2=L2

            DO 19 L2=1,LL2
            L=L+1
            ENCODE(10,101,NAM(L)) NAMPAR(L2),NAIND2
            IPAR(L)=IND2
   19       CONTINUE

   20       L1=L

            DO 21 L=L0,L1
            U(L)=FDAT(MPAR(L),IPAR(L))
            IF ((IVSAV.NE.0).AND.(IVSAV.EQ.IVSAV0).AND.(M.EQ.M0)) GO TO 22
            U0(L)=U(L)
   22       U(L)=U0(L)
            AA(L)=AAJ
            BB(L)=BBJ
            WERR(L)=WRJ
            LVAR(L)=.FALSE.
            WTVAR(L)=0.
            IF(.NOT.JWT) GO TO 21
            LVAR(L)=.TRUE.
            WTVAR(L)=WTVARJ
            VAL(L)=VALJ
            NVARWT=NVARWT+1
   21       CONTINUE
      C
            L = L1

             L = L1
    1       CONTINUE

            NCONTOT=NCONSTR+NVARWT
            FNORM=1./FLOAT(NCONTOT)
            M0=M
            NPAR=L
            KBT=L
            KBTS=KBT
            CALL DASH
            WRITE (3,102)
            WRITE (3,105) LIM1,(I,NAM(I),U(I),I=1,NPAR)
            CALL FCN(NPAR,G,AMIN,U,1)
            CALL DASH

            CALL MINNEW

            WRITE (3,105) LIM2,(I,NAM(I),U(I),I=1,NPAR)
            CALL FCN(NPAR,G,F,U,3)
            MINFLG=1
            CALL DASH
      C  RESTORE DEFAULTS IF SAVED
            MIFLG = .FALSE.
            NOPR = .FALSE.
```

```
      IF (SMINCAL)  RETURN
      DWORD7(1,1) = ISV1
      DWORD7(1,2) = ISV2
      DWORD7(2,2) = ISV3
      RETURN
  102 FORMAT (10X,*--- SOLV --- BETA-FUNCTION FITTING *)
  105 FORMAT (1H0,5X,A7,* VALUES OF VARIABLES*/
     1 (7X,I5,3X,A5,5X,E15.8))

      END
```

```
      SUBROUTINE SPLOAD                                                 SPLOAD     2
*   SPECIAL LOAD ROUTINE                                                SPLOAD     3
*     INPUT HAS P OR C IN COL. 1, FOLLOWED BY HOLLERITH DATA            SPLOAD     4
*     CONSTRUCTS A PAGE OR REM SYNCH INSTRUCTION                        SPLOAD     5
*     IF NOT IN MODE 2, EXECUTES PAGE AND/OR REMARK                     SPLOAD     6
*     WHEN IN MODE 3, IT IS HANDLED NORMALLY BY SWITCH                  SPLOAD     7
                                                                        SPLOAD     8
      DIMENSION ICOM(16)                                                SPLOAD     9
      COMMON/INSTR/OPNAME,NAME,OP,KA,KB,OBJA,OBJB,NXTM,TRA,LQ,NQ,LIN,   INSTR      2
     1              NIN,LFL,NFL,LBC,NBC,NTOT,IX,KIND,TYPE,ROWS,COLS,EXTR, INSTR    3
     2              M,NF,NB,NI,MSUBR,ISIGN                              INSTR      4
      INTEGER OPNAME,OP,OBJA,OBJB,TRA,TYPE,ROWS,COLS,EXTR               INSTR      5
                                                                        INSTR      6
      COMMON/STORE/LMAX,LINF,LFILE,MAX,MIN,ICARD(11)                    STORE      2
*  LMAX AND MAX ARE DIMENSIONS OF STORE AND INFF                        STORE      3
                                                                        STORE      4
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1          LDFLG,FIN                                               CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      COMMON/COPY/CPYSW                                                 COPY       2
      LOGICAL CPYSW                                                     COPY       3
                                                                        COPY       4
                                                                        SPLOAD    14
      INTEGER HPERIOD,HREM                                              SPLOAD    15
      DATA IB,HPERIOD,HREM/1H ,1H.,3HREM/                               SPLOAD    16
      DATA MASK1,MASK2/00777777777777777777B,55000000000000000000B/     SPLOAD    17
                                                                        SPLOAD    18
      INS = 1H                                                          SPLOAD    19
      IF (ISIGN.EQ.1HP) OPNAME=4HPAGE                                   SPLOAD    20
      IF (ISIGN.EQ.1HC) OPNAME = 4HREM                                  SPLOAD    21
      IF (ISIGN.EQ.HPERIOD)  OPNAME = HREM                              SPLOAD    22
      CALL STINFO(M,IB,OPNAME,0,0,0)                                    SPLOAD    23
      DECODE(80,2,ICARD(1)) (ICOM(I),I=1,16)                            SPLOAD    24
*    MASK OFF FIRST CHARACTER                                           SPLOAD    25
      ICOM(1) = ICOM(1).AND.MASK1                                       SPLOAD    26
      ICOM(1) = ICOM(1).OR.MASK2                                        SPLOAD    27
                                                                        SPLOAD    28
*                 RESERVE AND STORE                                     SPLOAD    29
      CALL RESRV(M,0,0,                                                 SPLOAD    30
     .  16,2,0,0,0,0,0,0                                                SPLOAD    31
     .)                                                                 SPLOAD    32
      CALL STDAT(M,2,1,16,ICOM)                                         SPLOAD    33
      IF (MODE.EQ.2) GO TO 10                                           SPLOAD    34
      IF (.NOT.CPYSW)  RETURN                                           SPLOAD    35
      IF (MODE.EQ.1)  GO TO 10                                          SPLOAD    36
      IF (ISIGN.EQ.1HP)  INS = ISIGN                                    SPLOAD    37
      GO TO 12                                                          SPLOAD    38
   10 IF (ISIGN.EQ.1HP)  WRITE (3,1)                                    SPLOAD    39
   12 WRITE (3,3) INS,(ICOM(I),I=1,16)                                  SPLOAD    40
      RETURN                                                            SPLOAD    41
                                                                        SPLOAD    42
    1 FORMAT (1H1)                                                      SPLOAD    43
    2 FORMAT (16A5)                                                     SPLOAD    44
    3 FORMAT (4X,A4,16A5)                                               SPLOAD    45
                                                                        SPLOAD    46
      END                                                               SPLOAD    47
```

```
      SUBROUTINE SRUN(IENTRY)                                           SRUN        2
                                                                        BLANK       2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK       3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ    1
      DIMENSION INFF(24,2000)                                           86MARSIZ    2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ    3
                                                                        BLANK       5
      COMMON/INSTR/OPNAME,NAME,OP,KA,KB,OBJA,OBJB,NXTM,TRA,LQ,NQ,LIN,   INSTR       2
     1              NIN,LFL,NFL,LBC,NBC,NTOT,IX,KIND,TYPE,ROWS,COLS,EXTR, INSTR     3
     2              M,NF,NB,NI,MSUBR,ISIGN                              INSTR       4
      INTEGER OPNAME,OP,OBJA,OBJB,TRA,TYPE,ROWS,COLS,EXTR               INSTR       5
                                                                        INSTR       6
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL      2
     1           LDFLG,FIN                                              CONTRL      3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL      4
                                                                        CONTRL      5
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL   DIM         2
                                                                        DIM         3
      COMMON/FLTN/IFL(15)                                               FLTN        2
                                                                        FLTN        3
      COMMON/STORE/LMAX,LINF,LFILE,MAX,MIN,ICARD(11)                    STORE       2
*  LMAX AND MAX ARE DIMENSIONS OF STORE AND INFF                        STORE       3
                                                                        STORE       4
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER   SWTCH       2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                          SWTCH       3
                                                                        SWTCH       4
      COMMON/TRKINT/G,EX,EY,OMSQ,SEND,VMX,VMN,DINT,BEG,DPR,JPR,         TRKINT      2
     1   THET,PLT,NPT,NZ,PMAX,NSIZ,ITITLE(6)                            TRKINT      3
      LOGICAL BEG,DPR,PLT                                               TRKINT      4
      COMMON /MATCH/ KAT,KBT,MPAR(30),IPAR(30),MSR,MTR,LTRTAB,FNORM,    MATCH       2
     1        ICON(30),ITYPE(30),NPOS(30),IPOS(7,30),BET0(30),WT(30),   MATCH       3
     2        SIG(30),NAMBFC(42),ERPR,IVSAV,LVAR(30),WTVAR(30),VAL(30), MATCH       4
     3        VARWTS,IPRPOS(6,30)                                       MATCH       5
      LOGICAL ERPR,LVAR,VARWTS                                          MATCH       6
      INTEGER R7,INAM(2)                                                SRUN       12
      COMMON/GRR/IERR,BSW,VSW,RSW,QSW,XSW                               GRR         2
      LOGICAL IERR,BSW,VSW,RSW,QSW,XSW                                  GRR         3
                                                                        GRR         4
      EQUIVALENCE (MIN,MS)                                              SRUN       14
      COMMON/SVNAM/NAMRUN                                               SVNAM       2
      DATA IBLNK/1H /,MXPMY,R7/1,5/,(INAM(I),I=1,2)/3H(1),3H(S)/        SRUN       16
      DATA IBLNK /1H /                                                  SRUN       17
      DATA MXPMY,R7/1,5/                                                SRUN       18
C                                                                       SRUN       19
      IF (OP.LT.0) RETURN                                               SRUN       20
      GO TO (4,5),IENTRY                                                SRUN       21
    4 CONTINUE                                                          SRUN       22
      WRITE (3,1) NAME,(ICARD(J),J=3,8)                                 SRUN       23
C  PUT ICARD INTO ITITLE FOR PLOT ROUTINES                              SRUN       24
      DO 6 J=1,6                                                        SRUN       25
    6 ITITLE(J) = ICARD(J+2)                                            SRUN       26
C  ISAV AND ISAV7  ARE NUMBER OF INFOS RESERVED FOR WORKING STORAGE     SRUN       27
    1 FORMAT (1H1,7X,15H     SYNCH RUN ,A5,10X,6A10//1X,130(1H=))       SRUN       28
C  LEND = LMAX = DIMENSION OF STORE                                     SRUN       29
C  MEND = MAX = DIMENSION OF M IN INFO                                  SRUN       30
C  RESERVE WORKING SPACE FOR MATRICES AND LENGTH                        SRUN       31
C  ZERO STORAGE                                                         SRUN       32
```

```
      DO 25 I=1,LMAX
   25 STORE(I) = 0.
      DO 26 I=1,MAX
      DO 26 J=1,24
   26 INFF(J,I)=0
      EMPTY=.FALSE.
      INDEF=.FALSE.
      XEQ=.TRUE.
      STOR=.TRUE.
      RSRV=.TRUE.
      ERROR=.FALSE.
      CYCSWT=.FALSE.
      IERR=.FALSE.
      PV = .FALSE.
      MSSW = .FALSE.
      VPR = .FALSE.
      MHS=.FALSE.
      MODS=.FALSE.
      ERPR=.TRUE.
      IVSAV=0
      MODE=1
      NF=0
      NB=0
      NI=0
      NAMRUN=NAME
      BEND=1.
      LFILE=1
      MS=MEND
      DO 3 I=1,ISAV
      CALL RESRV(MS,MXPMY,MINZER,
     . 0,0,0,0,0,0,0,0,0
     .)
      INFF(1,MS) = IBLNK
      INFF(2,MS) = IBLNK
      MS=MS-1
    3 CONTINUE
      MSIZE = 37
      I7 = ISAV7 - 2
      IF (ISAV7.EQ.0) GO TO 7
      DO 10 I=1,I7
      CALL RESRV(MS,R7,MINZER,
     . 0,0,0,0,0,0,0,0,0
     .)
      INFF (1,MS) = IBLNK
      INFF(2,MS) = IBLNK
      MS = MS - 1
   10 CONTINUE
C  SET UP INFO AND STORAGE FOR UNIT AND TRANSPOSE MATRICES
    7 CONTINUE
      MT = MS
      MUNIT = MS
      DO 15 I=1,2
      INFF(1,MS) = IBLNK
      INFF(2,MS) = INAM(I)
      CALL RESRV(MS,R7,MINZER,
     . 0,0,0,0,0,0,0,0,0
     .)
```

```
      15  MS = MS - 1                                                   SRUN      90
          MSYMPL = MS                                                   SRUN      91
C   DEFINE AND STORE MATRICES                                           SRUN      92
          CALL SETMAT(MT)                                               SRUN      93
*           RESTORE INFO FOR RUN                                        SRUN      94
          CALL WRTINF(OPNAME,MS)                                        SRUN      95
          RETURN                                                        SRUN      96
C   END OF RUN                                                          SRUN      97
    5     CONTINUE                                                      SRUN      98
          EMPTY=.FALSE.                                                 SRUN      99
          FIN = .TRUE.                                                  SRUN     100
          NSTO = LMAX - LFILE                                           SRUN     101
          NINFF = MAX - MIN                                             SRUN     102
          WRITE (3,11) NAME,OPNAME,KA,KB,LMAX,LFILE,NSTO,MAX,NINFF,MIN  SRUN     103
   11      FORMAT (/6H  *** ,1X,A5,2X,A5,1X,I3,1X,I3,1X,3H //,1X,       SRUN     104
         2 *CORE USE SUMMARY*,27X,*MAXIMUM*,17X,*USED*,8X,*UNUSED*/     SRUN     105
         3 34X,*STORE  (ELEMENT STORAGE)*,14X,I10,1X,*(LMAX)*,4X,I10,5X,I10/ SRUN     106
         4 34X,*INFF   (ELEMENT DEFINITIONS)*,10X,I10,1X,*(MAX)*,5X,I10,5X,  SRUN     107
         5  I10)                                                        SRUN     108
          WRITE (3,2) NAMRUN                                            SRUN     109
    2     FORMAT(/1X,130(1H=)/18H END OF SYNCH RUN ,A5)                 SRUN     110
          RETURN                                                        SRUN     111
          END                                                           SRUN     112
```

```
CARD NR. SEVERITY  DETAILS    DIAGNOSIS OF PROBLEM

    46      I                AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
```

```
      SUBROUTINE SSQMIN(M,N,F,X,E,ESCALE,IPRINT,MAXFUN)                 SSQMIN   2
      RETURN                                                            SSQMIN   3
      END                                                               SSQMIN   4
```

```
      SUBROUTINE PLOTT(N,C,X,Y)                                 TRKP683    3
      DIMENSION A(112,51),Z(6),X(1),Y(1)                        MY3TKPL   84
      COMMON/SAML/A,Z,XLOW,XRANGE,YLOW,YRANGE                   MY3TKPL   85
      IF( N .LT. 1 )RETURN                                      MY3TKPL   86
      DO 100 K=1,N                                              MY3TKPL   87
      XMULT =  ( X(K)-XLOW )/XRANGE                             MY3TKPL   88
      IF( XMULT .LT. 0.0 )GO TO 100                             MY3TKPL   89
      YMULT =  ( Y(K)-YLOW )/YRANGE                             MY3TKPL   90
      IF( YMULT .LT. 0.0 )GO TO 100                             MY3TKPL   91
      IT = INT( 110.0*XMULT +0.5 ) +1                           MY3TKPL   92
      I=IT                                                      MY3TKPL   93
      IF( I .GT. 111 )GO TO 100                                 MY3TKPL   94
      JT=  INT( 50.0*YMULT +0.5 ) +1                            MY3TKPL   95
      J=JT                                                      MY3TKPL   96
      IF( J .GT. 51 )GO TO 100                                  MY3TKPL   97
      A(I,J)=C                                                  MY3TKPL   98
  100 CONTINUE                                                  MY3TKPL   99
      RETURN                                                    MY3TKPL  100
      END                                                       MY3TKPL  101
```

```
      SUBROUTINE PAGTRKP(IC,K1)                               86PAG       1
      DIMENSION A(112,51),Z(6),X(1),Y(1)                      MY3TKPL   103
      COMMON/SAML/A,Z,XLOW,XRANGE,YLOW,YRANGE                 MY3TKPL   104
      WRITE(3,5555) K1                                        MY3TKPL   105
 5555 FORMAT(1H0,*PARTICLE NUMBER*,I5)                        MY3TKPL   106
      DO 250 K=1,51                                           MY3TKPL   107
      I=52-K                                                  MY3TKPL   108
      IP = MOD( I-1,5 )                                       MY3TKPL   109
      IF( IP ) 210,220,210                                    MY3TKPL   110
  220 CONTINUE                                                MY3TKPL   111
      DEL=YRANGE/10.00                                        MY3TKPL   112
      IM=I/5                                                  MY3TKPL   113
      FM=IM                                                   MY3TKPL   114
      XX=YLOW+FM*DEL                                          MY3TKPL   115
      IF( IC )222,221,222                                     MY3TKPL   116
221   WRITE(3,1221)XX,(A(J,I),J=1,112)                        MY3TKPL   117
1221  FORMAT(4X,F10.3,2H +,112A1)                             MY3TKPL   118
      GO TO 250                                               MY3TKPL   119
222   WRITE(3,1222)XX,(A(J,I),J=1,112)                        MY3TKPL   120
1222  FORMAT(4X,E10.3,2H +,112A1)                             MY3TKPL   121
      GO TO 250                                               MY3TKPL   122
210   WRITE(3,1210) (A(J,I),J=1,112)                          MY3TKPL   123
1210  FORMAT(15X,1HI,112A1)                                   MY3TKPL   124
  250 CONTINUE                                                MY3TKPL   125
      WRITE(3,1230)                                           MY3TKPL   126
 1230 FORMAT(15X,112(1H-))                                    MY3TKPL   127
      WRITE(3,1235)                                           MY3TKPL   128
 1235 FORMAT(16X,1HI,5(21X,1HI))                              MY3TKPL   129
      DEL=XRANGE/5.0                                          MY3TKPL   130
      DO 300 J=1,6                                            MY3TKPL   131
      Z(J)=XLOW+DEL*FLOAT(J-1)                                MY3TKPL   132
  300 CONTINUE                                                MY3TKPL   133
      IF(IC)400,500,400                                       MY3TKPL   134
  500 WRITE(3,1500) (Z(I),I=1,6)                              MY3TKPL   135
 1500 FORMAT(10X,F10.3,5(12X,F10.3))                          MY3TKPL   136
      GO TO 1499                                              MY3TKPL   137
  400 WRITE(3,1400)(Z(I),I=1,6)                               MY3TKPL   138
 1400 FORMAT(10X,E10.3,5(12X,E10.3))                          MY3TKPL   139
 1499 RETURN                                                  MY3TKPL   140
      END                                                     MY3TKPL   141
```

```
C ********************************TRKB********************************* 86TRKB      1
C BETATRON FUNCTION TRACKING ROUTINE 9-19-85                               86TRKB      2
C                                                                          86TRKB      3
      SUBROUTINE TRKB(M)                                                   86TRKB      4
                                                                           BLANK       2
      LEVEL 2, STORE,INFF,IWORK                                            BLANK       3
      COMMON STORE(48000),IWORK(10)                                        86MARSIZ    1
      DIMENSION INFF(24,2000)                                              86MARSIZ    2
      EQUIVALENCE (INFF,STORE)                                             86MARSIZ    3
                                                                           BLANK       5
      COMMON/BMI1/MI1(16000)                                               NV3BMI1     1
      COMMON/NELS1/NELS1                                                   BMI1L       2
      LEVEL 2,MI1                                                          BMI1L       3
                                                                           BMI1L       4
C                                                                          BMI1        3
      DIMENSION MI(1)                                                      TRKB        5
      EQUIVALENCE (MI,MI1),(NELS,NELS1)                                    TRKB        6
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,       CONTRL      2
     1          LDFLG,FIN                                                  CONTRL      3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN         CONTRL      4
                                                                           CONTRL      5
                                                                           TRKB        8
      COMMON /MATCH3/NOPR,NONU,FLAG,IPR,IPP,MINFLG,MIFLG,MATFLG,GLOBAL     MATCH3      2
      LOGICAL FLAG,NOPR,NONU,MIFLG,MATFLG,GLOBAL                           MATCH3      3
      COMMON/CDERIV/IFLAG,MR,NR,DP,NV(11),DAT(20),UZ,STOT,                 CDERIV      2
     1 DAT2(7),MP2FLG,NF,H0,LOCALPH,NUMALPH                                CDERIV      3
      DIMENSION BXI(9),BYI(9),V0(6),VI(6),VF(7),T(49)                      86TRKB      5
      DIMENSION BX(6),BY(6),BX0(6),BY0(6),RX(2,3),RY(2,3),RW(3),BETF(14)   86TRKB      6
      DIMENSION P(8), V(7)                                                 86TRKB      7
      EQUIVALENCE(BX,BETF),(BY,BETF(7)), (S,BETF(13)),(TH,BETF(14))        86TRKB      8
      EQUIVALENCE (V,VF(7))                                                86TRKB      9
      INTEGER BDAT,ELNAMEI,ELNAME                                          86TRKB     10
      EXTERNAL DER2,DER3,DER4,DELQ                                         86TRKB     11
      LOGICAL NOSTOR,MPFLG,VECFLAG                                         86TRKB     12
      LOGICAL RADFLG                                                       86TRKB     13
      DATA DZERO/0.0/                                                      86TRKB     14
      DATA MP2FLG/0/                                                       86TRKB     15
C                                                                          86TRKB     16
      DATA IBLANK/4H    /                                                  86TRKB     17
                                                                           86TRKB     18
C  INSTRUCTION TRKB-----TRACK BETATRON FUNCTIONS                           86TRKB     19
                                                                           86TRKB     20
C  T  TRKB  IR1 IR2//BLIN BET0 V0          IT1  IT2  ITBL IRAD S0  TH0     86TRKB     21
                                                                           86TRKB     22
C  RESERV  A TABLE FOR STORAGE OF BETA FUNCTIONS FROM POSITION IR1         86TRKB     23
C  THROUGH POSITION IR2 OF BET0.                                           86TRKB     24
C                                                                          86TRKB     25
C  TRACK BETA FUNCTIONS DEFINED BY [ BET0  IBET   //...]                   86TRKB     26
C  THROUGH BEAM LINE DEFINED BY [ BLIN  BML   //...] FROM IT1 TO IT2.      86TRKB     27
C                                                                          86TRKB     28
C  TRACK A PARTICLE VECTOR DEFINED BY [ V0  PVEC  //...]                   86TRKB     29
C  THROUGH THE BEAMLINE.                                                   86TRKB     30
C                                                                          86TRKB     31
C  BETA FUNCTION ARRAY IS STORED AT STORE(LQ3), LQ3=INFF(24,M),            86TRKB     32
C  WHERE B(I,IPOS+1) = QX,BX,AX,GX,X,DX, QY,BY,AY,GY,Y,DY, S,THETAX.       86TRKB     33
                                                                           86TRKB     34
C    ITBL = 1 --- INPUT BETAS FROM TRKB TABLE, AT IT1, EXCEPT              86TRKB     35
```

```
C                    ON 1ST CALL, AND/OR IF IT1=0, FROM IBET INPUT.
C     BET0 = /NAME/ --INITIAL BETAS FROM ARRAY/NAME/,FOR AN IBET,
C                     AND FROM BETA-FUNCTIONS OF /NAME/, IF A MATRIX.
C     IRAD = 1 --- OUTPUT PHASE ADVANCES IN RADIANS
C          = 0 ---                    IN UNITS OF 2PI
C
C   DEFAULTS ---
C
C     IF KA = KB = 0 (OR BLANK), A TABLE IS RESERVED SUFFICIENT TO STORE
C                                THE COMPLETE BEAM LINE.
C     IF KA = -1                 NO TABLE IS RESERVED, NO BETAS ARE STORE
C
C     IF KT1 = IT2 = 0 (OR BLANK), TRACKING IS DONE THROUGH COMPLETE BEAM
C

      RADFLG = .FALSE.
      IF (MINFLG.NE.4)  GO TO 23
      IF (MIFLG)  GO TO 152
      GO TO 4
 23   NOSTOR = .FALSE.
      TPI=4.*ACOS(DZERO)
      ELNAMEI=IBLANK
      KB=INFF(5,M)
      KA=INFF(4,M)
      LQ3 = INFF(24,M)
      NCOL = IDAT(M,5)
      IF (KA.GT.-1) GO TO 151
      NOSTOR = .TRUE.
      GO TO 152
 151  LTAB0 = LQ3 - NCOL*KA - 1
 152  MBML = MDAT(M,1)
      NDAT=INFF(17,MBML)
      CALL MIFILL(MBML,1,NDAT,NELS,MI)
      CALL MMM(M,NELS,MI)
      IF (MINFLG.EQ.4) GO TO 4
      IP1=IDAT(M,1)
      IP2=IDAT(M,2)
      ICALL=IDAT(M,6)
      IRAD = IDAT(M,4)
      SI = FDAT(M,1)
      TI = FDAT(M,2)
C   SET UP CONSTANTS FOR CALCULATING PSI IN DEGREES OR RADIANS
      IF (IRAD.EQ.1)  RADFLG=.TRUE.
      DNOM = TPI
      CCON = 1.
      IF (.NOT.RADFLG)   GO TO 153
      DNOM = 1.
      CCON = TPI
 153  CONTINUE
C  GET VECTOR
      NAMVEC = BDAT(M,3)
      VECFLAG = .FALSE.
      IF(NAMVEC.EQ.IBLANK) GO TO 160
      VECFLAG = .TRUE.
      MVEC = MDAT(M,3)
      CALL DATA(MVEC,1,1,6,VI)
 160  CONTINUE
```

```
      NAMBET=BDAT(M,2)
      NAME=INFF(2,M)

      ITABLE = IDAT(M,3)
      IF(ITABLE.EQ.1.AND..NOT.NOSTOR) GO TO 10
      GO TO 12

C 10  IF ((ICALL.EQ.0).OR.(IP1.EQ.0) ) GO TO 12
 10   CONTINUE
C  INITIAL BETAS FROM BETA FUNCTION TABLE OF TRKB INSTRUCTION,
C  EXCEPT THAT FOR FIRST CALL, OR IF IP1=0, THEY ARE TAKEN FROM IBET OR
      LOCBET=NCOL*(IP1-KA)+1
      MBET=M
      NTYP=8
      CALL DATA(MBET,NTYP,LOCBET+12,1,SI)
      CALL DATA(MBET,NTYP,LOCBET+13,1,TI)
      IF(VECFLAG) CALL DATA(MBET,NTYP,LOCBET+14,6,VI)
      GO TO 13

C  INITIAL BETAS FROM ARRAY SPECIFIED BY NAMBET
 12   LOCBET=1
      NTYP=1
      MBET=MDAT(M,2)

      NAMOP=INFF(1,MBET)

      IF(NAMOP.EQ.4HIBET) GO TO 13
C  INITIAL BETAS ARE BETAS OF MATRICES MBET.
      CALL BET(MBET,BXI,BYI,1)
      BXI(1)=0.
      BYI(1)=0.

      GO TO 16

 13   CALL DATA(MBET,NTYP,LOCBET  ,6,BXI)
      CALL DATA(MBET,NTYP,LOCBET+6,6,BYI)
C
 16   CONTINUE


 26   IF (IP1.EQ.0)  GO TO 9
      NELI = MI(IP1)
      ELNAMEI=MNAME(NELI)
  9   IF(IP1.LT.KA) NOSTOR=.TRUE.
      IF(IP2.GT.KB) NOSTOR=.TRUE.
      ICALL=1
      CALL STDAT(M,3,3,1,ICALL)


  4   CONTINUE

      S = SI
      TH = TI

      DO 3 I=1,6
      BX(I) = BXI(I)
```

```
      BY(I) = BYI(I)
      V(I) = VI(I)
    3 CONTINUE

      IF (NOPR) GO TO 7
      CALL DASH
      LNUM = 0
      IF (I2.GT.45)  WRITE (3,1004)
      IF(.NOT.VECFLAG) GO TO 406
      WRITE (3,1002)
      GO TO 407
  406 WRITE(3,1000)
  407 CONTINUE

      ELNAME=ELNAMEI
    7 CONTINUE
      I1 = IP1 + 1
      I2 = IP2

      IF (BX(4).EQ.0.) BX(4)=(1.+BX(3)*BX(3))/BX(2)
      IF (BY(4).EQ.0.)  BY(4)=(1.+BY(3)*BY(3))/BY(2)

C   LOOP OVER POSITIONS
   19 DO 1 IBS = I1,I2
      IPOS = IBS - 1
      IF(NOPR) GO TO 6
      IF(.NOT.VECFLAG) GO TO 190
      IF (MOD(LNUM,45).NE.0.OR.LNUM.EQ.0)  GO TO 191
      WRITE (3,1004)
      WRITE (3,1002)
      LNUM = 0
  191 IF (MOD(LNUM,5).EQ.0.AND.LNUM.NE.0)  WRITE (3,1010)
      WRITE(3,1003) IPOS,ELNAME,S,(BX(I),I=1,3),BX(5),BX(6),
     1              (BY(I),I=1,3),BY(5),BY(6),(V(I),I=1,4)
      LNUM = LNUM + 1
      GO TO 6
  190 IF (MOD(LNUM,45).NE.0.OR.LNUM.EQ.0)  GO TO 192
      WRITE (3,1004)
      WRITE (3,1000)
      LNUM = 0
  192 IF (MOD(LNUM,5).EQ.0.AND.LNUM.NE.0)  WRITE (3,1010)
      WRITE(3,1001) IPOS,ELNAME,S,(BX(I),I=1,3),BX(5),BX(6),
     1              (BY(I),I=1,3),BY(5),BY(6)
      LNUM = LNUM + 1
    6 NEL = MI(IBS)
      IF(NOSTOR) GO TO 8
      J=LTAB0+NCOL*IPOS
      DO 11 I=1,14
   11 STORE(J+I)=BETF(I)
      IF(.NOT.VECFLAG) GO TO 8
      DO 110 I=15,20
  110 STORE(J+I) = VF(I)
    8 CONTINUE

      ELNAME=MNAME(NEL)
      STOT = S
C
```

```
   18 CALL RXY(NEL,RX,RY,RW)

      IF(.NOT.VECFLAG) GO TO 180
      DO 181 I=1,6
      BX0(I)=BX(I)
      BY0(I)=BY(I)
  181 V0(I) = V(I)
      GO TO 182

  180 DO 2 I=1,6
      BX0(I)=BX(I)
    2 BY0(I)=BY(I)

  182 S = S + RW(1)
      TH = TH + RW(2)

      BX(2)=RX(1,1)*RX(1,1)*BX0(2)-2.*RX(1,1)*RX(1,2)*BX0(3)
     1         + RX(1,2)*RX(1,2)*BX0(4)

      BY(2)=RY(1,1)*RY(1,1)*BY0(2)-2.*RY(1,1)*RY(1,2)*BY0(3)
     1         + RY(1,2)*RY(1,2)*BY0(4)

      BX(3) = -RX(1,1)*RX(2,1)*BX0(2)
     1         + (RX(1,1)*RX(2,2)+RX(1,2)*RX(2,1))*BX0(3)
     2         - RX(1,2)*RX(2,2)*BX0(4)

      BY(3) = -RY(1,1)*RY(2,1)*BY0(2)
     1         + (RY(1,1)*RY(2,2)+RY(1,2)*RY(2,1))*BY0(3)
     2         - RY(1,2)*RY(2,2)*BY0(4)

      BX(4) = RX(2,1)*RX(2,1)*BX0(2) - 2.*RX(2,1)*RX(2,2)*BX0(3)
     1           + RX(2,2)*RX(2,2)*BX0(4)

      BY(4) = RY(2,1)*RY(2,1)*BY0(2) - 2.*RY(2,1)*RY(2,2)*BY0(3)
     1           + RY(2,2)*RY(2,2)*BY0(4)

      BX(5) = RX(1,1)*BX0(5) + RX(1,2)*BX0(6) + RX(1,3)

      BY(5) = RY(1,1)*BY0(5) + RY(1,2)*BY0(6) + RY(1,3)

      BX(6) = RX(2,1)*BX0(5) + RX(2,2)*BX0(6) + RX(2,3)

      BY(6) = RY(2,1)*BY0(5) + RY(2,2)*BY0(6) + RY(2,3)

      IF(NONU) GO TO 183

      BETAV=SQRT(BX(2)*BX0(2))
      BERAT=SQRT(BX0(2)/BX(2))
      SINE=RX(1,2)/BETAV
      COSINE=RX(1,1)*BERAT-BX0(3)*SINE
C    DNOM OR CCON SET TO TPI OR 1. DEPENDING ON DEGREES OR RADIANS.
      DNU = ATAN2(SINE,COSINE)/DNOM
      IF (DNU.LT.0)  DNU = DNU + CCON
      BX(1) = BX0(1) + DNU
      BETAV=SQRT(BY(2)*BY0(2))
      BERAT=SQRT(BY0(2)/BY(2))
      SINE=RY(1,2)/BETAV
```

```
          COSINE=RY(1,1)*BERAT-BY0(3)*SINE
          DNU = ATAN2(SINE,COSINE)/DNOM
          IF (DNU.LT.0)  DNU = DNU + CCON
          BY(1) = BY0(1) + DNU

    183   CONTINUE

          IF(.NOT.VECFLAG) GO TO 1
          CALL RTRV7(NEL,T,RW)
          CALL MXV7(T,V0,V)

    1     CONTINUE

    17    IPOS = I2
          MP2FLG = 0
          IF(NOSTOR) GO TO 5

          J = J + NCOL
          DO 14 I=1,14
    14    STORE(J+I)=BETF(I)
          IF(.NOT.VECFLAG) GO TO 5
          DO 140 I=15,20
    140   STORE(J+I) = VF(I)

    5     CONTINUE

          IF(NOPR) RETURN

          IF(.NOT.VECFLAG) GO TO 150

          WRITE(3,1003) IPOS,ELNAME,S,(BX(I),I=1,3),BX(5),BX(6),
         1              (BY(I),I=1,3),BY(5),BY(6),(V(I),I=1,4)
          GO TO 154
    150   WRITE(3,1001) IPOS,ELNAME,S,(BX(I),I=1,3),BX(5),BX(6),
         1              (BY(I),I=1,3),BY(5),BY(6)

    154   CALL DASH

          RETURN

    1000 FORMAT (" POS",11X,"S",7X,"QX",10X,"BX",8X,"AX",8X,"X",9X,
         1  "DX",14X,"QY",10X,"BY",8X,"AY",8X,"Y",9X,"DY")

    1001 FORMAT (I4,1X,A4,1X,F10.4,2(F10.6,F11.4,F11.6,F10.6,F10.6,6X))

    1002 FORMAT (" POS",10X,"S",7X,"QX",7X,"BX",7X,"AX",6X,"DX",6X,
         1  "DDX",6X,"QY",7X,"BY",7X,"AY",6X,"DY",6X,"DDY",4X,"X(MM)",
         2  1X,"DX(MR)",1X,"Y(MM)",1X,"DY(MR)")

    1003 FORMAT (I4,1X,A4,F10.4,2(F8.4,2F9.4,2F8.4,1X),2(F6.2,F7.3))
    1004 FORMAT (1H1)
    1010 FORMAT (1H )
          END
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
EQV/COMM	I	BMI1	NOT ALL ITEMS IN THIS COMMON BLOCK OCCUR IN LEVEL STATEMENTS.

```
      C ********************************TRKE********************************* 86TRKE      2
      C ENVELOPE TRACKING ROUTINE 9-19-85                                       86TRKE      3
      C                                                                         86TRKE      4
            SUBROUTINE TRKE(M)                                                  86TRKE      5
                                                                                BLANK       2
            LEVEL 2, STORE,INFF,IWORK                                           BLANK       3
            COMMON STORE(48000),IWORK(10)                                       86MARSIZ    1
            DIMENSION INFF(24,2000)                                             86MARSIZ    2
            EQUIVALENCE (INFF,STORE)                                            86MARSIZ    3
                                                                                BLANK       5
            COMMON/BMI1/MI1(16000)                                              NV3BMI1     1
            COMMON/NELS1/NELS1                                                  BMI1L       2
            LEVEL 2,MI1                                                         BMI1L       3
                                                                                BMI1L       4
      C                                                                         BMI1        3
            DIMENSION MI(1)                                                     86TRKE      8
            EQUIVALENCE (MI,MI1),(NELS,NELS1)                                   86TRKE      9
            COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,      CONTRL      2
           1          LDFLG,FIN                                                 CONTRL      3
            LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN        CONTRL      4
                                                                                CONTRL      5
            COMMON /MATCH3/NOPR,NONU,FLAG,IPR,IPP,MINFLG,MIFLG,MATFLG,GLOBAL    MATCH3      2
            LOGICAL FLAG,NOPR,NONU,MIFLG,MATFLG,GLOBAL                          MATCH3      3
            COMMON/CDERIV/IFLAG,MR,NR,DP,NV(11),DAT(20),UZ,STOT,                CDERIV      2
           1 DAT2(7),MP2FLG,NF,H0,LOCALPH,NUMALPH                               CDERIV      3
            DIMENSION BXI(9),BYI(9),V0(6),VI(6),VF(7),T(49)                     86TRKE     13
            DIMENSION BX(6),BY(6),BX0(6),BY0(6),RX(2,3),RY(2,3),RW(3),BETF(14)  86TRKE     14
            DIMENSION P(8), V(7)                                                86TRKE     15
            EQUIVALENCE(BX,BETF),(BY,BETF(7)), (S,BETF(13)),(TH,BETF(14))       86TRKE     16
            EQUIVALENCE (EPX,P(1)),(EPY,P(2)),(EPL,P(3)),(DPP,P(4)),(V,VF(7))   86TRKE     17
            INTEGER BDAT,ELNAMEI,ELNAME                                         86TRKE     18
            EXTERNAL DER2,DER3,DER4,DELQ                                        86TRKE     19
            LOGICAL NOSTOR,MPFLG,VECFLAG                                        86TRKE     20
            LOGICAL RADFLG                                                      86TRKE     21
            DATA DZERO/0.0/                                                     86TRKE     22
            DATA MP2FLG/0/                                                      86TRKE     23
      C                                                                         86TRKE     24
            DATA IBLANK/"    "/                                                 86TRKE     25
                                                                                86TRKE     26
      C  INSTRUCTION TRKE----TRACK BEAM ENVELOPES                                86TRKE     27
                                                                                86TRKE     28
      C  T  TRKE  IR1 IR2//BLIN BET0 V0   EPS    IT1  IT2  ITBL IADD EXCO       86TRKE     29
                                                                                86TRKE     30
      C  RESERV  A TABLE FOR STORAGE OF BETA FUNCTIONS FROM POSITION IR1        86TRKE     31
      C  THROUGH POSITION IR2 OF BET0.                                          86TRKE     32
      C                                                                         86TRKE     33
      C  TRACK BETA FUNCTIONS DEFINED BY [ BET0  IBET   //...]                  86TRKE     34
      C  THROUGH BEAM LINE DEFINED BY [ BLIN  BML   //...] FROM IT1 TO IT2,     86TRKE     35
      C  CALCULATE AND DISPLAY BEAM ENVELOPES.                                  86TRKE     36
      C  THE EMITTANCES ARE DEFINED BY [ EPS  BVAL   //...].                    86TRKE     37
      C                                                                         86TRKE     38
      C  TRACK A PARTICLE VECTOR DEFINED BY [ V0  PVEC  //...]                  86TRKE     39
      C  THROUGH THE BEAMLINE.                                                  86TRKE     40
      C                                                                         86TRKE     41
      C  BETA FUNCTION ARRAY IS STORED AT STORE(LQ3), LQ3=INFF(24,M),           86TRKE     42
      C  WHERE B(I,IPOS+1) = QX,BX,AX,GX,X,DX, QY,BY,AY,GY,Y,DY, S,THETAX.      86TRKE     43
                                                                                86TRKE     44
```

```
C    ITBL = 1 ---- INPUT BETAS FROM TRKB TABLE, AT IT1, EXCEPT
C                       ON 1ST CALL, AND/OR IF IT1=0, FROM IBET INPUT.
C    BET0 = /NAME/ ---INITIAL BETAS FROM ARRAY/NAME/,FOR AN IBET,
C                         AND FROM BETA-FUNCTIONS OF /NAME/, IF A MATRIX.
C    IADD = 0 ---- ADD XB AND XP IN QUADRATURE, ETC.
C    IADD = 1 ---- ADD XB AND XP ALGEBRAICALLY, ETC.
C
C  DEFAULTS ---
C
C    IF KA = KB = 0 (OR BLANK), A TABLE IS RESERVED SUFFICIENT TO STORE
C                                   THE COMPLETE BEAM LINE.
C    IF KA = -1                     NO TABLE IS RESERVED, NO BETAS ARE STORE
C
C    IF KT1 = IT2 = 0 (OR BLANK), TRACKING IS DONE THROUGH COMPLETE BEAM
C

      RADFLG = .FALSE.
      IF (MINFLG.NE.4)  GO TO 23
      IF (MIFLG)  GO TO 152
      GO TO 4
   23 NOSTOR = .FALSE.
      TPI=4.*ACOS(DZERO)
      ELNAMEI=IBLANK
      KB=INFF(5,M)
      KA=INFF(4,M)
      LQ3 = INFF(24,M)
      NCOL = IDAT(M,5)
      IF (KA.GT.-1) GO TO 151
      NOSTOR = .TRUE.
      GO TO 152
  151 LTAB0 = LQ3 - NCOL*KA - 1
  152 MBML = MDAT(M,1)

      NDAT=INFF(17,MBML)
      CALL MIFILL(MBML,1,NDAT,NELS,MI)
      CALL MMM(M,NELS,MI)
      IF (MINFLG.EQ.4) GO TO 4
      IP1=IDAT(M,1)
      IP2=IDAT(M,2)
      ICALL=IDAT(M,6)
      IADD = IDAT(M,4)
      SI = 0.0
      TI = 0.0
      IG = 1
      IF(IADD.EQ.0) IG = 2
      DNOM = TPI
      CCON = 1.

C  GET VECTOR
      NAMVEC = BDAT(M,3)
      VECFLAG = .FALSE.
      IF(NAMVEC.EQ.IBLANK) GO TO 160
      VECFLAG = .TRUE.
      MVEC = MDAT(M,3)
      CALL DATA(MVEC,1,1,6,VI)
  160 CONTINUE
```

```
      MBVAL = MDAT(M,4)
      CALL DATA(MBVAL,5,1,4,P)
      NAMBET=BDAT(M,2)
      NAME=INFF(2,M)

      ITABLE = IDAT(M,3)
      IF(ITABLE.EQ.1.AND..NOT.NOSTOR) GO TO 10
      GO TO 12

C 10   IF ((ICALL.EQ.0).OR.(IP1.EQ.0) ) GO TO 12
 10   CONTINUE
C  INITIAL BETAS FROM BETA FUNCTION TABLE OF TRKB INSTRUCTION,
C  EXCEPT THAT FOR FIRST CALL, OR IF IP1=0, THEY ARE TAKEN FROM IBET OR
      LOCBET=NCOL*(IP1-KA)+1
      MBET=M
      NTYP=8
      CALL DATA(MBET,NTYP,LOCBET+12,1,SI)
      CALL DATA(MBET,NTYP,LOCBET+13,1,TI)
      IF(VECFLAG) CALL DATA(MBET,NTYP,LOCBET+14,6,VI)
      GO TO 13

C  INITIAL BETAS FROM ARRAY SPECIFIED BY NAMBET
 12   LOCBET=1
      NTYP=1
      MBET=MDAT(M,2)

      NAMOP=INFF(1,MBET)

      IF(NAMOP.EQ.4HIBET) GO TO 13
C  INITIAL BETAS ARE BETAS OF MATRICES MBET.
      CALL BET(MBET,BXI,BYI,1)
      BXI(1)=0.
      BYI(1)=0.

      GO TO 16

 13   CALL DATA(MBET,NTYP,LOCBET  ,6,BXI)
      CALL DATA(MBET,NTYP,LOCBET+6,6,BYI)
C
 16   CONTINUE


C  GET EMITTANCES
      EPXCO = FDAT(M,1)
      EPYCO = FDAT(M,2)
      RATX = SQRT(EPXCO/EPX)
      RATY = SQRT(EPYCO/EPY)

 26   IF (IP1.EQ.0)  GO TO 9
      NELI = MI(IP1)
      ELNAMEI=MNAME(NELI)
 9    IF(IP1.LT.KA) NOSTOR=.TRUE.
      IF(IP2.GT.KB) NOSTOR=.TRUE.
      ICALL=1
      CALL STDAT(M,3,3,1,ICALL)

```

```
    4     CONTINUE

          S = SI
          TH = TI

          DO 3 I=1,6
          BX(I) = BXI(I)
          BY(I) = BYI(I)
          V(I) = VI(I)
    3     CONTINUE

          I1 = IP1 + 1
          I2 = IP2
          IF (NOPR) GO TO 7
          LNUM = 0
          IF (I2.GT.40) WRITE (3,4004)
          CALL DASH
          WRITE(3,4001) P(1),EPXCO,P(3),P(2),EPYCO,P(4)
          GO TO (404,405) IG
  404     WRITE (3,4006)
          GO TO 406
  405     WRITE (3,4008)
          IF(.NOT.VECFLAG) GO TO 406
          WRITE (3,4011)
          GO TO 407
  406     WRITE(3,4010)
  407     CONTINUE

          ELNAME=ELNAMEI
    7     CONTINUE

          IF (BX(4).EQ.0.) BX(4)=(1.+BX(3)*BX(3))/BX(2)
          IF (BY(4).EQ.0.)  BY(4)=(1.+BY(3)*BY(3))/BY(2)

C                      COMPUTE AND PRINT BEAM ENVELOPES
          X=SQRT(EPX*BX(2))
          Y=SQRT(EPY*BY(2))
          XP=BX(5)*DPP
          YP=BY(5)*DPP
          XCO=RATX*X
          YCO=RATY*Y
          XPR=SQRT(EPX*BX(4))
          YPR=SQRT(EPY*BY(4))
          XPPR=BX(6)*DPP
          YPPR=BY(6)*DPP
          XCOPR=RATX*XPR
          YCOPR=RATY*YPR
          GO TO (4116,4117) IG
 4116     XTOT = X + ABS(XP)
          XPRTOT = XPR + ABS(XPPR)
          YTOT = Y + ABS(YP)
          YPRTOT = YPR + ABS(YPPR)
          GO TO 4118
 4117     CONTINUE
          XTOT = SQRT( X * X + XP * XP )
          XPRTOT = SQRT( XPR*XPR + XPPR*XPPR)
          YTOT = SQRT(Y*Y + YP*YP)
```

```
            YPRTOT = SQRT(YPR*YPR + YPPR*YPPR)
      4118 CONTINUE
            XTOT=XTOT+XCO
            YTOT=YTOT+YCO
            XPRTOT=XPRTOT+XCOPR
            YPRTOT=YPRTOT+YCOPR

      C   LOOP OVER POSITIONS
       19   DO 1 IBS = I1,I2
            IPOS = IBS - 1
            IF(NOPR) GO TO 6
            IF(.NOT.VECFLAG) GO TO 190
            IF (MOD(LNUM,40).NE.0.OR.LNUM.EQ.0)  GO TO 191
            WRITE (3,4004)
            WRITE (3,4011)
            LNUM = 0
       191  IF (MOD(LNUM,40).NE.0.OR.LNUM.NE.0)  WRITE (3,4040)
            WRITE(3,4012) IPOS,ELNAME,S,X,XP,XCO,XTOT,Y,YP,YCO,YTOT,
          1                XPRTOT,YPRTOT,(V(I),I=1,4)
            LNUM = LNUM + 1
            GO TO 6
       190  IF (MOD(LNUM,40).NE.0.OR.LNUM.EQ.0)  GO TO 192
            WRITE (3,4004)
            WRITE (3,4010)
            LNUM = 0
       192  IF (MOD(LNUM,5).EQ.0.AND.LNUM.NE.0)  WRITE (3,4040)
            WRITE(3,4012) IPOS,ELNAME,S,X,XP,XCO,XTOT,Y,YP,YCO,YTOT,
          1                XPR,XPPR,XPRTOT,YPR,YPPR,YPRTOT
            LNUM = LNUM + 1
       6    NEL = MI(IBS)
            IF(NOSTOR) GO TO 8
            J=LTAB0+NCOL*IPOS
            DO 11 I=1,14
       11   STORE(J+I)=BETF(I)
            IF(.NOT.VECFLAG) GO TO 8
            DO 110 I=15,20
       110  STORE(J+I) = VF(I)
       8    CONTINUE

            ELNAME=MNAME(NEL)
            STOT = S
      C
       18   CALL RXY(NEL,RX,RY,RW)

            IF(.NOT.VECFLAG) GO TO 180
            DO 181 I=1,6
            BX0(I)=BX(I)
            BY0(I)=BY(I)
       181  V0(I) = V(I)
            GO TO 182

       180  DO 2 I=1,6
            BX0(I)=BX(I)
       2    BY0(I)=BY(I)

       182  S = S + RW(1)
            TH = TH + RW(2)
```

```
      BX(2)=RX(1,1)*RX(1,1)*BX0(2)-2.*RX(1,1)*RX(1,2)*BX0(3)
     1        + RX(1,2)*RX(1,2)*BX0(4)

      BY(2)=RY(1,1)*RY(1,1)*BY0(2)-2.*RY(1,1)*RY(1,2)*BY0(3)
     1        + RY(1,2)*RY(1,2)*BY0(4)

      BX(3) = -RX(1,1)*RX(2,1)*BX0(2)
     1        + (RX(1,1)*RX(2,2)+RX(1,2)*RX(2,1))*BX0(3)
     2        - RX(1,2)*RX(2,2)*BX0(4)

      BY(3) = -RY(1,1)*RY(2,1)*BY0(2)
     1        + (RY(1,1)*RY(2,2)+RY(1,2)*RY(2,1))*BY0(3)
     2        - RY(1,2)*RY(2,2)*BY0(4)

      BX(4) = RX(2,1)*RX(2,1)*BX0(2) - 2.*RX(2,1)*RX(2,2)*BX0(3)
     1          + RX(2,2)*RX(2,2)*BX0(4)

      BY(4) = RY(2,1)*RY(2,1)*BY0(2) - 2.*RY(2,1)*RY(2,2)*BY0(3)
     1          + RY(2,2)*RY(2,2)*BY0(4)

      BX(5) = RX(1,1)*BX0(5) + RX(1,2)*BX0(6) + RX(1,3)

      BY(5) = RY(1,1)*BY0(5) + RY(1,2)*BY0(6) + RY(1,3)

      BX(6) = RX(2,1)*BX0(5) + RX(2,2)*BX0(6) + RX(2,3)

      BY(6) = RY(2,1)*BY0(5) + RY(2,2)*BY0(6) + RY(2,3)

      IF(NONU) GO TO 183

      BETAV=SQRT(BX(2)*BX0(2))
      BERAT=SQRT(BX0(2)/BX(2))
      SINE=RX(1,2)/BETAV
      COSINE=RX(1,1)*BERAT-BX0(3)*SINE
C   DNOM OR CCON SET TO TPI OR 1. DEPENDING ON DEGREES OR RADIANS.
      DNU = ATAN2(SINE,COSINE)/DNOM
      IF (DNU.LT.0)  DNU = DNU + CCON
      BX(1) = BX0(1) + DNU
      BETAV=SQRT(BY(2)*BY0(2))
      BERAT=SQRT(BY0(2)/BY(2))
      SINE=RY(1,2)/BETAV
      COSINE=RY(1,1)*BERAT-BY0(3)*SINE
      DNU = ATAN2(SINE,COSINE)/DNOM
      IF (DNU.LT.0)  DNU = DNU + CCON
      BY(1) = BY0(1) + DNU

  183 CONTINUE

      IF(.NOT.VECFLAG) GO TO 184
      CALL RTRV7(NEL,T,RW)
      CALL MXV7(T,V0,V)

  184 CONTINUE

C                      COMPUTE AND PRINT BEAM ENVELOPES
      X=SQRT(EPX*BX(2))
```

```
      Y=SQRT(EPY*BY(2))
      XP=BX(5)*DPP
      YP=BY(5)*DPP
      XCO=RATX*X
      YCO=RATY*Y
      XPR=SQRT(EPX*BX(4))
      YPR=SQRT(EPY*BY(4))
      XPPR=BX(6)*DPP
      YPPR=BY(6)*DPP
      XCOPR=RATX*XPR
      YCOPR=RATY*YPR
      GO TO (5116,5117) IG
 5116 XTOT = X + ABS(XP)
      XPRTOT = XPR + ABS(XPPR)
      YTOT = Y + ABS(YP)
      YPRTOT = YPR + ABS(YPPR)
      GO TO 5118
 5117 CONTINUE
      XTOT = SQRT( X * X + XP * XP )
      XPRTOT = SQRT( XPR*XPR + XPPR*XPPR)
      YTOT = SQRT(Y*Y + YP*YP)
      YPRTOT = SQRT(YPR*YPR + YPPR*YPPR)
 5118 CONTINUE
      XTOT=XTOT+XCO
      YTOT=YTOT+YCO
      XPRTOT=XPRTOT+XCOPR
      YPRTOT=YPRTOT+YCOPR


 1    CONTINUE

 17   IPOS = I2
      MP2FLG = 0
      IF(NOSTOR) GO TO 5

      J = J + NCOL
      DO 14 I=1,14
 14   STORE(J+I)=BETF(I)
      IF(.NOT.VECFLAG) GO TO 5
      DO 140 I=15,20
 140  STORE(J+I) = VF(I)

 5    CONTINUE

      IF(NOPR) RETURN

      IF(.NOT.VECFLAG) GO TO 150

      WRITE(3,4012) IPOS,ELNAME,S,X,XP,XCO,XTOT,Y,YP,YCO,YTOT,
     1              XPRTOT,YPRTOT,(V(I),I=1,4)
      GO TO 153
 150  WRITE(3,4012) IPOS,ELNAME,S,X,XP,XCO,XTOT,Y,YP,YCO,YTOT,
     1              XPR,XPPR,XPRTOT,YPR,YPPR,YPRTOT
 153  CALL DASH

      RETURN
```

```
400         4001   FORMAT (3X,"BEAM ENVELOPES (MM,MRAD)"/                       86TRKE   387
               1   5X,"EMITTANCES (MM-MRAD) ---- EPSX = ",F10.6,5X,"EPSXCO = ",  86TRKE   388
               2   F10.6,5X,"EPSL = ",F13.6/30X,"EPSY = ",F10.6,5X,"EPSYCO = ",  86TRKE   389
               3   F10.6,5X,"SIGP = ",F10.6," (0/00)")                           86TRKE   390
                                                                                  86TRKE   391
405         4002 FORMAT(1H+,99X,"DISPLACEMENT = ",F5.2,1H*,"SIGMA")              86TRKE   392
            4006 FORMAT (5X,"SIGX = XB + XP + XCO,   ETC."/)                     86TRKE   393
            4008 FORMAT (5X,"SIGX = SQRT(XB*XB + XP*XP) + XCO,   ETC."/)         86TRKE   394
                                                                                  86TRKE   395
            4010 FORMAT(" POS",10X,"S",6X,"XB",6X,"XP",5X,"XCO",4X,"SIGX",3X,    86TRKE   396
410            1        "YB",6X,"YP",5X,"YCO",4X,"SIGY",4X,                      86TRKE   397
               2        "DXB",5X,"DXP",5X,"DSIGX",3X,"DYB",5X,"DYP",5X,"DSIGY")  86TRKE   398
                                                                                  86TRKE   399
            4011 FORMAT(" POS",10X,"S",6X,"XB",6X,"XP",5X,"XCO",4X,"SIGX",3X,    86TRKE   400
               1        "YB",6X,"YP",5X,"YCO",4X,"SIGY",4X,"DSIGX",3X,"DSIGY",   86TRKE   401
415            2        3X,"XCEN",4X,"DXCEN",3X,"YCEN",4X,"DYCEN")               86TRKE   402
                                                                                  86TRKE   403
            4012 FORMAT(I4,1X,A4,F10.4,2(F7.3,F8.3,2F7.3),6F8.3)                 86TRKE   404
            4004  FORMAT (1H1)                                                    86TRKE   405
            4040 FORMAT (1H )                                                     86TRKE   406
420              END                                                              86TRKE   407
```

```
CARD NR. SEVERITY  DETAILS   DIAGNOSIS OF PROBLEM

EQV/COMM    I      BMI1     NOT ALL ITEMS IN THIS COMMON BLOCK OCCUR IN LEVEL STATEMENTS.
    190     I               AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT
    219     I               AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT
    354     I               AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT
```

```
C ********************************TRKM********************************** 86TRKM     2
C MAP-TYPE ELEMENT TRACKING ROUTINE 9-19-85                               86TRKM     3
C                                                                         86TRKM     4
      SUBROUTINE TRKM(M)                                                  86TRKM     5
                                                                          BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                           BLANK      3
      COMMON STORE(48000),IWORK(10)                                       86MARSIZ   1
      DIMENSION INFF(24,2000)                                             86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                            86MARSIZ   3
                                                                          BLANK      5
      COMMON/BMI1/MI1(16000)                                              NV3BMI1    1
      COMMON/NELS1/NELS1                                                  BMI1L      2
      LEVEL 2,MI1                                                         BMI1L      3
                                                                          BMI1L      4
C                                                                         BMI1       3
      DIMENSION MI(1)                                                     86TRKM     8
      EQUIVALENCE (MI,MI1),(NELS,NELS1)                                   86TRKM     9
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,      CONTRL     2
     1          LDFLG,FIN                                                 CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN        CONTRL     4
                                                                          CONTRL     5
      COMMON /MATCH3/NOPR,NONU,FLAG,IPR,IPP,MINFLG,MIFLG,MATFLG,GLOBAL    MATCH3     2
      LOGICAL FLAG,NOPR,NONU,MIFLG,MATFLG,GLOBAL                          MATCH3     3
      COMMON/CDERIV/IFLAG,MR,NR,DP,NV(11),DAT(20),UZ,STOT,                CDERIV     2
     1 DAT2(7),MP2FLG,NF,H0,LOCALPH,NUMALPH                               CDERIV     3
      DIMENSION BXI(9),BYI(9),V0(6),VI(6),VF(7),T(49)                     86TRKM    13
      DIMENSION BX(6),BY(6),BX0(6),BY0(6),RX(2,3),RY(2,3),RW(3),BETF(14)  86TRKM    14
      DIMENSION P(8), V(7)                                                86TRKM    15
      EQUIVALENCE(BX,BETF),(BY,BETF(7)), (S,BETF(13)),(TH,BETF(14))       86TRKM    16
      EQUIVALENCE (V,VF(7))                                               86TRKM    17
      INTEGER BDAT,ELNAMEI,ELNAME                                         86TRKM    18
      EXTERNAL DER2,DER3,DER4,DELQ                                        86TRKM    19
      LOGICAL NOSTOR,MPFLG,VECFLAG                                        86TRKM    20
      LOGICAL RADFLG                                                      86TRKM    21
      DATA DZERO/0.0/                                                     86TRKM    22
      DATA MP2FLG/0/                                                      86TRKM    23
C                                                                         86TRKM    24
      DATA IBLANK/"    "/                                                 86TRKM    25
                                                                          86TRKM    26
C  INSTRUCTION TRKM---TRACK THROUGH ELEMENTS DEFINED BY MAP STATEMENTS    86TRKM    27
                                                                          86TRKM    28
C  T  TRKM  IR1 IR2//BLIN BET0 MAPK PAR     IT1  IT2  ITBL IRAD HSTEP     86TRKM    29
                                                                          86TRKM    30
C  RESERV  A TABLE FOR STORAGE OF BETA FUNCTIONS FROM POSITION IR1        86TRKM    31
C  THROUGH POSITION IR2 OF BET0.                                          86TRKM    32
C                                                                         86TRKM    33
C  TRACK BETA FUNCTIONS DEFINED BY [ BET0  IBET   //...]                  86TRKM    34
C  THROUGH BEAM LINE DEFINED BY [ BLIN  BML   //...] FROM IT1 TO IT2.     86TRKM    35
C                                                                         86TRKM    36
C  TRACK A PARTICLE VECTOR DEFINED BY [ V0  PVEC  //...]                  86TRKM    37
C  THROUGH THE BEAMLINE.                                                  86TRKM    38
C                                                                         86TRKM    39
C  BETA FUNCTION ARRAY IS STORED AT STORE(LQ3), LQ3=INFF(24,M),           86TRKM    40
C  WHERE B(I,IPOS+1) = QX,BX,AX,GX,X,DX, QY,BY,AY,GY,Y,DY, S,THETAX.      86TRKM    41
                                                                          86TRKM    42
C    ITBL = 1 --- INPUT BETAS FROM TRKB TABLE, AT IT1, EXCEPT             86TRKM    43
C                      ON 1ST CALL, AND/OR IF IT1=0, FROM IBET INPUT.     86TRKM    44
```

```
C    BET0 = /NAME/ --INITIAL BETAS FROM ARRAY/NAME/,FOR AN IBET,
C                     AND FROM BETA-FUNCTIONS OF /NAME/, IF A MATRIX.
C    IRAD = 1 --- OUTPUT PHASE ADVANCES IN RADIANS
C         = 0 ---                          IN UNITS OF 2PI
C
C  DEFAULTS ---
C
C    IF KA = KB = 0 (OR BLANK), A TABLE IS RESERVED SUFFICIENT TO STORE
C                               THE COMPLETE BEAM LINE.
C    IF KA = -1                 NO TABLE IS RESERVED, NO BETAS ARE STORE
C
C    IF KT1 = IT2 = 0 (OR BLANK), TRACKING IS DONE THROUGH COMPLETE BEAM
C

      RADFLG = .FALSE.
      IF (MINFLG.NE.4)  GO TO 23
      IF (MIFLG)  GO TO 152
      GO TO 4
   23 NOSTOR = .FALSE.
      TPI=4.*ACOS(DZERO)
      ELNAMEI=IBLANK
      KB=INFF(5,M)
      KA=INFF(4,M)
      LQ3 = INFF(24,M)
      NCOL = IDAT(M,5)
      IF (KA.GT.-1) GO TO 151
      NOSTOR = .TRUE.
      GO TO 152
  151 LTAB0 = LQ3 - NCOL*KA - 1
  152 MBML = MDAT(M,1)

      NDAT=INFF(17,MBML)
      CALL MIFILL(MBML,1,NDAT,NELS,MI)
      CALL MMM(M,NELS,MI)
      IF (MINFLG.EQ.4) GO TO 4
      IP1=IDAT(M,1)
      IP2=IDAT(M,2)
      ICALL=IDAT(M,6)
      SI = FDAT(M,2)
      TI = 0.
      H0 = FDAT(M,1)
      DAT2(7) = H0
      MAPK = BDAT(M,3)
      IRAD = IDAT(M,4)
C   SET UP CONSTANTS FOR CALCULATING PSI IN DEGREES OR RADIANS
      IF (IRAD.EQ.1)  RADFLG=.TRUE.
      DNOM = TPI
      CCON = 1.
      IF (.NOT.RADFLG)   GO TO 153
      DNOM = 1.
      CCON = TPI
  153 CONTINUE

      NAMBET=BDAT(M,2)
      NAMVEC = BDAT(M,3)
      NAMPAR = BDAT(M,4)
      NAME=INFF(2,M)
```

```
      ITABLE = IDAT(M,3)
      IF(ITABLE.EQ.1.AND..NOT.NOSTOR) GO TO 10
      GO TO 12

C 10  IF ((ICALL.EQ.0).OR.(IP1.EQ.0) ) GO TO 12
 10   CONTINUE
C  INITIAL BETAS FROM BETA FUNCTION TABLE OF TRKB INSTRUCTION,
C  EXCEPT THAT FOR FIRST CALL, OR IF IP1=0, THEY ARE TAKEN FROM IBET OR
      LOCBET=NCOL*(IP1-KA)+1
      MBET=M
      NTYP=8
      CALL DATA(MBET,NTYP,LOCBET+12,1,SI)
      CALL DATA(MBET,NTYP,LOCBET+13,1,TI)
      GO TO 13

C  INITIAL BETAS FROM ARRAY SPECIFIED BY NAMBET
 12   LOCBET=1
      NTYP=1
      MBET=MDAT(M,2)

      NAMOP=INFF(1,MBET)

      IF(NAMOP.EQ.4HIBET) GO TO 13
C  INITIAL BETAS ARE BETAS OF MATRICES MBET.
      CALL BET(MBET,BXI,BYI,1)
      BXI(1)=0.
      BYI(1)=0.

      GO TO 16

 13   CALL DATA(MBET,NTYP,LOCBET  ,6,BXI)
      CALL DATA(MBET,NTYP,LOCBET+6,6,BYI)

 16   CONTINUE

      IF (MAPK.EQ.4HMAP2) MP2FLG=1
      IF (MAPK.NE.3HELQ) GO TO 34
      NUM = 7
      GO TO 35
 34   DECODE(10,1002,MAPK) NUM
 1002 FORMAT(3X,I1)
 35   IF (NAMPAR.EQ.IBLANK) GO TO 26
      MPAR = MDAT(M,4)
      CALL DATA(MPAR,1,1,6,DAT2)

 26   IF (IP1.EQ.0)  GO TO 9
      NELI = MI(IP1)
      ELNAMEI=MNAME(NELI)
 9    IF(IP1.LT.KA) NOSTOR=.TRUE.
      IF(IP2.GT.KB) NOSTOR=.TRUE.
      ICALL=1
      CALL STDAT(M,3,3,1,ICALL)


 4    CONTINUE
      S = SI
```

```
      TH = TI

      DO 3 I=1,6
      BX(I) = BXI(I)
      BY(I) = BYI(I)
    3 CONTINUE
      I1 = IP1 + 1
      I2 = IP2

      IF (NOPR) GO TO 7
      IF (I2.GT.45)  WRITE (3,1004)
      LNUM = 0
      CALL DASH
  406 WRITE(3,1000)

      ELNAME=ELNAMEI
    7 CONTINUE

C  ALL ELEMENTS ARE DEFINED BY MAPS
   21 NR = 1
      IFLAG = 1
      NV(1) = 1
      S = 0.
      TH = 0.
      IF (NUM-3)  27,28,29

C  MAP2
   27 MR = 8
      GO TO 19

C  MAP3 -- TRANSVERSE AND LONGITUDINAL ENVELOPES
   28 MR = 6
      GO TO 19

C  MAP4 -- TRANSVERSE ENVELOPES
   29 MR = 4

C   LOOP OVER POSITIONS
   19 DO 1 IBS = I1,I2
      IPOS = IBS - 1
      IF(NOPR) GO TO 6

  190 IF (MOD(LNUM,45).NE.0.OR.LNUM.EQ.0)  GO TO 191
      WRITE (3,1004)
      WRITE (3,1000)
      LNUM = 0
  191 IF (MOD(LNUM,5).EQ.0.AND.LNUM.NE.0)  WRITE (3,1003)
      WRITE(3,1001) IPOS,ELNAME,S,(BX(I),I=1,3),BX(5),BX(6),
     1              (BY(I),I=1,3),BY(5),BY(6)
      LNUM = LNUM + 1
    6 NEL = MI(IBS)
      IF(NOSTOR) GO TO 8
      J=LTAB0+NCOL*IPOS
      DO 11 I=1,14
   11 STORE(J+I)=BETF(I)
    8 CONTINUE
```

```
      ELNAME=MNAME(NEL)
      STOT = S
C
      MAPK2=INFF(1,NEL)
      IF (MAPK2.EQ.MAPK) GO TO 20
      MAPK=MAPK2
      IF (MAPK.EQ.3HELQ) GO TO 22
      NUM=4
      GO TO 32
   22 NUM=7
      GO TO 36
   20 IF (NUM.EQ.7)  GO TO 36
      IF (NUM-3)  30,31,32

C   MAP2
  30  P(1) = BX(2)
      P(2) = BX(3)
      P(3) = BX(5)
      P(4) = BX(6)
      P(5) = BY(2)
      P(6) = BY(3)
      P(7) = BY(5)
      P(8) = BY(6)
      CALL DIFEQ(NEL,P,DER2)
      BX(2) = P(1)
      BX(3) = P(2)
      BX(5) = P(3)
      BX(6) = P(4)
      BY(2) = P(5)
      BY(3) = P(6)
      BY(5) = P(7)
      BY(6) = P(8)
      GO TO 33

C   MAP3
  31  P(1) = BX(2)
      P(2) = BX(3)
      P(3) = BY(2)
      P(4) = BY(3)
      P(5) = BX(5)
      P(6) = BX(6)
      P(7) = 0.
      CALL DIFEQ(NEL,P,DER3)
      BX(2) = P(1)
      BX(3) = P(2)
      BY(2) = P(3)
      BY(3) = P(4)
      BX(5) = P(5)
      BX(6) = P(6)
      GO TO 33

C   ELQ
  36  P(1) = BX(2)
      P(2) = BX(3)
      P(3) = BY(2)
      P(4) = BY(3)
      CALL DIFEQ(NEL,P,DELQ)
```

```
      GO TO 37

C  MAP4
 32   P(1) = BX(2)
      P(2) = BX(3)
      P(3) = BY(2)
      P(4) = BY(3)
      P(5) = 0.
      P(6) = 0.
      CALL DIFEQ(NEL,P,DER4)
 37   BX(2) = P(1)
      BX(3) = P(2)
      BY(2) = P(3)
      BY(3) = P(4)
 33   S = S + DAT(1)

 1    CONTINUE

 17   IPOS = I2
      MP2FLG = 0
      IF(NOSTOR) GO TO 5

      J = J + NCOL
      DO 14 I=1,14
 14   STORE(J+I)=BETF(I)

 5    CONTINUE

      IF(NOPR) RETURN

 150  WRITE(3,1001) IPOS,ELNAME,S,(BX(I),I=1,3),BX(5),BX(6),
     1              (BY(I),I=1,3),BY(5),BY(6)

 154  CALL DASH

      RETURN

 1000 FORMAT (" POS",11X,"S",7X,"QX",10X,"BX",8X,"AX",8X,"X",9X,
     1  "DX",14X,"QY",10X,"BY",8X,"AY",8X,"Y",9X,"DY")

 1001 FORMAT (I4,1X,A4,1X,F10.4,2(F10.6,F11.4,F11.6,F10.6,F10.6,6X))
 1004 FORMAT (1H1)
 1003 FORMAT (1H )
      END
```

```
CARD NR. SEVERITY  DETAILS    DIAGNOSIS OF PROBLEM

EQV/COMM    I      BMI1       NOT ALL ITEMS IN THIS COMMON BLOCK OCCUR IN LEVEL STATEMENTS.
```

```
                                                                         TRKCN      2
      SUBROUTINE TRKCN ( MFXPT, NEL, P, NP, IOP )                        TRKCN      3
                                                                         TRKCN      4
C  A ROUTINE TO CONTROL TRACKING THROUGH LINEAR-,SINGLE NON-LINEAR-,     TRKCN      5
C  OR PRD-TYPE PRODUCTS OF TRANSFORMATIONS.                              TRKCN      6
C  MFXPT = INDEX OF FXPT INSTRUCTION.  HERE IT IS USED TO REFERENCE      TRKCN      7
C  THE 7X7 MATRIX REPRESENTING THE LINEARIZED BEAM SYSTEM RELATIVE TO    TRKCN      8
C  THE REFERENCE RAY PP.  ON EACH CALL TO TRKCN,  MFXPT = NEL * MFXPT.   TRKCN      9
C  NEL = INDEX OF ELEMENT OF BEAM SYSTEM                                 TRKCN     10
C  PP = PARTICLE VECTOR                                                  TRKCN     11
C  NP = PARTICLE INDEX                                                   TRKCN     12
                                                                         TRKCN     13
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL    DIM        2
                                                                         DIM        3
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,     CONTRL     2
     1          LDFLG,FIN                                                CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN       CONTRL     4
                                                                         CONTRL     5
                                                                         BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                          BLANK      3
      COMMON STORE(48000),IWORK(10)                                      86MARSIZ   1
      DIMENSION INFF(24,2000)                                            86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                           86MARSIZ   3
                                                                         BLANK      5
      COMMON/ MQ/ ML   $ LEVEL 2,ML                                      TRKCNL     1
                                                                         TRKCNL     2
      EXTERNAL DER1,DER2,DER4,DER5,DER6,DER7                             TRKCN     17
      EXTERNAL DER3                                                      TRKCN     18
      DIMENSION P(7,1),T(49),PAR(25),ML(2)                               TRKCN     19
      DIMENSION RW(3)                                                    TRKCN     20
                                                                         TRKCN     21
      INTEGER RET,OPNAME,HSXTP,HMOVE,HNPOL                               TRKCN     22
      DATA RW,HSXTP,HMOVE/3*0.0,4HSXTP,4HMOVE/                           TRKCN     23
      DATA HNPOL/4HNPOL/                                                 TRKCN     24
                                                                         TRKCN     25
      NF=0                                                               TRKCN     26
      IF (IOP.NE.1) NF=M7END-5                                           TRKCN     27
      NN = NEL                                                           TRKCN     28
      KIND = INFF(20,NN)                                                 TRKCN     29
      IF (KIND.EQ.4HEDRF) GO TO 35                                       TRKCN     30
      IF (KIND.EQ.4HDKE ) GO TO 36                                       TRKCN     31
      IF (KIND.EQ.4HSOL ) GO TO 37                                       TRKCN     32
      IF (KIND.EQ.3HMAP) GO TO 20                                        TRKCN     33
      IF (KIND.NE.3HPRD) GO TO 30                                        TRKCN     34
C  PRD-TYPE PRODUCT                                                      TRKCN     35
      K = INFF(4,NN)                                                     TRKCN     36
      ASSIGN 21 TO RET                                                   TRKCN     37
      I = 1                                                              TRKCN     38
      MM = NEL                                                           TRKCN     39
       GO TO 22                                                          TRKCN     40
                                                                         TRKCN     41
C  SINGLE LINEAR TRANSFORMATION                                          TRKCN     42
 30   ASSIGN 45 TO RET                                                   TRKCN     43
       GO TO 40                                                          TRKCN     44
                                                                         TRKCN     45
C  SINGLE NON-LINEAR TRANSFORMATION                                      TRKCN     46
 20   ASSIGN 45 TO RET                                                   TRKCN     47
```

```
            IF (IOP.GT.1.AND.IOP.NE.5)  ASSIGN 43 TO RET                 TRKCN    48
            GO TO 221                                                    TRKCN    49
                                                                         TRKCN    50
      C  LOOP OVER PRD FACTORS                                           TRKCN    51
       21   I = I + 1                                                    TRKCN    52
            IF (I.GT.K) GO TO 45                                         TRKCN    53
       22   NN = MDAT(MM,I)                                              TRKCN    54
            KIND = INFF(20,NN)                                           TRKCN    55
            IF (KIND.NE.3HPRD) GO TO 222                                 TRKCN    56
            N2 = NN                                                      TRKCN    57
            K2 = 0                                                       TRKCN    58
            ASSIGN 321 TO RET                                            TRKCN    59
            KN = INFF(4,N2)                                              TRKCN    60
       321  K2 = K2 + 1                                                  TRKCN    61
            NN = MDAT(N2,K2)                                             TRKCN    62
            KIND = INFF(20,NN)                                           TRKCN    63
       322  IF (K2.EQ.KN) ASSIGN 21 TO RET                               TRKCN    64
            IF (KIND.NE.3HPRD) GO TO 222                                 TRKCN    65
            N3 = NN                                                      TRKCN    66
            K3 = 0                                                       TRKCN    67
            ASSIGN 421 TO RET                                            TRKCN    68
            KN3 = INFF(4,N3)                                             TRKCN    69
       421  K3 = K3 + 1                                                  TRKCN    70
            NN = MDAT(N3,K3)                                             TRKCN    71
            KIND = INFF(20,NN)                                           TRKCN    72
            IF (K3.EQ.KN3) ASSIGN 321 TO RET                             TRKCN    73
            IF (K2.EQ.KN.AND.K3.EQ.KN3) ASSIGN 21 TO RET                 TRKCN    74
            IF (KIND.NE.3HPRD) GO TO 222                                 TRKCN    75
            WRITE (3,1000)                                               TRKCN    76
       1000 FORMAT (* PRD INSTRUCTIONS NESTED MORE THAN THREE DEEP - ERROR*) TRKCN 77
            ERROR = .TRUE.                                               TRKCN    78
            RETURN                                                       TRKCN    79
       222  IF (KIND.NE.3HMAP) GO TO 40                                  TRKCN    80
                                                                         TRKCN    81
      *                  PICK UP SUBROUTINE NAME                         TRKCN    82
       221  OPNAME = INFF(1,NN)                                          TRKCN    83
            IF (OPNAME.NE.HSXTP)  GO TO 223                              TRKCN    84
            NUM = 11                                                     TRKCN    85
            GO TO 9                                                      TRKCN    86
       223  IF (OPNAME.EQ.HMOVE)  GO TO 27                               TRKCN    87
            IF (OPNAME.NE.HNPOL)  GO TO 25                               TRKCN    88
            NUM = 12                                                     TRKCN    89
            GO TO 9                                                      TRKCN    90
      *                  STRIP OFF MAP SUBROUTINE NUMBER                 TRKCN    91
       25   DECODE (10,100,OPNAME) NUM                                   TRKCN    92
       100  FORMAT (3X,I1,6X)                                            TRKCN    93
            IF (NUM.EQ.0) NUM = + 0                                      TRKCN    94
            NUM = NUM + 1                                                TRKCN    95
                                                                         TRKCN    96
      *                  GET DATA OF MAP INSTRUCTION.  TRANSMIT IN ARGUMENT. TRKCN 97
       9    NFL = INFF(15,NN)                                            TRKCN    98
            CALL DATA(NN,1,1,NFL,PAR)                                    TRKCN    99
            GO TO (10,11,12,13,14,15,16,17,18,19,23,28) NUM              TRKCN   100
       10   CALL MAP (P,PAR)                                             TRKCN   101
            GO TO RET,(21,43,45,321,421)                                 TRKCN   102
      C  MAP1 --- WIGGLER MAGNET                                         TRKCN   103
       11   CALL DIFEQ(NN,P,DER1)                                        TRKCN   104
```

```
      GO TO RET,(21,43,45,321,421)                                    TRKCN  105
C  MAP2                                                               TRKCN  106
 12   CALL DIFEQ(NN,P,DER2)                                           TRKCN  107
      GO TO RET,(21,43,45,321,421)                                    TRKCN  108
C  MAP3  INTEGRATE TRANSVERSE AND LONGITUDINAL ENVELOPES.             TRKCN  109
 13   CALL DIFEQ(NN,P,DER3)                                           TRKCN  110
      GO TO RET,(21,43,45,321,421)                                    TRKCN  111
C  MAP4 —— ENVELOPE INTEGRATION                                       TRKCN  112
 14   CALL DIFEQ(NN,P,DER4)                                           TRKCN  113
      GO TO RET,(21,43,45,321,421)                                    TRKCN  114
C  MAP5 —— SEXTUPOLE INTEGRATION                                      TRKCN  115
 15   CALL DIFEQ(NN,P,DER5)                                           TRKCN  116
      GO TO RET,(21,43,45,321,421)                                    TRKCN  117
 16   CALL DIFEQ (NN,P,DER6)                                          TRKCN  118
      GO TO RET,(21,43,45,321,421)                                    TRKCN  119
 17   CALL DIFEQ (NN,P,DER7)                                          TRKCN  120
      GO TO RET,(21,43,45,321,421)                                    TRKCN  121
 18   CALL MAP8(P,PAR)                                                TRKCN  122
      GO TO RET,(21,43,45,321,421)                                    TRKCN  123
 19   CALL MAP9(P,PAR)                                                TRKCN  124
      GO TO RET,(21,43,45,321,421)                                    TRKCN  125
                                                                      TRKCN  126
 43   CALL STOR7(MFXPT,P(1,2),RW)                                     TRKCN  127
      RETURN                                                          TRKCN  128
C                                                                     TRKCN  129
 23   INFF(4,NF)=INFF(4,NN)                                           TRKMPL   1
      INFF(5,NF)=INFF(5,NN)                                           TRKMPL   2
      CALL SXTP (NF,P,PAR)                                            TRKMPL   3
      GO TO 26                                                        TRKCN  131
 28   CALL NPOL(NF,NN,P,PAR)                                          TRKCN  132
      GO TO 26                                                        TRKCN  133
C                                                                     TRKCN  134
 27   CALL MOVE(NN,NF,P)                                              TRKCN  135
 26   IF (IOP.NE.1.AND.IOP.NE.5)  GO TO 42                            TRKCN  136
      GO TO RET,(21,43,45,321,421)                                    TRKCN  137
                                                                      TRKCN  138
C  CARRY OUT A LINEAR TRANSFORMATION                                  TRKCN  139
 40   IF (IOP.NE.1) GO TO 41                                          TRKCN  140
      CALL RTRV7(NN,T,RW)                                             TRKCN  141
      CALL MXV7(T,P,P)                                                TRKCN  142
      GO TO RET,(21,43,45,321,421)                                    TRKCN  143
 41   CALL REVMAT(NF,NN,P)                                            TRKCN  144
      IF (IOP.EQ.5)  GO TO 45                                         TRKCN  145
 42   ML(1) = MFXPT                                                   TRKCN  146
      ML(2) = NF                                                      TRKCN  147
      CALL MMM ( MFXPT, 2, ML )                                       TRKCN  148
      IF (IOP.NE.0)  CALL RTRV7(MFXPT,P(1,2),RW)                      TRKCN  149
      GO TO RET,(21,43,45,321,421)                                    TRKCN  150
C                                                                     TRKCN  151
 35   CONTINUE                                                        TRKCN  152
      CALL EDRF(NEL,P,NP)                                             TRKCN  153
      GO TO 45                                                        TRKCN  154
 36   CONTINUE                                                        TRKCN  155
      CALL DKE(NEL,P,NP)                                              TRKCN  156
      GO TO 45                                                        TRKCN  157
 37   CALL SOL(NEL,P,NP)                                              TRKCN  158
                                                                      TRKCN  159
```

```
        45  RETURN                                                  TRKCN    160
            END                                                     TRKCN    161
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
137	I	P	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
161	I	P	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

```
                                                                      TRKCN    162
      SUBROUTINE WBE(K,L)                                             WBE        2
C  WRITES BETATRON FUNCTIONS OF ELEMENTS MI(I),I=1,K                  WBE        3
                                                                      BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                       BLANK      3
      COMMON STORE(48000),IWORK(10)                                   86MARSIZ   1
      DIMENSION INFF(24,2000)                                         86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                        86MARSIZ   3
                                                                      BLANK      5
      COMMON/BMI/MI(5000)                                             BMIL       1
      COMMON/NELS/NELS                                                BMIL       2
      LEVEL 2,MI                                                      BMIL       3
                                                                      BMIL       4
                                                                      BMI        3
      DIMENSION BETX(9),BETY(9)                                       WBE        6
      CALL HED                                                        WBE        7
      IF (L.NE.0) GO TO 40                                            WBE        8
      WRITE (3,51)                                                    WBE        9
      GO TO 42                                                        WBE       10
   40 WRITE (3,52)                                                    WBE       11
   42 DO 45  I=1,K                                                    WBE       12
      M1=MI(I)                                                        WBE       13
      N=INFF(2,M1)                                                    WBE       14
      IF (L.NE.0) GO TO 50                                            WBE       15
      CALL BET(M1,BETX,BETY,1)                                        WBE       16
      BETX(1) = BETX(1) / 6.283185307                                 WBE       17
      BETY(1) = BETY(1) / 6.283185307                                 WBE       18
      WRITE (3,53) N,BETX(1),BETX(2),BETX(3),BETX(5),BETX(6),BETX(7), WBE       19
     1BETY(1),BETY(2),BETY(3),BETY(5),BETY(6),BETY(7)                 WBE       20
      GO TO 45                                                        WBE       21
   50 CALL BET(M1,BETX,BETY,0)                                        WBE       22
      BETX(1) = BETX(1) / 6.283185307                                 WBE       23
      BETY(1) = BETY(1) / 6.283185307                                 WBE       24
      WRITE (3,53) N,BETX(1),BETX(2),BETX(3),BETX(5),BETX(6),BETX(9), WBE       25
     1BETY(1),BETY(2),BETY(3),BETY(5),BETY(6),BETY(9)                 WBE       26
   45 CONTINUE                                                        WBE       27
      CALL HED                                                        WBE       28
      RETURN                                                          WBE       29
   51 FORMAT(    5X,19HBETATRON FUNCTIONS./          1H0,131HNAME     MU WBE    30
     1X/2PI    BETAX     ALPHAX        XEQ        DXEQ         WX        MUY/ WBE 31
     22PI    BETAY     ALPHAY        YEQ        DYEQ         WY     /1H0) WBE     32
   52 FORMAT( 5X,19HBETATRON FUNCTIONS./1H0,131HNAME     MUX/2PI    BETAX WBE    33
     1     ALPHAX        XEQ        DXEQ       XTRACE/2       MUY/2PI    BETAY  WBE 34
     2  ALPHAY        YEQ        DYEQ       YTRACE/2 /1H0)            WBE       35
   53 FORMAT(1X,A5,F12.8,F10.5,4F10.6,2X,F12.8,F10.5,4F10.6)          WBE       36
      END                                                             WBE       37
```

```
      SUBROUTINE WFLSR                                                  WFLSR      2
C   PRINTS CONTENTS OF INFF(J,I),J=1,24 AND I=MIN,MAX.                  WFLSR      3
C   IF ROUTINE CALLED BECAUSE OF ERROR, ONLY INFF IS PRINTED.           WFLSR      4
C   IF KB OF WFL IS NOT 0, STORE USED BY INSTRUCTIONS IS PRINTED.       WFLSR      5
C   IF KB NOT 0 AND KA=1, WORKING STORAGE IS ALSO PRINTED.              WFLSR      6
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/INSTR/OPNAME,NAME,OP,KA,KB,OBJA,OBJB,NXTM,TRA,LQ,NQ,LIN,   INSTR      2
     1             NIN,LFL,NFL,LBC,NBC,NTOT,IX,KIND,TYPE,ROWS,COLS,EXTR, INSTR      3
     2             M,NF,NB,NI,MSUBR,ISIGN                               INSTR      4
      INTEGER OPNAME,OP,OBJA,OBJB,TRA,TYPE,ROWS,COLS,EXTR               INSTR      5
                                                                        INSTR      6
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1          LDFLG,FIN                                               CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      COMMON/STORE/LMAX,LINF,LFILE,MAX,MIN,ICARD(11)                    STORE      2
*  LMAX AND MAX ARE DIMENSIONS OF STORE AND INFF                        STORE      3
                                                                        STORE      4
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL   DIM        2
                                                                        DIM        3
      LOGICAL ENDL                                                      WFLSR     12
C                                                                       WFLSR     13
      IF (MODE.EQ.3) GO TO 16                                           WFLSR     14
      ISGN = 1H                                                         WFLSR     15
      IF (OP.LT.0) ISGN = 1H-                                           WFLSR     16
      WRITE (3,35) ISGN,NAME,OPNAME,KA,KB                               WFLSR     17
 35   FORMAT (6H  *** ,A1,A5,2X,A5,1X,I3,1X,I3,5H //  ,6A10)            WFLSR     18
      IF (MODE.EQ.2) RETURN                                             WFLSR     19
      IF (OP.LT.0) RETURN                                               WFLSR     20
C   PRINT INFF                                                          WFLSR     21
 16   WRITE (3,24)                                                      WFLSR     22
 24    FORMAT (3X,*CONTENTS OF*,5X,*INFF(I,M)*,                         WFLSR     23
     1 18X,*OPNAME*,17X,*NAME*,19X,*OP*,19X,*KA*/                       WFLSR     24
     2 48X,*KB*,19X,*OBJA*,18X,*OBJB*,18X,*NXTM*/                       WFLSR     25
     3 48X,*TRA*,20X,*LQ*,20X,*NQ*,20X,*LIN*/                           WFLSR     26
     4 48X,*NIN*,20X,*LFL*,19X,*NFL*,19X,*LBC*/                         WFLSR     27
     5 48X,*NBC*,19X,*NTOT*,19X,*LQ2*,19X,*KIND*/                       WFLSR     28
     6 48X,*TYPE*,18X,*ROWS*,18X,*COLS*,19X,*LQ3*/)                     WFLSR     29
      NUM=MAX-MIN+1                                                     WFLSR     30
      MX=MAX                                                            WFLSR     31
      IF(.NOT.ERROR) GO TO 101                                          WFLSR     32
      NUM=10                                                            WFLSR     33
      MX=MIN+9                                                          WFLSR     34
 101  CONTINUE                                                          WFLSR     35
      DO 10 I=1,NUM                                                     WFLSR     36
      WRITE (3,22) INFF(2,MX),INFF(1,MX),MX,(INFF(J,MX),J=1,4)          WFLSR     37
 22   FORMAT (/2X,2(A5,2X),8H(M(OCT)=O5,8H),I=1,4),4(2X,O20))           WFLSR     38
      WRITE (3,23) MX,(INFF(J,MX),J=5,8)                                WFLSR     39
 23   FORMAT (16X,8H(M(DEC)=I5,8H),I=5,8),4(2X,O20))                    WFLSR     40
      J1 = 9                                                            WFLSR     41
      DO 31 K=1,4                                                       WFLSR     42
      J2 = J1+ 3                                                        WFLSR     43
```

```
      WRITE(3,30) J1,J2,(INFF(J,MX),J=J1,J2)
      J1 = J2 + 1
   30 FORMAT (21X,5H(I = ,I2,1H,I2,1H),5X,4(2X,O20))
   31 CONTINUE
      MX=MX-1
   10 CONTINUE
C  PRINT STOR
      IF (ERROR) RETURN
      IF (KB.EQ.0) RETURN
      LX=LFILE
      L = ISAV*(14+KADD)
      IF(KA.EQ.1) L=1
C  ENDL IS SWITCH TO TERMINATE PRINTING
      ENDL=.FALSE.
      WRITE (3,28)
   28 FORMAT (31H1    CONTENTS OF STORE IN OCTAL//
     141H  OCTAL LOCATION       PLUS    0  1  2  3/
     241H                               4  5  6  7//)
   17 J=L+3
      IF (J.LT.LX) GO TO 18
      J=LX
      ENDL=.TRUE.
   18 WRITE (3,25)  L,(STORE(I),I=L,J)
   25 FORMAT (5X,O6,5X,4(O20,2X))
      IF (ENDL) GO TO 20
      L=J+1
      J=L+3
      IF (J.LT.LX) GO TO 19
      J=LX
      ENDL=.TRUE.
   19 WRITE (3,26) (STORE(I),I=L,J)
   26 FORMAT (16X,4(O20,2X)//)
      IF (ENDL) GO TO 20
      L=J+1
      GO TO 17
   20 WRITE (3,27)
   27 FORMAT (1H1)
      RETURN
      END
```

```
      SUBROUTINE WMA(KA)                                                WMA        2
C WRITES MATRICES OF ELEMENTS MI(I),I=1,KA.                             WMA        3
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER   SWTCH      2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                          SWTCH      3
                                                                        SWTCH      4
      COMMON/BMI/MI(5000)                                               BMIL       1
      COMMON/NELS/NELS                                                  BMIL       2
      LEVEL 2,MI                                                        BMIL       3
                                                                        BMIL       4
                                                                        BMI        3
      DIMENSION RX(2,3),RY(2,3),RW(3),T(7,7)                            WMA        7
C  PRINT ALL MATRICES AS DEFINED BY SIZE                                WMA        8
C  IF SIZE ARBITRARY, PRINT IN 7X7 IF ANY ARE 7X7                       WMA        9
                                                                        WMA       10
      CALL DASH                                                         WMA       11
      IF (MSIZE.EQ.7) GO TO 300                                         WMA       12
      IF (MSIZE.EQ.3) GO TO 102                                         WMA       13
      DO 100 I=1,KA                                                     WMA       14
      M1 = IABS(MI(I))                                                  WMA       15
      KN = INFF(20,M1)                                                  WMA       16
      IF (KN.EQ.2HR7.OR.KN.EQ.3HROT) GO TO 300                          WMA       17
  100 CONTINUE                                                          WMA       18
C  PRINT ALL AS 3X3 MATRICES                                            WMA       19
  102 IF (BEND.NE.0.) GO TO 105                                         WMA       20
C  PRINT HEADING FOR MISALIGNMENT                                       WMA       21
      WRITE (3,11)                                                      WMA       22
      GO TO 106                                                         WMA       23
C  PRINT HEADING FOR TRANSFER MATRICES--NO MISALIGNMENT                 WMA       24
  105 WRITE (3,21)                                                      WMA       25
  106 DO 150 I=1,KA                                                     WMA       26
      M1 = MI(I)                                                        WMA       27
      N = MNAME(M1)                                                     WMA       28
      CALL RXY(M1,RX,RY,RW)                                             WMA       29
      IF (BEND.NE.0.) GO TO 110                                         WMA       30
      WRITE (3,12) (RX(1,J),J=1,3),(RY(1,J),J=1,3)                      WMA       31
      WRITE (3,13) N, (RX(2,J),J=1,3), (RY(2,J),J=1,3)                  WMA       32
      GO TO 120                                                         WMA       33
  110 WRITE (3,22) (RX(1,J),J=1,3),(RY(1,J),J=1,2)                      WMA       34
      WRITE (3,23) N, (RX(2,J),J=1,3), (RY(2,J),J=1,2)                  WMA       35
      W1 = RX(1,3)*RX(2,1) - RX(2,3)*RX(1,1)                            WMA       36
      W2 = RX(1,3)*RX(2,2) - RX(2,3)*RX(1,2)                            WMA       37
      WRITE (3,24) W1,W2,RW(3)                                          WMA       38
  120 WRITE (3,15) RW(1),RW(2)                                          WMA       39
  150 CONTINUE                                                          WMA       40
      GO TO 355                                                         WMA       41
C  PRINT ALL AS 7X7 MATRICES                                            WMA       42
  300 CONTINUE                                                          WMA       43
      WRITE (3,36)                                                      WMA       44
      DO 350 I=1,KA                                                     WMA       45
      M1 = MI(I)                                                        WMA       46
      N = MNAME(M1)                                                     WMA       47
```

```
      CALL RTRV7(M1,T,RW)
      DO 305 II=1,3
      WRITE (3,34) (T(II,J),J=1,7)
  305 CONTINUE
      WRITE (3,35) N,(T(4,J),J=1,7)
      DO 310 II=5,7
      WRITE (3,34) (T(II,J),J=1,7)
  310 CONTINUE
      WRITE (3,15) RW(1),RW(2)
  350 CONTINUE
  355 CALL DASH
      RETURN
   11 FORMAT (20H   TRANSFER MATRICES,25X,7HRX(I,J),40X,7HRY(I,J)//
     134X,1HX,11X,2HDX,11X,1H1,20X,1HY,12X,2HDY,11X,1H1/9X,7HELEMENT)
   12 FORMAT (27X,3F13.8,8X,3F13.8)
   13 FORMAT (10X,A5,12X,3F13.8,8X,3F13.8/30X,2H0.,11X,2H0.,11X,2H1.,19X
     1,2H0.,11X,2H0.,11X,2H1./)
   15 FORMAT (/30X,9HLENGTH = ,F13.8,5X,9H THETA = ,F13.8///)
   21 FORMAT (20H   TRANSFER MATRICES,29X,7HRX(I,J),35X,7HRY(I,J)//
     134X,1HX,9X,5HDX/DS,9X,3H-DS,8X,4HDP/P,15X,1HY,10X,5HDY/DS/
     29X,7HELEMENT)
   22 FORMAT (27X,2F13.8,3X,2H0.,8X,F13.8,3X,2F13.8)
   23 FORMAT (10X,A5,12X,2F13.8,3X,2H0.,8X,F13.8,3X,2F13.8)
   24 FORMAT (27X,2F13.8,3X,10H1.00000000,F13.8/30X,2H0.,11X,2H0.,11X,
     12H0.,11X,10H1.00000000)
   34 FORMAT (30X,7F13.8)
   35 FORMAT (10X,A5,15X,7F13.8)
   36 FORMAT (20H   TRANSFER MATRICES,53X,6HR(I,J)//36X,1HX,10X,5HDX/DS,
     110X,1HY,10X,5HDY/DS,9X,3H-DS,9X,4HDP/P,10X,1H1/9X,7HELEMENT)
      END
```

```
      SUBROUTINE WRTINF(INF,M)                                 WRTINF      2
C WRITES ONE COLUMN OF INF INTO INFF(M)                         WRTINF      3
                                                               BLANK       2
      LEVEL 2, STORE,INFF,IWORK                                BLANK       3
      COMMON STORE(48000),IWORK(10)                            86MARSIZ    1
      DIMENSION INFF(24,2000)                                  86MARSIZ    2
      EQUIVALENCE (INFF,STORE)                                 86MARSIZ    3
                                                               BLANK       5
      DIMENSION INF(1)                                         WRTINF      5
      DO 20 I=1,24                                             WRTINF      6
      IF(INF(I).EQ.1H*) GO TO 20                               WRTINF      7
      INFF(I,M)=INF(I)                                         WRTINF      8
   20 CONTINUE                                                 WRTINF      9
      RETURN                                                   WRTINF     10
      END                                                      WRTINF     11
```

```
      SUBROUTINE XEQCON(M)                                              XEQCON    2
C  CONTROL EXECUTION OF SYNCH INSTRUCTION BE ACT OR DELE               XEQCON    3
C  IF KB=1, KA ELEMENTS SHOULD BE CHANGED BEGINNING WITH ONE INPUT     XEQCON    4
C  IF KB = 0, INPUT CONTAINS KA ELEMENTS                                XEQCON    5
                                                                        BLANK     2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK     3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ  1
      DIMENSION INFF(24,2000)                                           86MARSIZ  2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ  3
                                                                        BLANK     5
      LOGICAL SEQ,ACT                                                   XEQCON    7
      INTEGER ELNUM                                                     XEQCON    8
      DIMENSION NDAT(50)                                                XEQCON    9
      DATA IBLNK/1H /                                                   XEQCON   10
                                                                        XEQCON   11
      SEQ = .FALSE.                                                     XEQCON   12
      KA = INFF(4,M)                                                    XEQCON   13
      KB = INFF(5,M)                                                    XEQCON   14
      NAM = INFF(1,M)                                                   XEQCON   15
      ACT = .TRUE.                                                      XEQCON   16
      IF (NAM.EQ.4HDELE) ACT = .FALSE.                                  XEQCON   17
      IF (KB.EQ.1) SEQ = .TRUE.                                         XEQCON   18
      KC = KA                                                           XEQCON   19
      IF (SEQ) KC = 1                                                   XEQCON   20
      CALL DATA(M,2,1,KC,NDAT)                                          XEQCON   21
      IF(SEQ) GO TO 15                                                  XEQCON   22
C  CHANGE EXECUTION MODE OF ALL ELEMENTS IN NDAT ARRAY                 XEQCON   23
      DO 10 I=1,KA                                                      XEQCON   24
      IF (NDAT(I).EQ.IBLNK)  GO TO 10                                   XEQCON   25
      NN = ELNUM(NDAT(I))                                               XEQCON   26
      IOP = IABS(INFF(3,NN))                                            XEQCON   27
      IF (.NOT.ACT) IOP = -IOP                                          XEQCON   28
      INFF(3,NN) = IOP                                                  XEQCON   29
   10 CONTINUE                                                          XEQCON   30
      RETURN                                                            XEQCON   31
   15 CONTINUE                                                          XEQCON   32
      NN = ELNUM(NDAT(1))                                               XEQCON   33
      DO 20 I=1,KA                                                      XEQCON   34
      IOP = IABS(INFF(3,NN))                                            XEQCON   35
      IF (.NOT.ACT) IOP = -IOP                                          XEQCON   36
      INFF(3,NN) = IOP                                                  XEQCON   37
      NN = INFF(8,NN)                                                   XEQCON   38
   20 CONTINUE                                                          XEQCON   39
      RETURN                                                            XEQCON   40
      END                                                               XEQCON   41
```

```
      SUBROUTINE ZAM0(NO,X,DERI,Y,F,T,HPRO)                             ZAM0   2
      COMMON /INTC/ IPMX,AREF,EMAX,SSSR,HFAC,SWAM,SWEX                  ZAM0   3
      COMMON /INTP/ HPR,XX,N,EUB,ELB,IP,IT,NRKS,SWIN                    ZAM0   4
      DIMENSION Y(1),F(1),T(8,1)                                        ZAM0   5
      LOGICAL SWAM,SWEX,SWIN                                            ZAM0   6
      INTEGER HFAC                                                      ZAM0   7
      DOUBLE PRECISION T,HPRO,HPR,XX                                    ZAM0   8
      DATA IPMX,AREF,EMAX,SSSR,HFAC,SWAM,SWEX                           ZAM0   9
     $    /1024,1.0,1.0E-6,100.0,2,.TRUE.,.TRUE./                       ZAM0  10
C                                                                       ZAM0  11
      HPR=HPRO                                                          ZAM0  12
      XX=DBLE(X)                                                        ZAM0  13
      N=NO                                                              ZAM0  14
      EUB=EMAX                                                          ZAM0  15
      ELB=EMAX/SSSR                                                     ZAM0  16
      IP=1                                                              ZAM0  17
      IT=0                                                              ZAM0  18
      NRKS=0                                                            ZAM0  19
      SWIN=SWEX                                                         ZAM0  20
      CALL DERI (X,Y,F)                                                 ZAM0  21
      DO 9 I=1,N                                                        ZAM0  22
      T(5,I)=DBLE(Y(I))                                                 ZAM0  23
    9 CONTINUE                                                          ZAM0  24
      RETURN                                                            ZAM0  25
      END                                                               ZAM0  26
```

```
      SUBROUTINE ZAM(X,DERI ,Y,F,T,SWPR)
     COMMON /INTC/ IPMX,AREF,EMAX,SSSR,HFAC,SWAM,SWEX
     COMMON /INTP/ HPR,XX,N,EUB,ELB,IP,IT,NRKS,SWIN
C
      DIMENSION Y(1),F(1),T(8,1)
     LOGICAL SWAM,SWEX,SWIN
     LOGICAL SWPR
     INTEGER HFAC
      DOUBLE PRECISION T,HPR,XX
      DOUBLE PRECISION D,H
 6000 FORMAT (36H0 CANNOT DECREASE H BECAUSE OF HMIN.    ,1PE16.8,I20)
C
 1    CONTINUE
      SWPR=.FALSE.
      TEST=0.0
      H=HPR/DBLE(FLOAT(IP*24))
      IF ((NRKS .LT. 3) .OR. (.NOT. SWAM)) GO TO 200
C
C     ADAMS-MOULTON STEP.
 100  CONTINUE
      DO 109 I=1,N
      D=DBLE(F(I))
      T(4,I)=D
      Y(I)=SNGL(T(5,I)+H*(
    X 55.0D0*D-59.0D0*T(3,I)+37.0D0*T(2,I)- 9.0D0*T(1,I) ))
 109  CONTINUE
      X=SNGL(XX+24.0D0*H)
     CALL DERI (X,Y,F)
      DO 119 I=1,N
      D=DBLE(F(I))
      D=(     T(5,I)+H*(
    X  9.0D0*D+19.0D0*T(4,I)- 5.0D0*T(3,I)+       T(2,I) ))
      T(6,I)=D
      E=ABS(SNGL(D)-Y(I))/14.0
      TEST=AMAX1(E/AMAX1(AREF,ABS(SNGL(D))),TEST)
 119  CONTINUE
C
      GO TO 300
C
C     ZONNEVELD STEP.
 200  CONTINUE
      DO 209 I=1,N
      D=DBLE(F(I))
      T(4,I)=D
C     1
      Y(I)=SNGL(T(5,I)+H*(
    X 12.0D0*D                                         ))
 209  CONTINUE
      X=SNGL(XX+12.0D0*H)
     CALL DERI (X,Y,F)
      DO 219 I=1,N
      D=DBLE(F(I))
      T(6,I)=D
C     2
      Y(I)=SNGL(T(5,I)+H*(
    X 12.0D0*D                                         ))
 219  CONTINUE
```

```
            CALL DERI (X,Y,F)
            DO 229 I=1,N
            D=DBLE(F(I))
            T(7,I)=D
      C     3
            Y(I)=SNGL(T(5,I)+H*(
           X 24.0D0*D                                          ))
        229 CONTINUE
            X=SNGL(XX+24.0D0*H)
            CALL DERI (X,Y,F)
            DO 239 I=1,N
            D=DBLE(F(I))
            T(8,I)=D
      C     4
            Y(I)=SNGL(T(5,I)+H*(
           X 3.75D0*T(4,I)+5.25D0*T(6,I)+9.75D0*T(7,I)-0.75D0*D ))
        239 CONTINUE
            X=SNGL(XX+18.0D0*H)
            CALL DERI (X,Y,F)
            DO 249 I=1,N
            D=DBLE(F(I))
            E=ABS(SNGL(H*(
           X -16.0D0*T(4,I)+48.0D0*T(6,I)+48.0D0*T(7,I)+48.0D0*T(8,I)
           X -128.0D0*D   )))
      C     5
            D=(    T(5,I)+H*(
           X 4.0D0*T(4,I)+ 8.0D0*T(6,I)+ 8.0D0*T(7,I)+ 4.0D0*T(8,I)
           X ))
            T(6,I)=D
            TEST=AMAX1(E/AMAX1(AREF,ABS(SNGL(D))),TEST)
        249 CONTINUE
      C
      C     BOTH ADAMS-MOULTON AND ZONNEVELD METHODS CONTINUE FROM HERE.
        300 CONTINUE
            X=SNGL(XX+24.0D0*H)
            IF (TEST .LE. EUB) GO TO 310
            IF (IP*HFAC .GT. IPMX) GO TO 309
      C
      C     REPEAT STEP WITH SMALLER H.
            NRKS=0
            IP=IP*HFAC
            IT=IT*HFAC
            DO 305 I=1,N
            Y(I)=SNGL(T(5,I))
            F(I)=SNGL(T(4,I))
        305 CONTINUE
            GO TO 1
      C
      C     CANNOT DECREASE H BECAUSE OF HMIN.
        309 CONTINUE
            IF (.NOT. SWIN) GO TO 310
            PRINT 6000, X,IPMX
            SWIN=.FALSE.
      C
        310 CONTINUE
      C
      C
```

```
C       ACCEPT CURRENT STEP.
C
C       XX STILL HAS NOT BEEN CHANGED SINCE ENTRY.
C       YY(XX) IS STILL IN T(5, ).
C       F(YY) IS IN T(4, ).
C
        IT=IT+1
        XX=XX+HPR/DBLE(FLOAT(IP))
        NRKS=MIN0(NRKS+1,4)
        DO 319 I=1,N
        D=T(6,I)
        T(5,I)=D
        Y(I)=SNGL(D)
  319   CONTINUE
        X=SNGL(XX)
       CALL DERI (X,Y,F)
        IF (IT .LT. IP) GO TO 320
C
C       X IS A MULTIPLE OF HPRINT.
        SWPR=.TRUE.
        IT=IT-IP
C
  320   CONTINUE
        IF (TEST .GE. ELB) GO TO 330
        IF (MOD(IP,HFAC)+MOD(IT,HFAC) .NE. 0) GO TO 330
C
C       PROCEED TO NEXT STEP WITH LARGER H, USING ZONNEVELD METHOD.
        NRKS=0
        IP=IP/HFAC
        IT=IT/HFAC
        RETURN
C
C
C       PROCEED TO NEXT STEP WITH SAME H.
  330   CONTINUE
        DO 339 I=1,N
        T(1,I)=T(2,I)
        T(2,I)=T(3,I)
        T(3,I)=T(4,I)
  339   CONTINUE
        RETURN
        END
```

```
                                                                  ZAM      158
      SUBROUTINE MINUITS                                          MINUITS    3
CC AUG. 17,1975                                                   MINUITS    4
      CALL MINNEW                                                 MINUITS    9
      RETURN                                                      MINUITS   10
      END                                                         MINUITS   11
```

```
               FUNCTION CALFCN(PVEC)                                             CALFCN     2
        CC          CALLED ONLY FROM IMPROV.  TRANSFORMS THE FUNCTION FCN        CALFCN     3
        CC          BY DIVIDING OUT THE QUADRATIC PART IN ORDER TO FIND FURTHER  CALFCN     4
        CC          MINIMA.    CALCULATES  (F-FMIN)/(X-XMIN)*V*(X-XMIN)          CALFCN     5
        CC                                                                       CALFCN     6
                COMMON                                                           COMMONU    2
              1/MINERR/ ERP(30)     ,ERN(30)                                     COMMONU    4
              2/PARINT/ X(15)       ,XT(15)     ,DIRIN(15)  ,MAXINT     ,NPAR    COMMONU    5
              3/PAREXT/ U(30)       ,NAM(30)    ,WERR(30)   ,MAXEXT     ,NU      COMMONU    6
              4/LIMITS/ ALIM(30)    ,BLIM(30)   ,LCODE(30)  ,LCORSP(30) ,LIMSET  COMMONU    7
              5/VARIAN/ V(15,15)                                                 COMMONU    8
              7/FIX   / IPFIX(15)   ,XS(15)     ,XTS(15)    ,WTS(15)   ,NPFIX    COMMONU    9
              C/CASC/ JH, JL, Y(16)                                              COMMONU   10
              F/DERIVA/ G(30)       ,G2(30)                                      COMMONU   11
              G/SIMVEC/ P(15,16)    ,PSTAR(15)  ,PSTST(15)  ,PBAR(15)   ,PRHO(15) COMMONU  12
              J/VARIAT/ VT(15,15)                                                COMMONU   13
              6/UNIT  / ISYSRD      ,ISYSWR     ,ISYSPU                          COMMONU   27
              8/TITLE / TITLE(13)   ,DATE(2)    ,ISW(7)     ,NBLOCK              COMMONU   28
              9/CONVER/ EPSI ,APSI  ,VTEST      ,NSTEPQ     ,NFCN       ,NFCNMX  COMMONU   29
              A/CARD  / CWORD       ,CWORD2     ,CWORD3     ,WORD7(7)            COMMONU   30
              B/MINIMA/ AMIN        ,UP         ,NEWMIN     ,ITAUR      ,SIGMA   COMMONU   31
              K/COMMND/ NNAME       ,CNAME(25)  ,CNAM2(25)  ,CNAM3(25)           COMMONU   32
               DIMENSION PVEC(15)                                                CALFCN     8
               CALL INTOEX(PVEC)                                                 CALFCN     9
               CALL FCN(NPAR,G,F,U,4)                                            CALFCN    10
               NFCN = NFCN + 1                                                   CALFCN    11
               DO 200 I= 1, NPAR                                                 CALFCN    12
               G(I) = 0.                                                         CALFCN    13
               DO 200 J= 1, NPAR                                                 CALFCN    14
           200 G(I) = G(I) + VT(I,J) * (XT(J)-PVEC(J))                           CALFCN    15
               DENOM = 0.                                                        CALFCN    16
               DO 210 I= 1, NPAR                                                 CALFCN    17
           210 DENOM = DENOM + G(I) * (XT(I)-PVEC(I))                            CALFCN    18
               IF (DENOM .LE. 0.)  ISW(2) = 0                                    CALFCN    19
               IF (DENOM .LE. 0.)  DENOM = 1.0                                   CALFCN    20
               CALFCN = (F-APSI) / DENOM                                         CALFCN    21
               RETURN                                                            CALFCN    22
               END                                                               CALFCN    23
```

```
              SUBROUTINE COMAND                                                    COMAND       2
      CC          READS THE COMMAND CARDS AND TAKES APPROPRIATE ACTION,            COMAND       3
      CC          EITHER DIRECTLY BY SKIPPING TO THE CORRESPONDING CODE IN         COMAND       4
      CC          COMAND, OR BY SETTING UP A CALL TO A SUBROUTINE                  COMAND       5
      CC                                                                           COMAND       6
               COMMON                                                              COMMONU      2
             1/MINERR/ ERP(30)     ,ERN(30)                                        COMMONU      4
             2/PARINT/ X(15)       ,XT(15)      ,DIRIN(15)  ,MAXINT     ,NPAR      COMMONU      5
             3/PAREXT/ U(30)       ,NAM(30)     ,WERR(30)   ,MAXEXT     ,NU        COMMONU      6
             4/LIMITS/ ALIM(30)    ,BLIM(30)    ,LCODE(30)  ,LCORSP(30) ,LIMSET    COMMONU      7
             5/VARIAN/ V(15,15)                                                    COMMONU      8
             7/FIX   / IPFIX(15)   ,XS(15)      ,XTS(15)    ,WTS(15)  ,NPFIX       COMMONU      9
             C/CASC/ JH, JL, Y(16)                                                 COMMONU     10
             F/DERIVA/ G(30)       ,G2(30)                                         COMMONU     11
             G/SIMVEC/ P(15,16)    ,PSTAR(15)   ,PSTST(15)  ,PBAR(15)   ,PRHO(15)  COMMONU     12
             J/VARIAT/ VT(15,15)                                                   COMMONU     13
             6/UNIT  / ISYSRD      ,ISYSWR      ,ISYSPU                            COMMONU     27
             8/TITLE / TITLE(13)   ,DATE(2)     ,ISW(7)     ,NBLOCK                COMMONU     28
             9/CONVER/ EPSI ,APSI  ,VTEST       ,NSTEPQ     ,NFCN       ,NFCNMX    COMMONU     29
             A/CARD  / CWORD       ,CWORD2      ,CWORD3     ,WORD7(7)              COMMONU     30
             B/MINIMA/ AMIN        ,UP          ,NEWMIN     ,ITAUR      ,SIGMA     COMMONU     31
             K/COMMND/ NNAME       ,CNAME(25)   ,CNAM2(25)  ,CNAM3(25)             COMMONU     32
              COMMON /MATCH2/ AA(30),BB(30),KBTS,SMINCAL,                          MATCH4       2
             1        DWORD(10),DWORD2(10),DWORD3(10),DWORD7(7,10),MINCOM          MATCH4       3
              LOGICAL SMINCAL                                                      MATCH4       4
              DIMENSION WORD8(8), GF(30)                                           COMAND      10
              EQUIVALENCE (WORD8(1), CWORD3)                                       COMAND      15
              DATA AGOOD,ABAD,ANONE/4HGOOD, 4H BAD, 4HNONE /                       COMAND      16
              IKARD = 0                                                            COMAND      17
           50 IKARD = IKARD + 1                                                    COMAND      18
              CWORD = DWORD(IKARD)                                                 COMAND      19
              CWORD2=DWORD2(IKARD)                                                 COMAND      20
              CWORD3=DWORD3(IKARD)                                                 COMAND      21
              DO 51 I=1,7                                                          COMAND      22
         51   WORD7(I)=DWORD7(I,IKARD)                                             COMAND      23
              K = 1                                                                COMAND      24
              DO 60 I= 1, 7                                                        COMAND      25
              IF (WORD7(I))  58,60,58                                              COMAND      26
           58 K = I + 1                                                            COMAND      27
           60 CONTINUE                                                             COMAND      28
              IF (CWORD .EQ. CNAME(5))  K = 2                                      COMAND      29
              WRITE (ISYSWR,5001) IKARD,CWORD,CWORD2,(WORD8(I),I=1,K)              COMAND      30
              WRITE (ISYSWR,5004)                                                  COMAND      31
              NFCNMX = WORD7(1) + 0.5                                              COMAND      32
              IF (NFCNMX .LE. 0)  NFCNMX = 1000                                    COMAND      34
              EPSI = WORD7(2)                                                      COMAND      39
              IF (EPSI .LE. 0.)  EPSI = 0.1 * UP                                   COMAND      40
              NEWMIN = 0                                                           COMAND      41
              ITAUR = 0                                                            COMAND      42
              ISW(1) = 0                                                           COMAND      43
              DO 80 I= 1, NNAME                                                    COMAND      44
              IF (CWORD .EQ. CNAME(I))  GO TO 90                                   COMAND      45
           80 CONTINUE                                                             COMAND      46
              WRITE (ISYSWR,5006)                                                  COMAND      47
              GO TO 50                                                             COMAND      48
           90 GO TO (100,200,300,400,500,600,700,1100,900,1000,1100,1200,1300,     COMAND      49
             * 1400,1500,1600,1700,1800,1402,2000,2100,2200,2300) , I              COMAND      50
```

```
      C                                             . . . . . . . . . . NEW MINIMUM   COMAND  51
   95 WRITE (ISYSWR,5011)                                                             COMAND  52
      ITAUR = 0                                                                       COMAND  53
      EPSI = 0.1*UP                                                                   COMAND  54
      C                                             . . . . . . . . . . . . MINIMIZE  COMAND  55
  100 NF = NFCN                                                                       COMAND  56
      CALL SIMPLX                                                                     COMAND  57
      IF (ISW(1) .GE. 1)  GO TO 50                                                    COMAND  58
      NFCNMX = NFCNMX + NF - NFCN                                                     COMAND  59
  150 VTEST = 0.04                                                                    COMAND  60
      GO TO 460                                                                       COMAND  61
      C                                             . . . . . . . . . . . . SEEK      COMAND  62
  200 CALL SEEK                                                                       COMAND  63
      GO TO 50                                                                        COMAND  64
      C                                             . . . . . . . . . . . . SIMPLEX   COMAND  65
  300 CALL SIMPLX                                                                     COMAND  66
      GO TO 50                                                                        COMAND  67
      C                                             . . . . . . . . . . . . MIGRAD    COMAND  68
  400 VTEST = WORD7(3)                                                                COMAND  69
      IF (VTEST .LE. 0.0)  VTEST = 0.01                                               COMAND  70
  460 NF = NFCN                                                                       COMAND  71
      APSI = EPSI                                                                     COMAND  72
      CALL MIGRAD                                                                     COMAND  73
      IF (ISW(2) .GT. 2)  GO TO  50                                                   COMAND  74
      IF (ISW(1) .EQ. 1)  GO TO 50                                                    COMAND  75
      NFCNMX = NFCNMX + NF - NFCN                                                     COMAND  76
      NF = NFCN                                                                       COMAND  77
      CALL SIMPLX                                                                     COMAND  78
      IF (ISW(1) .EQ. 1)  GO TO 50                                                    COMAND  79
      NFCNMX = NFCNMX + NF - NFCN                                                     COMAND  80
      CALL MIGRAD                                                                     COMAND  81
      GO TO 50                                                                        COMAND  82
      C                                             . . . . . . . . . . . . MINOS     COMAND  83
  500 IF (ISW(2) .LT. 1)  GO TO 550                                                   COMAND  84
      EPSI = 0.1 * UP                                                                 COMAND  85
      VTEST = 0.10                                                                    COMAND  86
      CALL MINOS                                                                      COMAND  87
      IF (NEWMIN .LT. 1)  GO TO 50                                                    COMAND  88
      GO TO 95                                                                        COMAND  89
  550 WRITE (ISYSWR,5006)                                                             COMAND  90
  555 WRITE (ISYSWR,5007)                                                             COMAND  91
      GO TO 50                                                                        COMAND  92
      C                                             . . . . . . . . . . . . PUNCH     COMAND  93
  600 CALL MPUNCH                                                                     COMAND  94
      GO TO 50                                                                        COMAND  95
      C                                             . . . . . . . . . . . . PRINTOUT  COMAND  96
  700 ISW(5) = WORD7(1) + 0.5                                                         COMAND  97
      GO TO 50                                                                        COMAND  98
      C                                             . . . . . . . . . . . . FIX       COMAND  99
  900 IT = WORD7(1) + 0.5                                                             COMAND 100
      IF (IT .LE. 0 .OR. IT .GT. NU)  GO TO 905                                       COMAND 101
      IF (LCORSP(IT) .LE. 0)  GO TO 905                                               COMAND 102
      CALL FIXPAR(IT, 0, ILAX)                                                        COMAND 103
      IF (ISW(2) .GT. 1)  CALL MPRINT(1,AMIN)                                         COMAND 104
      GO TO 50                                                                        COMAND 105
  905 WRITE (ISYSWR,5006)                                                             COMAND 106
      WRITE (ISYSWR,5012)                                                             COMAND 107
```

```
      GO TO 50
C                                           . . . . . . . . . . . RESTORE
 1000 IT = WORD7(1) + 0.5
      CALL RESTOR(IT)
      GO TO 50
C                                           . . . . . . . . . . . END, EXIT
 1100 IT = WORD7(1) + 0.5
      IF (ISW(4) .EQ. 1 .OR. IT .GT. 0)  GO TO 1150
CSY   IFLAG = 3
CSY   WRITE (ISYSWR,5009)
CSY   CALL FCN(NPAR,G,F,U,IFLAG)
      NFCN = NFCN + 1
 1150 IF (I .EQ. 11)  STOP
      RETURN
C                                           . . . . . . . . . . . GRADIENT
 1200 ISW(3) = 1
      IF (WORD7(1) .GT. 0.0)  GO TO 50
      DO 1230 I= 1, NU
 1230 GF(I) = 0.0
      CALL INTOEX(X)
      CALL FCN(NPAR,GF,AMIN,U,2)
      NFCN = NFCN + 1
      CALL DERIVE(GF,G2)
      ISW(3) = 0
      CALL DERIVE(G, G2)
      WRITE (ISYSWR,5013)
      ISW(3) = 1
      DO 1250 I= 1, NU
      LC = LCORSP(I)
      IF (LC .EQ. 0)  GO TO 1250
      AWD = AGOOD
      IF (ABS(GF(LC)-G(LC)) .GT. ABS(G2(LC)))  AWD = ABAD
      IF (GF(LC) .EQ. 0.)  AWD = ANONE
      IF (AWD .NE. AGOOD)  ISW(3) = 0
      WRITE(ISYSWR,5014) I,NAM(I),GF(LC),G(LC),G2(LC),AWD
 1250 CONTINUE
      IF (ISW(3) .EQ. 0)  WRITE (ISYSWR,5015)
      GO TO 50
C                                           . . . . . . . . . . . CALL FCN
 1300 IFLAG = WORD7(1)
      IF (IFLAG .EQ. 3)  ISW(4) = 1
      CALL FCN(NPAR,G,F,U,IFLAG)
      NFCN = NFCN + 1
      IF(IFLAG.LE.5) GO TO 50
      CALL EXTOIN(X)
      CALL FCN(NPAR,G,AMIN,U,4)
      NFCN=NFCN+1
      IF (ISW(2) .LE. 1)  GO TO 1350
      ISW(2) = 1
      WRITE (ISYSWR, 5010)
 1350 CALL MPRINT(1,AMIN)
      GO TO 50
C                                           . . . . . . . . . . . MATOUT
 1400 IF(ISW(2).GE.2) GO TO 1405
C                                           . . . . . . . . . . . HESSE
 1402 CALL HESSE
      CALL MPRINT(1, AMIN)
```

```
 1405 CALL MATOUT(0.0 , 1)
      GO TO 50
C                                     . . . . . . . . . . . COVARIANCE
 1500 NRAPE = WORD7(1) + 0.5
      IF (NRAPE .NE. NPAR)  GO TO 1550
      READ (ISYSRD,5002)  ((V(I,J),I=1,NRAPE),J=1,NRAPE)
      ISW(2) = 3
      CALL MATOUT(0.0 , 1)
      CALL MPRINT(1,AMIN)
      GO TO 50
 1550 WRITE (ISYSWR,5006)
      WRITE (ISYSWR,5008)
      NRAP2 = NRAPE**2
      READ (ISYSRD,5002)  (G(1),I=1,NRAP2)
      GO TO 50
C                                     . . . . . . . . . . . IMPROVE
 1600 CONTINUE
      IF (ISW(2) .LT. 2)  GO TO 550
      CALL IMPROV
      IF (NEWMIN .EQ. 1)  GO TO 150
      GO TO 50
C                                     . . . . . . . . . . . ERROR DEF
 1700 CONTINUE
      UP = WORD7(1)
      IF (UP .LE. 0.)  UP = 1.0
      IF (ISW(2) .GE. 1)  CALL MPRINT(1,AMIN)
      GO TO 50
C                                     . . . . . . . . . . . PAGE
 1800 WRITE (ISYSWR,5005)
      GO TO 50
C                                     . . . . . . . . . . . CONTOUR
 2000 CONTINUE
      IF (ISW(2) .LT. 1)  GO TO 550
      KE1 = WORD7(1)
      KE2 = WORD7(2)
      IF (KE1 .LE. 0 .OR. KE2 .LE. 0)  GO TO 905
      IF (KE1 .GT. NU .OR. KE2 .GT. NU)  GO TO 905
      IF (LCORSP(KE1) .LE. 0 .OR. LCORSP(KE2) .LE. 0)  GO TO 905
      NFCNMX = 1000
      CALL CONTOU
      CALL MPRINT(1,AMIN)
      IF (NEWMIN .LE. 0)  GO TO 50
      GO TO 95
C                                     . . . . . . . . . . . STANDARD
 2100 CALL STAND
      GO TO 50
C                                     . . . . . . . . . . . RELEASE
 2200 CONTINUE
      DO 2220 IRL=1,7
      KRL = WORD7(IRL)
      IF (KRL .EQ. 0)  GO TO 50
      KRL = -IABS(KRL)
      CALL RESTOR(KRL)
 2220 CONTINUE
      GO TO 50
C                                     . . . . . . BLANK COMMAND CARD
 2300 CONTINUE
```

```
      GO TO 50                                                          COMAND
 5000 FORMAT (2A4, A2, 7F10.0)                                          COMAND
 5001 FORMAT (1H 10(1H*)/1H 3H*** I3,4H****,2A4,A2,7F15.5)              COMAND
 5002 FORMAT (7F10.0,10X)                                               COMAND
 5003 FORMAT (13E10.2)                                                  COMAND
 5004 FORMAT (1H 10(1H*))                                               COMAND
 5005 FORMAT (1H1)                                                      COMAND
 5006 FORMAT (1H+10X  17H(COMMAND IGNORED))                             COMAND
 5007 FORMAT ( 118H0THE ABOVE COMMAND CANNOT BE EXECUTED BECAUSE COVARIA COMAND
     1NCE MATRIX WAS NEITHER CALCULATED NOR SUPPLIED, OR WAS DESTROYED./ COMAND
     2/)                                                                COMAND
 5008 FORMAT( 102H SIZE OF COVARIANCE MATRIX TO BE READ DOES NOT CORRESP COMAND
     1OND  TO NUMBER OF CURRENTLY VARIABLE PARAMETERS./)                COMAND
 5009 FORMAT (  28H0CALL TO FCN WITH IFLAG = 3 /)                       COMAND
 5010 FORMAT (  51H NEW START POINT ASSUMED     COVARIANCE MATRIX LOST)  COMAND
 5011 FORMAT (  50H0NEW MINIMUM FOUND.  GO BACK TO MINIMIZATION STEP./1H COMAND
     1 ,         60(1H=)/60X1HV/60X1HV/60X1HV/57X7HVVVVVVV/58X5HVVVVV/59 COMAND
     2X          3HVVV/60X1HV///)                                       COMAND
 5012 FORMAT (  91H0THE ABOVE COMMAND CANNOT BE EXECUTED BECAUSE IT REQU COMAND
     1ESTS A PARAMETER THAT IS NOT VARIABLE.//)                         COMAND
 5013 FORMAT (  37H0CHECK OF GRADIENT CALCULATION IN FCN 12X    9HPARAME COMAND
     1TER          6X    9HG(IN FCN) 3X    9HG(MINUIT) 4X    5HERROR 5X  COMAND
     2  9HAGREEMENT)                                                    COMAND
 5014 FORMAT (45X I5,2X A10,3E12.4,4X A4)                               COMAND
 5015 FORMAT (  54H0MINUIT DOES NOT ACCEPT DERIVATIVE CALCULATIONS BY FC COMAND
     1N)                                                                COMAND
      END                                                               COMAND
```

```
      SUBROUTINE CONTOU                                                 CONTOU     2
CC        FINDS POINTS LYING ON CONTOURS OF A GIVEN FCN VALUE,          CONTOU     3
CC        AS A FUNCTION OF TWO VARIABLE PARAMETERS SPECIFIED BY THE     CONTOU     4
CC        CONTOUR COMMAND.   THE CONTOURS ARE THEN PLOTTED BY PLTCON.   CONTOU     5
CC                                                                      CONTOU     6
       COMMON                                                           COMMONU    2
     1/MINERR/ ERP(30)    ,ERN(30)                                      COMMONU    4
     2/PARINT/ X(15)      ,XT(15)     ,DIRIN(15)  ,MAXINT     ,NPAR     COMMONU    5
     3/PAREXT/ U(30)      ,NAM(30)    ,WERR(30)   ,MAXEXT     ,NU       COMMONU    6
     4/LIMITS/ ALIM(30)   ,BLIM(30)   ,LCODE(30)  ,LCORSP(30) ,LIMSET   COMMONU    7
     5/VARIAN/ V(15,15)                                                 COMMONU    8
     7/FIX   / IPFIX(15)  ,XS(15)     ,XTS(15)    ,WTS(15)   ,NPFIX     COMMONU    9
     C/CASC/ JH, JL, Y(16)                                              COMMONU   10
     F/DERIVA/ G(30)      ,G2(30)                                       COMMONU   11
     G/SIMVEC/ P(15,16)   ,PSTAR(15)  ,PSTST(15)  ,PBAR(15)   ,PRHO(15) COMMONU   12
     J/VARIAT/ VT(15,15)                                                COMMONU   13
     6/UNIT  / ISYSRD     ,ISYSWR     ,ISYSPU                           COMMONU   27
     8/TITLE / TITLE(13)  ,DATE(2)    ,ISW(7)     ,NBLOCK               COMMONU   28
     9/CONVER/ EPSI ,APSI ,VTEST      ,NSTEPQ     ,NFCN       ,NFCNMX   COMMONU   29
     A/CARD  / CWORD      ,CWORD2     ,CWORD3     ,WORD7(7)             COMMONU   30
     B/MINIMA/ AMIN       ,UP         ,NEWMIN     ,ITAUR      ,SIGMA    COMMONU   31
     K/COMMND/ NNAME      ,CNAME(25)  ,CNAM2(25)  ,CNAM3(25)            COMMONU   32
      COMMON /CONPTS/ NSPT, SPT(3,300)                                  CONTOU     8
      DIMENSION DIG(5)                                                  CONTOU     9
      DATA CROSS,ANEWMN,DIG / 1HX, 1H*, 1H1, 1H2, 1H3, 1H4, 1H5 /       CONTOU    10
      DATA NSTMX,THMAX,MINPT,TEN,NPTDIM / 10, 0.4, 10, 100., 298 /      CONTOU    11
      ALLOW(IEXT) = (U(IEXT)-ALIM(IEXT)) *(BLIM(IEXT)-U(IEXT))          CONTOU    12
      KE1 = WORD7(1)                                                    CONTOU    13
      KE2 = WORD7(2)                                                    CONTOU    14
      NCON = WORD7(3)                                                   CONTOU    15
      IF (NCON .LE. 0)  NCON = 2                                        CONTOU    16
      IF (NCON .GT. 5)  NCON=5                                          CONTOU    17
      SPT(1,1) = U(KE1)                                                 CONTOU    18
      SPT(2,1) = U(KE2)                                                 CONTOU    19
      SPT(3,1) = 1H                                                     CONTOU    20
      NSPT = 1                                                          CONTOU    21
      NEWMIN = 0                                                        CONTOU    22
      KI1 = LCORSP(KE1)                                                 CONTOU    23
      KI2 = LCORSP(KE2)                                                 CONTOU    24
      IF (ISW(2) .LT. 2)  GO TO 40                                      CONTOU    25
      CALL UCOPY(V,VT,MAXINT**2)                                        CONTOU    26
      CALL VERMIN(VT,MAXINT,MAXINT,NPAR,IERR)                           CONTOU    27
      IF (IERR .LE. 0)  GO TO 60                                        CONTOU    28
C             MATRIX NOT POSITIVE-DEFINITE                              CONTOU    29
      WRITE (ISYSWR,35)                                                 CONTOU    30
   35 FORMAT (  40H0COVARIANCE MATRIX NOT POSITIVE-DEFINITE)            CONTOU    31
   40 WRITE (ISYSWR,42)                                                 CONTOU    32
   42 FORMAT (  26H0ONLY DIAGONAL ERRORS USED)                          CONTOU    33
      A11 = V(KI1,KI1) * UP                                             CONTOU    34
      A22 = V(KI2,KI2) * UP                                             CONTOU    35
      A12 = 0.0                                                         CONTOU    36
      GO TO 75                                                          CONTOU    37
C          INVERT 2X2 SUBMATRIX BY HAND                                 CONTOU    38
   60 CONTINUE                                                          CONTOU    39
      DETR = VT(KI1,KI1)*VT(KI2,KI2) - VT(KI1,KI2)**2                   CONTOU    40
      IF (DETR .LE. 0.0)  GO TO 40                                      CONTOU    41
      DETR = UP/DETR                                                    CONTOU    42
```

```
      A11 = VT(KI2,KI2) * DETR
      A22 = VT(KI1,KI1) * DETR
      A12 =-VT(KI1,KI2) * DETR
   75 CONTINUE
      RHO = A12/SQRT(A11*A22)
      XMIN = U(KE1)
      YMIN = U(KE2)
      IF (LCODE(KE1) .GT. 1)  GO TO 110
      ALIM(KE1) = XMIN - TEN*WERR(KE1)
      BLIM(KE1) = XMIN + TEN*WERR(KE1)
      EXIN1 = 1.0
      GO TO 115
  110 EXIN1 = 0.5*(BLIM(KE1)-ALIM(KE1))*COS(X(KI1))
  115 CONTINUE
      IF (LCODE(KE2) .GT. 1)  GO TO 120
      ALIM(KE2) = YMIN - TEN*WERR(KE2)
      BLIM(KE2) = YMIN + TEN*WERR(KE2)
      EXIN2 = 1.0
      GO TO 125
  120 CONTINUE
      EXIN2 = 0.5* (BLIM(KE2)-ALIM(KE2)) * COS(X(KI2))
  125 CONTINUE
      A11 = A11 * EXIN1**2
      A22 = A22 * EXIN2**2
      A12 = A12 * ABS(EXIN1*EXIN2)
      CIRCUM = 4.0*SQRT(A11 + A22)
      IFLAG = 4
C                                      . CALCULATE THE CONTOURS
      DO 800 ICON= 1, NCON
      SYM = DIG(ICON)
      ANP = 50 - 5*(NCON-ICON)
      DCA = 6.2832/ANP
      RAVE = FLOAT(ICON) * CIRCUM / ANP
      RMAXAB = 3.0*RAVE
      RMIN = 0.1*RAVE
      DIFF = UP* FLOAT(ICON**2)
      AIM = AMIN + DIFF
      TOLERN = 0.05*DIFF
      NPTMX = 200
      IF (NPTDIM-NSPT .LT. NPTMX)  NPTMX = NPTDIM-NSPT
      NPT = 0
      IF (NPTMX .LT. 1)  GO TO 810
      WRITE (ISYSWR, 1000)  KE1, KE2, SYM,DIFF
C                                      . . SET UP FIRST POINT LONG ARM
      STEP2 = FLOAT(ICON) * SQRT(A11*(1.-RHO**2))
      U(KE1) = XMIN + STEP2
      IF (ALLOW(KE1) .LT. 0.)  GO TO 731
      CALL FCN(NPAR, G, F2, U, IFLAG)
      NFCN = NFCN + 1
      IF (F2 .LE. AMIN)  GO TO 741
      IF (ISW(5) .GE. 2)  WRITE (ISYSWR,900)
      THETB = 1.5708 - A12/SQRT(A11**2 + A12**2)
      THET1 = THETB
      R = 10.0 * RMAXAB
      XPT = U(KE1) - R*COS(THET1)
      YPT = U(KE2) - R*SIN(THET1)
      SLOPE = 2.0 * (F2-AMIN) / (STEP2* SIN(THET1))
```

```
      TOLER = TOLERN
      NSTEP = 0
      RMAX = RAVE
      GO TO 407
C                                       . . . . . . . . CUT STEP LENGTH R
  350 THET1 = 0.25 * (3.0*THETB + THET1)
      R = R * 0.5
      RMAX = R
      TOLER = TOLERN*R/RMAXAB
      SLOP = SLOPE
      NSTEP = 0
C                                       . . . . . . . FIND NEXT POINT
  400 U(KE1) = XPT + R*COS(THET1)
      U(KE2) = YPT + R*SIN(THET1)
      IF (ALLOW(KE1) .LT. 0. .OR. ALLOW(KE2) .LT. 0.)  GO TO 731
      CALL FCN(NPAR,G,F2,U,IFLAG)
      NFCN = NFCN + 1
      IF (F2 .LE. AMIN)  GO TO 741
C                                       . CHECK IF TWO TRIES ARE NEEDED
      F1 = F2
      THET2 = THET1
      AFA = ABS(F2-AIM)
      IF (AFA .GT. 10.*TOLER .AND. R .GT. RMIN)  GO TO 350
      IF (AFA .LT. 0.2*TOLER)  GO TO 420
  407 DTHET = (F2-AIM) / (R*SLOPE)
      THMIN = 0.001*RAVE/R
      IF (ABS(DTHET) .GT. THMAX)  DTHET = SIGN(THMAX,DTHET)
      IF (ABS(DTHET) .LT. THMIN)  DTHET = SIGN(THMIN,DTHET)
C                                       . SUBSEQUENT TRIALS FOR NEXT PT
  410 CONTINUE
      F1 = F2
      THET2 = THET1 + DTHET
      U(KE1) = XPT + R*COS(THET2)
      U(KE2) = YPT + R*SIN(THET2)
      IF (ALLOW(KE1) .LT. 0. .OR. ALLOW(KE2) .LT. 0.)  GO TO 731
      CALL FCN(NPAR,G,F2,U,IFLAG)
      NFCN = NFCN + 1
      IF (F2 .LE. AMIN)  GO TO 741
      SLOPE = (F1-F2)/(DTHET*R)
      IF (SLOPE .GT. 0.)  GO TO 420
      IF (NPT .GT. 0)  GO TO 415
      SLOPE = 2.0 * (AMIN-F2) /(SIN(THET2) *(XMIN-U(KE1)))
      GO TO 420
  415 IF (R .GT. RMIN)  GO TO 350
      SLOPE = SLOP
  420 CHANG = (F2-AIM)/(SLOPE*R)
      IF (ABS(CHANG) .GT. THMAX)  CHANG = SIGN(THMAX, CHANG)
      THETA = THET2 + CHANG
C                        CHECK IF BEST TRIAL POINT IS ACCURATE ENOUGH
      DELF = AMIN1(ABS(F2-AIM), ABS(F1-AIM))
      IF (DELF .LT. TOLER)  GO TO 430
      IF (NPT .EQ. 0)  GO TO 425
      IF (NSTEP .EQ. 1 .AND. R .GT. RMIN)  GO TO 350
  425 NSTEP = NSTEP + 1
      IF (NSTEP .GT. NSTMX)  GO TO 751
      THET1 = THET2
      DTHET = THETA - THET1
```

```
      GO TO 410
C                                         . . .  ACCEPT NEW POINT
  430 XPT = XPT + R*COS(THETA)
      YPT = YPT + R*SIN(THETA)
      U(KE1) = XPT
      U(KE2) = YPT
      IF (ALLOW(KE1) .LT. 0. .OR. ALLOW(KE2) .LT. 0.)  GO TO 731
      THETAB = ABS(THETA-THETB)
      NSTEP = 0
  440 IF (NPT .GT. 1)  GO TO 445
      THBEG = THETA
      XBEG = XPT
      YBEG = YPT
  445 CONTINUE
      NPT = NPT + 1
      NSPT = NSPT + 1
      SPT(1,NSPT) = XPT
      SPT(2,NSPT) = YPT
      SPT(3,NSPT) = SYM
      XPM = XPT - XMIN
      YPM = YPT - YMIN
      IF (ISW(5) .LT. 2)  GO TO 450
      THDEG = THETA*180. / 3.14159265
      THDEG = AMOD(THDEG,360.0)
      WRITE (ISYSWR,901) NPT,XPT,YPT,XPM,YPM,DELF,R,THDEG,NFCN
  450 CONTINUE
C                                         .  TEST IF CONTOUR COMPLETED
      IF (NPT .LT. MINPT)  GO TO 500
      IF ((YPT-YBEG)**2 +(XPT-XBEG)**2 .GT. R**2)  GO TO 500
      IF (COS(THETA-THBEG) .GT. 0.5)  GO TO 600
  500 IF (NPT .GE. NPTMX)  GO TO 761
C                                         . . . . .  ESTIMATE NEW POINT
      RINV = THETAB**2 / (R*DCA**2)
      IF (RINV .GT. 1.0/RMAX)  GO TO 522
      R2 = RMAX
      GO TO 540
  522 R2 = 1.0/RINV
      IF (R2 .LT. RMIN)  R2 = RMIN
  540 CHANG = R2*RINV*DCA
      IF (CHANG .GT. THMAX)  CHANG = THMAX
      IF (THETB .GT. THETA)  CHANG = -CHANG
      THET1 = THETA + CHANG
      THETB = THETA
      R = R2
      RMAX = AMIN1(2.0*R , RMAXAB)
      TOLER = TOLERN * R/RMAXAB
      SLOP = SLOPE
      GO TO 400
C                                         . CONTOUR COMPLETED NORMALLY
  600 WRITE (ISYSWR,605)  SYM,NPT
  605 FORMAT (   9H0CONTOUR A3,  18H IS COMPLETED WITHI5,   7H POINTS//)
      GO TO 790
C                                         . . . . .  ERROR RETURNS
  731 WRITE (ISYSWR, 735)  U(KE1), U(KE2)
  735 FORMAT (  34H0CONTOUR LEAVING ALLOWED REGION AT2E16.7)
      GO TO 780
  741 WRITE (ISYSWR, 745)
```

```
  745 FORMAT (  38H0NEW ABSOLUTE MINIMUM FOUND BY CONTOUR/)            CONTOU  214
      NEWMIN = 1                                                       CONTOU  215
      NSPT = NSPT + 1                                                  CONTOU  216
      SPT(1,NSPT) = U(KE1)                                             CONTOU  217
      SPT(2,NSPT) = U(KE2)                                             CONTOU  218
      SPT(3,NSPT) = ANEWMN                                             CONTOU  219
      AMIN = F2                                                        CONTOU  220
      CALL EXTOIN(X)                                                   CONTOU  221
      CALL MPRINT(0,AMIN)                                              CONTOU  222
      GO TO 810                                                        CONTOU  223
  751 WRITE (ISYSWR, 755)                                              CONTOU  224
  755 FORMAT (  39H0TOO MANY ITERATIONS TO FIND NEXT POINT)            CONTOU  225
      GO TO 780                                                        CONTOU  226
  761 WRITE (ISYSWR, 765)                                              CONTOU  227
  765 FORMAT (  45H0MEMORY OVERFLOW.  TOO MANY POINTS IN CONTOUR)      CONTOU  228
  780 WRITE (ISYSWR,1001)  NPT                                         CONTOU  229
  790 U(KE1) = XMIN                                                    CONTOU  230
      U(KE2) = YMIN                                                    CONTOU  231
  800 CONTINUE                                                         CONTOU  232
C                                      . . . . . END MAIN LOOP . . . . CONTOU  233
  810 CONTINUE                                                         CONTOU  234
      WRITE (ISYSWR,1002)  KE1,KE2,UP,KE2                              CONTOU  235
      CALL PLTCON(NSPT,SPT)                                            CONTOU  236
      WRITE (ISYSWR,1003)  KE1                                         CONTOU  237
      RETURN                                                           CONTOU  238
  900 FORMAT ( 8X    2HPT 9X    1HX 13X    1HY 10X    6HX-XMIN 6X    6HY CONTOU  239
     1-YMIN 7X    8HACCURACY 4X    10HSEPARATION 5X    5HTHETA 3X    5HCA CONTOU  240
     2LLS)                                                             CONTOU  241
  901 FORMAT (I10,2F14.7,2X,2F12.7,F13.6,2X, F12.7,F10.1,I8)           CONTOU  242
 1000 FORMAT (  15H0PARAMETERS NO.I4,   4H ANDI4,10X   8HCONTOUR A3,  12 CONTOU  243
     1H  F = FMIN +F11.6)                                              CONTOU  244
 1001 FORMAT (  55H CONTOUR NOT COMPLETED BECAUSE OF ABOVE ERROR CONDITI CONTOU  245
     1ON/1X 54(1H*),I5,  43H POINTS ON THE CONTOUR HAVE BEEN DETERMINED/ CONTOU  246
     2/)                                                               CONTOU  247
 1002 FORMAT(36H1  FUNCTION CONTOURS FOR PARAMETERS  2I4, 20X,         CONTOU  248
     1 20H  CONTOUR 1 = FMIN +    F10.5/12H0  PARAMETER  I3)           CONTOU  249
 1003 FORMAT (25H+                PARAMETER    I3,1H,/)                CONTOU  250
      END                                                              CONTOU  251
```

```
      SUBROUTINE  DERIVE(GG,GG2)                                        DERIVE     2
CC        CALCULATES THE FIRST DERIVATIVES OF FCN (GG),                 DERIVE     3
CC        EITHER BY FINITE DIFFERENCES OR BY TRANSFORMING THE USER-     DERIVE     4
CC        SUPPLIED DERIVATIVES TO INTERNAL COORDINATES,                 DERIVE     5
CC        ACCORDING TO WHETHER ISW(3) IS ZERO OR ONE.                   DERIVE     6
CC        IF ISW(3) = 0, AN ERROR ESTIMATE GG2  IS AVAILABLE            DERIVE     7
CC                                                                      DERIVE     8
       COMMON                                                           COMMONU    2
     1/MINERR/ ERP(30)     ,ERN(30)                                     COMMONU    4
     2/PARINT/ X(15)       ,XT(15)      ,DIRIN(15)  ,MAXINT     ,NPAR   COMMONU    5
     3/PAREXT/ U(30)       ,NAM(30)     ,WERR(30)   ,MAXEXT     ,NU     COMMONU    6
     4/LIMITS/ ALIM(30)    ,BLIM(30)    ,LCODE(30)  ,LCORSP(30) ,LIMSET COMMONU    7
     5/VARIAN/ V(15,15)                                                 COMMONU    8
     7/FIX   / IPFIX(15)   ,XS(15)      ,XTS(15)    ,WTS(15)   ,NPFIX   COMMONU    9
     C/CASC/ JH, JL, Y(16)                                              COMMONU   10
     F/DERIVA/ G(30)       ,G2(30)                                      COMMONU   11
     G/SIMVEC/ P(15,16)    ,PSTAR(15)   ,PSTST(15)  ,PBAR(15)   ,PRHO(15) COMMONU   12
     J/VARIAT/ VT(15,15)                                                COMMONU   13
     6/UNIT  / ISYSRD      ,ISYSWR      ,ISYSPU                         COMMONU   27
     8/TITLE / TITLE(13)   ,DATE(2)     ,ISW(7)     ,NBLOCK             COMMONU   28
     9/CONVER/ EPSI ,APSI  ,VTEST       ,NSTEPQ     ,NFCN       ,NFCNMX COMMONU   29
     A/CARD  / CWORD       ,CWORD2      ,CWORD3     ,WORD7(7)           COMMONU   30
     B/MINIMA/ AMIN        ,UP          ,NEWMIN     ,ITAUR      ,SIGMA  COMMONU   31
     K/COMMND/ NNAME       ,CNAME(25)   ,CNAM2(25)  ,CNAM3(25)          COMMONU   32
      DIMENSION GG(30),GG2(15),GY(30)                                   DERIVE    11
      IF (ISW(3) .EQ. 1)  GO TO 100                                     DERIVE    16
      IFLAG = 4                                                         DERIVE    17
      DO 46  I=1,NPAR                                                   DERIVE    18
      EPS = 0.1 * ABS(DIRIN(I))                                         DERIVE    19
      IF (ISW(2) .GE. 1) EPS = EPS + 0.005*SQRT(V(I,I)*UP)              DERIVE    20
      IF (EPS .LT.  1.0E-8*ABS(X(I)))  EPS = 1.0E-8*X(I)                DERIVE    21
      XTF = X(I)                                                        DERIVE    22
      X(I) = XTF + EPS                                                  DERIVE    23
      CALL INTOEX(X)                                                    DERIVE    24
      CALL FCN(NPAR,GY,FS1,U ,IFLAG)                                    DERIVE    25
      NFCN=NFCN+1                                                       DERIVE    26
      X(I) = XTF - EPS                                                  DERIVE    27
      CALL INTOEX(X)                                                    DERIVE    28
      CALL FCN(NPAR,GY,FS2,U,IFLAG)                                     DERIVE    29
      NFCN=NFCN+1                                                       DERIVE    30
C                                             . . . . . . . FIRST DERIVATIVE DERIVE    31
      GG(I)= (FS1-FS2)/(2.0*EPS)                                        DERIVE    32
C                                             . . . ERROR ON FIRST DERIVATIVE DERIVE    33
      GG2(I)= (FS1+FS2-2.0*AMIN)/(2.0*EPS)                              DERIVE    34
      X(I) = XTF                                                        DERIVE    35
   46 CONTINUE                                                          DERIVE    36
      CALL INTOEX(X)                                                    DERIVE    37
      GO TO 200                                                         DERIVE    38
C                                             .  DERIVATIVES CALC BY FCN DERIVE    39
  100 DO 150 I= 1, NU                                                   DERIVE    40
      LC=LCORSP(I)                                                      DERIVE    41
      IF (LC .LT. 1)  GO TO 150                                         DERIVE    42
      IF (LCODE(I) .GT. 1)  GO TO 120                                   DERIVE    43
      GG(LC)=GG(I)                                                      DERIVE    44
      GO TO 150                                                         DERIVE    45
  120 DD = (BLIM(I)-ALIM(I))*0.5 *COS(X(LC))                            DERIVE    46
      GG(LC)=GG(I)*DD                                                   DERIVE    47
```

```
  150 CONTINUE                                              DERIVE   48
  200 RETURN                                                DERIVE   49
      END                                                   DERIVE   50
```

```
              SUBROUTINE EXTOIN(PINT)                                        EXTOIN     2
CC               TRANSFORMS THE EXTERNAL PARAMETER VALUES X  TO INTERNAL     EXTOIN     3
CC               VALUES IN THE DENSE ARRAY PINT.   FUNCTION PINTF IS USED.   EXTOIN     4
CC                                                                           EXTOIN     5
               COMMON                                                        COMMONU    2
             1/MINERR/ ERP(30)     ,ERN(30)                                  COMMONU    4
             2/PARINT/ X(15)       ,XT(15)      ,DIRIN(15)  ,MAXINT     ,NPAR   COMMONU    5
             3/PAREXT/ U(30)       ,NAM(30)     ,WERR(30)   ,MAXEXT     ,NU     COMMONU    6
             4/LIMITS/ ALIM(30)    ,BLIM(30)    ,LCODE(30)  ,LCORSP(30) ,LIMSET COMMONU    7
             5/VARIAN/ V(15,15)                                              COMMONU    8
             7/FIX   / IPFIX(15)   ,XS(15)      ,XTS(15)    ,WTS(15)   ,NPFIX  COMMONU    9
             C/CASC/ JH, JL, Y(16)                                           COMMONU   10
             F/DERIVA/ G(30)       ,G2(30)                                   COMMONU   11
             G/SIMVEC/ P(15,16)    ,PSTAR(15)   ,PSTST(15)  ,PBAR(15)   ,PRHO(15) COMMONU   12
             J/VARIAT/ VT(15,15)                                             COMMONU   13
             6/UNIT  / ISYSRD      ,ISYSWR      ,ISYSPU                      COMMONU   27
             8/TITLE / TITLE(13)   ,DATE(2)     ,ISW(7)     ,NBLOCK          COMMONU   28
             9/CONVER/ EPSI ,APSI  ,VTEST       ,NSTEPQ     ,NFCN       ,NFCNMX COMMONU   29
             A/CARD  / CWORD       ,CWORD2      ,CWORD3     ,WORD7(7)        COMMONU   30
             B/MINIMA/ AMIN        ,UP          ,NEWMIN     ,ITAUR      ,SIGMA  COMMONU   31
             K/COMMND/ NNAME       ,CNAME(25)   ,CNAM2(25)  ,CNAM3(25)       COMMONU   32
              DIMENSION PINT(2)                                              EXTOIN     7
              LIMSET=0                                                       EXTOIN     8
              DO 100  I= 1, NU                                               EXTOIN     9
              J = LCORSP(I)                                                  EXTOIN    10
              IF ( J )  100,100,50                                           EXTOIN    11
           50 PINT(J) = PINTF(U(I),I)                                        EXTOIN    12
          100 CONTINUE                                                       EXTOIN    13
              RETURN                                                         EXTOIN    14
              END                                                            EXTOIN    15
```

```
      SUBROUTINE FIXPAR(I2,KODE,ILAX)                                   FIXPAR     2
CC        REMOVES PARAMETER I2 FROM THE INTERNAL (VARIABLE) PARAMETER   FIXPAR     3
CC        LIST, AND ARRANGES THE REST OF THE LIST TO FILL THE HOLE.     FIXPAR     4
CC        IF KODE=0, I2 IS AN EXTERNAL NUMBER, OTHERWISE INTERNAL.      FIXPAR     5
CC        ILAX IS RETURNED AS THE EXTERNAL NUMBER OF THE PARAMETER.     FIXPAR     6
CC                                                                      FIXPAR     7
       COMMON                                                           COMMONU    2
     1/MINERR/ ERP(30)   ,ERN(30)                                       COMMONU    4
     2/PARINT/ X(15)     ,XT(15)    ,DIRIN(15) ,MAXINT    ,NPAR         COMMONU    5
     3/PAREXT/ U(30)     ,NAM(30)   ,WERR(30)  ,MAXEXT    ,NU           COMMONU    6
     4/LIMITS/ ALIM(30)  ,BLIM(30)  ,LCODE(30) ,LCORSP(30) ,LIMSET      COMMONU    7
     5/VARIAN/ V(15,15)                                                 COMMONU    8
     7/FIX   / IPFIX(15) ,XS(15)    ,XTS(15)   ,WTS(15)  ,NPFIX         COMMONU    9
     C/CASC/ JH, JL, Y(16)                                              COMMONU   10
     F/DERIVA/ G(30)     ,G2(30)                                        COMMONU   11
     G/SIMVEC/ P(15,16)  ,PSTAR(15) ,PSTST(15) ,PBAR(15)  ,PRHO(15)     COMMONU   12
     J/VARIAT/ VT(15,15)                                                COMMONU   13
     6/UNIT  / ISYSRD    ,ISYSWR    ,ISYSPU                             COMMONU   27
     8/TITLE / TITLE(13) ,DATE(2)   ,ISW(7)    ,NBLOCK                  COMMONU   28
     9/CONVER/ EPSI ,APSI ,VTEST    ,NSTEPQ    ,NFCN      ,NFCNMX       COMMONU   29
     A/CARD  / CWORD     ,CWORD2    ,CWORD3    ,WORD7(7)                COMMONU   30
     B/MINIMA/ AMIN      ,UP        ,NEWMIN    ,ITAUR     ,SIGMA        COMMONU   31
     K/COMMND/ NNAME     ,CNAME(25) ,CNAM2(25) ,CNAM3(25)               COMMONU   32
      DIMENSION V1(2),YY(15)                                            FIXPAR    10
      EQUIVALENCE (V(1,1) , V1(1))                                      FIXPAR    15
      IF (KODE)  250,50,150                                             FIXPAR    16
C                                         . EXT PARAM NO. SPECIFIED     FIXPAR    17
   50 I = I2                                                            FIXPAR    18
      IF (I .GT. NU)  GO TO 70                                          FIXPAR    19
      IF (I .LT. 1)  GO TO 70                                           FIXPAR    20
   60 IF (LCORSP(I))  70,70,80                                          FIXPAR    21
C                                  ERROR RETURN   PARAM ALREADY FIXED   FIXPAR    22
   70 ILAX = 0                                                          FIXPAR    23
      WRITE (ISYSWR,500)  I                                             FIXPAR    24
      GO TO 300                                                         FIXPAR    25
   80 LC = LCORSP(I)                                                    FIXPAR    26
      IT = LC                                                           FIXPAR    27
      LCORSP(I) = 0                                                     FIXPAR    28
      ILAX = I                                                          FIXPAR    29
      NPAR = NPAR - 1                                                   FIXPAR    30
      NPFIX = NPFIX + 1                                                 FIXPAR    31
      IPFIX(NPFIX) = I                                                  FIXPAR    32
      XS(NPFIX) = X(LC)                                                 FIXPAR    33
      XTS(NPFIX) = XT(LC)                                               FIXPAR    34
      EPS = ABS(DIRIN(LC)) * 10.                                        FIXPAR    35
      IF (ISW(2) .GE. 1)  EPS = EPS + SQRT(ABS(V(LC,LC))*UP)            FIXPAR    36
      IF (EPS .LT. 1.0E-10*ABS(X(LC)))  EPS = 1.0E-8*X(LC)              FIXPAR    37
      WTS(NPFIX) = EPS*0.1                                              FIXPAR    38
      DO 100  IK= I, NU                                                 FIXPAR    39
      IF (LCORSP(IK))  100,100,85                                       FIXPAR    40
   85 LC = LCORSP(IK) - 1                                               FIXPAR    41
      LCORSP(IK) = LC                                                   FIXPAR    42
      X(LC) = X(LC+1)                                                   FIXPAR    43
      XT(LC) = XT(LC+1)                                                 FIXPAR    44
      DIRIN(LC) = DIRIN(LC+1)                                           FIXPAR    45
  100 CONTINUE                                                          FIXPAR    46
      IF (ISW(2) .GT. 1)  GO TO 250                                     FIXPAR    47
```

```
      ISW(2) = 0
      GO TO 300
C                                        . INT PARAM NO. SPECIFIED
  150 CONTINUE
      DO 200  IQ= 1, NU
      IF (LCORSP(IQ) .NE. I2)  GO TO 200
      I = IQ
      GO TO 60
  200 CONTINUE
      GO TO 70
C         REMOVE ONE ROW AND ONE COLUMN FROM VARIANCE MATRIX
  250 KON = 0
      IF (NPAR .LE. 0)  GO TO 300
      KON2 = 0
      MPAR = NPAR + 1
      DO 260 I= 1, MPAR
  260 YY(I)=V(I,IT)
      DO 294 I= 1, MPAR
      IF (I.EQ.IT)  GO TO 294
      KON2 = KON2 + 1
      DO 292 J= 1, MPAR
      IF (J .EQ. IT)  GO TO 292
      KON = KON + 1
      V1(KON)=V(J,I) - YY(J)*YY(I)/YY(IT)
  292 CONTINUE
      KON = MAXINT*KON2
  294 CONTINUE
C         CHECK FOR WELL-BEHAVED FINAL MATRIX
      DO 295 I= 1, NPAR
      IF (V(I,I) .LE. 0.)  GO TO 296
      DO 295 J= 1, NPAR
      IF (I .EQ. J)  GO TO 295
      IF (V(I,J)**2 .GE. V(I,I)*V(J,J))  V(I,J) = 0.
  295 CONTINUE
      GO TO 300
  296 ISW(2) = 0
      WRITE (ISYSWR, 501)
  300 RETURN
  500 FORMAT (  28H0ERROR IN FIXPAR.  PARAMETERI3,  18H WAS NOT VARIABLE
     1./)
  501 FORMAT (  72H0COVARIANCE MATRIX WAS ILL-CONDITIONED AND HAS BEEN D
     1ESTROYED BY FIXPAR./)
      END
```

```
      FUNCTION FTIME(BIDON)
CC        GIVES THE ELAPSED JOB TIME IN FLOATING-POINT MINUTES
CC        BY CALLING SOME INSTALLATION-DEPENDENT SUBROUTINE
CC
      CALL SECOND(SEC)
      FTIME=SEC
      RETURN
      END
```

```
      SUBROUTINE HESSE                                                  HESSE      2
CC        CALCULATES THE FULL SECOND-DERIVATIVE MATRIX OF FCN           HESSE      3
CC        BY TAKING FINITE DIFFERENCES.   INCLUDES SOME SAFEGUARDS      HESSE      4
CC        AGAINST NON-POSITIVE-DEFINITE MATRICES, AND IT MAY SET        HESSE      5
CC        OFF-DIAGONAL ELEMENTS TO ZERO IN ATTEMPT TO FORCE POSITIVENESS HESSE      6
CC                                                                      HESSE      7
       COMMON                                                           COMMONU    2
     1/MINERR/ ERP(30)     ,ERN(30)                                     COMMONU    4
     2/PARINT/ X(15)       ,XT(15)     ,DIRIN(15)  ,MAXINT     ,NPAR    COMMONU    5
     3/PAREXT/ U(30)       ,NAM(30)    ,WERR(30)   ,MAXEXT     ,NU      COMMONU    6
     4/LIMITS/ ALIM(30)    ,BLIM(30)   ,LCODE(30)  ,LCORSP(30) ,LIMSET  COMMONU    7
     5/VARIAN/ V(15,15)                                                 COMMONU    8
     7/FIX   / IPFIX(15)   ,XS(15)     ,XTS(15)    ,WTS(15)   ,NPFIX    COMMONU    9
     C/CASC/ JH, JL, Y(16)                                              COMMONU   10
     F/DERIVA/ G(30)       ,G2(30)                                      COMMONU   11
     G/SIMVEC/ P(15,16)    ,PSTAR(15)  ,PSTST(15)  ,PBAR(15)   ,PRHO(15) COMMONU   12
     J/VARIAT/ VT(15,15)                                                COMMONU   13
     6/UNIT  / ISYSRD      ,ISYSWR     ,ISYSPU                          COMMONU   27
     8/TITLE / TITLE(13)   ,DATE(2)    ,ISW(7)     ,NBLOCK              COMMONU   28
     9/CONVER/ EPSI  ,APSI ,VTEST      ,NSTEPQ     ,NFCN       ,NFCNMX  COMMONU   29
     A/CARD  / CWORD       ,CWORD2     ,CWORD3     ,WORD7(7)            COMMONU   30
     B/MINIMA/ AMIN        ,UP         ,NEWMIN     ,ITAUR      ,SIGMA   COMMONU   31
     K/COMMND/ NNAME       ,CNAME(25)  ,CNAM2(25)  ,CNAM3(25)           COMMONU   32
      DIMENSION YY(15),GY(30)                                           HESSE     10
      DATA DFWANT,DFZERO,DFMIN,DFMAX/ 0.01, 0.00000001, 0.001, 0.1 /    HESSE     15
      WRITE (ISYSWR, 500)                                               HESSE     16
      IFLAG = 4                                                         HESSE     17
      NPFN = NFCN                                                       HESSE     18
      NPARD = NPAR                                                      HESSE     19
C                                           . . . . . . DIAGONAL ELEMENTS . HESSE  20
      MDIAG = 0                                                         HESSE     21
      DO 100 ID= 1, NPARD                                               HESSE     22
      I = ID + NPAR - NPARD                                             HESSE     23
      D = 0.02* ABS(DIRIN(I))                                           HESSE     24
      IF (ISW(2) .GE. 1)  D = 0.02* SQRT(ABS(V(I,I))*UP)                HESSE     25
      IF (D .LT.  1.0E-8 *ABS(X(I)))  D = 1.0E-8 * ABS(X(I))            HESSE     26
      DO 20 J= 1, NPAR                                                  HESSE     27
   20 V(I,J) = 0.                                                       HESSE     28
      ICYC = 0                                                          HESSE     29
   40 DIRIN(I) = D                                                      HESSE     30
      XTF = X(I)                                                        HESSE     31
      X(I) = XTF + D                                                    HESSE     32
      CALL INTOEX(X)                                                    HESSE     33
      CALL FCN(NPAR, GY, FS1, U, IFLAG)                                 HESSE     34
      NFCN = NFCN + 1                                                   HESSE     35
      X(I) = XTF - D                                                    HESSE     36
      CALL INTOEX(X)                                                    HESSE     37
      CALL FCN (NPAR, GY, FS2, U, IFLAG)                                HESSE     38
      NFCN = NFCN + 1                                                   HESSE     39
      X(I) = XTF                                                        HESSE     40
C                                          CHECK IF STEP SIZES APPROPRIATE HESSE  41
      ICYC = ICYC + 1                                                   HESSE     42
      IF (ICYC .GE. 4)  GO TO 55                                        HESSE     43
      DF = AMAX1(ABS(FS1-AMIN),ABS(FS2-AMIN))/UP                        HESSE     44
      IF (DF .GT. DFMIN)  GO TO 45                                      HESSE     45
      IF (DF .GT. DFZERO)  GO TO 50                                     HESSE     46
      D = D*1000.                                                       HESSE     47
```

```
      GO TO 40                                                          HESSE  48
   45 IF (DF .LT. DFMAX)  GO TO 55                                      HESSE  49
   50 CHAN = SQRT(DFWANT/DF)                                            HESSE  50
      IF (CHAN .LT. 0.001)  CHAN = 0.001                                HESSE  51
      D = D*CHAN                                                        HESSE  52
      GO TO 40                                                          HESSE  53
   55 CONTINUE                                                          HESSE  54
C                                       GET FIRST AND SECOND DERIVATIVE HESSE  55
      G(I) = (FS1-FS2)/(2.0 * D)                                        HESSE  56
      G2(I) = (FS1 + FS2 - 2.0*AMIN) / D**2                             HESSE  57
      YY(I) = FS1                                                       HESSE  58
      IF (ABS(G(I))+ ABS(G2(I)).GT. 1.0E-30)  GO TO 80                  HESSE  59
C                                     FIX A PARAMETER IF  G = G2 = 0.0 . HESSE  60
      IF (ITAUR .GE. 1)  GO TO 85                                       HESSE  61
      ISW(2) = 0                                                        HESSE  62
      CALL FIXPAR(I, 1, IFIX)                                           HESSE  63
      WRITE (ISYSWR, 460) IFIX, NAM(IFIX), G(I), G2(I)                  HESSE  64
      IF (NPAR .EQ. 0)  MDIAG = 1                                       HESSE  65
      GO TO 100                                                         HESSE  66
   80 IF (G2(I) .GT. 1.0E-30)  GO TO 90                                 HESSE  67
   85 MDIAG = 1                                                         HESSE  68
      WRITE (ISYSWR,510)  I                                             HESSE  69
   90 V(I,I) = G2(I)                                                    HESSE  70
  100 CONTINUE                                                          HESSE  71
      CALL INTOEX(X)                                                    HESSE  72
      IF (MDIAG .EQ. 1)  GO TO 390                                      HESSE  73
      ISW(2) = 1                                                        HESSE  74
C                                       . . . .  OFF-DIAGONAL ELEMENTS  HESSE  75
      IF (NPAR .EQ. 1)  GO TO 214                                       HESSE  76
      NPARM1 = NPAR - 1                                                 HESSE  77
      DO 200 I= 1, NPARM1                                               HESSE  78
      IP1 = I + 1                                                       HESSE  79
      DO 180 J= IP1, NPAR                                               HESSE  80
      IF (NFCNMX-NFCN+NPFN .LT. NPAR)  GO TO 210                        HESSE  81
      XTI = X(I)                                                        HESSE  82
      XTJ = X(J)                                                        HESSE  83
      X(I) = XTI + DIRIN(I)                                             HESSE  84
      X(J) = XTJ + DIRIN(J)                                             HESSE  85
      CALL INTOEX(X)                                                    HESSE  86
      CALL FCN(NPAR, GY, FS1, U, IFLAG)                                 HESSE  87
      NFCN = NFCN + 1                                                   HESSE  88
      X(I) = XTI                                                        HESSE  89
      X(J) = XTJ                                                        HESSE  90
      ELEM = (FS1+AMIN-YY(I)-YY(J)) / (DIRIN(I)*DIRIN(J))               HESSE  91
      IF (ELEM**2 .LT. G2(I)*G2(J)) GO TO 170                           HESSE  92
      ELEM = 0.                                                         HESSE  93
      WRITE (ISYSWR, 470)  I,J                                          HESSE  94
  170 V(I,J) = ELEM                                                     HESSE  95
      V(J,I) = ELEM                                                     HESSE  96
  180 CONTINUE                                                          HESSE  97
  200 CONTINUE                                                          HESSE  98
      GO TO 214                                                         HESSE  99
  210 J = J - 1                                                         HESSE 100
      WRITE (ISYSWR, 490)  I,J                                          HESSE 101
  214 CALL INTOEX(X)                                                    HESSE 102
      CALL VERMIN(V,MAXINT,MAXINT,NPAR,IFAIL)                           HESSE 103
      IF (IFAIL .LT. 1)  GO TO 222                                      HESSE 104
```

```
      WRITE (ISYSWR,520)
C                                         . . DIAGONAL MATRIX ONLY . . .
  216 WRITE (ISYSWR,540)
      ISW(2) = 1
      DO 220 I= 1, NPAR
      DO 218 J= 1, NPAR
  218 V(I,J) = 0.
  220 V(I,I) = 1.0/G2(I)
      MDIAG = 0
      GO TO 223
  222 WRITE (ISYSWR, 480)
      ISW(2) = 2
C                                         . . . . . . . . CALCULATE  E D M
  223 DO 225 I= 1, NPAR
      DO 225 J= 1, NPAR
  225 V(I,J) = 2.0 * V(I,J)
      SIGMA = 0.
      DO 250 I= 1, NPAR
      IF (V(I,I) .GT. 0.)  GO TO 228
      WRITE (ISYSWR,510)  I
      MDIAG = 1
  228 R = 0.
      DO 240 J= 1, NPAR
      IF (I .EQ. J)  GO TO 230
      IF (V(I,J)**2 .LT. ABS(V(I,I)*V(J,J)))  GO TO 230
      WRITE (ISYSWR, 470)  I,J
      V(I,J) = 0.
      V(J,I) = 0.
  230 CONTINUE
  240 R = R + V(I,J) * G(J)
  250 SIGMA = SIGMA + 0.5 *R *G(I)
      IF (MDIAG .EQ. 1)  GO TO 390
      IF (SIGMA .GT. 0.)  GO TO 400
      WRITE (ISYSWR,530)
      GO TO 216
  390 ISW(2) = 0
  400 RETURN
  460 FORMAT(10H0PARAMETER I3,2H, A10,  16H, HAS BEEN FIXED 9X   19HFIRS
     1T DERIVATIVE IS E11.3/1X40(1H*),8X   20HSECOND DERIVATIVE IS E11.3
     2/)
  470 FORMAT (  71H COVARIANCE MATRIX NOT POSITIVE-DEFINITE.    FAULTY E
     1LEMENT IN POSITION2I3)
  480 FORMAT (  35H SECOND DERIVATIVE MATRIX INVERTED )
  490 FORMAT (  76H0CALL LIMIT IN HESSE.  OFF-DIAGONAL ELEMENTS CALCULAT
     1ED  ONLY UP TO POSITION2I3/)
  500 FORMAT (  37H START SECOND DERIVATIVE CALCULATION )
  510 FORMAT (  17H DIAGONAL ELEMENT I5,   20H IS ZERO OR NEGATIVE)
  520 FORMAT (  23H MATRIX INVERSION FAILS)
  530 FORMAT (  29H MATRIX NOT POSITIVE-DEFINITE)
  540 FORMAT (  30H ONLY DIAGONAL MATRIX PRODUCED)
      END
```

```
      SUBROUTINE IMPROV                                                 IMPROV     2
CC        ATTEMPTS TO IMPROVE ON A GOOD LOCAL MINIMUM BY FINDING A      IMPROV     3
CC        BETTER ONE.   THE QUADRATIC PART OF FCN IS REMOVED BY CALFCN  IMPROV     4
CC        AND THIS TRANSFORMED FUNCTION IS MINIMIZED USING THE SIMPLEX  IMPROV     5
CC        METHOD FROM SEVERAL RANDOM STARTING POINTS.                   IMPROV     6
CC        REF. -- GOLDSTEIN AND PRICE, MATH.COMP. 25, 569 (1971)        IMPROV     7
CC                                                                      IMPROV     8
       COMMON                                                           COMMONU    2
     1/MINERR/ ERP(30)     ,ERN(30)                                     COMMONU    4
     2/PARINT/ X(15)       ,XT(15)      ,DIRIN(15)  ,MAXINT     ,NPAR   COMMONU    5
     3/PAREXT/ U(30)       ,NAM(30)     ,WERR(30)   ,MAXEXT     ,NU     COMMONU    6
     4/LIMITS/ ALIM(30)    ,BLIM(30)    ,LCODE(30)  ,LCORSP(30) ,LIMSET COMMONU    7
     5/VARIAN/ V(15,15)                                                 COMMONU    8
     7/FIX   / IPFIX(15)   ,XS(15)      ,XTS(15)    ,WTS(15)   ,NPFIX   COMMONU    9
     C/CASC/ JH, JL, Y(16)                                              COMMONU   10
     F/DERIVA/ G(30)       ,G2(30)                                      COMMONU   11
     G/SIMVEC/ P(15,16)    ,PSTAR(15)   ,PSTST(15)  ,PBAR(15)   ,PRHO(15) COMMONU   12
     J/VARIAT/ VT(15,15)                                                COMMONU   13
     6/UNIT  / ISYSRD      ,ISYSWR      ,ISYSPU                         COMMONU   27
     8/TITLE / TITLE(13)   ,DATE(2)     ,ISW(7)     ,NBLOCK             COMMONU   28
     9/CONVER/ EPSI  ,APSI  ,VTEST      ,NSTEPQ     ,NFCN       ,NFCNMX COMMONU   29
     A/CARD  / CWORD       ,CWORD2      ,CWORD3     ,WORD7(7)           COMMONU   30
     B/MINIMA/ AMIN        ,UP          ,NEWMIN     ,ITAUR      ,SIGMA  COMMONU   31
     K/COMMND/ NNAME       ,CNAME(25)   ,CNAM2(25)  ,CNAM3(25)          COMMONU   32
      DATA ALPHA,BETA,GAMMA / 1.0, 0.5 , 2.0 /                          IMPROV    10
      IF (NPAR .LE. 0)  RETURN                                          IMPROV    11
      ITAUR = 1                                                         IMPROV    12
      EPSI = 0.1*UP                                                     IMPROV    13
      NPFN=NFCN                                                         IMPROV    14
      NLOOP = WORD7(2)                                                  IMPROV    15
      IF (NLOOP .LE. 0)  NLOOP = NPAR + 4                               IMPROV    16
      NPARP1=NPAR+1                                                     IMPROV    17
      WG = 1.0/FLOAT(NPAR)                                              IMPROV    18
      IFLAG=4                                                           IMPROV    19
      SIGSAV = SIGMA                                                    IMPROV    20
      APSI = AMIN                                                       IMPROV    21
      DO 2 I= 1, NPAR                                                   IMPROV    22
      XT(I) = X(I)                                                      IMPROV    23
      G2(I) = SQRT(V(I,I)*UP)                                           IMPROV    24
      DO 2 J= 1, NPAR                                                   IMPROV    25
    2 VT(I,J) = V(I,J)                                                  IMPROV    26
      CALL VERMIN(VT,MAXINT,MAXINT,NPAR,IFAIL)                          IMPROV    27
      IF (IFAIL .GE. 1)  GO TO 280                                      IMPROV    28
      LOOP = 0                                                          IMPROV    29
    3 DO 4 I= 1, NPAR                                                   IMPROV    30
      DIRIN(I) = 2.0*G2(I)                                              IMPROV    31
    4 X(I)= XT(I)+2.0*DIRIN(I)*(RGEN(I)-0.5)                            IMPROV    32
      LOOP = LOOP + 1                                                   IMPROV    33
      REG = 2.0                                                         IMPROV    34
      WRITE (ISYSWR, 1040) LOOP                                         IMPROV    35
    5 AMIN = CALFCN(X)                                                  IMPROV    36
C                                         . . . . SET UP  RANDOM SIMPLEX IMPROV    37
      JL = NPARP1                                                       IMPROV    38
      JH = NPARP1                                                       IMPROV    39
      Y(NPARP1) = AMIN                                                  IMPROV    40
      AMAX = AMIN                                                       IMPROV    41
      DO 15 I= 1, NPAR                                                  IMPROV    42
```

```
      XI = X(I)
      X(I)= XI-DIRIN(I)*(RGEN(I)-0.5)
      Y(I) = CALFCN(X)
      IF (Y(I) .GE. AMIN)  GO TO 7
      AMIN = Y(I)
      JL = I
    7 IF (Y(I) .LE. AMAX)  GO TO 8
      AMAX = Y(I)
      JH = I
    8 CONTINUE
      DO 10 J= 1, NPAR
   10 P(J,I) = X(J)
      P(I,NPARP1) = XI
   15 X(I) = XI
      SIGMA = AMIN
      SIG2 = SIGMA
C                                  . . . . . . . .  START MAIN LOOP
   50 CONTINUE
      IF (AMIN .LT. 0.)  GO TO 95
      IF (ISW(2) .LT. 2)  GO TO 280
      EP = 0.1*AMIN
      IF (SIG2 .LT. EP   .AND. SIGMA.LT.EP  )     GO TO 100
      SIG2 = SIGMA
      IF ((NFCN-NPFN) .GT. NFCNMX)  GO TO 300
C         CALCULATE NEW POINT * BY REFLECTION
      DO 60 I= 1, NPAR
      PB = 0.
      DO 59 J= 1, NPARP1
   59 PB = PB + WG * P(I,J)
      PBAR(I) = PB - WG * P(I,JH)
   60 PSTAR(I)=(1.+ALPHA)*PBAR(I)-ALPHA*P(I,JH)
      YSTAR = CALFCN(PSTAR)
      IF(YSTAR.GE.AMIN) GO TO 70
C         POINT * BETTER THAN JL, CALCULATE NEW POINT **
      DO 61 I=1,NPAR
   61 PSTST(I)=GAMMA*PSTAR(I)+(1.-GAMMA)*PBAR(I)
      YSTST = CALFCN(PSTST)
   66 IF (YSTST .LT. Y(JL))  GO TO 67
      CALL RAZZIA(YSTAR,PSTAR)
      GO TO 50
   67 CALL RAZZIA(YSTST,PSTST)
      GO TO 50
C         POINT * IS NOT AS GOOD AS JL
   70 IF (YSTAR .GE. Y(JH))  GO TO 73
      JHOLD = JH
      CALL RAZZIA(YSTAR,PSTAR)
      IF (JHOLD .NE. JH)  GO TO 50
C         CALCULATE NEW POINT **
   73 DO 74 I=1,NPAR
   74 PSTST(I)=BETA*P(I,JH)+(1.-BETA)*PBAR(I)
      YSTST = CALFCN(PSTST)
      IF(YSTST.GT.Y(JH)) GO TO 5
C     POINT ** IS BETTER THAN JH
      IF (YSTST .LT. AMIN)  GO TO 67
      CALL RAZZIA(YSTST,PSTST)
      GO TO 50
C                                  . . . . . . .  END MAIN LOOP
```

```
   95 WRITE (ISYSWR,1000)
      REG = 0.1
C                                  . . . . . ASK IF POINT IS NEW
  100 CALL INTOEX(X)
      CALL FCN(NPAR,G,AMIN,U,4)
      NFCN = NFCN + 1
      DO 120 I= 1, NPAR
      DIRIN(I) = REG*G2(I)
      IF (ABS(X(I)-XT(I)) .GT. DIRIN(I)) GO TO 150
  120 CONTINUE
      GO TO 230
  150 NFCNMX = NFCNMX + NPFN - NFCN
      NPFN = NFCN
      CALL SIMPLX
      IF (AMIN .GE. APSI)  GO TO 325
      DO 220 I= 1, NPAR
      DIRIN(I) = 0.1 *G2(I)
      IF (ABS(X(I)-XT(I)) .GT. DIRIN(I)) GO TO 250
  220 CONTINUE
  230 IF (AMIN .LT. APSI)  GO TO 350
      GO TO 325
C                                  . . . . . . . TRULY NEW MINIMUM
  250 NEWMIN = 1
      ISW(2) = 0
      ITAUR = 0
      NFCNMX = NFCNMX + NPFN - NFCN
      WRITE (ISYSWR,1030)
      RETURN
C                                  . . . RETURN TO PREVIOUS REGION
  280 WRITE (ISYSWR,1020)
      ISW(2) = 0
      GO TO 325
  300 ISW(1) = 1
  325 DO 330 I= 1, NPAR
      DIRIN(I) = 0.01*G2(I)
  330 X(I) = XT(I)
      AMIN = APSI
      SIGMA = SIGSAV
  350 CALL INTOEX(X)
      WRITE (ISYSWR,1010)
      IF (ISW(2) .LT. 2)  GO TO 380
      IF (LOOP .LT. NLOOP .AND. ISW(1) .LT. 1)  GO TO 3
      ISW(2) = 3
  380 CALL MPRINT (1,AMIN)
      RETURN
 1000 FORMAT (  54H AN IMPROVEMENT ON THE PREVIOUS MINIMUM HAS BEEN FOUN
     1D)
 1010 FORMAT (  51H IMPROVE HAS RETURNED TO REGION OF ORIGINAL MINIMUM)
 1020 FORMAT (  44H0COVARIANCE MATRIX WAS NOT POSITIVE-DEFINITE)
 1030 FORMAT (  38H0IMPROVE HAS FOUND A TRULY NEW MINIMUM/1H 37(1H*)/)
 1040 FORMAT (  18H0START ATTEMPT NO.I2,  20H TO FIND NEW MINIMUM)
      END
```

```
      SUBROUTINE INTOEX(PINT)                                           INTOX      2
CC        TRANSFORMS FROM INTERNAL COORDINATES (PINT) TO EXTERNAL       INTOX      3
CC        PARAMETERS (U).   THE MINIMIZING ROUTINES WHICH WORK IN       INTOX      4
CC        INTERNAL COORDINATES CALL THIS ROUTINE BEFORE CALLING FCN.    INTOX      5
       COMMON                                                           COMMONU    2
     1/MINERR/ ERP(30)     ,ERN(30)                                     COMMONU    4
     2/PARINT/ X(15)       ,XT(15)      ,DIRIN(15)  ,MAXINT     ,NPAR   COMMONU    5
     3/PAREXT/ U(30)       ,NAM(30)     ,WERR(30)   ,MAXEXT     ,NU     COMMONU    6
     4/LIMITS/ ALIM(30)    ,BLIM(30)    ,LCODE(30)  ,LCORSP(30) ,LIMSET COMMONU    7
     5/VARIAN/ V(15,15)                                                 COMMONU    8
     7/FIX   / IPFIX(15)   ,XS(15)      ,XTS(15)    ,WTS(15)   ,NPFIX   COMMONU    9
     C/CASC/ JH, JL, Y(16)                                              COMMONU   10
     F/DERIVA/ G(30)       ,G2(30)                                      COMMONU   11
     G/SIMVEC/ P(15,16)    ,PSTAR(15)   ,PSTST(15)  ,PBAR(15)   ,PRHO(15) COMMONU 12
     J/VARIAT/ VT(15,15)                                                COMMONU   13
     6/UNIT  / ISYSRD      ,ISYSWR      ,ISYSPU                         COMMONU   27
     8/TITLE / TITLE(13)   ,DATE(2)     ,ISW(7)     ,NBLOCK             COMMONU   28
     9/CONVER/ EPSI  ,APSI ,VTEST       ,NSTEPQ     ,NFCN       ,NFCNMX COMMONU   29
     A/CARD  / CWORD       ,CWORD2      ,CWORD3     ,WORD7(7)           COMMONU   30
     B/MINIMA/ AMIN        ,UP          ,NEWMIN     ,ITAUR      ,SIGMA  COMMONU   31
     K/COMMND/ NNAME       ,CNAME(25)   ,CNAM2(25)  ,CNAM3(25)          COMMONU   32
      DIMENSION PINT(2)                                                 INTOX      7
      DO 100 I= 1, NU                                                   INTOX      8
      J = LCORSP(I)                                                     INTOX      9
      IF ( J )  100,100,50                                              INTOX     10
   50 CONTINUE                                                          INTOX     11
      IF (LCODE(I) .EQ. 1)  GO TO 80                                    INTOX     12
      AL = ALIM(I)                                                      INTOX     13
      U(I) = AL + 0.5 *(SIN(PINT(J)) +1.0) * (BLIM(I) -AL)              INTOX     14
      GO TO 100                                                         INTOX     15
   80 U(I) = PINT(J)                                                    INTOX     16
  100 CONTINUE                                                          INTOX     17
      RETURN                                                            INTOX     18
      END                                                               INTOX     19
```

```
      SUBROUTINE MATOUT(TRACE,KODE)                                     MATOUT   2
CC        PRINTS THE COVARIANCE MATRIX V.     CALCULATES AND PRINTS THE MATOUT   3
CC        INDIVIDUAL CORRELATION COEFFICIENTS AND GLOBAL CORRELATIONS.  MATOUT   4
CC                                                                      MATOUT   5
       COMMON                                                           COMMONU  2
     1/MINERR/ ERP(30)    ,ERN(30)                                      COMMONU  4
     2/PARINT/ X(15)      ,XT(15)     ,DIRIN(15)  ,MAXINT     ,NPAR     COMMONU  5
     3/PAREXT/ U(30)      ,NAM(30)    ,WERR(30)   ,MAXEXT     ,NU       COMMONU  6
     4/LIMITS/ ALIM(30)   ,BLIM(30)   ,LCODE(30)  ,LCORSP(30) ,LIMSET   COMMONU  7
     5/VARIAN/ V(15,15)                                                 COMMONU  8
     7/FIX   / IPFIX(15)  ,XS(15)     ,XTS(15)    ,WTS(15)   ,NPFIX     COMMONU  9
     C/CASC/ JH, JL, Y(16)                                              COMMONU 10
     F/DERIVA/ G(30)      ,G2(30)                                       COMMONU 11
     G/SIMVEC/ P(15,16)   ,PSTAR(15)  ,PSTST(15)  ,PBAR(15)   ,PRHO(15) COMMONU 12
     J/VARIAT/ VT(15,15)                                                COMMONU 13
     6/UNIT  / ISYSRD     ,ISYSWR     ,ISYSPU                           COMMONU 27
     8/TITLE / TITLE(13)  ,DATE(2)    ,ISW(7)     ,NBLOCK               COMMONU 28
     9/CONVER/ EPSI  ,APSI ,VTEST     ,NSTEPQ     ,NFCN       ,NFCNMX   COMMONU 29
     A/CARD  / CWORD      ,CWORD2     ,CWORD3     ,WORD7(7)             COMMONU 30
     B/MINIMA/ AMIN       ,UP         ,NEWMIN     ,ITAUR      ,SIGMA    COMMONU 31
     K/COMMND/ NNAME      ,CNAME(25)  ,CNAM2(25)  ,CNAM3(25)            COMMONU 32
      DIMENSION VLINE(15)                                               MATOUT   8
      IF (ISW(2) .LT. 2)  RETURN                                        MATOUT  13
      WRITE (ISYSWR,600)                                                MATOUT  14
      IF (TRACE .NE. 0.0)  WRITE (ISYSWR,610)  TRACE                    MATOUT  15
      IF (NPAR .EQ. 0)  GO TO 250                                       MATOUT  16
      ISWTR = ISW(5) - ITAUR                                            MATOUT  17
      IF (ISWTR .LT. 2)  GO TO 120                                      MATOUT  18
C                                     . . INTERNAL COVARIANCE MATRIX    MATOUT  19
      DO 100 I= 1, NPAR                                                 MATOUT  20
  100 WRITE (ISYSWR,620)  (V(I,J), J=1,I)                               MATOUT  21
      WRITE (ISYSWR,630)                                                MATOUT  22
  120 CONTINUE                                                          MATOUT  23
C                                     . . . . . CORRELATION COEFFS. .   MATOUT  24
      IF (KODE .LT. 1)  GO TO 500                                       MATOUT  25
      IF (NPAR .LE. 1)  GO TO 500                                       MATOUT  26
      WRITE (ISYSWR, 650)                                               MATOUT  27
      NPARM = MIN0(NPAR-1, 18)                                          MATOUT  28
      WRITE (ISYSWR,690)  (ID,ID=1,NPARM)                               MATOUT  29
      DO 200 I= 2, NPAR                                                 MATOUT  30
      IM = I-1                                                          MATOUT  31
      DO 170 J= 1, IM                                                   MATOUT  32
  170 VLINE(J) = V(I,J)/SQRT(ABS(V(I,I)*V(J,J)))                        MATOUT  33
  200 WRITE (ISYSWR,660)  I,(VLINE(IZ),IZ=1,IM)                         MATOUT  34
  250 CONTINUE                                                          MATOUT  35
      WRITE (ISYSWR, 630)                                               MATOUT  36
C                                     GLOBAL CORRELATION COEFFS . . .   MATOUT  37
      CALL UCOPY(V,P,MAXINT**2)                                         MATOUT  38
      CALL VERMIN(P,MAXINT,MAXINT,NPAR,IERR)                            MATOUT  39
      IF(IERR .GT. 0)  RETURN                                           MATOUT  40
      WRITE (ISYSWR,670)                                                MATOUT  41
      DO 400 I= 1, NU                                                   MATOUT  42
      L = LCORSP(I)                                                     MATOUT  43
      IF (L .EQ. 0)  GO TO 400                                          MATOUT  44
      GCC = 1.0 - 1.0/(V(L,L)*P(L,L))                                   MATOUT  45
      WRITE(ISYSWR,680)  I, NAM(I), GCC                                 MATOUT  46
  400 CONTINUE                                                          MATOUT  47
```

```
  500 RETURN
  600 FORMAT(/11X  26HINTERNAL COVARIANCE MATRIX)
  610 FORMAT (1H+43X  28H LAST FRACTIONAL CHANGE WAS  F10.6)
  620 FORMAT (1H 9X11E10.3)
  630 FORMAT (1H )
  650 FORMAT (11X  24HCORRELATION COEFFICIENTS)
  660 FORMAT (8X,I3,18F6.3/(20F6.3))
  670 FORMAT (64X   18HGLOBAL CORRELATION/ 54X   25HPARAMETER     COEFFI
     1CIENT)
  680 FORMAT (47X,I5,2X,A10,F13.5)
  690 FORMAT (8X,4HINT.,I4,17I6)
      END
```

```
      SUBROUTINE MIDATA                                                  MIDATA     2
CC        READS THE DATA CARDS (TITLE CARD AND PARAMETER CARDS)          MIDATA     3
CC        AND SETS UP THE STARTING PARAMETER LISTS.                      MIDATA     4
CC        CONTROL THEN PASSES TO COMAND FOR READING THE COMMAND CARDS.   MIDATA     5
CC                                                                       MIDATA     6
       COMMON                                                            COMMONU    2
     1/MINERR/ ERP(30)      ,ERN(30)                                     COMMONU    4
     2/PARINT/ X(15)        ,XT(15)      ,DIRIN(15)  ,MAXINT     ,NPAR   COMMONU    5
     3/PAREXT/ U(30)        ,NAM(30)     ,WERR(30)   ,MAXEXT     ,NU     COMMONU    6
     4/LIMITS/ ALIM(30)     ,BLIM(30)    ,LCODE(30)  ,LCORSP(30) ,LIMSET COMMONU    7
     5/VARIAN/ V(15,15)                                                  COMMONU    8
     7/FIX   / IPFIX(15)    ,XS(15)      ,XTS(15)    ,WTS(15)   ,NPFIX   COMMONU    9
     C/CASC/ JH, JL, Y(16)                                               COMMONU   10
     F/DERIVA/ G(30)        ,G2(30)                                      COMMONU   11
     G/SIMVEC/ P(15,16)     ,PSTAR(15)   ,PSTST(15)  ,PBAR(15)   ,PRHO(15) COMMONU 12
     J/VARIAT/ VT(15,15)                                                 COMMONU   13
     6/UNIT  / ISYSRD       ,ISYSWR      ,ISYSPU                         COMMONU   27
     8/TITLE / TITLE(13)    ,DATE(2)     ,ISW(7)     ,NBLOCK             COMMONU   28
     9/CONVER/ EPSI  ,APSI  ,VTEST       ,NSTEPQ     ,NFCN       ,NFCNMX COMMONU   29
     A/CARD  / CWORD        ,CWORD2      ,CWORD3     ,WORD7(7)           COMMONU   30
     B/MINIMA/ AMIN         ,UP          ,NEWMIN     ,ITAUR      ,SIGMA  COMMONU   31
     K/COMMND/ NNAME        ,CNAME(25)   ,CNAM2(25)  ,CNAM3(25)          COMMONU   32
      COMMON /MATCH2/ AA(30),BB(30),KBTS,SMINCAL,                        MATCH4     2
     1        DWORD(10),DWORD2(10),DWORD3(10),DWORD7(7,10),MINCOM        MATCH4     3
      LOGICAL SMINCAL                                                    MATCH4     4
C                                        . INITIALIZE NEW DATA BLOCK . . MIDATA     9
      NBLOCK = NBLOCK + 1                                                MIDATA    10
      VERSN = 1.75                                                       MIDATA    11
      WRITE (ISYSWR,1004)  VERSN,NBLOCK                                  MIDATA    12
      WRITE (ISYSWR,1005)                                                MIDATA    13
      DO 50 I= 1, 7                                                      MIDATA    14
   50 ISW(I) = 0                                                         MIDATA    15
      SIGMA = 0.                                                         MIDATA    16
      WRITE (ISYSWR,1005)                                                MIDATA    17
      NPFIX = 0                                                          MIDATA    18
      NINT = 0                                                           MIDATA    19
      NU = 0                                                             MIDATA    20
      NPAR = 0                                                           MIDATA    21
      IFATAL = 0                                                         MIDATA    22
      WRITE (ISYSWR,1005)                                                MIDATA    23
      DO 100 I= 1, MAXEXT                                                MIDATA    24
      ERP(I) = 0.0                                                       MIDATA    25
      ERN(I) = 0.0                                                       MIDATA    26
      LCODE(I) = 0                                                       MIDATA    27
  100 LCORSP (I) = 0                                                     MIDATA    28
      UP = 1.0                                                           MIDATA    29
      ISW(5) = 1                                                         MIDATA    30
C                                     . . . READ PARAMETER CARDS . .     MIDATA    31
      DO 200 K=1,KBTS                                                    MIDATA    32
      NAMK=NAM(K)                                                        MIDATA    33
      UK=U(K)                                                            MIDATA    34
      WK=WERR(K)                                                         MIDATA    35
      A=AA(K)                                                            MIDATA    36
      B=BB(K)                                                            MIDATA    37
      NU = MAX0(NU,K)                                                    MIDATA    38
      IF (K .LE. MAXEXT)  GO TO 115                                      MIDATA    39
      IFATAL = IFATAL + 1                                                MIDATA    40
```

```
      WRITE (ISYSWR,1009)  K,MAXEXT
      WRITE (ISYSWR,1002)  K,NAMK,UK,WK,A,B
      GO TO 160
  115 CONTINUE
      IF (WK .GT. 0.0)  GO TO 122
C                                     . . . FIXED PARAMETER . . . . .
      WRITE (ISYSWR, 1002)  K,NAMK,UK
      LCODE(K) = 0
      GO TO 160
C                                     . . . VARIABLE PARAMETER . . .
  122 WRITE (ISYSWR, 1002)  K,NAMK,UK,WK,A,B
      NINT = NINT + 1
      IF  (A)  140,130,140
  130 IF  (B)  140,135,140
  135 LCODE(K) = 1
      GO TO 160
  140 IF (B-A)  145,142,150
  142 IFATAL = IFATAL + 1
      WRITE (ISYSWR,1010)
      GO TO 150
  145 SAV = B
      B = A
      A = SAV
  150 ALIM(K) = A
      BLIM(K) = B
      LCODE(K) = 4
      IF ((B-U(K))*(U(K)-A))  153,155,160
  153 IFATAL = IFATAL + 1
      WRITE (ISYSWR,1011)
      GO TO 160
  155 WRITE (ISYSWR,1006)
  160 CONTINUE
  200 CONTINUE
C                                     . . . END PARAMETER CARDS
C                                     . .   . STOP IF FATAL ERROR
  250 WRITE (ISYSWR,1005)
      IF (NINT .LE. MAXINT)  GO TO 253
      WRITE (ISYSWR,1008)  NINT,MAXINT
      IFATAL = IFATAL + 1
  253 IF (IFATAL .LE. 0)  GO TO 280
      WRITE (ISYSWR,1013)  IFATAL
      STOP
C                                     CALCULATE STEP SIZES DIRIN
  280 NPAR = 0
      DO 300 K= 1, NU
      IF (LCODE(K) .LE. 0)  GO TO 300
      NPAR = NPAR + 1
      LCORSP(K) = NPAR
      SAV = U(K)
      X(NPAR) = PINTF(SAV,K)
      XT(NPAR) = X(NPAR)
      SAV2 = SAV + WERR(K)
      VPLU = PINTF(SAV2,K) - X(NPAR)
      SAV2 = SAV - WERR(K)
      VMINU = PINTF(SAV2,K) - X(NPAR)
      DIRIN(NPAR) = 0.5 * (ABS(VPLU) +ABS(VMINU))
  300 CONTINUE
```

```
              RETURN                                                          MIDATA     98
         1110 FORMAT (12X, 15A4, 25X 4HTIME,F7.3)                             MIDATA     99
C...      THE FORMAT BELOW IS MACHINE-DEPENDENT. (A10) , (A6,4X) , ETC.       MIDATA    100
         1001 FORMAT (F10.0, A10, 4F10.0)                                     MIDATA    101
         1002 FORMAT (10XI10,5X,A10,2F15.6,5X,2E15.4)                         MIDATA    102
         1003 FORMAT (  53H WARNING           - ABOVE LIMITS HAVE BEEN REVERSED. MIDATA 103
             1)                                                               MIDATA    104
         1004 FORMAT (1H1/50X 21(1H*)/50X                                     MIDATA    105
             1       21H*   D506   MINUITS  *                                 MIDATA    107
             2/50X11H*   VERSION F6.2,4H   */50X16H* DATA BLOCK NO. I3,2H *)  MIDATA    112
         1005 FORMAT (12X96(1H*))                                             MIDATA    113
         1006 FORMAT (  49H WARNING           - ABOVE PARAMETER IS AT LIMIT )  MIDATA    114
         1007 FORMAT ( 106H WARNING  *******  - PARAMETER REQUESTED ON FOLLOWING MIDATA 115
             1 CARD HAS ALREADY APPEARED.  PREVIOUS VALUES IGNORED.)          MIDATA    116
         1008 FORMAT (  46H0   TOO MANY VARIABLE PARAMETERS.  YOU REQUESTI5/  49 MIDATA 117
             1H   THIS VERSION OF MINUIT IS ONLY DIMENSIONED FORI4//)         MIDATA    118
         1009 FORMAT (  30H0FATAL ERROR. PARAMETER NUMBER I11,  29H GREATER THAN MIDATA 119
             1 ALLOWED MAXIMUM I4)                                            MIDATA    120
         1010 FORMAT (  47H FATAL ERROR. UPPER AND LOWER LIMITS ARE EQUAL.)    MIDATA    121
         1011 FORMAT (  38H FATAL ERROR. PARAMETER OUTSIDE LIMITS/)           MIDATA    122
         1012 FORMAT (  43H0FATAL ERROR. MORE THAN 200 PARAMETER CARDS /)     MIDATA    123
         1013 FORMAT (/I5,  41H FATAL ERRORS ON PARAMETER CARDS.  ABORT.//)   MIDATA    124
              END                                                             MIDATA    125
```

```
      SUBROUTINE  MIGRAD                                                MIGRAD     2
CC        PERFORMS A LOCAL FUNCTION MINIMIZATION USING BASICALLY THE   MIGRAD     3
CC        METHOD OF DAVIDON-FLETCHER-POWELL AS MODIFIED BY FLETCHER    MIGRAD     4
CC        REF. -- FLETCHER, COMP.J. 13,317 (1970)   "SWITCHING METHOD" MIGRAD     5
CC                                                                      MIGRAD     6
       COMMON                                                           COMMONU    2
     1/MINERR/ ERP(30)    ,ERN(30)                                      COMMONU    4
     2/PARINT/ X(15)      ,XT(15)     ,DIRIN(15)  ,MAXINT     ,NPAR     COMMONU    5
     3/PAREXT/ U(30)      ,NAM(30)    ,WERR(30)   ,MAXEXT     ,NU       COMMONU    6
     4/LIMITS/ ALIM(30)   ,BLIM(30)   ,LCODE(30)  ,LCORSP(30) ,LIMSET   COMMONU    7
     5/VARIAN/ V(15,15)                                                 COMMONU    8
     7/FIX   / IPFIX(15)  ,XS(15)     ,XTS(15)    ,WTS(15)   ,NPFIX     COMMONU    9
     C/CASC/ JH, JL, Y(16)                                              COMMONU   10
     F/DERIVA/ G(30)      ,G2(30)                                       COMMONU   11
     G/SIMVEC/ P(15,16)   ,PSTAR(15)  ,PSTST(15)  ,PBAR(15)   ,PRHO(15) COMMONU   12
     J/VARIAT/ VT(15,15)                                                COMMONU   13
     6/UNIT  / ISYSRD     ,ISYSWR     ,ISYSPU                           COMMONU   27
     8/TITLE / TITLE(13)  ,DATE(2)    ,ISW(7)     ,NBLOCK               COMMONU   28
     9/CONVER/ EPSI ,APSI ,VTEST      ,NSTEPQ     ,NFCN       ,NFCNMX   COMMONU   29
     A/CARD  / CWORD      ,CWORD2     ,CWORD3     ,WORD7(7)             COMMONU   30
     B/MINIMA/ AMIN       ,UP         ,NEWMIN     ,ITAUR      ,SIGMA    COMMONU   31
     K/COMMND/ NNAME      ,CNAME(25)  ,CNAM2(25)  ,CNAM3(25)            COMMONU   32
      DIMENSION GS(30) , R(15),XXS(15), FLNU(15), VG(15), VII(15)       MIGRAD     9
      DATA SLAMIN,SLAMAX,TLAMIN,TLAMAX/0.2, 3.0, 0.05, 6.0/             MIGRAD    14
      IF (NPAR .LE. 0)  RETURN                                          MIGRAD    15
      ISWTR = ISW(5) - ITAUR                                            MIGRAD    16
      NPFN = NFCN                                                       MIGRAD    17
      PARN=NPAR                                                         MIGRAD    18
      RHO2 = 10.*APSI                                                   MIGRAD    19
      ROSTOP = 1.0E-5 * APSI                                            MIGRAD    20
      TRACE=1.                                                          MIGRAD    21
      IFLAG=4                                                           MIGRAD    22
      IF (ISW(3) .EQ. 1)  IFLAG = 2                                     MIGRAD    23
      FS = AMIN                                                         MIGRAD    24
      IF (ITAUR .LT. 1)  WRITE (ISYSWR,470)  ROSTOP, APSI, VTEST        MIGRAD    25
      GO TO 2                                                           MIGRAD    26
    1 WRITE (ISYSWR,520)                                                MIGRAD    27
C                                     . . . . STEP SIZES DIRIN . . .    MIGRAD    28
    2 NPARD = NPAR                                                      MIGRAD    29
      DO  3 I= 1, NPAR                                                  MIGRAD    30
      D = 0.02* ABS(DIRIN(I))                                           MIGRAD    31
      IF (ISW(2) .GE. 1)  D = 0.02* SQRT(ABS(V(I,I))*UP)                MIGRAD    32
      IF (D .LT.  1.0E-8 *ABS(X(I)))  D = 1.0E-8 * X(I)                 MIGRAD    33
    3 DIRIN(I) = D                                                      MIGRAD    34
C                                     . . . . . . STARTING GRADIENT     MIGRAD    35
      NTRY = 0                                                          MIGRAD    36
    4 NEGG2 = 0                                                         MIGRAD    37
      DO 10  ID= 1, NPARD                                               MIGRAD    38
      I = ID + NPAR - NPARD                                             MIGRAD    39
      D = DIRIN(I)                                                      MIGRAD    40
      XTF = X(I)                                                        MIGRAD    41
      X(I) = XTF + D                                                    MIGRAD    42
      CALL  INTOEX(X)                                                   MIGRAD    43
      CALL FCN(NPAR,G,FS1,U,4)                                          MIGRAD    44
      NFCN = NFCN + 1                                                   MIGRAD    45
      X(I) = XTF - D                                                    MIGRAD    46
      CALL  INTOEX(X)                                                   MIGRAD    47
```

```
      CALL FCN(NPAR,G,FS2,U,4)
      NFCN = NFCN + 1
      X(I) = XTF
      GS(I) = (FS1-FS2)/(2.0 * D)
      G2(I) = (FS1 + FS2 - 2.0*AMIN) / D**2
      IF (G2(I) .GT. 1.0E-30)  GO TO 10
C                                        . . . SEARCH IF G2 .LE. 0. . .
      WRITE (ISYSWR,520)
      NEGG2 = NEGG2 + 1
      NTRY = NTRY + 1
      IF (NTRY .GT. 4)  GO TO 230
      D = 50.*ABS(DIRIN(I))
      XBEG = XTF
      IF (GS(I) .LT. 0.)  DIRIN(I) = -DIRIN(I)
      KG = 0
      NF = 0
      NS = 0
    5 X(I) = XTF + D
      CALL INTOEX(X)
      CALL FCN(NPAR,G,F,U,4)
      NFCN = NFCN + 1
      IF (F .LE. AMIN)  GO TO 6
C         FAILURE
      IF (KG .EQ. 1)  GO TO 8
      KG = -1
      NF = NF + 1
      D = -0.4*D
      IF (NF .LT. 10)  GO TO 5
      D = 1000.*D
      GO TO 7
C         SUCCESS
    6 XTF = X(I)
      D = 3.0*D
      AMIN = F
      KG = 1
      NS = NS + 1
      IF (NS .LT. 10)  GO TO 5
      IF (AMIN .LT. FS)  GO TO 8
      D = 0.001*D
    7 XTF = XBEG
      G2(I) = 1.0
      NEGG2 = NEGG2 - 1
    8 X(I) = XTF
      DIRIN(I) = 0.1*D
      FS = AMIN
   10 CONTINUE
      IF (NEGG2 .GE. 1)  GO TO 4
      NTRY = 0
      MATGD = 1
C                                        . . . . . . DIAGONAL MATRIX
      IF (ISW(2) .GT. 1)  GO TO 15
   11 NTRY = 1
      MATGD = 0
      DO 13 I= 1, NPAR
      DO 12 J= 1, NPAR
   12 V(I,J) = 0.
   13 V(I,I) = 2.0/G2(I)
```

```
      C                                  . . . GET SIGMA AND SET UP LOOP MIGRAD  105
         15 SIGMA = 0.                                                   MIGRAD  106
            DO 18 I= 1, NPAR                                             MIGRAD  107
            IF (V(I,I) .LE. 0.)  GO TO 11                                MIGRAD  108
            RI = 0.                                                      MIGRAD  109
            DO 17 J= 1, NPAR                                             MIGRAD  110
            XXS(I) = X(I)                                                MIGRAD  111
         17 RI= RI+ V(I,J) * GS(J)                                       MIGRAD  112
         18 SIGMA = SIGMA + GS(I) *RI *0.5                               MIGRAD  113
            IF (SIGMA .GE. 0.)  GO TO 20                                 MIGRAD  114
            WRITE (ISYSWR,520)                                           MIGRAD  115
            IF (NTRY.EQ.0)  GO TO 11                                     MIGRAD  116
            ISW(2) = 0                                                   MIGRAD  117
            GO TO 230                                                    MIGRAD  118
         20 ISW(2) = 1                                                   MIGRAD  119
            ITER = 0                                                     MIGRAD  120
            CALL INTOEX(X)                                               MIGRAD  121
            IF (ISWTR .GE. 1)  CALL MPRINT(0,AMIN)                       MIGRAD  122
            IF (ISWTR .GE. 2)  CALL MATOUT(0.0, 1)                       MIGRAD  123
      C                                  . . . . . .  START MAIN LOOP    MIGRAD  124
         24 CONTINUE                                                     MIGRAD  125
            GDEL = 0.                                                    MIGRAD  126
            DO 30  I=1,NPAR                                              MIGRAD  127
            RI = 0.                                                      MIGRAD  128
            DO 25 J=1,NPAR                                               MIGRAD  129
         25 RI = RI + V(I,J) *GS(J)                                      MIGRAD  130
            DIRIN(I) = -0.5*RI                                           MIGRAD  131
            GDEL = GDEL + DIRIN(I)*GS(I)                                 MIGRAD  132
      C                                  .LINEAR SEARCH ALONG -VG   . . . MIGRAD  133
         30 X(I) =XXS(I) + DIRIN(I)                                      MIGRAD  134
            CALL INTOEX(X)                                               MIGRAD  135
            CALL FCN (NPAR, G, F, U, 4)                                  MIGRAD  136
            NFCN=NFCN+1                                                  MIGRAD  137
      C                                  . QUADR INTERP USING SLOPE GDEL MIGRAD  138
            DENOM = 2.0*(F-AMIN-GDEL)                                    MIGRAD  139
            IF (DENOM .LE. 0.)  GO TO 35                                 MIGRAD  140
            SLAM = -GDEL/DENOM                                           MIGRAD  141
            IF (SLAM .GT. SLAMAX)  GO TO 35                              MIGRAD  142
            IF (SLAM .LT. SLAMIN)  SLAM=SLAMIN                           MIGRAD  143
            GO TO 40                                                     MIGRAD  144
         35 SLAM = SLAMAX                                                MIGRAD  145
         40 IF (ABS(SLAM-1.0) .LT. 0.1)  GO TO 70                        MIGRAD  146
            DO 45 I= 1, NPAR                                             MIGRAD  147
         45 X(I) =XXS(I) + SLAM*DIRIN(I)                                 MIGRAD  148
            CALL INTOEX(X)                                               MIGRAD  149
            CALL FCN(NPAR,G,F2,U,4)                                      MIGRAD  150
            NFCN = NFCN + 1                                              MIGRAD  151
      C                                  . QUADR INTERP USING 3 POINTS   MIGRAD  152
            AA = FS/SLAM                                                 MIGRAD  153
            BB = F/(1.0-SLAM)                                            MIGRAD  154
            CC = F2/ (SLAM*(SLAM-1.0))                                   MIGRAD  155
            DENOM = 2.0*(AA+BB+CC)                                       MIGRAD  156
            IF (DENOM .LE. 0.)  GO TO 48                                 MIGRAD  157
            TLAM = (AA*(SLAM+1.0) + BB*SLAM + CC)/DENOM                  MIGRAD  158
            IF (TLAM .GT. TLAMAX)  GO TO 48                              MIGRAD  159
            IF (TLAM .LT. TLAMIN)  TLAM=TLAMIN                           MIGRAD  160
            GO TO 50                                                     MIGRAD  161
```

```
   48 TLAM = TLAMAX
   50 CONTINUE
      DO 51 I= 1, NPAR
   51 X(I) = XXS(I)+TLAM*DIRIN(I)
      CALL INTOEX(X)
      CALL FCN(NPAR,G,F3,U,4)
      NFCN = NFCN + 1
      IF (F.GE.AMIN .AND. F2.GE.AMIN .AND. F3.GE.AMIN) GO TO 200
      IF (F .LT. F2 .AND. F .LT. F3)  GO TO 61
      IF (F2 .LT. F3)  GO TO 58
   55 F = F3
      SLAM = TLAM
      GO TO 65
   58 F = F2
      GO TO 65
   61 SLAM = 1.0
   65 DO 67 I= 1, NPAR
      DIRIN(I) = DIRIN(I)*SLAM
   67 X(I) = XXS(I) + DIRIN(I)
   70 AMIN = F
      ISW(2) = 2
      IF (SIGMA+FS-AMIN .LT. ROSTOP)  GO TO 170
      IF (SIGMA+RHO2+FS-AMIN .GT. APSI)  GO TO 75
      IF (TRACE .LT. VTEST)  GO TO 170
   75 CONTINUE
      IF (NFCN-NPFN .GE. NFCNMX)  GO TO 190
      ITER = ITER + 1
      IF (ISWTR.GE. 3 .OR.(ISWTR.EQ. 2 .AND. MOD(ITER,10) .EQ.1))
     *     CALL MPRINT(0,AMIN)
C                                             . . . GET GRADIENT AND SIGMA .
      IF (ISW(3) .NE. 1)  GO TO 80
      CALL INTOEX(X)
      CALL FCN(NPAR,G,AMIN,U,IFLAG)
      NFCN = NFCN + 1
   80 CALL DERIVE(G,G2)
      RHO2 = SIGMA
      SIGMA = 0.
      GVG = 0.
      DELGAM = 0.
      DO 100 I= 1, NPAR
      RI = 0.
      VGI = 0.
      DO 90 J= 1, NPAR
      VGI = VGI + V(I,J)*(G(J)-GS(J))
   90 RI = RI + V(I,J) *G (J)
      R(I) = RI * 0.5
      VG(I) = VGI*0.5
      GAMI = G(I) - GS(I)
      GVG = GVG + GAMI*VG(I)
      DELGAM = DELGAM + DIRIN(I)*GAMI
  100 SIGMA = SIGMA + G(I)*R(I)
      IF (SIGMA .LT. 0.)  GO TO 1
      IF (GVG .LE. 0.)  GO TO 105
      IF (DELGAM .LE. 0.)  GO TO 105
      GO TO 107
  105 IF (SIGMA .LT. 0.1*ROSTOP)  GO TO 170
      GO TO 1
```

```
  107 CONTINUE
C                                              .  UPDATE COVARIANCE MATRIX
      TRACE=0.
      DO 120 I= 1, NPAR
      VII(I) = V(I,I)
      DO  120  J=1,NPAR
      D = DIRIN(I)*DIRIN(J)/DELGAM - VG(I)*VG(J)/GVG
  120 V(I,J) = V(I,J) + 2.0*D
      IF (DELGAM .LE. GVG)  GO TO 135
      DO 125 I= 1, NPAR
  125 FLNU(I) = DIRIN(I)/DELGAM - VG(I)/GVG
      DO 130 I= 1, NPAR
      DO 130 J= 1, NPAR
  130 V(I,J) = V(I,J) + 2.0*GVG*FLNU(I)*FLNU(J)
  135 CONTINUE
      DO 140 I= 1, NPAR
  140 TRACE = TRACE + ((V(I,I)-VII(I))/(V(I,I)+VII(I)))**2
      TRACE = SQRT(TRACE/PARN)
      IF (ISWTR .GE. 4)  CALL MATOUT(TRACE, 0)
      CALL UCOPY(X,XXS,NPAR)
      CALL UCOPY(G,GS,NPAR)
      FS = F
      GO TO 24
C                                              . . . . . .  END MAIN LOOP
  170 WRITE(ISYSWR,500)
      ISW(2) = 3
      IF(ISWTR .GE. 0)  CALL MPRINT(1-ITAUR,AMIN)
      ISWTR = ISWTR - 3*ITAUR
      IF (ISWTR .GT. 0)  CALL MATOUT(TRACE,1)
      IF (ITAUR .GT. 0)  GO TO 435
      IF (MATGD .GT. 0)  GO TO 435
      NPARGD = NPAR*(NPAR+5)/2
      IF (NFCN-NPFN .GE. NPARGD)  GO TO 435
      WRITE (ISYSWR,180)
  180 FORMAT (  55H COVARIANCE MATRIX INACCURATE.  MINUIT WILL RECALCULA
     1TE)
      CALL HESSE
      CALL MPRINT(1,AMIN)
      CALL MATOUT(0.0, 1)
      IF (ISW(2) .GE. 2)  ISW(2) = 3
      GO TO 435
  190 ISW(1) = 1
      GO TO 230
  200 WRITE (ISYSWR,650)
      CALL UCOPY(XXS,X,NPAR)
      ISW(2) = 1
      IF (SIGMA .LT. ROSTOP)  GO TO 170
      IF (MATGD .GT. 0)  GO TO 2
  230 WRITE (ISYSWR,510)
      CALL  INTOEX(X)
      CALL MPRINT(1-ITAUR, AMIN)
      ISWTR = ISW(5) - ITAUR*3
      IF (ISWTR .LT. 1)  GO TO 435
      IF (ISW(2) .LE. 1)  GO TO 435
      CALL MATOUT(TRACE,1)
  435 RETURN
  470 FORMAT (  37H START MIGRAD MINIMIZATION.          8X  66HCONVERG
```

```
     1ENCE CRITERIA  --  ESTIMATED DISTANCE TO MINIMUM (EDM) .LT.E9.2/45 MIGRAD
     2X  11HOR EDM .LT.E9.2,  46H AND FRACTIONAL CHANGE IN VARIANCE MATR MIGRAD
     3IX .LT.E9.2)                                                       MIGRAD
  500 FORMAT (  34H MIGRAD MINIMIZATION HAS CONVERGED )                  MIGRAD
  510 FORMAT (  39H MIGRAD TERMINATED WITHOUT CONVERGENCE )              MIGRAD
  520 FORMAT (11X  43HCOVARIANCE MATRIX IS NOT POSITIVE-DEFINITE.)       MIGRAD
  650 FORMAT (  34H MIGRAD FAILS TO FIND IMPROVEMENT )                   MIGRAD
      END                                                                MIGRAD
```

```
               SUBROUTINE MINNEW                                              MINNEW      2
      CC           THIS IS THE MAIN PROGRAM, DISGUISED AS A SUBROUTINE FOR    MINNEW      3
      CC           REASONS OF COMPATIBILITY BETWEEN SYSTEMS.    IT INITIALIZES MINNEW     4
      CC           SOME CONSTANTS IN COMMON (INCLUDING THE LOGICAL I/O UNIT NOS.) MINNEW  5
      CC           WHICH WOULD HAVE TO BE IN BLOCK DATA FOR MANY COMPILERS.   MINNEW      6
      CC           THEN VERIFIES THAT FCN GIVES THE SAME VALUE WHEN CALLED    MINNEW      7
      CC           TWICE WITH THE SAME ARGUMENTS, AND PASSES CONTROL TO COMAND. MINNEW    8
      CC                                                                      MINNEW      9
                COMMON                                                        COMMONU     2
              1/MINERR/ ERP(30)     ,ERN(30)                                  COMMONU     4
              2/PARINT/ X(15)       ,XT(15)      ,DIRIN(15)  ,MAXINT     ,NPAR      COMMONU  5
              3/PAREXT/ U(30)       ,NAM(30)     ,WERR(30)   ,MAXEXT     ,NU        COMMONU  6
              4/LIMITS/ ALIM(30)    ,BLIM(30)    ,LCODE(30)  ,LCORSP(30) ,LIMSET    COMMONU  7
              5/VARIAN/ V(15,15)                                              COMMONU     8
              7/FIX   / IPFIX(15)   ,XS(15)      ,XTS(15)    ,WTS(15)   ,NPFIX      COMMONU  9
              C/CASC/ JH, JL, Y(16)                                           COMMONU    10
              F/DERIVA/ G(30)       ,G2(30)                                   COMMONU    11
              G/SIMVEC/ P(15,16)    ,PSTAR(15)   ,PSTST(15)  ,PBAR(15)   ,PRHO(15) COMMONU  12
              J/VARIAT/ VT(15,15)                                             COMMONU    13
              6/UNIT  / ISYSRD      ,ISYSWR      ,ISYSPU                      COMMONU    27
              8/TITLE / TITLE(13)   ,DATE(2)     ,ISW(7)     ,NBLOCK          COMMONU    28
              9/CONVER/ EPSI  ,APSI ,VTEST       ,NSTEPQ     ,NFCN       ,NFCNMX    COMMONU 29
              A/CARD  / CWORD       ,CWORD2      ,CWORD3     ,WORD7(7)        COMMONU    30
              B/MINIMA/ AMIN        ,UP          ,NEWMIN     ,ITAUR      ,SIGMA     COMMONU 31
              K/COMMND/ NNAME       ,CNAME(25)   ,CNAM2(25)  ,CNAM3(25)       COMMONU    32
               DATA RET1,RET2,RET3/ 4H RET, 4HRETU, 4HRETR /                  MINNEW     11
               DATA CNAME /        4HMINI ,4HSEEK ,4HSIMP ,4HMIGR ,4HMINO ,4HPUNC  MINNEW 12
              7            ,4HPRIN ,4HEND  ,4HFIX  ,4HREST ,4HEXIT ,4HGRAD ,4HCALL  MINNEW 13
              4            ,4HMATO ,4HCOVA ,4HIMPR ,4HERRO ,4HPAGE ,4HHESS ,4HCONT  MINNEW 14
              1            ,4HSTAN ,4HRELE ,4H     ,4H     ,4HEND  /          MINNEW     15
               DATA CNAM2 /        4HMIZE ,4H     ,4HLEX  ,4HAD   ,4HS    ,4HH     MINNEW 16
              7            ,4HTOUT ,4H     ,4H     ,4HORE  ,4H     ,4HIENT ,4H FCN  MINNEW 17
              4            ,4HUT   ,4HRIAN ,4HOVE  ,4HR DE ,4H     ,4HE    ,4HOUR   MINNEW 18
              1            ,4HDARD ,4HASE  ,4H     ,4H     ,4HRETU /          MINNEW     19
               DATA CNAM3 /        2H     ,2H     ,2H     ,2H     ,2H     ,2H      MINNEW 20
              7            ,2H     ,2H     ,2H     ,2H     ,2H     ,2H     ,2H      MINNEW 21
              4            ,2H     ,2HCE   ,2H     ,2HF    ,2H     ,2H     ,2H      MINNEW 22
              1            ,2H     ,2H     ,2H     ,2H     ,2HRN   /          MINNEW     23
               DATA NNAME, NBLOCK / 25, 0 /                                   MINNEW     24
               DATA ISYSRD,ISYSWR,ISYSPU,MAXINT,MAXEXT/2,3,98,15,30/          MINNEW     26
           210 FORMAT (  5( 8X,2A4,A2))                                       MINNEW     27
           110 CONTINUE                                                       MINNEW     28
               NFCN = 1                                                       MINNEW     29
               CALL MIDATA                                                    MINNEW     30
               CALL INTOEX(X)                                                 MINNEW     31
      C        WRITE (ISYSWR,120)                                             MINNEW     32
           120 FORMAT (/20H0FIRST ENTRY TO FCN    )                           MINNEW     33
      C        CALL FCN(NPAR,G,AMIN,U,1)                                      MINNEW     34
               CALL FCN(NPAR,G,AMIN,U,4)                                      MINNEW     35
               CALL MPRINT(1,AMIN)                                            MINNEW     36
               CALL FCN(NPAR,G,F    ,U,4)                                     MINNEW     37
               IF  (F .NE. AMIN)  GO TO 160                                   MINNEW     38
               NFCN = 3                                                       MINNEW     39
               CALL COMAND                                                    MINNEW     40
               IF (CWORD2.EQ.RET1 .OR.CWORD2.EQ.RET2 .OR.CWORD2.EQ.RET3) RETURN  MINNEW  41
               GO TO 110                                                      MINNEW     42
           160 CONTINUE                                                       MINNEW     43
```

```
      WRITE (ISYSWR,880) AMIN, F                                        MINNEW    44
      STOP                                                              MINNEW    45
  880 FORMAT (  62H0FOR THE ABOVE VALUES OF THE PARAMETERS, FCN IS TIME- MINNEW    46
     1DEPENDENT/   4H0F = E22.14,  15H FOR FIRST CALL/   4H F =E22.14,  MINNEW    47
     211H FOR SECOND)                                                   MINNEW    48
      END                                                               MINNEW    49
```

```
      SUBROUTINE MINOS                                                  MINOS      2
CC        PERFORMS A MINOS ERROR ANALYSIS ON THOSE PARAMETERS FOR       MINOS      3
CC        WHICH IT IS REQUESTED ON THE MINOS COMMAND CARD.              MINOS      4
CC        THE PARAMETER IN QUESTION IS VARIED, AND THE MINIMUM OF THE   MINOS      5
CC        FUNCTION WITH RESPECT TO THE OTHER PARAMETERS IS FOLLOWED     MINOS      6
CC        UNTIL IT CROSSES THE VALUE FMIN+UP.                           MINOS      7
CC                                                                      MINOS      8
       COMMON                                                           COMMONU    2
     1/MINERR/ ERP(30)   ,ERN(30)                                       COMMONU    4
     2/PARINT/ X(15)     ,XT(15)    ,DIRIN(15) ,MAXINT    ,NPAR         COMMONU    5
     3/PAREXT/ U(30)     ,NAM(30)   ,WERR(30)  ,MAXEXT    ,NU           COMMONU    6
     4/LIMITS/ ALIM(30)  ,BLIM(30)  ,LCODE(30) ,LCORSP(30) ,LIMSET      COMMONU    7
     5/VARIAN/ V(15,15)                                                 COMMONU    8
     7/FIX   / IPFIX(15) ,XS(15)    ,XTS(15)   ,WTS(15)  ,NPFIX         COMMONU    9
     C/CASC/ JH, JL, Y(16)                                              COMMONU   10
     F/DERIVA/ G(30)     ,G2(30)                                        COMMONU   11
     G/SIMVEC/ P(15,16)  ,PSTAR(15) ,PSTST(15) ,PBAR(15)  ,PRHO(15)     COMMONU   12
     J/VARIAT/ VT(15,15)                                                COMMONU   13
     6/UNIT  / ISYSRD    ,ISYSWR    ,ISYSPU                             COMMONU   27
     8/TITLE / TITLE(13) ,DATE(2)   ,ISW(7)    ,NBLOCK                  COMMONU   28
     9/CONVER/ EPSI ,APSI ,VTEST    ,NSTEPQ    ,NFCN      ,NFCNMX       COMMONU   29
     A/CARD  / CWORD     ,CWORD2    ,CWORD3    ,WORD7(7)                COMMONU   30
     B/MINIMA/ AMIN      ,UP        ,NEWMIN    ,ITAUR     ,SIGMA        COMMONU   31
     K/COMMND/ NNAME     ,CNAME(25) ,CNAM2(25) ,CNAM3(25)               COMMONU   32
      DIMENSION LMI(30), KIRSCH(5),XDEV(15), W(30),SPT(3,20)            MINOS     11
      DATA APOS,ANEG,ADOT,ASTAR / 4HPOSI,4HNEGA,1H., 1H* /              MINOS     16
      IF (NPAR .LE. 0)  GO TO 700                                       MINOS     17
C                                 . . UNPACK PARAMETER REQUESTS .       MINOS     18
      MARC = 0                                                          MINOS     19
      DO 5 I=1,30                                                       MINOS     20
    5 LMI(I)=0                                                          MINOS     21
      KNT = 0                                                           MINOS     22
      DO  20 I= 2, 7                                                    MINOS     23
      LIME = WORD7(I) + 0.5                                             MINOS     24
      IF (LIME .EQ. 0)  GO TO  20                                       MINOS     25
      DO  10 K= 1, 5                                                    MINOS     26
      K2 = 6 - K                                                        MINOS     27
      LEMON = MOD(LIME,100)                                             MINOS     28
      KIRSCH(K2) = LEMON                                                MINOS     29
   10 LIME = LIME/100                                                   MINOS     30
      DO  15 J= 1, 5                                                    MINOS     31
      LEMON = KIRSCH(J)                                                 MINOS     32
      IF (LEMON .GT. MAXEXT)  GO TO  14                                 MINOS     33
      IF (LEMON .EQ. 0)  GO TO  15                                      MINOS     34
      IF (LCORSP(LEMON) .EQ. 0)  GO TO  14                              MINOS     35
      KNT = KNT + 1                                                     MINOS     36
      LMI(KNT) = LEMON                                                  MINOS     37
      GO TO  15                                                         MINOS     38
   14 MARC = 1                                                          MINOS     39
   15 CONTINUE                                                          MINOS     40
   20 CONTINUE                                                          MINOS     41
      IF (KNT .GT. 0)  GO TO  40                                        MINOS     42
      DO  30 I= 1, MAXEXT                                               MINOS     43
      IF (LCORSP(I) .LT. 1)  GO TO  30                                  MINOS     44
      KNT = KNT + 1                                                     MINOS     45
      LMI(KNT) = I                                                      MINOS     46
      IF (KNT .EQ. 30)  GO TO  40                                       MINOS     47
```

```
   30 CONTINUE
   40 CONTINUE
      IF (MARC .EQ. 1)  WRITE (ISYSWR,811)
      WRITE (ISYSWR, 810)  (LMI(IZ3),IZ3=1,KNT)
C                                         . . SAVE AND PREPARE START VALS
      SIGSAV = SIGMA
      TOLER = EPSI
      APSI = EPSI*0.5
      ITAUR = 1
      ABEST=AMIN
      AIM = AMIN + UP
      NSAVE = NFCNMX
      MPAR=NPAR
      DO 130 I= 1, MPAR
      XT(I) = X(I)
      DO 125 J= 1, MPAR
  125 VT(I,J) = V(I,J)
  130 CONTINUE
      DO 135 I= 1, NU
      ERP(I) = 0.
      ERN(I) = 0.
  135 W(I) = WERR(I)
      KNT = 0
C                                         . . . . . START MAIN LOOP . . .
  150 KNT = KNT + 1
      ISW(1) = 0
      NLIMIT = NFCN + NSAVE
      IF (KNT .GT. 30)  GO TO 590
      IF (LMI(KNT) .LT. 1)  GO TO 590
      ILAX = LMI(KNT)
      ERP(ILAX) = 0.
      IT = LCORSP(ILAX)
      XTIT = XT(IT)
      CALL INTOEX(XT)
      UT = U(ILAX)
      SPT(1,1) = UT
      SPT(2,1) = AIM
      SPT(3,1) = 1H
      SPT(1,2) = UT
      SPT(2,2) = ABEST
      SPT(3,2) = ASTAR
      SPT(1,3) = UT
      SPT(2,3) = AIM + 0.2*(AIM-ABEST)
      SPT(3,3) = ADOT
      NSPT = 3
      IF (LCODE(ILAX) .GT. 1)  GO TO 160
      ALIM(ILAX) = UT -100.*W(ILAX)
      BLIM(ILAX) = UT +100.*W(ILAX)
  160 CONTINUE
      XUNIT = SQRT(UP/VT(IT,IT))
      MARC = 0
      DO 162 I= 1, MPAR
      IF (I .EQ. IT)  GO TO 162
      MARC = MARC + 1
      XDEV(MARC) = XUNIT*VT(IT,I)
  162 CONTINUE
      CALL FIXPAR (IT,1,ILAX)
```

```
      SIG = 1.0
      ASIG = APOS
      DULIM = BLIM(ILAX) - UT
      IF(ISW(2).LT.1) GO TO 460
C                                   . SIG=SIGN OF ERROR BEING CALCD
  165 WRITE (ISYSWR,806)  ASIG,ILAX,NAM(ILAX)
      ITER = 0
      LIMSET = 0
      DU1 = SIG*W(ILAX)
      IF (ABS(DU1) .LE. DULIM)  GO TO 180
      LIMSET = 1
      DU1 = SIG * DULIM
      IF (DULIM .LT. 1.0E-3*W(ILAX))    GO TO 440
  180 U(ILAX) = UT + DU1
      IF (NPAR .EQ. 0)  GO TO 205
      FAC = DU1/W(ILAX)
      DO 185 I= 1, NPAR
  185 X(I) = XT(I) + FAC*XDEV(I)
  200 CALL INTOEX (X)
  205 WRITE (ISYSWR,801)  ILAX,UT,DU1,U(ILAX)
      CALL FCN(NPAR,G,AMIN,U,4)
      NFCN = NFCN + 1
      NFCNMX = NLIMIT - NFCN
      CALL MIGRAD
      IF (AMIN .LT. ABEST)  GO TO 650
      IF (ISW(1) .GE. 1)  GO TO 450
      IF (ISW(2) .GE. 2)  GO TO 240
      NFCNMX = NLIMIT - NFCN
      CALL SIMPLX
      IF (AMIN .LT. ABEST)  GO TO 650
      IF (ISW(1) .GE. 1)  GO TO 450
      NFCNMX = NLIMIT - NFCN
      CALL MIGRAD
      IF (AMIN .LT. ABEST)  GO TO 650
      IF (ISW(1) .GE. 1)  GO TO 450
      IF (ISW(2) .LT. 2)  GO TO 460
  240 CREM = AMIN - ABEST
      NSPT = NSPT + 1
      SPT(1,NSPT) = U(ILAX)
      SPT(2,NSPT) = AMIN
      SPT(3,NSPT) = ASIG
      IF (CREM .LE. 0.0)  GO TO 650
      SQUC = SQRT(UP/CREM)
      IF (ABS(AMIN-AIM) .LT. TOLER)  GO TO 400
C                                 . . ANOTHER ITERATION NECESSARY
      ITER = ITER + 1
      IF (ITER .GT. 6)  GO TO 430
      IF (ITER .EQ. 1)  GO TO 270
C       CHECK PREVIOUS ITERATION TO AVOID OSCILLATING
      SQUC2 = SQUC
      IF ((SQUC2-1.0) * (SQUC1-1.0) .GT. 0.0)  GO TO 270
      SQUC = 0.65*SQUC2 + 0.35
      SQUC11 = 1.0/SQUC1
      IF ((SQUC11-SQUC)*(SQUC-1.0) .GT. 0.0)  GO TO 270
      WRITE (ISYSWR,260)
  260 FORMAT (  44H MINOS IS HAVING TROUBLE WITH THIS PARAMETER)
      SQUC = 0.5*SQUC11 + 0.5
```

```
  270 CONTINUE
      SQUC1 = SQUC
      DU1 = DU1 * SQUC
      IF (ABS(DU1) .LE. DULIM)  GO TO 280
      IF (LIMSET .EQ. 1)  GO TO 440
      LIMSET = 1
      DU1 = SIG*DULIM
  280 U(ILAX) = UT + DU1
      DO 290 I= 1, NPAR
  290 X(I) = XT(I) + SQUC*(X(I)-XT(I))
      GO TO 200
C                                       . ERROR SUCCESSFULLY CALCULATED
  400 EROS = DU1 * SQUC
      WRITE (ISYSWR,808)  ASIG,ILAX,NAM(ILAX),EROS
  410 WRITE (ISYSWR,812)
      IF (SIG .GT. 0.)  GO TO 420
      ERN(ILAX) = EROS
      GO TO 500
  420 ERP(ILAX) = EROS
      SIG = -1.0
      ASIG = ANEG
      DULIM = UT - ALIM(ILAX)
      GO TO 165
C                                       . . . . . . . . . FAILURE RETURNS
  430 WRITE (ISYSWR, 809)
      EROS = 0.
      GO TO 410
  440 WRITE (ISYSWR, 807)  ASIG,ILAX,NAM(ILAX),DULIM
      EROS = 0.
      GO TO 410
  450 WRITE (ISYSWR, 802)  NSAVE
      GO TO 500
  460 WRITE (ISYSWR, 805)
C                                       . . PARAMETER FINISHED. RESET V
  500 CONTINUE
      CALL RESTOR(1)
      ISW(2) = 3
      DO 560 I= 1, MPAR
      DO 550 J= 1, MPAR
  550 V(I,J) = VT(I,J)
  560 CONTINUE
      IF (NSPT .LT. 9)  GO TO 150
      NSPT3 = NSPT - 3
      WRITE (ISYSWR,813)  ILAX,NAM(ILAX),NSPT3
      CALL PLTCON(NSPT,SPT)
      WRITE(ISYSWR,814)  ILAX
      GO TO 150
C                                       . . . . PRINTOUT FINAL VALUES .
  590 CALL UCOPY(XT,X,MPAR)
      CALL INTOEX (XT)
      SIGMA = SIGSAV
      AMIN = ABEST
      CALL MPRINT(2,AMIN)
      CALL MATOUT(0.0, 1)
      GO TO 700
C                                       . . . NEW MINIMUM FOUND . . . .
  650 NEWMIN = 1
```

```
      ISW(2) = 0
      CALL RESTOR(1)
      CALL EXTOIN(X)
      DO 670 I= 1, NPAR
  670 DIRIN(I) = SQRT(VT(I,I)*UP)
  700 RETURN
  801 FORMAT (  10H0PARAMETERI4,   7H SET TOE11.3,3H + E10.3,3H = E12.3)
  802 FORMAT (  87H0CALL LIMIT EXCEEDED.  CALCULATION OF MINOS ERROR FOR
     1 THIS PARAMETER REQUIRES MORE THANI5,  11H FCN CALLS./)
  805 FORMAT (/45X   46HMINOS ERROR NOT CALCULATED FOR THIS PARAMETER.)
  806 FORMAT (  18H0DETERMINATION OF A4,  30HTIVE MINOS ERROR FOR PARAME
     1TERI3, 2X A10)
  807 FORMAT (/45X4HTHE A4,  29HTIVE MINOS ERROR OF PARAMETERI3,   2H, A
     110,      9H, EXCEEDS E12.4)
  808 FORMAT (/45X4HTHE A4,  29HTIVE MINOS ERROR OF PARAMETERI3,   2H, A
     110,      4H, IS E12.4)
  809 FORMAT (/45X  48HTOO MANY ITERATIONS. SEE PLOT BELOW FOR DETAILS.)
  810 FORMAT (  38H MINOS ERRORS REQUESTED FOR PARAMETERS30I3//)
  811 FORMAT (  55H THERE ARE MISTAKES IN THE MINOS COMMAND CARD JUST RE
     1AD)
  812 FORMAT (45X, 75(1H*))
  813 FORMAT (50H1MINOS FINDS NON-PARABOLIC BEHAVIOR FOR PARAMETER ,I3,
     12H, A10/28H ERROR CALCULATION REQUIRED ,I3,63H MINIMIZATIONS.
     2A SUMMARY OF POINTS FOUND IS PLOTTED BELOW.  /13H0   FCN VALUE )
  814 FORMAT (20H+          PARAMETER  I3,1H,/)
      END
```

```
      SUBROUTINE MPRINT  (IKODE,FVAL)                                   MPRINT     2
CC        PRINTS THE VALUES OF THE PARAMETERS AT THE TIME OF THE CALL.  MPRINT     3
CC        ALSO PRINTS OTHER RELEVANT INFORMATION SUCH AS FUNCTION VALUE, MPRINT    4
CC        ESTIMATED DISTANCE TO MINIMUM, PARAMETER ERRORS, STEP SIZES.  MPRINT     5
CC        ACCORDING TO THE VALUE OF IKODE,THE PRINTOUT IS LONG FORMAT,  MPRINT     6
CC        SHORT FORMAT, OR MINOS FORMAT (0,1,2)                         MPRINT     7
CC                                                                      MPRINT     8
       COMMON                                                           COMMONU    2
     1/MINERR/ ERP(30)     ,ERN(30)                                     COMMONU    4
     2/PARINT/ X(15)       ,XT(15)      ,DIRIN(15)   ,MAXINT     ,NPAR  COMMONU    5
     3/PAREXT/ U(30)       ,NAM(30)     ,WERR(30)    ,MAXEXT     ,NU    COMMONU    6
     4/LIMITS/ ALIM(30)    ,BLIM(30)    ,LCODE(30)   ,LCORSP(30) ,LIMSET COMMONU   7
     5/VARIAN/ V(15,15)                                                 COMMONU    8
     7/FIX   / IPFIX(15)   ,XS(15)      ,XTS(15)     ,WTS(15)   ,NPFIX  COMMONU    9
     C/CASC/ JH, JL, Y(16)                                              COMMONU   10
     F/DERIVA/ G(30)       ,G2(30)                                      COMMONU   11
     G/SIMVEC/ P(15,16)    ,PSTAR(15)   ,PSTST(15)   ,PBAR(15)   ,PRHO(15) COMMONU 12
     J/VARIAT/ VT(15,15)                                                COMMONU   13
     6/UNIT  / ISYSRD      ,ISYSWR      ,ISYSPU                         COMMONU   27
     8/TITLE / TITLE(13)   ,DATE(2)     ,ISW(7)      ,NBLOCK            COMMONU   28
     9/CONVER/ EPSI ,APSI  ,VTEST       ,NSTEPQ      ,NFCN       ,NFCNMX COMMONU  29
     A/CARD  / CWORD       ,CWORD2      ,CWORD3      ,WORD7(7)          COMMONU   30
     B/MINIMA/ AMIN        ,UP          ,NEWMIN      ,ITAUR      ,SIGMA COMMONU   31
     K/COMMND/ NNAME       ,CNAME(25)   ,CNAM2(25)   ,CNAM3(25)         COMMONU   32
C                                     . GET TIME AND PRINT HEADINGS .   MPRINT    10
      TI    = FTIME(BIDON)                                              MPRINT    11
      IF (IKODE .EQ. 2)  GO TO 10                                       MPRINT    12
      WRITE (ISYSWR,1000)                                               MPRINT    13
      GO TO 18                                                          MPRINT    14
C                 IKODE = 2        MINOS PRINTOUT HAS SPECIAL HEADINGS. MPRINT    15
   10 WRITE (ISYSWR,1006)  TITLE,DATE                                   MPRINT    16
   18 E = SIGMA                                                         MPRINT    17
      KOUNT = 0                                                         MPRINT    18
C                                     . . . LOOP OVER PARAMETERS . .    MPRINT    19
      DO 200 I= 1, NU                                                   MPRINT    20
      IF (NAM (I))  20,200,20                                           MPRINT    21
   20 L = LCORSP(I)                                                     MPRINT    22
      IF (L .EQ. 0)  GO TO 55                                           MPRINT    23
C              VARIABLE PARAMETER.  CALCULATE EXTERNAL ERROR IF V EXISTS MPRINT   24
      IF (ISW(2) .LT. 1)  GO TO 27                                      MPRINT    25
      DX = SQRT(ABS(V(L,L)*UP))                                         MPRINT    26
      IF (LCODE(I) .LE. 1)  GO TO 26                                    MPRINT    27
      AL = ALIM(I)                                                      MPRINT    28
      BA = BLIM(I) - AL                                                 MPRINT    29
      DU1 = AL + 0.5 *(SIN(X(L)+DX) +1.0) * BA - U(I)                   MPRINT    30
      DU2 = AL + 0.5 *(SIN(X(L)-DX) +1.0) * BA - U(I)                   MPRINT    31
      IF (DX .GT. 1.0)  DU1 = BA                                        MPRINT    32
      DX = 0.5 * (ABS(DU1) + ABS(DU2))                                  MPRINT    33
   26 WERR(I) = DX                                                      MPRINT    34
   27 X1 = X(L)                                                         MPRINT    35
      X2 = DIRIN(L)                                                     MPRINT    36
      IF (IKODE .LT. 2)  GO TO 29                                       MPRINT    37
      X1 = ERP(I)                                                       MPRINT    38
      X2 = ERN(I)                                                       MPRINT    39
   29 IF (KOUNT)  30,30,40                                              MPRINT    40
   30 KOUNT = 1                                                         MPRINT    41
      WRITE (ISYSWR,1001) FVAL,NFCN,TI,E, L,I,NAM(I),U(I),WERR(I),X1,X2 MPRINT    42
```

```
      GO TO 45                                                          MPRINT      43
   40 WRITE (ISYSWR,1002)   L,I,NAM(I),U(I),WERR(I),X1,X2               MPRINT      44
   45 IF (LCODE(I) .LE. 1)  GO TO 200                                   MPRINT      45
      IF (ABS(COS(X(L))) .LT. 0.001)  WRITE (ISYSWR,1004)               MPRINT      46
      GO TO 200                                                         MPRINT      47
C                            FIXED PARAMETER.  PRINT ONLY IF IKODE .GT.0 MPRINT      48
   55 IF (IKODE .EQ. 0)  GO TO 200                                      MPRINT      49
      IF (KOUNT)  60,60,70                                              MPRINT      50
   60 KOUNT = 1                                                         MPRINT      51
      WRITE (ISYSWR,1001) FVAL,NFCN,TI,E, L,I,NAM(I),U(I)               MPRINT      52
      GO TO 200                                                         MPRINT      53
   70 WRITE (ISYSWR,1003)                    I,NAM(I),U(I)              MPRINT      54
  200 CONTINUE                                                          MPRINT      55
      IF (IKODE.GE.1 .AND.ISW(2).GE.1)  WRITE (ISYSWR,1005)  UP         MPRINT      56
      RETURN                                                            MPRINT      57
 1000 FORMAT(/4X"FCN VALUE"5X"CALLS"4X"TIME"5X"EDM"7X"INT.EXT. PARAMETER MAY117+     15
     1"5X"VALUE"12X"ERROR"6X"INTERN.VALUE  INT.STEP SIZE")              MAY117+     16
 1001 FORMAT(E15.7, I7, F9.3, E11.2, I6, I4, 1XA10, E17.8,3E14.5)       MAY117+     17
 1002 FORMAT(42X                     I6, I4, 1XA10, E17.8,3E14.5)       MAY117+     18
 1003 FORMAT (1H  47X  I4,1XA10,4E14.5)                                 MPRINT      63
 1004 FORMAT (1H 52X  42HWARNING -   - ABOVE PARAMETER IS AT LIMIT.)    MPRINT      64
 1005 FORMAT (/45X   40HERRORS CORRESPOND TO FUNCTION CHANGE OF F10.4)  MPRINT      65
 1006 FORMAT (1H140X36HRESULTS OF FULL MINOS ERROR ANALYSIS/41X36(1H*)/ MPRINT      66
     1       /20X 15A4//80X   9HPARABOLIC4X  26H.. ... MINOS ERRORS .... MPRINT      67
     2..           / 4X9HFCN VALUE5X5HCALLS4X4HTIME4X6H EDM   4X   75HIN MPRINT      68
     3T.EXT. PARAMETER     VALUE         ERROR         POSITIVE      NEGA MPRINT      69
     4TIVE   )                                                          MPRINT      70
      END                                                               MPRINT      71
```

```
      SUBROUTINE MPUNCH                                                 MPUNCH     2
CC        PUNCHES CURRENT PARAMETER VALUES AND STEP SIZES ONTO CARDS    MPUNCH     3
CC        IN FORMAT WHICH CAN BE REREAD BY MINUIT FOR RESTARTING.       MPUNCH     4
CC        THE COVARIANCE MATRIX IS ALSO PUNCHED IF IT EXISTS.           MPUNCH     5
CC                                                                      MPUNCH     6
       COMMON                                                           COMMONU    2
     1/MINERR/ ERP(30)   ,ERN(30)                                       COMMONU    4
     2/PARINT/ X(15)     ,XT(15)    ,DIRIN(15) ,MAXINT    ,NPAR         COMMONU    5
     3/PAREXT/ U(30)     ,NAM(30)   ,WERR(30)  ,MAXEXT    ,NU           COMMONU    6
     4/LIMITS/ ALIM(30)  ,BLIM(30)  ,LCODE(30) ,LCORSP(30) ,LIMSET      COMMONU    7
     5/VARIAN/ V(15,15)                                                 COMMONU    8
     7/FIX   / IPFIX(15) ,XS(15)    ,XTS(15)   ,WTS(15)  ,NPFIX         COMMONU    9
     C/CASC/ JH, JL, Y(16)                                              COMMONU   10
     F/DERIVA/ G(30)     ,G2(30)                                        COMMONU   11
     G/SIMVEC/ P(15,16)  ,PSTAR(15) ,PSTST(15) ,PBAR(15)  ,PRHO(15)     COMMONU   12
     J/VARIAT/ VT(15,15)                                                COMMONU   13
     6/UNIT  / ISYSRD    ,ISYSWR    ,ISYSPU                             COMMONU   27
     8/TITLE / TITLE(13) ,DATE(2)   ,ISW(7)    ,NBLOCK                  COMMONU   28
     9/CONVER/ EPSI ,APSI ,VTEST    ,NSTEPQ    ,NFCN      ,NFCNMX       COMMONU   29
     A/CARD  / CWORD     ,CWORD2    ,CWORD3    ,WORD7(7)                COMMONU   30
     B/MINIMA/ AMIN      ,UP        ,NEWMIN    ,ITAUR     ,SIGMA        COMMONU   31
     K/COMMND/ NNAME     ,CNAME(25) ,CNAM2(25) ,CNAM3(25)               COMMONU   32
      DIMENSION VC(7)                                                   MPUNCH     8
      TI = FTIME(DUMMY)                                                 MPUNCH     9
      WRITE (ISYSPU,1002)                                               MPUNCH    10
      WRITE (ISYSPU,1000)  TITLE,DATE,TI                                MPUNCH    11
      DO 200 I= 1, NU                                                   MPUNCH    12
      IF (NAM (I))  20,200,20                                           MPUNCH    13
C    (AVOID PUNCHING MINUS ZERO)                                        MPUNCH    14
   20 IF (WERR(I) .EQ. 0.0) WERR(I) = 0.0                               MPUNCH    15
      IF (LCODE(I) .GT. 1)  GO TO 100                                   MPUNCH    16
C         PARAMETER WITHOUT LIMITS                                      MPUNCH    17
      WRITE (ISYSPU,1001)  I,NAM(I),U(I),WERR(I)                        MPUNCH    18
      GO TO 200                                                         MPUNCH    19
C         PARAMETER WITH LIMITS                                         MPUNCH    20
  100 CONTINUE                                                          MPUNCH    21
      WRITE (ISYSPU,1001) I,NAM(I),U(I),WERR(I),ALIM(I),BLIM(I)         MPUNCH    22
  200 CONTINUE                                                          MPUNCH    23
      WRITE (ISYSPU,1002)                                               MPUNCH    24
      IF (ISW(2) .LT. 3)  GO TO 300                                     MPUNCH    25
      WRITE (ISYSPU,1003)  NPAR                                         MPUNCH    26
      K = 0                                                             MPUNCH    27
      KC = 0                                                            MPUNCH    28
      DO 250 I= 1, NPAR                                                 MPUNCH    29
      DO 250 J= 1, NPAR                                                 MPUNCH    30
      K = K + 1                                                         MPUNCH    31
      VC(K) = V(I,J)                                                    MPUNCH    32
      IF (K .NE. 7)  GO TO 250                                          MPUNCH    33
      K = 0                                                             MPUNCH    34
      KC = KC + 1                                                       MPUNCH    35
      WRITE(ISYSPU,1004)  VC,TI,NBLOCK,KC                               MPUNCH    36
  250 CONTINUE                                                          MPUNCH    37
      IF (K .EQ. 0)  GO TO 300                                          MPUNCH    38
      KP1 = K + 1                                                       MPUNCH    39
      DO 260 I= KP1, 7                                                  MPUNCH    40
  260 VC(I) = 0.                                                        MPUNCH    41
      KC = KC + 1                                                       MPUNCH    42
```

```
      WRITE(ISYSPU,1004)  VC,TI,NBLOCK,KC
  300 RETURN
 1000 FORMAT (15A4,10X,F6.3)
 1001 FORMAT (I10,A10,4E10.4)
 1002 FORMAT (1H )
 1003 FORMAT (10HCOVARIANCE I10)
 1004 FORMAT (7E10.3,F6.3,2I2)
      END
```

```
      FUNCTION PINTF(PEXTI,I)                                           PINTF      2
CC        CALCULATES THE INTERNAL PARAMETER VALUE PINTF CORRESPONDING   PINTF      3
CC        TO THE EXTERNAL VALUE PEXTI FOR PARAMETER I.                  PINTF      4
CC                                                                      PINTF      5
       COMMON                                                           COMMONU    2
     1/MINERR/ ERP(30)    ,ERN(30)                                      COMMONU    4
     2/PARINT/ X(15)      ,XT(15)     ,DIRIN(15)  ,MAXINT     ,NPAR     COMMONU    5
     3/PAREXT/ U(30)      ,NAM(30)    ,WERR(30)   ,MAXEXT     ,NU       COMMONU    6
     4/LIMITS/ ALIM(30)   ,BLIM(30)   ,LCODE(30)  ,LCORSP(30) ,LIMSET   COMMONU    7
     5/VARIAN/ V(15,15)                                                 COMMONU    8
     7/FIX   / IPFIX(15)  ,XS(15)     ,XTS(15)    ,WTS(15)  ,NPFIX      COMMONU    9
     C/CASC/ JH, JL, Y(16)                                              COMMONU   10
     F/DERIVA/ G(30)      ,G2(30)                                       COMMONU   11
     G/SIMVEC/ P(15,16)   ,PSTAR(15)  ,PSTST(15)  ,PBAR(15)   ,PRHO(15) COMMONU   12
     J/VARIAT/ VT(15,15)                                                COMMONU   13
     6/UNIT  / ISYSRD     ,ISYSWR     ,ISYSPU                           COMMONU   27
     8/TITLE / TITLE(13)  ,DATE(2)    ,ISW(7)     ,NBLOCK               COMMONU   28
     9/CONVER/ EPSI ,APSI ,VTEST      ,NSTEPQ     ,NFCN       ,NFCNMX   COMMONU   29
     A/CARD  / CWORD      ,CWORD2     ,CWORD3     ,WORD7(7)             COMMONU   30
     B/MINIMA/ AMIN       ,UP         ,NEWMIN     ,ITAUR      ,SIGMA    COMMONU   31
     K/COMMND/ NNAME      ,CNAME(25)  ,CNAM2(25)  ,CNAM3(25)            COMMONU   32
      DATA BIG, SMALL / 1.570796326795 , -1.570796326795 /              PINTF      7
      IGO = LCODE(I)                                                    PINTF      8
      GO TO (100,200,300,400),IGO                                       PINTF      9
C--       IGO = 1  MEANS NO LIMITS                                      PINTF     10
  100 PINTF = PEXTI                                                     PINTF     11
      GO TO 800                                                         PINTF     12
  200 CONTINUE                                                          PINTF     13
  300 CONTINUE                                                          PINTF     14
C--       IGO = 4  MEANS THERE ARE TWO LIMITS                           PINTF     15
  400 ALIMI = ALIM(I)                                                   PINTF     16
      BLIMI = BLIM(I)                                                   PINTF     17
      IF (PEXTI-ALIMI)  440,500,460                                     PINTF     18
  440 A = SMALL                                                         PINTF     19
  450 PINTF = A                                                         PINTF     20
      PEXTI = ALIMI + 0.5* (BLIMI-ALIMI) *(SIN(A) +1.0)                 PINTF     21
      LIMSET=1                                                          PINTF     22
C     WRITE (ISYSWR,241) I                                              PINTF     23
      GO TO 800                                                         PINTF     24
  460 IF (BLIMI-PEXTI)  470,520,480                                     PINTF     25
  470 A = BIG                                                           PINTF     26
      GO TO 450                                                         PINTF     27
  480 YY=2.0*(PEXTI-ALIMI)/(BLIMI-ALIMI) - 1.0                          PINTF     28
      PINTF = ATAN(YY/SQRT(1.0- YY**2) )                                PINTF     29
      GO TO 800                                                         PINTF     30
  500 PINTF = SMALL                                                     PINTF     31
      GO TO 800                                                         PINTF     32
  520 PINTF = BIG                                                       PINTF     33
  800 RETURN                                                            PINTF     34
  241 FORMAT (  19H WARNING - VARIABLEI3,  46H HAS BEEN BROUGHT BACK INS PINTF     35
     1IDE LIMITS BY PINTF.)                                             PINTF     36
      END                                                               PINTF     37
```

```
      SUBROUTINE PLTCON(NSPT,SPT)                                       PLTCON     2
CC        PLOTS POINTS IN ARRAY SPT ONTO ONE PAGE WITH LABELLED AXES    PLTCON     3
CC        NSPT IS THE NUMBER OF POINTS TO BE PLOTTED                    PLTCON     4
CC        SPT(1,I) = X-COORD. OF ITH POINT                              PLTCON     5
CC        SPT(2,I) = Y-COORD. OF ITH POINT                              PLTCON     6
CC        SPT(3,I) = CHARACTER TO BE PLOTTED AT THIS POSITION           PLTCON     7
CC                                                                      PLTCON     8
       COMMON                                                           COMMONU    2
     1/MINERR/ ERP(30)    ,ERN(30)                                      COMMONU    4
     2/PARINT/ X(15)      ,XT(15)     ,DIRIN(15)  ,MAXINT     ,NPAR     COMMONU    5
     3/PAREXT/ U(30)      ,NAM(30)    ,WERR(30)   ,MAXEXT     ,NU       COMMONU    6
     4/LIMITS/ ALIM(30)   ,BLIM(30)   ,LCODE(30)  ,LCORSP(30) ,LIMSET   COMMONU    7
     5/VARIAN/ V(15,15)                                                 COMMONU    8
     7/FIX   / IPFIX(15)  ,XS(15)     ,XTS(15)    ,WTS(15)   ,NPFIX     COMMONU    9
     C/CASC/ JH, JL, Y(16)                                              COMMONU   10
     F/DERIVA/ G(30)      ,G2(30)                                       COMMONU   11
     G/SIMVEC/ P(15,16)   ,PSTAR(15)  ,PSTST(15)  ,PBAR(15)   ,PRHO(15) COMMONU   12
     J/VARIAT/ VT(15,15)                                                COMMONU   13
     6/UNIT  / ISYSRD     ,ISYSWR     ,ISYSPU                           COMMONU   27
     8/TITLE / TITLE(13)  ,DATE(2)    ,ISW(7)     ,NBLOCK               COMMONU   28
     9/CONVER/ EPSI  ,APSI ,VTEST     ,NSTEPQ     ,NFCN       ,NFCNMX   COMMONU   29
     A/CARD  / CWORD      ,CWORD2     ,CWORD3     ,WORD7(7)             COMMONU   30
     B/MINIMA/ AMIN       ,UP         ,NEWMIN     ,ITAUR      ,SIGMA    COMMONU   31
     K/COMMND/ NNAME      ,CNAME(25)  ,CNAM2(25)  ,CNAM3(25)            COMMONU   32
      DIMENSION SPT(3,2) , ALINE(100), SAV(3)                           PLTCON    10
      DIMENSION FMT(6), FMTIN(12)                                       PLTCON    11
      DATA DOT,SLASH,AI,ZERO,ABLANK/1H., 1H/, 1HI, 1H0, 1H /            PLTCON    12
      DATA MAXNX,MAXNY/ 98, 56/                                         PLTCON    13
      DATA (FMT(IU),IU=1,6)/4H(5X,,6HF15.7,,1H ,6HF15.7,,1H ,6HF15.7)/  PLTCON    14
      DATA (FMTIN(IU),IU=1,12)/1H , 4H 5X,, 4H10X,, 4H15X,, 4H20X,,     PLTCON    15
     *      4H25X,, 4H30X,, 4H35X,, 4H40X,, 4H45X,, 4H50X,, 4H55X,/     PLTCON    16
      IF (NSPT .LE. 1)  RETURN                                          PLTCON    17
      XBEST = SPT(1,1)                                                  PLTCON    18
      YBEST = SPT(2,1)                                                  PLTCON    19
      CHBEST = SPT(3,1)                                                 PLTCON    20
C         ORDER THE POINTS BY DECREASING Y                              PLTCON    21
      KM1 = NSPT - 1                                                    PLTCON    22
      DO 150 I= 1, KM1                                                  PLTCON    23
      IQUIT = 0                                                         PLTCON    24
      NI = NSPT - I                                                     PLTCON    25
      DO 140 J= 1, NI                                                   PLTCON    26
      IF (SPT(2,J) - SPT(2,J+1))  120, 140, 140                         PLTCON    27
  120   DO 130 I3= 1, 3                                                 PLTCON    28
        SAV(I3) = SPT(I3,J)                                             PLTCON    29
        SPT(I3,J) = SPT(I3,J+1)                                         PLTCON    30
  130   SPT(I3,J+1) = SAV(I3)                                           PLTCON    31
      IQUIT = 1                                                         PLTCON    32
  140 CONTINUE                                                          PLTCON    33
      IF (IQUIT)  160, 160, 150                                         PLTCON    34
  150 CONTINUE                                                          PLTCON    35
  160 CONTINUE                                                          PLTCON    36
C         FIND EXTREME VALUES                                           PLTCON    37
      XMAX = SPT(1,1)                                                   PLTCON    38
      XMIN = XMAX                                                       PLTCON    39
      DO 200 I= 1, NSPT                                                 PLTCON    40
        IF (SPT(1,I) .GT. XMAX)  XMAX = SPT(1,I)                        PLTCON    41
        IF (SPT(1,I) .LT. XMIN)  XMIN = SPT(1,I)                        PLTCON    42
```

```
  200 CONTINUE
      CALL BINSIZ(XMIN,XMAX,MAXNX,XMIN,XMAX,NX,BWIDX)
      ANX = NX
      YMAX = SPT(2,1)
      YMIN = SPT(2,NSPT)
      IF (YMAX .EQ. YBEST)  YMAX=YBEST + 1.0
      CALL BINSIZ(YMIN,YMAX,MAXNY,YMIN,YMAX,NY,BWIDY)
      ANY = NY
      IF (CHBEST .EQ. ABLANK)  GO TO 50
      XBEST = 0.5 * (XMAX+XMIN)
      YBEST = 0.5 * (YMAX+YMIN)
   50 CONTINUE
C         FIND SCALE CONSTANTS
      AX = 1.0/BWIDX
      AY = 1.0/BWIDY
      BX = -AX*XMIN + 2.0
      BY = -AY*YMIN - 2.0
C         CONVERT POINTS TO GRID POSITIONS
      DO 300 I= 1, NSPT
      SPT(1,I) = AX*SPT(1,I) + BX
  300 SPT(2,I) = ANY-AY*SPT(2,I) - BY
      NXBEST = AX*XBEST + BX
      NYBEST = ANY  - AY*YBEST - BY
C         PRINT THE CONTOURS
      NY = ANY + 1.
      NX = ANX + 2.
      ISP1 = 1
      LINODD = 1
      DO 400 I= 1, NY
      DO 310 IBK= 1, NX
  310 ALINE (IBK) = ABLANK
      ALINE(1) = DOT
      ALINE(NX) = DOT
      ALINE(NXBEST) = DOT
      IF (I.NE.1 .AND. I.NE.NYBEST .AND. I.NE.NY)  GO TO 320
      DO 315 J= 1, NX
  315 ALINE(J) = DOT
  320 CONTINUE
      YPRT = YMAX - FLOAT(I-1)*BWIDY
      IF (ISP1 .GT. NSPT)  GO TO 350
        DO 340 K= ISP1,NSPT
      KS = SPT(2,K)
      IF (KS .GT. I)  GO TO 345
      IX = SPT(1,K)
  340 ALINE(IX) = SPT(3,K)
        ISP1 = NSPT + 1
        GO TO 350
  345   ISP1 = K
  350 CONTINUE
      IF (LINODD .EQ. 1 .OR. I .EQ. NY)  GO TO 380
      LINODD = 1
      WRITE (ISYSWR, 1007)       (ALINE(IW),IW=1,NX)
      GO TO 400
  380 WRITE (ISYSWR, 1001) YPRT, (ALINE(IW),IW=1,NX)
      LINODD = 0
  400 CONTINUE
      DO 410 IBK= 1, NX
```

```
  410 ALINE (IBK) = ABLANK                                        PLTCON   100
      ALINE(1) = SLASH                                            PLTCON   101
      ALINE(NXBEST) = AI                                          PLTCON   102
      ALINE(NX-1) = SLASH                                         PLTCON   103
      WRITE (ISYSWR, 1007)      (ALINE(IW),IW=1,NX)               PLTCON   104
      N1 = NXBEST/5 - 1                                           PLTCON   105
      IF (N1 .LT. 1)  N1 = 1                                      PLTCON   106
      IF (N1 .GT. 12)  N1 = 12                                    PLTCON   107
      N2 = NX/5 - N1 - 3                                          PLTCON   108
      IF (N2 .GT. 0)  GO TO 420                                   PLTCON   109
      IF (N1 .GT. 1)  N1= N1-1                                    PLTCON   110
      N2 = 1                                                      PLTCON   111
  420 IF (N2 .GT. 12)  N2 = 12                                    PLTCON   112
      FMT(3) = FMTIN(N1)                                          PLTCON   113
      FMT(5) = FMTIN(N2)                                          PLTCON   114
      WRITE (ISYSWR,FMT )  XMIN,XBEST,XMAX                        PLTCON   115
      WRITE (ISYSWR,1002)  BWIDX                                  PLTCON   116
  500 RETURN                                                      PLTCON   117
 1001 FORMAT (F15.7,3H ..,100A1)                                  PLTCON   118
 1002 FORMAT (40X,13HONE COLUMN =   F13.7)                        PLTCON   119
 1003 FORMAT (5XA10)                                              PLTCON   120
 1007 FORMAT (18X , 100A1)                                        PLTCON   121
      END                                                         PLTCON   122
```

```
            SUBROUTINE RAZZIA(YNEW,PNEW)                                     RAZZIA       2
      CC        CALLED ONLY BY SIMPLEX (AND IMPROV) TO ADD A NEW POINT       RAZZIA       3
      CC        AND REMOVE AN OLD ONE FROM THE CURRENT SIMPLEX, AND GET THE  RAZZIA       4
      CC        ESTIMATED DISTANCE TO MINIMUM.                               RAZZIA       5
      CC                                                                     RAZZIA       6
             COMMON                                                          COMMONU      2
           1/MINERR/ ERP(30)     ,ERN(30)                                    COMMONU      4
           2/PARINT/ X(15)       ,XT(15)     ,DIRIN(15)  ,MAXINT     ,NPAR   COMMONU      5
           3/PAREXT/ U(30)       ,NAM(30)    ,WERR(30)   ,MAXEXT     ,NU     COMMONU      6
           4/LIMITS/ ALIM(30)    ,BLIM(30)   ,LCODE(30)  ,LCORSP(30) ,LIMSET COMMONU      7
           5/VARIAN/ V(15,15)                                                COMMONU      8
           7/FIX   / IPFIX(15)   ,XS(15)     ,XTS(15)    ,WTS(15)  ,NPFIX    COMMONU      9
           C/CASC/ JH, JL, Y(16)                                             COMMONU     10
           F/DERIVA/ G(30)       ,G2(30)                                     COMMONU     11
           G/SIMVEC/ P(15,16)    ,PSTAR(15)  ,PSTST(15)  ,PBAR(15)   ,PRHO(15) COMMONU   12
           J/VARIAT/ VT(15,15)                                               COMMONU     13
           6/UNIT  / ISYSRD      ,ISYSWR     ,ISYSPU                         COMMONU     27
           8/TITLE / TITLE(13)   ,DATE(2)    ,ISW(7)     ,NBLOCK             COMMONU     28
           9/CONVER/ EPSI ,APSI  ,VTEST      ,NSTEPQ     ,NFCN       ,NFCNMX COMMONU     29
           A/CARD  / CWORD       ,CWORD2     ,CWORD3     ,WORD7(7)           COMMONU     30
           B/MINIMA/ AMIN        ,UP         ,NEWMIN     ,ITAUR      ,SIGMA  COMMONU     31
           K/COMMND/ NNAME       ,CNAME(25)  ,CNAM2(25)  ,CNAM3(25)          COMMONU     32
            DIMENSION PNEW(15)                                               RAZZIA       8
            DO 10 I=1,NPAR                                                   RAZZIA       9
       10   P(I,JH)=PNEW(I)                                                  RAZZIA      10
            Y(JH)=YNEW                                                       RAZZIA      11
            IF(YNEW.GE.AMIN) GO TO 18                                        RAZZIA      12
            DO 15 I=1,NPAR                                                   RAZZIA      13
       15   X(I)=PNEW(I)                                                     RAZZIA      14
            CALL INTOEX(X)                                                   RAZZIA      15
            AMIN=YNEW                                                        RAZZIA      16
            JL=JH                                                            RAZZIA      17
       18   CONTINUE                                                         RAZZIA      18
            JH=1                                                             RAZZIA      19
            NPARP1=NPAR+1                                                    RAZZIA      20
       20   DO 25 J=2,NPARP1                                                 RAZZIA      21
            IF (Y(J) .GT. Y(JH))  JH = J                                     RAZZIA      22
       25   CONTINUE                                                         RAZZIA      23
            SIGMA = Y(JH) - Y(JL)                                            RAZZIA      24
            IF (SIGMA .LE. 0.)  GO TO 45                                     RAZZIA      25
            US = 1.0/SIGMA                                                   RAZZIA      26
            DO 35 I= 1, NPAR                                                 RAZZIA      27
            PBIG = P(I,1)                                                    RAZZIA      28
            PLIT = PBIG                                                      RAZZIA      29
            DO 30 J= 2, NPARP1                                               RAZZIA      30
            IF (P(I,J) .GT. PBIG)  PBIG = P(I,J)                             RAZZIA      31
            IF (P(I,J) .LT. PLIT)  PLIT = P(I,J)                             RAZZIA      32
         30 CONTINUE                                                         RAZZIA      33
            DIRIN(I) = PBIG - PLIT                                           RAZZIA      34
            IF (ITAUR .LT. 1 )  V(I,I) = 0.5*(V(I,I) +US*DIRIN(I)**2)        RAZZIA      35
         35 CONTINUE                                                         RAZZIA      36
         40 RETURN                                                           RAZZIA      37
         45 WRITE (ISYSWR, 1000)  NPAR                                       RAZZIA      38
            GO TO 40                                                         RAZZIA      39
       1000 FORMAT (  59H0***** FUNCTION VALUE DOES NOT SEEM TO DEPEND ON ANY RAZZIA      40
           1OF THE I3,   20H VARIABLE PARAMETERS/15X   58HVERIFY THAT STEP SIZ RAZZIA    41
           2ES ARE BIG ENOUGH AND CHECK FCN LOGIC./1X 81(1H*)/1X 81(1H*)///)  RAZZIA     42
```

```
      END                                                          RAZZIA      43
```

```
      SUBROUTINE RESTOR(K)                                              RESTOR     2
CC        RESTORES A FIXED PARAMETER TO VARIABLE STATUS                 RESTOR     3
CC        BY INSERTING IT INTO THE INTERNAL PARAMETER LIST AT THE       RESTOR     4
CC        APPROPRIATE PLACE.                                            RESTOR     5
CC                                                                      RESTOR     6
       COMMON                                                           COMMONU    2
     1/MINERR/ ERP(30)    ,ERN(30)                                      COMMONU    4
     2/PARINT/ X(15)      ,XT(15)    ,DIRIN(15)  ,MAXINT    ,NPAR       COMMONU    5
     3/PAREXT/ U(30)      ,NAM(30)   ,WERR(30)   ,MAXEXT    ,NU         COMMONU    6
     4/LIMITS/ ALIM(30)   ,BLIM(30)  ,LCODE(30)  ,LCORSP(30) ,LIMSET    COMMONU    7
     5/VARIAN/ V(15,15)                                                 COMMONU    8
     7/FIX   / IPFIX(15)  ,XS(15)    ,XTS(15)    ,WTS(15)   ,NPFIX      COMMONU    9
     C/CASC/ JH, JL, Y(16)                                              COMMONU   10
     F/DERIVA/ G(30)      ,G2(30)                                       COMMONU   11
     G/SIMVEC/ P(15,16)   ,PSTAR(15) ,PSTST(15)  ,PBAR(15)  ,PRHO(15)   COMMONU   12
     J/VARIAT/ VT(15,15)                                                COMMONU   13
     6/UNIT  / ISYSRD     ,ISYSWR    ,ISYSPU                            COMMONU   27
     8/TITLE / TITLE(13)  ,DATE(2)   ,ISW(7)     ,NBLOCK                COMMONU   28
     9/CONVER/ EPSI ,APSI ,VTEST     ,NSTEPQ     ,NFCN      ,NFCNMX     COMMONU   29
     A/CARD  / CWORD      ,CWORD2    ,CWORD3     ,WORD7(7)              COMMONU   30
     B/MINIMA/ AMIN       ,UP        ,NEWMIN     ,ITAUR     ,SIGMA      COMMONU   31
     K/COMMND/ NNAME      ,CNAME(25) ,CNAM2(25)  ,CNAM3(25)             COMMONU   32
C--       K = 0 MEANS RESTORE ALL PARAMETERS                            RESTOR     8
C--       K = 1 MEANS RESTORE THE LAST PARAMETER FIXED                  RESTOR     9
C--       K = -I MEANS RESTORE EXTERNAL PARAMETER I (IF POSSIBLE)       RESTOR    10
C--       IQ = FIX-LOCATION WHERE INTERNAL PARAMETERS WERE STORED       RESTOR    11
C--       IR = EXTERNAL NUMBER OF PARAMETER BEING RESTORED              RESTOR    12
C--       IS = INTERNAL NUMBER OF PARAMETER BEING RESTORED              RESTOR    13
      IF (K .GT. 1)  WRITE (ISYSWR,510)                                 RESTOR    14
      IF (NPFIX .LT. 1)  WRITE (ISYSWR,500)                             RESTOR    15
      IF (K.EQ.1 .OR. K.EQ.0)  GO TO 50                                 RESTOR    16
C                    RELEASE PARAMETER WITH SPECIFIED EXTERNAL NUMBER   RESTOR    17
      KA = IABS(K)                                                      RESTOR    18
      IF (LCORSP(KA) .EQ. 0)  GO TO 15                                  RESTOR    19
      WRITE (ISYSWR,540)                                                RESTOR    20
  540 FORMAT (49H0ERROR.  PARAMETER SPECIFIED IS ALREADY VARIABLE. /)   RESTOR    21
      RETURN                                                            RESTOR    22
   15 IF (NPFIX .LT. 1)  GO TO 21                                       RESTOR    23
      DO 20 IK= 1, NPFIX                                                RESTOR    24
      IF (IPFIX(IK) .EQ. KA)  GO TO 24                                  RESTOR    25
   20 CONTINUE                                                          RESTOR    26
   21 WRITE (ISYSWR,530)                                                RESTOR    27
  530 FORMAT (53H0ERROR.  PARAMETER SPECIFIED HAS NEVER BEEN VARIABLE./) RESTOR    28
      RETURN                                                            RESTOR    29
   24 IF (IK .EQ. NPFIX)  GO TO 50                                      RESTOR    30
C                    MOVE SPECIFIED PARAMETER TO END OF LIST            RESTOR    31
      IPSAV = IPFIX(IK)                                                 RESTOR    32
      XSSAV = XS(IK)                                                    RESTOR    33
      XTSSAV = XTS(IK)                                                  RESTOR    34
      WTSSAV = WTS(IK)                                                  RESTOR    35
      IKP1 = IK + 1                                                     RESTOR    36
      DO 30 I= IKP1,NPFIX                                               RESTOR    37
      IPFIX(I-1) = IPFIX(I)                                             RESTOR    38
      XS(I-1) = XS(I)                                                   RESTOR    39
      XTS(I-1) = XTS(I)                                                 RESTOR    40
   30 WTS(I-1) = WTS(I)                                                 RESTOR    41
      IPFIX(NPFIX) = IPSAV                                              RESTOR    42
```

```
      XS(NPFIX) = XSSAV
      XTS(NPFIX) = XTSSAV
      WTS(NPFIX) = WTSSAV
C                   RESTORE LAST PARAMETER IN LIST  -- IPFIX(NPFIX)
   50 CONTINUE
      IF (NPFIX .LT. 1)  GO TO 300
      IR = IPFIX(NPFIX)
      IS = 0
      DO 100 IJ= IR, NU
      IK = NU + IR - IJ
      IF (LCORSP(IK))  100,100,85
   85 LC = LCORSP(IK) + 1
      IS = LC - 1
      LCORSP(IK) = LC
      X(LC) = X(LC-1)
      XT(LC) = XT(LC-1)
      DIRIN(LC) = DIRIN(LC-1)
  100 CONTINUE
      NPAR = NPAR + 1
      IF (IS .EQ. 0)   IS = NPAR
      LCORSP(IR) = IS
      IQ = NPFIX
      X(IS) = XS(IQ)
      XT(IS) = XTS(IQ)
      DIRIN(IS) = WTS(IQ)
      NPFIX = NPFIX - 1
      ISW(2) = 0
      IF (ITAUR .LT. 1)  WRITE(ISYSWR,520) IR,NAM(IR)
      IF (K.EQ.0)  GO TO 50
  300 RETURN
  500 FORMAT (  61H0ERROR IN CALL TO RESTOR.  THERE ARE NO MORE FIXED PA
     1RAMETERS/)
  510 FORMAT (  52H0ERROR IN CALL TO RESTOR.  ARGUMENT GREATER THAN ONE/
     1)
  520 FORMAT (20X,   9HPARAMETER,I4,2H, ,A10,  22H RESTORED TO VARIABLE.
     1)
      END
```

```
          SUBROUTINE SEEK                                                     SEEK          2
CC           PERFORMS A ROUGH-MINIMIZATION BY MONTE CARLO SEARCH.            SEEK          3
CC           EACH TIME A NEW MINIMUM IS FOUND, THE SEARCH AREA IS SHIFTED    SEEK          4
CC           TO BE CENTERED AT THE BEST VALUE.   RANDOM POINTS ARE CHOSEN     SEEK          5
CC           UNIFORMLY OVER A HYPERCUBE DETERMINED BY CURRENT STEP SIZES.    SEEK          6
CC                                                                           SEEK          7
          COMMON                                                             COMMONU       2
         1/MINERR/ ERP(30)      ,ERN(30)                                     COMMONU       4
         2/PARINT/ X(15)        ,XT(15)     ,DIRIN(15)  ,MAXINT     ,NPAR    COMMONU       5
         3/PAREXT/ U(30)        ,NAM(30)    ,WERR(30)   ,MAXEXT     ,NU      COMMONU       6
         4/LIMITS/ ALIM(30)     ,BLIM(30)   ,LCODE(30)  ,LCORSP(30) ,LIMSET  COMMONU       7
         5/VARIAN/ V(15,15)                                                  COMMONU       8
         7/FIX   / IPFIX(15)    ,XS(15)     ,XTS(15)    ,WTS(15)  ,NPFIX     COMMONU       9
         C/CASC/ JH, JL, Y(16)                                               COMMONU      10
         F/DERIVA/ G(30)        ,G2(30)                                      COMMONU      11
         G/SIMVEC/ P(15,16)     ,PSTAR(15)  ,PSTST(15)  ,PBAR(15)   ,PRHO(15) COMMONU     12
         J/VARIAT/ VT(15,15)                                                 COMMONU      13
         6/UNIT  / ISYSRD       ,ISYSWR     ,ISYSPU                          COMMONU      27
         8/TITLE / TITLE(13)    ,DATE(2)    ,ISW(7)     ,NBLOCK              COMMONU      28
         9/CONVER/ EPSI ,APSI   ,VTEST      ,NSTEPQ     ,NFCN       ,NFCNMX  COMMONU      29
         A/CARD  / CWORD        ,CWORD2     ,CWORD3     ,WORD7(7)            COMMONU      30
         B/MINIMA/ AMIN         ,UP         ,NEWMIN     ,ITAUR      ,SIGMA   COMMONU      31
         K/COMMND/ NNAME        ,CNAME(25)  ,CNAM2(25)  ,CNAM3(25)           COMMONU      32
          DIMENSION AMID(15) , N(15)                                         SEEK         10
          WRITE (ISYSWR,1000)                                                SEEK         15
          NUMBER = WORD7(1) + 0.5                                            SEEK         16
          IF (NUMBER .LE. 0)  NUMBER = 10 * NPAR                             SEEK         17
          IFLAG = 4                                                          SEEK         18
C--          INITIAL VALUES                                                  SEEK         19
          DO 100 J= 1, NU                                                    SEEK         20
          NI = LCORSP(J)                                                     SEEK         21
          IF (NI .LE. 0)  GO TO 100                                          SEEK         22
          N(NI) = J                                                          SEEK         23
          AMID(NI) = U(J)                                                    SEEK         24
      100 CONTINUE                                                           SEEK         25
          NCYCL = 0                                                          SEEK         26
C                 MONTE CARLO SEARCH OVER ENTIRE VARIABLE PARAMETER SPACE    SEEK         27
          DO 650 INUM= 1, NUMBER                                             SEEK         28
          DO 200 I2= 1, NPAR                                                 SEEK         29
          I3 = N(I2)                                                         SEEK         30
C                                         RANDOM POINTS IN UNIFORM DISTR.    SEEK         31
      180 XPLS=2.0*(RGEN(I2)-0.5)                                            SEEK         32
          U(I3) = AMID(I2) + XPLS*WERR(I3)                                   SEEK         33
          IF (LCODE(I3) .LE. 1)  GO TO 200                                   SEEK         34
          IF (U(I3) .GT. BLIM(I3) .OR. U(I3) .LT. ALIM(I3))  GO TO 180       SEEK         35
      200 CONTINUE                                                           SEEK         36
          CALL FCN(NPAR,G,F,U,IFLAG)                                         SEEK         37
          NFCN = NFCN + 1                                                    SEEK         38
          IF (F .GE. AMIN)  GO TO 650                                        SEEK         39
          AMIN = F                                                           SEEK         40
          DO 500 I= 1, NPAR                                                  SEEK         41
          J = N(I)                                                           SEEK         42
      500 AMID(I) = U(J)                                                     SEEK         43
          NCYCL = NCYCL + 1                                                  SEEK         44
          IF (ISW(5) .LT. 2)  GO TO 650                                      SEEK         45
          CALL EXTOIN(X)                                                     SEEK         46
          IF (ISW(5).GE. 3 .OR. MOD(NCYCL,10).EQ.1)  CALL MPRINT(0,AMIN)     SEEK         47
```

```
      650 CONTINUE                                              SEEK        48
C                            SEARCH FINISHED. SET U TO BEST VALUES SEEK    49
          DO 800 I= 1, NPAR                                     SEEK        50
          NI = N(I)                                             SEEK        51
      800 U(NI) = AMID(I)                                       SEEK        52
          CALL EXTOIN(X)                                        SEEK        53
          WRITE (ISYSWR,1005)                                   SEEK        54
          CALL MPRINT(1,AMIN)                                   SEEK        55
          RETURN                                                SEEK        56
     1000 FORMAT (  35H SEEK -- MONTE CARLO MINIMUM SEARCH )    SEEK        57
     1005 FORMAT (  28H0BEST VALUE FOUND IN SEEK IS )           SEEK        58
          END                                                   SEEK        59
```

```
            SUBROUTINE SIMPLX                                               SIMPLX     2
      CC        PERFORMS A MINIMIZATION USING THE SIMPLEX METHOD OF NELDER  SIMPLX     3
      CC        AND MEAD (REF. -- COMP. J. 7,308 (1965)).                   SIMPLX     4
      CC                                                                    SIMPLX     5
             COMMON                                                         COMMONU    2
           1/MINERR/ ERP(30)     ,ERN(30)                                   COMMONU    4
           2/PARINT/ X(15)       ,XT(15)      ,DIRIN(15)  ,MAXINT     ,NPAR COMMONU    5
           3/PAREXT/ U(30)       ,NAM(30)     ,WERR(30)   ,MAXEXT     ,NU   COMMONU    6
           4/LIMITS/ ALIM(30)    ,BLIM(30)    ,LCODE(30)  ,LCORSP(30) ,LIMSET COMMONU  7
           5/VARIAN/ V(15,15)                                               COMMONU    8
           7/FIX   / IPFIX(15)   ,XS(15)      ,XTS(15)    ,WTS(15)   ,NPFIX COMMONU    9
           C/CASC/ JH, JL, Y(16)                                            COMMONU   10
           F/DERIVA/ G(30)       ,G2(30)                                    COMMONU   11
           G/SIMVEC/ P(15,16)    ,PSTAR(15)   ,PSTST(15)  ,PBAR(15)   ,PRHO(15) COMMONU 12
           J/VARIAT/ VT(15,15)                                              COMMONU   13
           6/UNIT  / ISYSRD      ,ISYSWR      ,ISYSPU                       COMMONU   27
           8/TITLE / TITLE(13)   ,DATE(2)     ,ISW(7)     ,NBLOCK           COMMONU   28
           9/CONVER/ EPSI ,APSI  ,VTEST       ,NSTEPQ     ,NFCN       ,NFCNMX COMMONU 29
           A/CARD  / CWORD       ,CWORD2      ,CWORD3     ,WORD7(7)         COMMONU   30
           B/MINIMA/ AMIN        ,UP          ,NEWMIN     ,ITAUR      ,SIGMA COMMONU  31
           K/COMMND/ NNAME       ,CNAME(25)   ,CNAM2(25)  ,CNAM3(25)        COMMONU   32
            DIMENSION XBEST(15)                                             SIMPLX     7
            LOGICAL CONVER                                                  SIMPLX     8
            DATA ALPHA,BETA,GAMMA,RHOMIN,RHOMAX / 1.0, 0.5, 2.0, 4.0, 8.0/  SIMPLX     9
            IF (NPAR .LE. 0)  RETURN                                        SIMPLX    10
            NPFN=NFCN                                                       SIMPLX    11
            NPARP1=NPAR+1                                                   SIMPLX    12
            RHO1 = 1.0 + ALPHA                                              SIMPLX    13
            RHO2 = RHO1 + ALPHA*GAMMA                                       SIMPLX    14
            WG = 1.0/FLOAT(NPAR)                                            SIMPLX    15
            IFLAG=4                                                         SIMPLX    16
            WRITE(ISYSWR,100) EPSI                                          SIMPLX    17
            DO 2 I= 1, NPAR                                                 SIMPLX    18
            IF (ISW(2) .GE. 1)  DIRIN(I) = SQRT(V(I,I)*UP)                  SIMPLX    19
            IF (ABS(DIRIN(I)) .LT. 1.0E-10*ABS(X(I))) DIRIN(I)=1.0E-8*X(I)  SIMPLX    20
            IF(ITAUR.LT. 1)  V(I,I) =    DIRIN(I)**2/UP                     SIMPLX    21
          2 CONTINUE                                                        SIMPLX    22
            IF (ITAUR .LT. 1)  ISW(2) = 1                                   SIMPLX    23
      C**       CHOOSE THE INITIAL SIMPLEX USING SINGLE-PARAMETER SEARCHES  SIMPLX    24
          1 CONTINUE                                                        SIMPLX    25
            CONVER=.TRUE.                                                   SIMPLX    26
            YNPP1 = AMIN                                                    SIMPLX    27
            JL = NPARP1                                                     SIMPLX    28
            Y(NPARP1) = AMIN                                                SIMPLX    29
            ABSMIN = AMIN                                                   SIMPLX    30
            DO 10 I= 1, NPAR                                                SIMPLX    31
            AMING = AMIN                                                    SIMPLX    32
            PBAR(I) = X(I)                                                  SIMPLX    33
            BESTX = X(I)                                                    SIMPLX    34
            KG = 0                                                          SIMPLX    35
            NS = 0                                                          SIMPLX    36
            NF = 0                                                          SIMPLX    37
          4 X(I) = BESTX + DIRIN(I)                                         SIMPLX    38
            CALL INTOEX(X)                                                  SIMPLX    39
            CALL FCN(NPAR,G, F, U, 4)                                       SIMPLX    40
            NFCN = NFCN + 1                                                 SIMPLX    41
            IF (F .LE. AMING)  GO TO 6                                      SIMPLX    42
```

```
C        FAILURE
         IF (KG .EQ. 1)  GO TO 8
         KG = -1
         NF = NF + 1
         DIRIN(I) = DIRIN(I) * (-0.4)
         IF (NF .LT. 3)  GO TO 4
         NS = 6
C        SUCCESS
       6 BESTX = X(I)
         DIRIN(I) = DIRIN(I) * 3.0
         AMING = F
         KG = 1
         NS = NS + 1
         IF (NS .LT. 6)  GO TO 4
C        LOCAL MINIMUM FOUND IN ITH DIRECTION
       8 Y(I) = AMING
         IF (AMING .LT. ABSMIN)  JL = I
         IF (AMING .LT. ABSMIN)  ABSMIN = AMING
         X(I) = BESTX
         XBEST(I)=BESTX
         DO 9 K= 1, NPAR
       9 P(K,I) = X(K)
      10 CONTINUE
         JH = NPARP1
         AMIN=Y(JL)
         CALL RAZZIA(YNPP1,PBAR)
         DO 20 I= 1, NPAR
      20 X(I) = P(I,JL)
         CALL INTOEX(X)
         IF (ISW(5) .GE. 1)  CALL MPRINT(0,AMIN)
         SIGMA = SIGMA * 10.
         SIG2 = SIGMA
         IGNAL = 0
         NCYCL=0
C                                          . . . . .  START MAIN LOOP
      50 CONTINUE
         IF (IGNAL .GE. 10)  GO TO 1
         IF (SIG2 .LT. EPSI .AND. SIGMA.LT.EPSI)     GO TO 76
         SIG2 = SIGMA
         IF ((NFCN-NPFN) .GT. NFCNMX)  GO TO 78
C        CALCULATE NEW POINT * BY REFLECTION
         DO 60 I= 1, NPAR
         PB = 0.
         DO 59 J= 1, NPARP1
      59 PB = PB + WG * P(I,J)
         PBAR(I) = PB - WG * P(I,JH)
      60 PSTAR(I)=(1.+ALPHA)*PBAR(I)-ALPHA*P(I,JH)
         CALL INTOEX(PSTAR)
         CALL FCN(NPAR,G,YSTAR,U,4)
         NFCN=NFCN+1
         IF(YSTAR.GE.AMIN) GO TO 70
C        POINT * BETTER THAN JL, CALCULATE NEW POINT **
         DO 61 I=1,NPAR
      61 PSTST(I)=GAMMA*PSTAR(I)+(1.-GAMMA)*PBAR(I)
         CALL INTOEX(PSTST)
         CALL FCN(NPAR,G,YSTST,U,4)
         NFCN=NFCN+1
```

```
C          TRY A PARABOLA THROUGH PH, PSTAR, PSTST.  MIN = PRHO
      Y1 = (YSTAR-Y(JH)) * RHO2
      Y2 = (YSTST-Y(JH)) * RHO1
      RHO = 0.5 * (RHO2*Y1 -RHO1*Y2) / (Y1 -Y2)
      IF (RHO .LT. RHOMIN)  GO TO 66
      IF (RHO .GT. RHOMAX)  RHO = RHOMAX
      DO 64 I= 1, NPAR
   64 PRHO(I) = RHO*PSTAR(I) + (1. -RHO)*P(I,JH)
      CALL INTOEX(PRHO)
      CALL FCN(NPAR, G, YRHO, U, 4)
      NFCN = NFCN + 1
      IF (YRHO .LT. Y(JL) .AND. YRHO .LT. YSTST)  GO TO 65
      IF (YSTST .LT. Y(JL))  GO TO 67
      IF (YRHO .GT. Y(JL))  GO TO 66
C          ACCEPT MINIMUM POINT OF PARABOLA, PRHO
   65 CALL RAZZIA (YRHO,PRHO)
      IGNAL = MAX0(IGNAL-2, 0)
      GO TO 68
   66 IF (YSTST .LT. Y(JL))  GO TO 67
      IGNAL = MAX0(IGNAL-1, 0)
      CALL RAZZIA(YSTAR,PSTAR)
      GO TO 68
   67 IGNAL = MAX0(IGNAL-2, 0)
  675 CALL RAZZIA(YSTST,PSTST)
   68 NCYCL=NCYCL+1
      IF (ISW(5) .LT. 2)  GO TO 50
      IF (ISW(5) .GE. 3 .OR. MOD(NCYCL, 10) .EQ. 0) CALL MPRINT(0,AMIN)
      GO TO 50
C          POINT * IS NOT AS GOOD AS JL
   70 IF (YSTAR .GE. Y(JH))  GO TO 73
      JHOLD = JH
      CALL RAZZIA(YSTAR,PSTAR)
      IF (JHOLD .NE. JH)  GO TO 50
C          CALCULATE NEW POINT **
   73 DO 74 I=1,NPAR
   74 PSTST(I)=BETA*P(I,JH)+(1.-BETA)*PBAR(I)
      CALL INTOEX (PSTST)
      CALL FCN(NPAR,G,YSTST,U,4)
      NFCN=NFCN+1
      IF(YSTST.GT.Y(JH)) GO TO 1
C     POINT ** IS BETTER THAN JH
      IF (YSTST .LT. AMIN)  GO TO 675
      IGNAL = IGNAL + 1
      CALL RAZZIA(YSTST,PSTST)
      GO TO 50
C                                      . . . . . . .  END MAIN LOOP
   76 WRITE(ISYSWR,120)
      GO TO 80
   78 WRITE(ISYSWR,130)
      ISW(1) = 1
      CONVER=.FALSE.
   80 DO 82 I=1,NPAR
      PB = 0.
      DO 81 J=1,NPARP1
   81 PB = PB + WG * P(I,J)
   82 PBAR(I) = PB - WG * P(I,JH)
      CALL INTOEX(PBAR)
```

```
      CALL FCN(NPAR,G,YPBAR,U,IFLAG)
      NFCN=NFCN+1
      IF (YPBAR .LT. AMIN)  CALL RAZZIA(YPBAR,PBAR)
      CALL INTOEX(X)
      IF (NFCNMX+NPFN-NFCN .LT. 3*NPAR)  GO TO 90
      IF (SIGMA .GT. 2.0*EPSI)  GO TO 1
   90 IF(CONVER) GO TO 902
      CALL INTOEX(XBEST)
  902 CALL MPRINT(1-ITAUR, AMIN)
      RETURN
  100 FORMAT(       37H START SIMPLEX MINIMIZATION          8X    65HCON
     1VERGENCE CRITERION -- ESTIMATED DISTANCE TO MINIMUM (EDM) .LT.E10.
     22 )
  120 FORMAT(1H ,  34HSIMPLEX MINIMIZATION HAS CONVERGED)
  130 FORMAT(1H ,  38HSIMPLEX TERMINATES WITHOUT CONVERGENCE)
      END
```

```
      SUBROUTINE STAND
CC       OPTIONAL USER-SUPPLIED SUBROUTINE IS CALLED WHENEVER THE
CC       COMMAND "STANDARD" APPEARS.
CC
      RETURN
      END
```

```
      SUBROUTINE VERMIN(A,L,M,N,IFAIL)                                  VERMIN     2
CC        INVERTS A SYMMETRIC MATRIX.   MATRIX IS FIRST SCALED TO       VERMIN     3
CC        HAVE ALL ONES ON THE DIAGONAL (EQUIVALENT TO CHANGE OF UNITS) VERMIN     4
CC        BUT NO PIVOTING IS DONE SINCE MATRIX IS POSITIVE-DEFINITE.    VERMIN     5
CC                                                                      VERMIN     6
       COMMON                                                           COMMONU    2
     1/MINERR/ ERP(30)   ,ERN(30)                                       COMMONU    4
     2/PARINT/ X(15)     ,XT(15)    ,DIRIN(15) ,MAXINT    ,NPAR         COMMONU    5
     3/PAREXT/ U(30)     ,NAM(30)   ,WERR(30)  ,MAXEXT    ,NU           COMMONU    6
     4/LIMITS/ ALIM(30)  ,BLIM(30)  ,LCODE(30) ,LCORSP(30) ,LIMSET      COMMONU    7
     5/VARIAN/ V(15,15)                                                 COMMONU    8
     7/FIX   / IPFIX(15) ,XS(15)    ,XTS(15)   ,WTS(15)   ,NPFIX        COMMONU    9
     C/CASC/ JH, JL, Y(16)                                              COMMONU   10
     F/DERIVA/ G(30)     ,G2(30)                                        COMMONU   11
     G/SIMVEC/ P(15,16)  ,PSTAR(15) ,PSTST(15) ,PBAR(15)  ,PRHO(15)     COMMONU   12
     J/VARIAT/ VT(15,15)                                                COMMONU   13
     6/UNIT  / ISYSRD    ,ISYSWR    ,ISYSPU                             COMMONU   27
     8/TITLE / TITLE(13) ,DATE(2)   ,ISW(7)    ,NBLOCK                  COMMONU   28
     9/CONVER/ EPSI ,APSI ,VTEST    ,NSTEPQ    ,NFCN      ,NFCNMX       COMMONU   29
     A/CARD  / CWORD     ,CWORD2    ,CWORD3    ,WORD7(7)                COMMONU   30
     B/MINIMA/ AMIN      ,UP        ,NEWMIN    ,ITAUR     ,SIGMA        COMMONU   31
     K/COMMND/ NNAME     ,CNAME(25) ,CNAM2(25) ,CNAM3(25)               COMMONU   32
      DIMENSION A(L,M) , PP(15) , Q(15) , S(15)                         VERMIN     9
      IFAIL=0                                                           VERMIN    14
      IF (N .LT. 1)  GO TO 100                                          VERMIN    15
      IF (N .GT. MAXINT)  GO TO 100                                     VERMIN    16
C                   SCALE MATRIX BY SQRT OF DIAG ELEMENTS               VERMIN    17
      DO 8  I=1,N                                                       VERMIN    18
      SI = A(I,I)                                                       VERMIN    19
      IF (SI) 100,100,8                                                 VERMIN    20
    8 S(I) = 1.0/SQRT(SI)                                               VERMIN    21
      DO 20 I= 1, N                                                     VERMIN    22
      DO 20 J= 1, N                                                     VERMIN    23
   20 A(I,J) = A(I,J) *S(I)*S(J)                                        VERMIN    24
C                                     . . . START MAIN LOOP . . . .     VERMIN    25
      DO 65 I=1,N                                                       VERMIN    26
      K = I                                                             VERMIN    27
C                   PREPARATION FOR ELIMINATION STEP1                   VERMIN    28
      Q(K)=1./A(K,K)                                                    VERMIN    29
      PP(K) = 1.0                                                       VERMIN    30
      A(K,K)=0.0                                                        VERMIN    31
      KP1=K+1                                                           VERMIN    32
      KM1=K-1                                                           VERMIN    33
      IF(KM1)100,50,40                                                  VERMIN    34
   40 DO 49 J=1,KM1                                                     VERMIN    35
      PP(J)=A(J,K)                                                      VERMIN    36
      Q(J)=A(J,K)*Q(K)                                                  VERMIN    37
   49 A(J,K)=0.                                                         VERMIN    38
   50 IF(K-N)51,60,100                                                  VERMIN    39
   51 DO 59 J=KP1,N                                                     VERMIN    40
      PP(J)=A(K,J)                                                      VERMIN    41
      Q(J)=-A(K,J)*Q(K)                                                 VERMIN    42
   59 A(K,J)=0.0                                                        VERMIN    43
C                   ELIMINATION PROPER                                  VERMIN    44
   60 DO 65 J=1,N                                                       VERMIN    45
      DO 65 K=J,N                                                       VERMIN    46
   65 A(J,K)=A(J,K)+PP(J)*Q(K)                                          VERMIN    47
```

```
C                       ELEMENTS OF LEFT DIAGONAL AND UNSCALING
      DO 70 J= 1, N
      DO 70 K= 1, J
      A(K,J) = A(K,J) *S(K)*S(J)
   70 A(J,K) = A(K,J)
      RETURN
C                       FAILURE RETURN
  100 IFAIL=1
      RETURN
      END
```

```
      SUBROUTINE UCOPY(A,X,N)                                           UCOPY      2
      DIMENSION A(N),X(N)                                               UCOPY      3
C     COMMENT COPIES N WORDS FROM A INTO X, THE BEGINNING               UCOPY      4
C     OF A MAY OVERLAP THE END OF X, BUT NOT VICE-VERSA.                UCOPY      5
C     MUST HAVE N.GE.0 .                                                UCOPY      6
      IF(N.LE.0) RETURN                                                 UCOPY      7
      DO 100 I= 1,N,1                                                   UCOPY      8
        X(I)= A(I)                                                      UCOPY      9
  100 CONTINUE                                                          UCOPY     10
      RETURN                                                            UCOPY     11
      END                                                               UCOPY     12
```

```
      SUBROUTINE BINSIZ(A1,A2,NAA,BL,BH,NB,BWID)
C        SUBROUTINE TO DETERMINE REASONABLE HISTOGRAM INTERVALS
C        GIVEN ABSOLUTE UPPER AND LOWER BOUNDS A1 AND A2
C        AND DESIRED MAXIMUM NUMBER OF BINS NAA
C        PROGRAM MAKES REASONABLE BINNING FROM BL TO BH OF WIDTH BWID
C        F. JAMES, AUGUST, 1974
C        TYPED BY E. CLOSE 9/75 FROM LISTING FROM CERN THAT CAME
C        WITH THE CODE.
      AL= AMIN1(A1,A2)
      AH= AMAX1(A1,A2)
      IF(AL.EQ.AH) AH= AL+1.
C        IF(NAA.EQ.-1), PROGRAM USES BWID INPUT FROM CALLING ROUTIINE
      IF(NAA.EQ.-1) GOTO 150
   10 NA=NAA-1
      IF(NA.LT.1) NA= 1
C        GET NOMINAL BIN WIDTH IN EXPON FORM
   20 AWID=(AH-AL)/FLOAT(NA)
      LOG= ALOG10(AWID)
      IF(AWID.LE.1.0) LOG= LOG-1
      SIGFIG= AWID*(10.00**(-LOG))
C        ROUND MANTISSA UP TO 2,2.5,5,OR 10
      IF(SIGFIG .GT. 2.0) GOTO 40
      SIGRND= 2.0
      GOTO 100
   40 IF(SIGFIG.GT.2.5) GOTO 50
      SIGRND= 2.5
      GOTO 100
   50 IF(SIGFIG.GT.5.0) GOTO 60
      SIGRND= 5.0
      GOTO 100
   60 SIGRND=1.0
      LOG= LOG+1
  100 CONTINUE
      BWID= SIGRND*10.0**LOG
      GOTO 200
C        GET NEW BOUNDS FROM NEW WIDTH BWID
  150 IF(BWID.LE.0.) GOTO 10
  200 CONTINUE
      ALB= AL/BWID
      LWID= ALB
      IF(ALB.LT.0.0) LWID= LWID-1
      BL= BWID* FLOAT(LWID)
      ALB= AH/BWID+1.0
      KWID= ALB
      IF(ALB.LT.0.0) KWID= KWID-1
      BH= BWID*FLOAT(KWID)
      NB= KWID-LWID
      IF(NAA.GT.5) GOTO 240
      IF(NAA.EQ.-1) RETURN
C        REQUEST FOR ONE BIN IS DIFFICULT CASE
      IF(NAA.GT.1 .OR. NB .EQ. 1) RETURN
      BWID= BWID*2.0
      NB= 1
      RETURN
  240 IF(2*NB.NE.NAA) RETURN
      NA= NA+1
      GOTO 20
```

```
      END                                                             BINSIZ      59
```

```
B N L   SCOPE 2.1.5  590BB

SYS DEVICES 844/14/PF FLS=200K FLL=1747K MXS=160K MXL=1301K MXB=1301B

HH.MM.SS CPU SECOND ORIGIN
         16:47:41  CVX STATION SYNLSGU
         30"COURANT "::DUA1:[COURANT.SYN.PROG]SYNLIST.OUT
16.50.17 00000.003 MFZ.     -SYNLS,STMFZ,T77.
16.50.17 00000.003 JOB.     -ACCOUNT,COURANT,3,*.
16.50.22 00000.027 JOB.     -ATTACH,OLDPL,BSY86,ID=EDC.
          16.50.25 00000.003 MFZ.     -SJRSN,T400,P7000,SN.
          16.50.40 00000.109 LOD.     -DISPLAY,R1.                  NUMBER OF RETRIEVALS ATTEMPTED.
          16.50.42 00000.113 USR.           0    0B
          16.50.42 00000.113 LOD.     -PRELOD.                      SET UP TO RETRIEVE FILE
          16.50.44 00000.125 JOB.     -LOADPF,RET.                  RETRIEVE FILE
          16.50.45 00000.129 MFZ.        JM260 - STAGE NT IN  LFN=DUMTAPE VSN=0X9242
                    16.54.32 MFC.       16.48.33.VSN(DUMTAPE=X9242)
                    16.54.32 MFC.       16.49.33.REQUEST(DUMTAPE,NT,PE,US,NORING,U,E,S,VSN=0X9242)
                    16.54.32 MFC.       16.49.33.( NT 041 ASSIGNED)
                    16.54.32 MFC.       16.49.33.-SP- PRE STAGING    (PARTIAL)
                    16.54.32 MFC.       16.49.33.NT41 VOLUME SERIAL NUMBER IS  0X9242
                    16.54.32 MFC.       16.52.16.-SP- SKIPPED 3234, STAGED  167, BLOCK(S)
                    16.54.33 MFC.       16.52.16.NO. WORDS=     85478  NO. OF EOR =     1
                    16.54.33 STP.        JM511 - WORDS  READ-       247254B
          16.54.35 00000.133 USR.        PFLOD - BEGIN FILE DUMTAPE RECOVERY
          16.54.35 00000.156 USR.        PFLOD - ---LOAD BEGINS AGAIN ----------AT  BN =           0B--
          16.54.35 00000.157 USR.        PFLOD - LOADING BEGINS WITH FILE   112
          16.54.43 00000.312 USR.        PFLOD -       EDC CYCLE   1 BSY86
          16.54.43 00000.312 USR.        PFLOD -      1 PERMANENT FILES LOADED
          16.54.43 00000.313 USR.        PFLOD - LOADPF ERROR COUNT       0
          16.54.43 00000.313 USR.        PFLOD - END LOADPF
          16.54.43 00000.314 LOD.     -SET,R2=0.
          16.54.43 00000.318 JOB.     -EXIT(U)
16.54.52 00000.038 MFZ.        PF254 - CYCLE    1 ATTACHED FROM SN=SYSTEM
16.54.52 00000.038 LOD.     -UPDATE,Q,L=1.
16.55.00 00000.625 USR.               3 NON-FATAL ERRORS
16.55.00 00000.627 USR.       UPDATE COMPLETE.
16.55.01 00000.628 LOD.     -FTN,I,P,R=0.
16.58.04 00003.759 USR.          3.127 CP SECONDS COMPILATION TIME
16.58.04 00003.760 LOD.     -UPDATE,Q,L=1.
16.58.47 00008.508 USR.               3 NON-FATAL ERRORS
16.58.48 00008.510 USR.       UPDATE COMPLETE.
16.58.48 00008.511 LOD.     -FTN,I,P,R=0.
17.17.21 00041.459 USR.         32.945 CP SECONDS COMPILATION TIME
17.17.21 00041.461 MFZ.        JM166 - MAXIMUM USER SCM        47000B WORDS
17.17.21 00041.461 MFZ.        JM167 - MAXIMUM USER LCM        20000B WORDS
17.17.21 00041.461 MFZ.        JM170 - MAXIMUM JS+IO LCM         103B BUFFERS
17.17.21 00041.461 MFZ.        RM770 - MAXIMUM ACTIVE FILES                4
17.17.21 00041.461 MFZ.        RM771 - OPEN/CLOSE CALLS                   70
17.17.21 00041.462 MFZ.        RM772 - DATA TRANSFER CALLS           233,290
17.17.21 00041.462 MFZ.        RM773 - CONTROL/POSITIONING CALLS       3,433
17.17.21 00041.462 MFZ.        RM774 - BM DATA TRANSFER CALLS         27,315
17.17.21 00041.462 MFZ.        RM775 - BM CONTROL/POSITIONING CALLS    2,006
17.17.21 00041.462 MFZ.        RM776 - QUEUE MANAGER CALLS             5,145
17.17.21 00041.463 MFZ.        RM777 - RECALL CALLS                    4,620
17.17.21 00041.463 MFZ.        SCM                 827.165  KWS
17.17.21 00041.463 MFZ.        LCM                  19.912  KWS
17.17.21 00041.463 MFZ.        I/O                   1.619  MW
17.17.21 00041.463 MFZ.        RMS                  14.211  MWS
17.17.21 00041.464 MFZ.        USER                 19.526  SEC
```

```
17.17.21 00041.464 MFZ.      JOB                41.466  SEC
17.17.21 00041.464 MFZ.      DIO            14 412.857  KW
17.17.21 00041.464 MFZ.      RU                108.287
17.17.21 00041.464 MFZ.      BNL CCUS            5.831
17.17.21 00041.465 MFZ.      CCUS COMPUTED AT UPPER TIER RATE
17.17.21 00041.465 MFZ.      SC050 - 007376 SC/LC SWAPS
17.17.21 00041.465 MFZ.      EJ  END OF JOB  -  PN000003
```

```
      SUBROUTINE SSSR(M,P,KF,KG)                                        SSSR      2
C  SETS UP MATRICES FOR A LONG(PI) OR COLLINS(PI/2) STRAIGHT SECTIONS   SSSR      3
C   MATRICES GENERATED AS FOLLOWS                                       SSSR      4
C      QF = FOCUSSING MAGNET AT M-1.                                    SSSR      5
C      QD = DEFICUSSING MAGNET AT M-2.                                  SSSR      6
C      QS = END DRIFT SPACE (HALF CENTER DRIFT SPACE) AT M-3 (PI)       SSSR      7
C      QS = HALF CENTER DRIFT SPACE AT M-4 (COLLINS), AND THE STRAIGHT  SSSR      8
C           SECTION MAGNET Q AT M.                                      SSSR      9
C      QT = DRIFT SPACE BETWEEN F AND D MAGNETS AT M-4 (PI), AND THE    SSSR     10
C           STRAIGHT SECTION MAGNET Q AT M.                             SSSR     11
C      QE = END DRIFT SPACE AT M-3 (COLLINS)                            SSSR     12
C                                                                       SSSR     13
C   Q CAN TAKE EITHER OF TWO FORMS DEPENDING ON VALUE OF KF.            SSSR     14
C     IF KF=1, Q = PRODUCT OF  ...QS QD QT QF QS QS QD QT QF QS. (DFDF) SSSR     15
C     IF KF=2, Q = PRODUCT OF  ...QS QF QT QD QS QS QF QT QD QS. (FDFD) SSSR     16
C           FOR LONG (PI) STRAIGHT SECTION                              SSSR     17
C           FOR COLLINS STRAIGHT SECTION                                SSSR     18
C     IF KF=1, Q = PRODUCT OF  ...QE QD QS QS QF QE. (DF)               SSSR     19
C     IF KF=2, Q = PRODUCT OF  ...QE QF QS QS QD QE. (FD)               SSSR     20
      COMMON/BMI/MI(5000)                                               BMIL      1
      COMMON/NELS/NELS                                                  BMIL      2
      LEVEL 2,MI                                                        BMIL      3
                                                                        BMIL      4
                                                                        BMI       3
      EQUIVALENCE (E,T)                                                 SSSR     22
      LOGICAL COLL                                                      SSSR     23
      DIMENSION P(6)                                                    SSSR     24
C                                                                       SSSR     25
      COLL = .FALSE.                                                    SSSR     26
      IF (KG.EQ.1)  COLL = .TRUE.                                       SSSR     27
C  SET UP FOR PI SECTION FIRST                                          SSSR     28
      S = P(4)/2.                                                       SSSR     29
      T = P(5)                                                          SSSR     30
      P(2) = ABS(P(2))                                                  SSSR     31
      P(4) = 0.                                                         SSSR     32
      P(5) = 0.                                                         SSSR     33
      P(6) = 0.                                                         SSSR     34
C  SET UP QF                                                            SSSR     35
      L = M - 1                                                         SSSR     36
      CALL STDAT(L,1,1,6,P)                                             SSSR     37
      CALL MAGNET(L,P)                                                  SSSR     38
      MI(4) = L                                                         SSSR     39
C  SET UP QD                                                            SSSR     40
      L = M - 2                                                         SSSR     41
      CALL STDAT(L,1,1,6,P)                                             SSSR     42
      CALL EXCHM(L+1,L)                                                 SSSR     43
      MI(2) = L                                                         SSSR     44
C  SET UP QS                                                            SSSR     45
      L = M - 3                                                         SSSR     46
      CALL DRIFT (L,S)                                                  SSSR     47
      MI(1) = L                                                         SSSR     48
      MI(5) = L                                                         SSSR     49
C  SET UP QT OR QE  (T IS REALLY E IF COLLINS)                          SSSR     50
      L = M - 4                                                         SSSR     51
      CALL DRIFT (L,T)                                                  SSSR     52
      MI(3) = L                                                         SSSR     53
      IF (KF.EQ.1)  GO TO 2                                             SSSR     54
```

```
C   AN FD OR FDFD
      MI(2) = M - 1
      MI(4) = M - 2
 2    IF (COLL)  GO TO 4
C  COMPLETE MI LIST
      DO 3 I=1,5
 3    MI(I+5) = MI(I)
      IM = 10
      GO TO 5
C   MAKE CHANGES FOR COLLINS STRAIGHT SECTION
 4    ISAV = MI(1)
      MI(1) = MI(3)
      MI(3) = ISAV
      ISAV = MI(4)
      MI(4) = MI(5)
      MI(5) = ISAV
      MI(6) = MI(1)
      IM = 6
 5    L = LM
      CALL MMM(L,IM,MI)
C  SAVE THE MILIST AS DATA FOR Q
C  PI SECTION HAS ONE PIECE OF INTEGER DATA, SO MILIST BEGINS AT 2.
      DO 6 I=1,IM
 6    CALL REPINT(M,I,MI(I))
      RETURN
      END
```

```
              SUBROUTINE STAR(M,KA)                                    STAR       2
      C THE MATRIX M = MATRIX N RAISED TO POWER K (M=N**K)             STAR       3
                                                                       STAR       4
              COMMON/BMI/MI(5000)                                      BMIL       1
              COMMON/NELS/NELS                                         BMIL       2
              LEVEL 2,MI                                               BMIL       3
                                                                       BMIL       4
                                                                       BMI        3
              N=MDAT(M,1)                                              STAR       6
              DO 20 I=1,KA                                             STAR       7
              MI(I)=N                                                  STAR       8
         20   CONTINUE                                                 STAR       9
              CALL MMM(M,KA,MI)                                        STAR      10
              RETURN                                                   STAR      11
              END                                                      STAR      12
```

```
      SUBROUTINE STDAT(M,I,J,K,A)                                   STDAT      2
C  REPLACES K DATA OF TYPE I FROM ARRAY A INTO INFF(J,M)            STDAT      3
*   IF I=1, FL. PT.    IF I=2, BCD.   IF I=3, INTEGER.              STDAT      4
C  IF I =4,SKIP. IF I=5, STORE AT LQ                                STDAT      5
C  IF I=7,STORE AT LQ2.  IF I=8, STORE AT LQ3                       STDAT      6
                                                                    BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                     BLANK      3
      COMMON STORE(48000),IWORK(10)                                 86MARSIZ   1
      DIMENSION INFF(24,2000)                                       86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                      86MARSIZ   3
                                                                    BLANK      5
      DIMENSION A(1),LOC(10)                                        STDAT      8
      IF (I.EQ.4) RETURN                                            STDAT      9
      LOC(1) = INFF(14,M)                                           STDAT     10
      LOC(2)=INFF(16,M)                                             STDAT     11
      LOC(3)=INFF(12,M)                                             STDAT     12
      LOC(5) = INFF(10,M)                                           STDAT     13
      LOC(7) = INFF(19,M)                                           STDAT     14
      LOC(8)=INFF(24,M)                                             STDAT     15
      LOCI = LOC(I)+J-2                                             STDAT     16
      DO 10 L=1,K                                                   STDAT     17
      LOCI = LOCI+1                                                 STDAT     18
   10 STORE(LOCI) = A(L)                                            STDAT     19
      RETURN                                                        STDAT     20
      END                                                           STDAT     21
```

```
            SUBROUTINE STINFO( M0,NAME0,OPNAM0,KA0,KB0,NAME1 )          STINFO     2
      C  STORES NAME, OPNAME, OP, KA, KB  IN INFO(M0)                   STINFO     3
      C  EXAMPLE---- CALL STINFO( MH,5HROVER,5HMMM  ,6,0,5HFIDO )       STINFO     4
      C          ----  NAME IS ROVER.                                   STINFO     5
      C               CALL STINFO(-MH,4HOX  ,5HMMM  ,6,0,5HFIDO  )      STINFO     6
      C          ---  NAME IS FOX.                                      STINFO     7
      C  IF M0 IS +, NAME IS   NAME0.  IF M0 IS -, NAME IS NAME0 PRECEDED BY STINFO     8
      C  THE FIRST LETTER OF NAME1.                                     STINFO     9
                                                                        BLANK      2
            LEVEL 2, STORE,INFF,IWORK                                   BLANK      3
            COMMON STORE(48000),IWORK(10)                               86MARSIZ   1
            DIMENSION INFF(24,2000)                                     86MARSIZ   2
            EQUIVALENCE (INFF,STORE)                                    86MARSIZ   3
                                                                        BLANK      5
            INTEGER OPNUM,OPNAM0                                        STINFO    11
            IF (M0) 2,1,1                                               STINFO    12
          1 INFF(2,M0) = NAME0                                          STINFO    13
            GO TO 3                                                     STINFO    14
          2 M0= -M0                                                     STINFO    15
            ENCODE(10,102,INFF(2,M0)) NAME1,NAME0                       STINFO    16
        102 FORMAT (A1,A4,5X)                                           STINFO    17
          3 INFF(1,M0) = OPNAM0                                         STINFO    18
            INFF(3,M0) = OPNUM(OPNAM0)                                  STINFO    19
            INFF(4,M0) = KA0                                            STINFO    20
            INFF(5,M0) = KB0                                            STINFO    21
            RETURN                                                      STINFO    22
            END                                                         STINFO    23
```

```
      SUBROUTINE STOR7(M,T,RW)                                          STOR7      2
C  STORES 7X7 MATRICES                                                  STOR7      3
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1          LDFLG,FIN                                               CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL   DIM        2
                                                                        DIM        3
      DIMENSION T(49),TINV(49),RX(2,3),RY(2,3),RW(3),RPQ(4)             STOR7      7
                                                                        STOR7      8
      IF(M)4,10,10                                                      STOR7      9
    4 M = - M                                                           STOR7     10
      CALL TINV7(TINV,T)                                                STOR7     11
      RW(1) = -RW(1)                                                    STOR7     12
      RW(2) = -RW(2)                                                    STOR7     13
      RW(3) = TINV(40)                                                  STOR7     14
      DO 5 I=1,49                                                       STOR7     15
      T(I) = TINV(I)                                                    STOR7     16
    5 CONTINUE                                                          STOR7     17
   10 LQ = INFF(10,M)                                                   STOR7     18
      NQ = INFF(11,M)                                                   STOR7     19
      KIND = INFF(20,M)                                                 STOR7     20
      IF (KIND.EQ.3HMXY) GO TO 11                                       STOR7     21
C  CHECK STORAGE                                                        STOR7     22
      NQQ = 49 + KADD                                                   STOR7     23
      IF (NQQ.GT.NQ) GO TO 45                                           STOR7     24
      CALL STDAT(M,5,1,3,RW)                                            STOR7     25
      NJ = LQ + KADD - 1                                                STOR7     26
      DO 20 I=1,49                                                      STOR7     27
      STORE(NJ + I) = T(I)                                              STOR7     28
   20 CONTINUE                                                          STOR7     29
      RETURN                                                            STOR7     30
   45 MN = INFF(2,M)                                                    STOR7     31
      WRITE (3,1) MN                                                    STOR7     32
      ERROR = .TRUE.                                                    STOR7     33
    1 FORMAT (61H *** ERROR *** STORAGE   REQUEST EXCEEDS STORAGE RESERV STOR7     34
     1E FOR A6)                                                         STOR7     35
      RETURN                                                            STOR7     36
   11 CALL CONVMAT(3,T,RX,RY,RW,RPQ)                                    STOR7     37
      CALL STXY(M,RX,RY,RW)                                             STOR7     38
      RETURN                                                            STOR7     39
      END                                                               STOR7     40
```

```
      SUBROUTINE STRFCN(X,BKDIF)                                        STRFCN   2
C  FUNCTION EVALUATION FOR STR4 INSTRUCTION                             STRFCN   3
C                                                                       WRK      2
      COMMON/WRK/BZ,BQ,AZ,DPSI,SM,SL,PSI,WW,WT,BETW,P,Q,                WRK      3
     1      T,U,BKZ,BKZP,CZ,SZ,CHZB,SHZB,PHI,BK,SA,BL,BM,               WRK      4
     2      AQ,RHO,BETS,PSIWS,BKZI,BKZPI,G(6),AB,C,S,CHB,               WRK      5
     3      SHB,E,EI,SZCB,SZBC,CZBC,CZCB,SZSB,SZBS,CZBS,CZSB,           WRK      6
     4      ZW,ZT,BKI,Y,Z,Y1,Y2,Y3,Y4                                   WRK      7
C                                                                       WRK      8
      EQUIVALENCE (BKS,BZ)                                              STRFCN   5
      PHI = X                                                           STRFCN   6
      C = COS(PHI)                                                      STRFCN   7
      S = SIN(PHI)                                                      STRFCN   8
      E = EXP(PHI)                                                      STRFCN   9
      EI = 1.0/E                                                        STRFCN  10
      SHB = 0.5*(E-EI)                                                  STRFCN  11
      CHB = 0.5*(E+EI)                                                  STRFCN  12
C                                                                       STRFCN  13
      SZCB = SZ*CHB                                                     STRFCN  14
      SZBC = SHZB*C                                                     STRFCN  15
      CZBC = CHZB*C                                                     STRFCN  16
      CZCB = CZ*CHB                                                     STRFCN  17
      SZSB = SZ*SHB                                                     STRFCN  18
      SZBS = SHZB*S                                                     STRFCN  19
      CZBS = CHZB*S                                                     STRFCN  20
      CZSB = CZ*SHB                                                     STRFCN  21
C                                                                       STRFCN  22
      ZW = BKZ*SZSB + BKZP*SZBS*P - T*(CZBS + BKZP*SL*SZBS)             STRFCN  23
      ZT = -(CZSB + U*CZBS) + SL*(BKZ*SZSB - BKZP*U*SZBS)               STRFCN  24
     1      + Q*BKZP*SZBS                                               STRFCN  25
      Y1 = BKZ*SZCB - BKZP*P*SZBC + T*(CZBC + BKZP*SL*SZBC)             STRFCN  26
      Y2 = CZCB - U*CZBC - SL*(BKZ*SZCB + BKZP*U*SZBC) + Q*BKZP*SZBC    STRFCN  27
      Y3 = CZSB - P*CZBS + T*( BKZPI*SZBS + SL*CZBS)                    STRFCN  28
      Y4 = -BKZI*SZSB - BKZPI*U*SZBS - SL*(CZSB + U*CZBS) + Q*CZBS      STRFCN  29
      BK = (Y1*ZT + Y2*ZW) /(Y3*ZT + Y4*ZW)                             STRFCN  30
      BKI = 1.0/BK                                                      STRFCN  31
      SA = (CZSB - P*CZBS - BKI*(BKZ*SZCB - BKZP*P*SZBC) -              STRFCN  32
     1       T*(BKI*CZBC - BKZPI*SZBS - SL*CZBS + BKZP*BKI*SL*SZBC))    STRFCN  33
     2        /ZW                                                       STRFCN  34
      BL = (CZCB - P*CZBC -BKI*(BKZ*SZSB + BKZP*P*SZBS) -               STRFCN  35
     1      SA*(BKZ*SZCB + BKZP*P*SZBC) + T*(SZBC*BKZPI + BKI*CZBS +    STRFCN  36
     2      (SL + SA)*CZBC + BKZP*BKI*SL*SZBS + BKZP*SL*SA*SZBC ))/     STRFCN  37
     3      (BK*(BKI*(BKZ*SZCB + BKZP*P*SZBC) - (CZSB + P*CZBS)         STRFCN  38
     4      + SA*(BKZ*SZSB - BKZP*P*SZBS) + T*(-BKI*CZBC +              STRFCN  39
     5      BKZPI*SZBS - BKI*BKZP*SL*SZBC + (SA + SL)*CZBS +            STRFCN  40
     6      BKZP*SA*SL*SZBS )))                                         STRFCN  41
      BM = PHI/BK                                                       STRFCN  42
      BKDIF = BK - BKS                                                  STRFCN  43
      RETURN                                                            STRFCN  44
      END                                                               STRFCN  45
```

```
      SUBROUTINE STRN(M,NCELL,PAR)                                      STRN       2
C  DESIGN MATCHING STRAIGHT SECTION                                     STRN       3
C  TWO OPTIONS    1 CALCULATE BETAS AND ALPHAS AT POS                   STRN       4
C                 2 READ IN BETAS AND ALPHAS                            STRN       5
C  DATA INPUT FORMAT                                                    STRN       6
C               NAME    STRN OPT     POS    BLBX                        STRN       7
C                                    MU         TRIAL LEN   GRAD   RHO  STRN       8
C     (IF OPTION 2)                  BETAX     BETAY        ALPHAX ALPHAY STRN     9
C   K=KE.KIA.KA.KS.KS.KB.KIB.KE                                         STRN      10
C                                                                       STRN      11
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/BMI/MI(5000)                                               BMIL       1
      COMMON/NELS/NELS                                                  BMIL       2
      LEVEL 2,MI                                                        BMIL       3
                                                                        BMIL       4
                                                                        BMI        3
C                                                                       STRN      14
      DIMENSION RW(3)                                                   STRN      15
      DIMENSION NCELL(3),PAR(8),BX(9),BY(9),RX(2,3),RY(2,3),PX(2,2),    STRN      16
     1          PY(2,2),UX(2,3),UY(2,3)                                 STRN      17
      INTEGER C,BLBX,POS,BDAT                                           STRN      18
      REAL LZ,LTOT                                                      STRN      19
      DATA EPSK,ITMX/.0000001,25/                                       STRN      20
      DATA IFO,IDFO,IBL/9HFOCUSSING,2HDE,2H  /                          STRN      21
C                                                                       STRN      22
      KPN = INFF(4,M)                                                   STRN      23
      POS = BDAT(M,1)                                                   STRN      24
      C = MDAT(M,1)                                                     STRN      25
      BLBX = MDAT(M,2)                                                  STRN      26
      XYM = PAR(1)                                                      STRN      27
      SVM = PAR(2)                                                      STRN      28
      OKK = PAR(3)                                                      STRN      29
      RHO = PAR(4)                                                      STRN      30
      OK = SQRT(OKK/RHO)                                                STRN      31
      GO TO (101,103),KPN                                               STRN      32
  101 CALL BET(C,BX,BY,0)                                               STRN      33
      GO TO 110                                                         STRN      34
C  GET BETAS AND ALPHAS FROM INPUT                                      STRN      35
  103 BX(2) = PAR(5)                                                    STRN      36
      BY(2) = PAR(6)                                                    STRN      37
      BX(3) = PAR(7)                                                    STRN      38
      BY(3) = PAR(8)                                                    STRN      39
      BX(4) = (1.+BX(3)*BX(3))/BX(2)                                    STRN      40
      BY(4) = BX(4)                                                     STRN      41
  110 CALL HED                                                          STRN      42
      WRITE (3,501) POS                                                 STRN      43
      WRITE (3,502) XYM,OKK,RHO,BX(2),BY(2),BX(3),BY(3)                 STRN      44
      XYM = XYM * 6.283185307                                           STRN      45
C   XYM IS NOW PSI OF WRITE UP                                          STRN      46
      BX(1) = XYM                                                       STRN      47
      BY(1) = XYM                                                       STRN      48
      CSI = COS(XYM)                                                    STRN      49
```

```
      SSI = SIN(XYM)
C  COMPUT PX AND PY (P AND Q).  PY IS Q OF WRITE UP
      RBX = SQRT(BX(2))
      RBY = SQRT(BY(2))
      PX(1,1) = (RBY*(CSI+BX(3)*SSI))/RBX
      PX(1,2) = RBX*RBY*SSI
      PX(2,1) = ((BX(3)*BY(3)-1.)*SSI+(BX(3)+BY(3))*CSI)/(RBX*RBY)
      PX(2,2) = (RBX*(CSI+BY(3)*SSI))/RBY
      PY(1,1) = PX(2,2)
      PY(2,2) = PX(1,1)
      PY(1,2) = PX(1,2)
      PY(2,1) = PX(2,1)
C  GET KIA (UX,UY)  (U AND V OF WRITE UP).  STORE KIA IN M-1
      CALL RXY(BLBX,RX,RY,RW)
      L = M -  1
C                        STORE OMEGA = 0 IN STORAGE OF L
      OMEG = 0.
      CALL REPFLT(L,4,OMEG)
      CALL STXY(L,RX,RY,RW)
      LZ = RW(1)
      DO 160 J=1,2
      DO 158 I=1,2
      UX(I,J) = RX(I,J)
      UY(I,J) = RY(I,J)
  158 CONTINUE
  160 CONTINUE
C  COMPUT KIB = WI = WX,WY(OLD,WIX,WIY)
      L = M - 2
      CALL REF(L,BLBX)
      CALL RXY(L,RX,RY,RW)
      DO 164 I=1,2
      DO 162 J=1,2
      RXS = RX(I,J)
      RX(I,J) = RY(I,J)
  162 RY(I,J) = RXS
  164 CONTINUE
      CALL REPFLT(L,4,OMEG)
      CALL STXY(L,RX,RY,RW)
C  SET UP ITERATION FOR LENGTH
      NIT = 1
      OM = SVM
      DCS = 0.
      DELM = SVM/5.
      OMM = 0.
      SOK = 1.
  185 PHIO = OK=OM
      CS = COS(PHIO)
      SN = SIN(PHIO)
      EP = EXP(PHIO)
      EM = 1./EP
      CB = (EP+EM)/2.
      SB = CB-EM
C  COMPUTE F AND G
C  TRY D MAGNET FOR KA
      SOK = 1.
  196 F1 = SB*UX(1,1)+PX(2,2)*SN*UY(1,1)
      F2 = -PX(2,1)*SN*UY(1,1)
```

```
      F3 = CB*UX(2,1)-PX(2,2)*CS*UY(2,1)
      F4 = PX(2,1)*CS*UY(2,1)
      G1 = SB*UX(1,1)-PX(1,1)*SN*UY(1,1)+PX(2,1)*SN*UY(1,2)
      G2 = SB*UX(1,2)+PX(2,2)*SN*UY(1,2)-PX(1,2)*SN*UY(1,1)
      G3 = CB*UX(2,1)+PX(1,1)*CS*UY(2,1)-PX(2,1)*CS*UY(2,2)
      G4 = CB*UX(2,2)-PX(2,2)*CS*UY(2,2)+PX(1,2)*CS*UY(2,1)
C  COMPUT A,B,C DISCRIMINANT,DEN
 207  AFG = F1*G1+F2*G2
      BFG = (F1*G3+F3*G1+F2*G4+F4*G2)*SOK
      CFG = F3*G3+F4*G4
      DCS = BFG*BFG-4.*AFG*CFG
      IF (SOK.LT.0..OR.DCS.GE.0.)  GO TO 2085
C  OTHERWISE TRY F MAGNET FOR KA
 2077 SOK = -1.
      SS = SN
      SN = SB
      SB = SS
      SS = CS
      CS = CB
      CB = SS
      GO TO 196
 2085 DEN = 1.*AFG
      IF (DCS.LT.0.)  GO TO 2086
      DIS = SQRT(DCS)
C  COMPUTE K(MO)  CALLED TK
      TK = 0.
      TKP = (-BFG+DIS)/DEN
      IF (TKP.GE.0.)  GO TO 2185
      IF (SOK.LT.0.) GO TO 220
      GO TO 2077
 2086 OMM = OMM + DELM
      OM = OMM
      NIT = NIT + 1
      GO TO 185
 2185 TK = TKP
      TKM = (-BFG-DIS)/DEN
      IF (TKM.LE.0.) GO TO 220
      DKP = ABS(OK-TKP)
      DKM = ABS(OK-TKM)
      IF (DKM.LT.DKP)  TK=TKM
C  COMPARE TK AND OK,NIT AND ITMX
 220  DK = TK-OK
      AD = ABS(DK)
      IF (AD.LE.EPSK.OR.NIT.GE.ITMX)  GO TO 313
C  COMPUTE AND ADJUST NEW OM
      CM = TK*OM/OK
      TM = .5*OM
      IF (CM.LT.TM) CM=TM
      TM = 2.*OM
      IF (CM.GT.TM) CM=TM
      IF (NIT.EQ.1)  GO TO 311
      IF (AD.LT.SD)  GO TO 311
      CM = (CM+OM)/2.
 311  OM = CM
      SD = AD
      NIT = NIT + 1
      GO TO 185
```

```
C  COMPUTE MAGNET KB,KA.  STORE IN M-3,M-4
 313  BX(1) = OM
 3131 BX(2) = SOK*OKK
      BX(3) = RHO
      BX(4) = 0.
      BX(5) = 0.
      BX(6) = 0.
      L = M - 3
      CALL STDAT(L,1,1,6,BX)
      CALL MAGNET(L,BX)
      BX(2) = -SOK*OKK
      L = M - 4
      CALL STDAT(L,1,1,6,BX)
      CALL MAGNET(L,BX)
      MI(1) = M - 1
      MI(2) = M - 4
      L = M - 5
      CALL REPFLT(L,4,OMEG)
      CALL MMM(L,2,MI)
C   UX AND UY ARE NOW TO BE X,Y PARTITIONS OF WD(OLD UIJ,VIJ)
      CALL RXY(L,UX,UY,RW)
      L = M - 6
      MI(1) = M - 2
      MI(2) = M - 3
      CALL REPFLT(L,4,OMEG)
      CALL MMM(L,2,MI)
      ELE = (UX(2,1)+PX(1,1)*UY(2,1)-PX(2,1)*UY(2,2))/(PX(2,1)*UY(2,1))
      DLD = -UX(2,1)+PX(1,1)*UY(2,1)-PX(2,1)*(UY(2,2)+ELE*UY(2,1))
      ELD = UX(1,1)-PX(1,1)*UY(1,1)+PX(2,1)*(UY(1,2)+ELE*UY(1,1))
      ELD = ELD/DLD
      DLD = 2. * ELD
      IPR = IBL
      IF (SOK.GT.0.) IPR = IDFO
      WRITE(3,504)IPR,IFO
 3317 L = M - 7
      CALL DRIFT(L,ELE)
      L = L - 8
      CALL DRIFT(L,ELD)
      MI(1) = M - 7
      MI(2) = M -1
      MI(3) = M - 4
      MI(4) = M - 8
      MI(5) = M - 8
      MI(6) = M - 3
      MI(7) = M - 2
      MI(8) = M - 7
      CALL MMM(M,8,MI)
      DLH = DLD/2.
      CALL RXY(M,RX,RY,RW)
      LTOT = RW(1)
      WRITE (3,505) ELE,LZ,OM,DLH,DLD,LTOT
      WRITE (3,503) NIT
      CALL HED
      RETURN
 501  FORMAT (/* MATCHING STRAIGHT SECTION (INSERTION OR HALF CELL*
     1  * REPLACEMENT) AT POSITION  *,A5)
 502  FORMAT(2X,*MU*,6X,*= *,F13.8,5X,*GRAD*,4X,*= *,F13.8,5X,*RHO*,
```

```
      1   5X,*= *,F13.8/2X,*BETAX   = *,F13.8,5X,*BETAY   = *,F13.8,     STRN   221
      1   5X,*ALPHAX  = *,F13.8,5X,*ALPHAY  = *,F13.8/)                  STRN   222
  503 FORMAT (3X,I5,* ITERATIONS TO COMPUTE QUADRUPOLE LENGTH*)          STRN   223
  504 FORMAT (2X,*FIRST MAGNET KA IS *,A2,A10)                           STRN   224
  505 FORMAT (3X,*LEND  = *,F13.8,6X,*LO    = *,F13.8,6X,*LQUAD = *,     STRN   225
      1   F13.8/3X,*LDRF/2= *,F13.8,6X,*LDRF  = *,F13.8,6X,*LTOT  = *,   STRN   226
      2   F13.8)                                                         STRN   227
      END                                                                STRN   228
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
40	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

```
      SUBROUTINE STRPI(M)                                               STRPI      2
C  DESIGN A LONG STRAIGHT SECTION TO MATCH A SPECIFIED CELL.            STRPI      3
C  INPUT CARDS...                                                       STRPI      4
C  A(N)    STRPI    K    A(M)                                           STRPI      5
C                        B0        BQ       AVAC      LTO      LSEP     STRPI      6
C  N IS THE NUMBER OF THE LONG STRAIGHT SECTION.                        STRPI      7
C  M IS THE NUMBER OF THE CELL.                                         STRPI      8
C  K IS THE STRAIGHT-SECTION TYPE INDICATOR.(K=1 FOR DFDF,K=2 FOR FDFD) STRPI      9
C  J IS THE POINT OF INSERTION INTO THE CELL.                           STRPI     10
C  THIS ROUTINE ASSUMES THAT THE CELL MATRICES ARE STORED AS ...CELL,   STRPI     11
C  CELL1,CELL2,...,CELL12, AND THAT BETAMAX OCCURS AT CELL POSITION S.  STRPI     12
C  IF M IS A SYMMETRIC CELL (CELL, ALTC), ITS RHO IS PARAMETER 3.       STRPI     13
C  IF M IS ASYMMETRIC (CFD,DCFD), IS RHO IS PARAMETER 4.                STRPI     14
C                                                                       STRPI     15
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/GRR/IERR,BSW,VSW,RSW,QSW,XSW                               GRR        2
      LOGICAL IERR,BSW,VSW,RSW,QSW,XSW                                  GRR        3
                                                                        GRR        4
      COMMON/AGS/ALPHA,BETA,RHO,BZ,AVAC,BMAX                            AGS        2
C                                                                       AGS        3
      COMMON/STR/GAM,A,LK,BK,F,LM,BL,LL,BQ,LTO,SMUD,CMUD                STR        2
      REAL LK,LM,LL,LTO                                                 STR        3
C                                                                       STR        4
      EXTERNAL LPAR                                                     STRPI     20
      REAL LDRF                                                         STRPI     21
      INTEGER C,CI,P,GAM,BDAT,ELNUM,HCFD                                STRPI     22
      DIMENSION BX(9),BY(9),PAR(5)                                      STRPI     23
      EQUIVALENCE (PAR,BX)                                              STRPI     24
      DATA HCFD/3HCFD/                                                  STRPI     25
C                                                                       STRPI     26
C  NAME OF CELL FROM INPUT AND ITS INDEX                                STRPI     27
      C = BDAT(M,1)                                                     STRPI     28
      CI = ELNUM(C)                                                     STRPI     29
      P = INFF(5,M)                                                     STRPI     30
      L = CI - P                                                        STRPI     31
      KI = INFF(4,M)                                                    STRPI     32
C  RETRIEVE INPUT                                                       STRPI     33
      CALL DATA(M,1,1,5,PAR)                                            STRPI     34
C  SET BZERO,BQ,AVAC,LTO,LSEP                                           STRPI     35
      B0 = PAR(1)                                                       STRPI     36
      BQ = PAR(2)                                                       STRPI     37
      AVAC = PAR(3)                                                     STRPI     38
      LTO = PAR(4)                                                      STRPI     39
      LSEP = PAR(5)                                                     STRPI     40
      CALL BET(L,BX,BY,0)                                               STRPI     41
      ALPHA = BX(3)                                                     STRPI     42
      BETA = BX(2)                                                      STRPI     43
C  COMPUTE BETMAX FOR THE CELL                                          STRPI     44
      L = CI - 2                                                        STRPI     45
      CALL BET(L,BX,BY,0)                                               STRPI     46
      BETMAX = BX(2)                                                    STRPI     47
C  GET THE VALUE OF RHO FOR THE CELL                                    STRPI     48
```

```
      IOP = INFF(1,CI)
      IF (IOP.EQ.HCFD)  GO TO 51
C  CELL, ALTC,DCFD
      I = 3
      GO TO 52
C  CFD
 51   I = 4
 52   RHO = FDAT(CI,I)
      BL = LTO/2.
      F = SQRT(LSEP*BL)
      S = LSEP + BL
      A = AVAC
      AS = A*SQRT((BETA/BETMAX)*((1.+ALPHA*S/BETA)**2+(S/BETA)**2))
      LK = BQ/(AS*B0)
      BK = SQRT(LK/RHO)
      LM = 1./(F*BK**2)
      THETA = BK*LM
      GAM = P
      CALL GRT(1,THETA,1,LPAR)
      IF (IERR)  RETURN
      LDRF = 2.*BL
      GRAD = BQ/AS
      CALL HED
      WRITE(3,3) C,ALPHA,BETA,BETMAX,RHO,B0,BQ,AVAC,AS,LM,LK,LDRF,LSEP,
     1           LTO,BL,GRAD
      CALL HED
      PAR(1) = LM
      PAR(2) = LK
      PAR(3) = RHO
      PAR(4) = LDRF
      PAR(5) = LSEP
      CALL SSSR(M,PAR,KI,0)
      RETURN
  3   FORMAT(* LONG STRAIGHT SECTION IN CELL *,A1,* .*/1H0,5X,*ALPHA=*,
     1      F12.8,*,*,7X,*BETA=*,F12.8,*,*,4X,*BETMAX=*,F12.8,*,*,7X,
     2      *RHO=*,F12.8/9X,*B0=*,F12.8,*,*,9X,*BQ=*,F12.8,*,*,6X,
     3      *AVAC=*,F12.8,*,*,8X,*AS=*,F12.8/7X,*LMAG=*,F12.8,*,*,10X,
     4      *K=*,F12.8,*,*,6X,*LDRF=*,F12.8,*,*,6X,*LSEP=*,F12.8,/
     5      7X,*LTOT*,F12.8,*,*,7X,*LEND=*,F12.8,*,*,6X,*GRAD=*,F12.8,
     6      2X,*KG/M*)
      END
```

```
      SUBROUTINE STR2(M)                                              STR2         2
C  DESIGN A COLLINS LONG STRAIGHT SECTION TO MATCH A SPECIFIED CELL.  STR2         3
C THE MU VALUE OF THE STRAIGHT SECTION MAY BE SPECIFIED BY THE USER.  STR2         4
C INPUT CARDS...                                                      STR2         5
C A(M)      STR2    K      A(N)     BZ      BQ     AVAC     MU/2PI    STR2         6
C M IS THE NUMBER OF THE FIRST CONSTITUENT MATRIX OF STRT. SECTION    STR2         7
C N IS THE NUMBER OF THE CELL                                         STR2         8
C THIS ROUTINE ASSUMES THAT THE CELL MATRICES ARE STORED AS ... CELL, STR2         9
C CELL1,CELL2,...,CELL12,AND THAT BETAMAX OCCURS AT CELL POSITION 2.  STR2        10
C                                                                     STR2        11
                                                                      BLANK        2
      LEVEL 2, STORE,INFF,IWORK                                       BLANK        3
      COMMON STORE(48000),IWORK(10)                                   86MARSIZ     1
      DIMENSION INFF(24,2000)                                         86MARSIZ     2
      EQUIVALENCE (INFF,STORE)                                        86MARSIZ     3
                                                                      BLANK        5
      COMMON/BMI/MI(5000)                                             BMIL         1
      COMMON/NELS/NELS                                                BMIL         2
      LEVEL 2,MI                                                      BMIL         3
                                                                      BMIL         4
                                                                      BMI          3
      COMMON/GRR/IERR,BSW,VSW,RSW,QSW,XSW                             GRR          2
      LOGICAL IERR,BSW,VSW,RSW,QSW,XSW                                GRR          3
                                                                      GRR          4
      COMMON/AGS/ALPHA,BETA,RHO,BZ,AVAC,BMAX                          AGS          2
C                                                                     AGS          3
      COMMON/STR/GAM,A,LK,BK,F,LM,BL,LL,BQ,LTO,SMUD,CMUD              STR          2
      REAL LK,LM,LL,LTO                                               STR          3
C                                                                     STR          4
C                                                                     STR2        17
      EXTERNAL PARSR                                                  STR2        18
      DIMENSION PAR(5),BX(9),BY(9)                                    STR2        19
      EQUIVALENCE (LTO,LTOT),(E,PAR,BX)                               STR2        20
      INTEGER C,CI,BDAT,ELNUM,HCFD                                    STR2        21
      REAL MUD,LTOT,LEN                                               STR2        22
      DATA HCFD/3HCFD/                                                STR2        23
C NAME OF CELL FROM INPUT AND ITS INDEX                               STR2        24
      C = BDAT(M,1)                                                   STR2        25
      CI = ELNUM(C)                                                   STR2        26
      KI = INFF(4,M)                                                  STR2        27
C RETRIEVE INPUT DATA                                                 STR2        28
      CALL DATA(M,1,1,4,PAR)                                          STR2        29
C SET BZERO,BQ,AVAC,MUD                                               STR2        30
      BZ = PAR(1)                                                     STR2        31
      BQ = PAR(2)                                                     STR2        32
      AVAC = PAR(3)                                                   STR2        33
      MUD = 6.283185307*PAR(4)                                        STR2        34
C DETERMINE INSERTION POINT, EITHER 4 OR 10                           STR2        35
      IF (KI.EQ.2) GO TO 50                                           STR2        36
      INSP = 4                                                        STR2        37
      GO TO 49                                                        STR2        38
   50 INSP = 10                                                       STR2        39
C COMPUTE ALPHA, BETA AT INSERTION POINT                              STR2        40
   49 L = CI - INSP                                                   STR2        41
      CALL BET(L,BX,BY,0)                                             STR2        42
      PALPH = BX(3)                                                   STR2        43
      ALPHA = ABS(PALPH)                                              STR2        44
```

```
      BETA = BX(2)
      GAM = BX(4)
C  COMPUTE BMAX FOR THE CELL
      L = CI - 2
      CALL BET(L,BX,BY,0)
      BMAX = BX(2)
C  GET VALUE OF RHO FOR THE CELL
      IOP = INFF(1,CI)
      IF (IOP.EQ.HCFD)  GO TO 51
C  CELL,ALTC OR DCFD
      I = 3
      GO TO 52
C   CFD
 51   I = 4
 52   RHO = FDAT(CI,I)
      SMUD = SIN(MUD)
      CMUD = COS(MUD)
      A = AVAC
C  COMPUTE FIRST GUESS FOR THETA
      LK = BQ/(A*BZ)
      BK = SQRT(LK/RHO)
      F = ALPHA/GAM
      LM = 1./(F*(BK*BK))
      THET = BK * LM
C  CALL ROOT FINDING ROUTINE
      CALL GRT (1,THET,1,PARSR)
      IF (IERR)  RETURN
      LTOT = BL+2*LL+2.*LM
      G = BQ/A
      CALL HED
      WRITE(3,1) C,PALPH,BETA,RHO,BZ,BQ,AVAC,LM,LK,BL,LL,LTOT,A,G
      CALL HED
      PAR(1) = LM
      PAR(2) = LK
      PAR(3) = RHO
      PAR(4) = BL
      PAR(5) = LL
      CALL SSSR(M,PAR,KI,1)
C  ON RETURNING, THE MI LIST CONTAINS THE LIST FOR THE CELL WITH THE
C  CSS INSETED. I.E., QK, WHETE Q IS THE CELL NAME PREFIX.
C  QK IS COMPUTED AND SAVED.
      J = 2 - KI
      L = CI - 13
      MI(2) = L
      MI(12) = L
      L = CI - 14
      MI(J+5) = L
      MI(J+8) = L
      L = CI -15
      MI(1) = L
      MI(J+6) = L
      MI(J+7) = L
      MI(13) = L
      L = CI - 16
      MI(3) = L
      MI(J+4) = L
      MI(J+9) = L
```

```
      MI(11) = L
      IF (KI.EQ.2) GO TO 53
      MI(4) = M
      GO TO 54
   53 MI(10) = M
   54 L = M - 5
      CALL MMM(L,13,MI)
C  STORE THE MI LIST
      E = 0.
      J = 0
      DO 55 I =1,13
      J = MI(I)
      CALL  REPINT(L,I,J)
   55 CONTINUE
      RETURN
    1 FORMAT (* COLLINS STRAIGHT SECTION IN CELL*,A1,* .*/1H0,5X,
     1 *ALPHA=*F12.8,*,*,7X,*BETA=*,F12.8,*,*,7X,*RHO=*,F12.8/
     2 9X,*B0=*,F12.8,*,*,9X,*BQ=*,F12.8,*,*,6X,*AVAC=*,F12.8/
     3 6X,* LMAG=*,F12.8,*,*,10X,*K=*,F12.8,*,*,6X,*LDRF=*,F12.8,
     4 *,*,6X,*LEND=*,F12.8/6X,* LTOT=*,F12.8,*,*,9X,*AQ=*,F12.8,
     5 *,*,6X,*GRAD=*,F12.8,*  KG/M*)
      END
```

```
      SUBROUTINE STR4(M,NCELL,PAR)                                      STR4       2
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/GRR/IERR,BSW,VSW,RSW,QSW,XSW                               GRR        2
      LOGICAL IERR,BSW,VSW,RSW,QSW,XSW                                  GRR        3
                                                                        GRR        4
      COMMON/BMI/MI(5000)                                               BMIL       1
      COMMON/NELS/NELS                                                  BMIL       2
      LEVEL 2,MI                                                        BMIL       3
                                                                        BMIL       4
                                                                        BMI        3
C                                                                       WRK        2
      COMMON/WRK/BZ,BQ,AZ,DPSI,SM,SL,PSI,WW,WT,BETW,P,Q,                WRK        3
     1      T,U,BKZ,BKZP,CZ,SZ,CHZB,SHZB,PHI,BK,SA,BL,BM,               WRK        4
     2      AQ,RHO,BETS,PSIWS,BKZI,BKZPI,G(6),AB,C,S,CHB,               WRK        5
     3      SHB,E,EI,SZCB,SZBC,CZBC,CZCB,SZSB,SZBS,CZBS,CZSB,           WRK        6
     4      ZW,ZT,BKI,Y,Z,Y1,Y2,Y3,Y4                                   WRK        7
C                                                                       WRK        8
      DIMENSION BX(9),BY(9),PAR(4),PARQ(7),PARR(7),NCELL(3)             STR4       7
      EQUIVALENCE (BX,G),(BZ,BKS)                                       STR4       8
      LOGICAL QMAG,RMAG                                                 STR4       9
      INTEGER PI,QI,RI,ELNUM,HMAG                                       STR4      10
      EXTERNAL STRFCN                                                   STR4      11
      DATA HMAG/3HMAG/                                                  STR4      12
C                                                                       STR4      13
C  INPUT...K   STR4     P    Q    R    MUS/2PI    L    M                STR4      14
C                                                                       STR4      15
      PSI = PAR(1)*6.283185307                                          STR4      16
      SL = PAR(2)                                                       STR4      17
      SM = PAR(3)                                                       STR4      18
C  FIND INDEX OF PREVIOUSLY DEFINED ELEMENTS                            STR4      19
      PI = ELNUM(NCELL(1))                                              STR4      20
      QI = ELNUM(NCELL(2))                                              STR4      21
      RI = ELNUM(NCELL(3))                                              STR4      22
      INAME = MNAME(M)                                                  STR4      23
      QMAG = .FALSE.                                                    STR4      24
      RMAG = .FALSE.                                                    STR4      25
C  IF INPUT ELEMENTS Q AND R ARE MAGNETS, TURN ON MAG SWITCH            STR4      26
C  IF NOT MAGNETS, IT IS ASSUMED THEY ARE CELLS                         STR4      27
      IQP = INFF(1,QI)                                                  STR4      28
      IRP = INFF(1,RI)                                                  STR4      29
      IF (IQP.EQ.HMAG)  QMAG = .TRUE.                                   STR4      30
      IF (IRP.EQ.HMAG)  RMAG = .TRUE.                                   STR4      31
C  GET PARAMETERS FOR OUTER AND INNER ELEMENTS---IN MAG SEQUENCE.       STR4      32
      IF (QMAG)  GO TO 9                                                STR4      33
C  Q IS CELL.  PUT INTO MAG SEQUENCE                                    STR4      34
      CALL DATA(QI,1,1,7,PARQ)                                          STR4      35
      PARQ(4) = PARQ(7)                                                 STR4      36
      PARQ(5) = PARQ(6)                                                 STR4      37
      PARQ(6) = PARQ(5)                                                 STR4      38
      GO TO 10                                                          STR4      39
C  Q IS MAGNET                                                          STR4      40
 9    CALL DATA(QI,1,1,6,PARQ)                                          STR4      41
```

```
 10   IF (RMAG)  GO TO 15
C  R IS CELL.  PUT INTO MAG SEQUENCE
      CALL DATA(RI,1,1,7,PARR)
      PARR(4) = PARR(7)
      PARR(5) = PARR(6)
      PARR(6) = PARR(5)
      GO TO 20
C  R IS MAGNET
 15   CALL DATA(RI,1,1,6,PARR)
C  COMPUTE BETATRON FUNCTIONS
 20   IF (PAR(4).NE.0.)  PARR(1) = PAR(4)
      CALL BET(PI,BX,BY,0)
      ALPHX = BX(3)
      ALPHY = BY(3)
      BETW = BX(2)
      BETS = BY(2)
C
      SZ = SIN(PSI)
      CZ = COS(PSI)
C  REQUIRED MATRIX ELEMENTS OF COMPLETE REPLACEMENT ARRAY
C     (HORIZONTAL PLANE)  M11=P, M12=0, M21=T, M22=U
      P = (CZ + BX(3)*SZ) * BY(7)/BX(7)
      Q = BX(7) * BY(7) * SZ
      T = -((1. - BX(3)*BY(3)) * SZ - (BX(3) + BY(3))*CZ)/(BX(7)*BY(7))
      U = (CZ + BY(3)*SZ) * BX(7)/BY(7)
C
      BX(1) = ABS(PARQ(2))
      RHO = PARQ(3)
      BX(4) = 1.0/RHO
      BKZ = SQRT((BX(1) + BX(4)) / RHO)
      BKZP = SQRT(ABS(-BX(1) + BX(4)) / RHO)
      BKZI = 1.0/BKZ
      BKZPI = 1.0/BKZP
C
      PHI = BKZ*SM
      CZ = COS(PHI)
      SZ = SIN(PHI)
      Z = BKZP*SM
      E = EXP(Z)
      EI = 1.0/E
      CHZB = 0.5*(E + EI)
      SHZB = 0.5*(E - EI)
C  SET UP KOF AND KOD
      BX(1) = SM
        DO 25 I = 2,6
 25     BX(I) = PARQ(I)
      BX(2) = ABS(BX(2))
      L = M - 3
      CALL STDAT(L,1,1,6,BX)
      CALL MAGNET (L,BX)
      MI(2) = L
      L = M - 4
      BX(2) = -BX(2)
      CALL STDAT(L,1,1,6,BX)
      CALL MAGNET(L,BX)
      MI(9) = L
C  SET UP KE
```

```
                L = M - 5
                CALL DRIFT(L,SL)
                MI(1) = L
                MI(10) = L
          C  SET UP MATRIX LIST FOR REST OF MODIFIED HALF-CELL.
                L = M - 7
                MI(3) = L
                MI(8) = L
                MI(4) = M - 2
                L = M - 6
                MI(5) = L
                MI(6) = L
                MI(7) = M - 1
                  DO 30 I = 1,6
           30     G(I) = PARR(I)
                RHO = G(3)
                BKS = SQRT(ABS((G(2) + G(4)/G(3)) / G(3)))
                PHI = BKS*G(1)
                CALL GRT(1,PHI,7,STRFCN)
                IF (IERR) RETURN
                CALL DRIFT(M-7,SA)
                CALL DRIFT(M-6,BL)
                G(1) = BM
                G(2) = ABS(G(2))
                CALL STDAT(M-1,1,1,6,G)
                CALL MAGNET (M-1,G)
                G(2) = -G(2)
                CALL STDAT(M-2,1,1,6,G)
                CALL MAGNET(M-2,G)
                CALL MMM(M,10,MI)
          C  OUTPUT
                CALL HED
                SPRO = -G(2)
                DRF = 2.0*BL
                BX(1) = 2.0*(SL + SM + SA + BM + BL)
                PHAD = PAR(1)*360.
                WRITE(3,2) INAME,NCELL(1)
                WRITE(3,3) PAR(1),PHAD,INAME,INAME,NCELL(2),SM,PSI,INAME,INAME,
               1  NCELL(3),BM
                WRITE(3,4) BETW,ALPHX,SL,BL,BX(1)
                WRITE(3,5) BETS,ALPHY,SA,DRF
                CALL HED
          C
            2   FORMAT(1X,*4-ELEMENT ANTISYMMETRIC LONG STRAIGHT SECTION ARRAY  *,
               1  A1,*   , REPLACES LATTICE FROM  *A3,*  TO NEXT ANTI HOMOLOGOUS *
               2 *POINT.*)
            3   FORMAT(3X,* PHASE ADVANCE = *,F9.5,* 2PI*/
               1  18X,*= *,F9.5,* DEG.*,10X,*OUTER ELEMENTS *,A1,*0F,*,A1,
               2  *OD ARE LIKE MAGNETS OF *A3,*, BUT WITH LENGTH MI = *,F14.8,*.*/
               3  18X,*= *,F9.5,* RAD.*,10X,*INNER ELEMENTS *,A1,*IF,*,A1,
               4  *ID ARE LIKE MAGNETS OF *,A3,*, BUT WITH LENGTH M2 = *,F14.8,
               5  *.*)
            4   FORMAT(4X,*BETAX = *,F14.8,2X,*ALPHAX = *,F14.8,*  LEND = *,F14.8,
               1  *  LDRF/2 = *,F14.8,11X,*LTOT = *,F14.8)
            5   FORMAT(4X,*BETAY = *,F14.8,*  ALPHAY = *,F14.8,*  LSEP = *,
               1  F14.8,*  LDRF   = *,F14.8)
                RETURN
```

```
      END                                                          STR4      156
```

```
      SUBROUTINE STXY(M,RX,RY,RW)                                       STXY       2
C  STORES RX AND RY MATRICES IN LQ(M). DIMENSIONS ARE (2,3) FOR EACH.   STXY       3
C  IF ELEMENT NUMBER IS -, STORE INVERSE                                STXY       4
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1          LDFLG,FIN                                               CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER   SWTCH      2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                          SWTCH      3
                                                                        SWTCH      4
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL   DIM        2
                                                                        DIM        3
      DIMENSION RX(2,3),RY(2,3),AX(2,3),AY(2,3),RW(3)                   STXY       9
      DIMENSION RPQ(4),T(49)                                            STXY      10
      EQUIVALENCE (P1,RPQ(1)),(P2,RPQ(2)),(Q1,RPQ(3)),(Q2,RPQ(4))       STXY      11
C                                                                       STXY      12
      IF (M.GE.0) GO TO 7                                               STXY      13
    4 M = - M                                                           STXY      14
      AX(2,1) = -RX(2,1)                                                STXY      15
      AX(1,1) = RX(2,2)                                                 STXY      16
      AX(2,2) = RX(1,1)                                                 STXY      17
      AX(1,2) = -RX(1,2)                                                STXY      18
      AX(1,3) = RX(1,2) * RX(2,3) - RX(2,2)*RX(1,3)                     STXY      19
      AX(2,3) = RX(2,1) *RX(1,3) - RX(1,1)*RX(2,3)                      STXY      20
      AY(1,1) = RY(2,2)                                                 STXY      21
      AY(2,2) = RY(1,1)                                                 STXY      22
      AY(1,2) = - RY(1,2)                                               STXY      23
      AY(2,1) = - RY(2,1)                                               STXY      24
      AY(1,3) = RY(1,2)*RY(2,3) - RY(2,2)*RY(1,3)                       STXY      25
      AY(2,3) = RY(2,1)*RY(1,3) - RY(1,1)*RY(2,3)                       STXY      26
      DO 5 J=1,3                                                        STXY      27
      DO 5 I=1,2                                                        STXY      28
      RX(I,J) = AX(I,J)                                                 STXY      29
      RY(I,J) = AY(I,J)                                                 STXY      30
    5 CONTINUE                                                          STXY      31
      RW(1) = -RW(1)                                                    STXY      32
      RW(2) = -RW(2)                                                    STXY      33
      IF (BEND.NE.0.) RW(3) = -RW(3)                                    STXY      34
    7 CONTINUE                                                          STXY      35
      P1 = RX(1,3)*RX(2,1) - RX(2,3)*RX(1,1)                            STXY      36
      P2 = RX(1,3)*RX(2,2) - RX(2,3)*RX(1,2)                            STXY      37
      Q1 = RY(1,3)*RY(2,1) - RY(2,3)*RY(1,1)                            STXY      38
      Q2 = RY(1,3)*RY(2,2) - RY(2,3)*RY(1,2)                            STXY      39
      LQ = INFF(10,M)                                                   STXY      40
      NQ = INFF(11,M)                                                   STXY      41
      KIND = INFF(20,M)                                                 STXY      42
      IF (KIND.NE.3HMXY) GO TO 11                                       STXY      43
                                                                        STXY      44
C  CHECK STORAGE REQUIREMENT                                            STXY      45
      NNQ = 12 + KADDR                                                  STXY      46
      IF (NNQ.GT.NQ) GO TO 20                                           STXY      47
```

```
      NJ = LQ + KADDR
      NI = NJ + 6
      DO 10 IJ=1,3
      DO 10 II=1,2
      STORE(NJ)=RX(II,IJ)
      STORE(NI)=RY(II,IJ)
      NJ=NJ+1
      NI=NI+1
   10 CONTINUE
      CALL STDAT(M,5,1,3,RW)
      CALL STDAT(M,5,4,4,RPQ)
      RETURN

   20 WRITE (3,1) M
    1 FORMAT (55H *** ERROR *** INSUFFICIENT STORAGE RESERVED AT INDEX
     1I5)
      ERROR=.TRUE.
      RETURN

   11 CALL CONVMAT(7,T,RX,RY,RW,RPQ)
      CALL STOR7(M,T,RW)
      RETURN
      END
```

```
                                                                              STXY     71
      SUBROUTINE SUBR(IENTRY)                                                 SUBR      2
C        SUBROUTINE LOGIC. INSTRUCTION /Q /SUB /... MEANS Q IS THE NAME       SUBR      3
C        OF A S/R CONSISTING OF INSTRUCTIONS FOLLOWING UP TO AN END           SUBR      4
C        INSTRUCTION. Q MAY LATER BE REQUESTED BY... /  /CALL/K/ /Q  .        SUBR      5
                                                                              BLANK     2
      LEVEL 2, STORE,INFF,IWORK                                               BLANK     3
      COMMON STORE(48000),IWORK(10)                                           86MARSIZ  1
      DIMENSION INFF(24,2000)                                                 86MARSIZ  2
      EQUIVALENCE (INFF,STORE)                                                86MARSIZ  3
                                                                              BLANK     5
      COMMON/INSTR/OPNAME,NAME,OP,KA,KB,OBJA,OBJB,NXTM,TRA,LQ,NQ,LIN,         INSTR     2
     1             NIN,LFL,NFL,LBC,NBC,NTOT,IX,KIND,TYPE,ROWS,COLS,EXTR,      INSTR     3
     2             M,NF,NB,NI,MSUBR,ISIGN                                     INSTR     4
      INTEGER OPNAME,OP,OBJA,OBJB,TRA,TYPE,ROWS,COLS,EXTR                     INSTR     5
                                                                              INSTR     6
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,          CONTRL    2
     1          LDFLG,FIN                                                     CONTRL    3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN            CONTRL    4
                                                                              CONTRL    5
      COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER         SWTCH     2
      LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                                SWTCH     3
                                                                              SWTCH     4
      COMMON/STORE/LMAX,LINF,LFILE,MAX,MIN,ICARD(11)                          STORE     2
*   LMAX AND MAX ARE DIMENSIONS OF STORE AND INFF                             STORE     3
                                                                              STORE     4
      COMMON/COPY/CPYSW                                                       COPY      2
      LOGICAL CPYSW                                                           COPY      3
                                                                              COPY      4
      LOGICAL MOSS                                                            SUBR     12
      INTEGER BDAT,ELNUM                                                      SUBR     13
      DIMENSION INSUB(10),ITNO(10)                                            SUBR     14
      LOGICAL XSW                                                             SUBR     15
                                                                              SUBR     16
      XSW = .FALSE.                                                           SUBR     17
      IF (OP.LT.0) XSW = .TRUE.                                               SUBR     18
      MOSS = .FALSE.                                                          SUBR     19
C                                                                             SUBR     20
      EMPTY=.FALSE.                                                           SUBR     21
      IF(IENTRY.EQ.3) EMPTY=.TRUE.                                            SUBR     22
      GO TO (60,70,80),IENTRY                                                 SUBR     23
   60 CONTINUE                                                                SUBR     24
      IF (.NOT.CPYSW)  GO TO 61                                               SUBR     25
      WRITE (3,51) NAME,OPNAME,KA,KB,(ICARD(IW),IW=3,8)                       SUBR     26
      CALL HED                                                                SUBR     27
   61 IF (MODE.EQ.1)  GO TO 1                                                 SUBR     28
      ERROR=.TRUE.                                                            SUBR     29
      WRITE (3,52)                                                            SUBR     30
      RETURN                                                                  SUBR     31
   51 FORMAT (/6H  *** ,1X,A5,2X,A5,1X,I3,1X,I3,5H //  ,6A10)                 SUBR     32
   52 FORMAT   (76H ERROR***SUB INSTRUCTIONS MUST NOT BE NESTED DURING        SUBR     33
     1DEFINITION            )                                                 SUBR     34
   54 FORMAT (6H  *** ,1X,A5,2X,A5,1X,I3,1X,I3,5H //  ,6A10)                  SUBR     35
    1 MODE =2                                                                 SUBR     36
      XEQ=.FALSE.                                                             SUBR     37
      MSUBR=M                                                                 SUBR     38
      CALL RESRV(M,0,0,                                                       SUBR     39
```

```
     . 4,3,0,0,0,0,0,0
     .)
      RETURN
C  END ENTRY
 70   CONTINUE
      GO TO (4,2,3),MODE
 4    CONTINUE
      IF (MSSW) GO TO 25
      ERROR = .TRUE.
      WRITE (3,53)
      RETURN
 53   FORMAT (70H ERROR****INSTRUCTION NAMED END MUST BE PRECEDED BY A S
     1UB INSTRUCTION.)
C  SET TRA TO MSUBR
 2    CALL INFW(MSUBR,9,M)
      IF (.NOT.CPYSW)  GO TO 30
      WRITE (3,54) NAME,OPNAME,KA,KB,(ICARD(IW),IW=3,8)
      CALL HED
      CALL SKIP1
 30   MODE = 1
      XEQ=.TRUE.
      RETURN
C  MOSS CONTROLS MODE TO ALLOW PROPER FLOW OF PROGRAM WHEN MESH OR
C  VPAR ARE OPERATIVE.
 25   MOSS = .TRUE.
 3    MSUBR = INFF(9,M)
      KT = IDAT(MSUBR,3)
      ICOUNT = IDAT(MSUBR,4)
      ICOUNT = ICOUNT -1
      KT = KT + 1
      CALL REPINT(MSUBR,3,KT)
      CALL REPINT(MSUBR,4,ICOUNT)
      CALL REPINT(MSUBR,1,0)
      CALL REPINT(MSUBR,2,0)
      NAMESR=MNAME(MSUBR)
      IF(ICOUNT-1)5,6,7
 5    MCALL = INFF(9,MSUBR)
      ISNB=ISNB-1
      M=MCALL
      IF (MOSS) GO TO 8
      IF(MODS) MODE=1
      IF(.NOT.MODS) MODE=INFF(19,MCALL)
      GO TO (8,9,10),MODE
 8    XEQ=.TRUE.
      STOR=.TRUE.
      RSRV=.TRUE.
      RETURN
 9    STOR=.TRUE.
      RSRV=.TRUE.
      XEQ=.FALSE.
      RETURN
 10   STOR=.FALSE.
      RSRV=.FALSE.
      XEQ=.TRUE.
      RETURN
 6    CALL REPINT(MSUBR,2,1)
 7    TRA=MSUBR-1
```

```
          TRASW=.TRUE.
          DO 20 IS=1,10
          IST = IS
          IF (INSUB(IS).EQ.NAMESR) GO TO 21
   20     CONTINUE
C  ERROR HERE AS NAME SHOULD MATCH.  PRINT MESSAGE AND CONTINUE
          WRITE (3,105) NAMESR
  105     FORMAT (/5X,A5,29H SUBROUTINE NOT IN PRINT LIST//)
          CALL HED
          RETURN
   21     ITNO(IST)=KT
          IGO = 2
          GO TO 200
   22     RETURN
C  PRINT ITERATION NUMBERS OF SUBROUTINES
C
  200     CONTINUE
          IPMAX = ISNB-1
          IF (IPMAX.EQ.0) WRITE (3,102) INSUB(1),ITNO(1)
  102      FORMAT (/9X,4HSUB.,2X,A5,7H, ITER.,2X,I3)
          IF (IPMAX.EQ.1) WRITE (3,100) INSUB(1),ITNO(1),INSUB(2),ITNO(2)
  100     FORMAT (/9X,4HSUB.,2X,A5,7H, ITER.,2X,I3,10X,4HSUB.,2X,A5,
         1 7H, ITER.,2X,I3)
          IF (IPMAX.GE.2) GO TO 104
          GO TO 110
  104     CONTINUE
          DO 201 IS=2,IPMAX
          WRITE (3,101) INSUB(IS),ITNO(IS),INSUB(IS+1),ITNO(IS+1)
  101     FORMAT (15X,A5,9X,I3,16X,A5,9X,I3)
  201     CONTINUE
  110     CONTINUE
           GO TO (55,22),IGO
C
C  CALL ENTRY
   80     CONTINUE
          IF(MODE.EQ.3) GO TO 11
          IF (MHS) GO TO 11
          IF (VPR) GO TO 11
          IF (CPYSW)  CALL SKIP1
          CALL INP(M,0,0,
         .  1,2,0,0,0,0,0,0
         .)
          IF (CPYSW)  CALL SKIP1
          IF(MODE.EQ.2) RETURN
          IF (XSW) RETURN
C   MODE IS 1.  BEGINNING OF SUBROUTINE LOOP OR LOOPS
C  ZERO ARRAYS FOR STORAGE OF NAMES AND ITERATION NUMBERS.
   81     CONTINUE
          DO 12 KS=1,10
          INSUB(KS)=0
          ITNO(KS)=0
   12     CONTINUE
          ISNB=0
   11     NAMESR = BDAT(M,1)
          MSUBR = ELNUM(NAMESR)
          IF (MHS) GO TO 55
          IF (VPR) GO TO 55
```

```
          ISNB = ISNB+1                                             SUBR     154
           INSUB(ISNB)=NAMESR                                       SUBR     155
          ITNO(ISNB)=1                                              SUBR     156
     82   CONTINUE                                                  SUBR     157
          IF (MODE.NE.3) GO TO 55                                   SUBR     158
          IGO = 1                                                   SUBR     159
          GO TO 200                                                 SUBR     160
     55   CALL INFW(M,9,MSUBR)                                      SUBR     161
C  SETS SWITCH FOR INCREMENT TO RESTORE ORIGINAL VALUE              SUBR     162
          CALL REPINT(MSUBR,2,0)                                    SUBR     163
C  SET BEGINNING OF COUNT FOR ITERATION NUMBER PRINT                SUBR     164
          CALL REPINT(MSUBR,3,1)                                    SUBR     165
C  SET TOTAL COUNT FOR ITERATIONS THROUGH THE SUBROUTINE            SUBR     166
          IF (KA.LT.1) KA = 1                                       SUBR     167
          CALL REPINT(MSUBR,4,KA)                                   SUBR     168
          IF (KA.EQ.1) CALL REPINT(MSUBR,2,1)                       SUBR     169
          STOR=.FALSE.                                              SUBR     170
          RSRV=.FALSE.                                              SUBR     171
          TRASW=.TRUE.                                              SUBR     172
          TRA=MSUBR-1                                               SUBR     173
C  PUT MODE INTO IX OF INFO                                         SUBR     174
          IF (.NOT.MSSW) CALL INFW(MODE,19,M)                       SUBR     175
C  SET SWITCH FOR INCREMENT TO STORE ORIGINAL VALUE                 SUBR     176
          CALL REPINT(MSUBR,1,1)                                    SUBR     177
          MODE=3                                                    SUBR     178
          RETURN                                                    SUBR     179
          END                                                       SUBR     180
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
41	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
63	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
100	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
146	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

```
      SUBROUTINE SUM(K,S,SM)
      DIMENSION S(1)
      SM=0.
      DO 10 I=1,K
      SM = SM + S(I)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SWITCH                                                 SWITCH     2
                                                                        SWITCH     3
*   CONTROLS CALLS TO DIFFERENT SWITCHES OR OVERLAYS,ACCORDING TO       SWITCH     4
* OP OF SYNCH INSTRUCTION.                                              SWITCH     5
      COMMON/INSTR/OPNAME,NAME,OP,KA,KB,OBJA,OBJB,NXTM,TRA,LQ,NQ,LIN,   INSTR      2
     1             NIN,LFL,NFL,LBC,NBC,NTOT,IX,KIND,TYPE,ROWS,COLS,EXTR, INSTR     3
     2             M,NF,NB,NI,MSUBR,ISIGN                               INSTR      4
      INTEGER OPNAME,OP,OBJA,OBJB,TRA,TYPE,ROWS,COLS,EXTR               INSTR      5
                                                                        INSTR      6
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1          LDFLG,FIN                                               CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      COMMON/GRR/IERR,BSW,VSW,RSW,QSW,XSW                               GRR        2
      LOGICAL IERR,BSW,VSW,RSW,QSW,XSW                                  GRR        3
                                                                        GRR        4
*  IF OPNAME=NULL, IT IS A P OR C CARD.  IT NEEDS SPECIAL HANDLING TO   SWITCH     9
* CONSTRUCT A SYNCH PAGE OR REM INSTRUCTION.                            SWITCH    10
      IF (OPNAME.NE.4HNULL) GO TO 5                                     SWITCH    11
      CALL SPLOAD                                                       SWITCH    12
      RETURN                                                            SWITCH    13
 5    CONTINUE                                                          SWITCH    14
      XSW = .FALSE.                                                     SWITCH    15
*    IF OP IS NEGATIVE, EXECUTION IS TO BE DELETED.  TURN ON XSW.       SWITCH    16
      IF (OP.LT.0) XSW = .TRUE.                                         SWITCH    17
      IF ((OP.LE.0).AND.(MODE.EQ.3)) RETURN                             SWITCH    18
*   TURN OP TO POSITIVE IF LOADING IS REQUIRED                          SWITCH    19
      IF (XSW) OP = -OP                                                 SWITCH    20
*   IF OP WAS ILLEGAL, IT WAS SET TO 0 BY OPNUM.  ERROR SWITCH IS ON    SWITCH    21
      IF (OP.EQ.0) RETURN                                               SWITCH    22
*  IF OP IS FIN, INCR OR REPL, IGNORE ERROR.  GO TO NEXT CASE.          SWITCH    23
      IF (OP.EQ.2.OR.OP.EQ.7.OR.OP.EQ.8) IERR = .FALSE.                 SWITCH    24
      IF (IERR) RETURN                                                  SWITCH    26
      IERR = .FALSE.                                                    SWITCH    27
      IF (OP.GT.50) GO TO 10                                            SWITCH    28
      CALL OVERLAY(5LSYNCH,1,0,6HRECALL)                                SWITCH    29
      GO TO 20                                                          SWITCH    30
 10    IF (OP.GT.64) GO TO 11                                           SWITCH    31
      CALL OVERLAY(5LSYNCH,3,0,6HRECALL)                                SWITCH    32
      GO TO 20                                                          SWITCH    33
 11   IF (OP.GE.100) GO TO 12                                           MY3SWTH    1
      CALL OVERLAY(5LSYNCH,4,0,6HRECALL)                                SWITCH    35
      GO TO 20                                                          SWITCH    36
 12   IF (OP.GT.150) GO TO 14                                           SWITCH    37
      CALL OVERLAY(5LSYNCH,5,0,6HRECALL)                                SWITCH    38
      GO TO 20                                                          SWITCH    39
   14 IF (OP.GT.160) GO TO 15                                           SWITCH    40
      CALL OVERLAY(5LSYNCH,6,0,6HRECALL)                                SWITCH    41
      GO TO 20                                                          SWITCH    42
   15 IF (OP.GT.200) GO TO 16                                           SWITCH    43
      CALL OVERLAY (5LSYNCH,7,0,6HRECALL)                               SWITCH    44
 20   RETURN                                                            SWITCH    45
*                 IF OP.GT.200 IT IS AN ERROR                           SWITCH    46
 16   WRITE (3,50) OP,OPNAME                                            SWITCH    47
 50   FORMAT (5X,3H***,I5,* IS AN INVALID OP NUMBER.    SYNCH INSTRUCTIO SWITCH    48
     1N NAMED *A5)                                                      SWITCH    49
      ERROR = .TRUE.                                                    SWITCH    50
```

```
      RETURN                                                    SWITCH      51
      END                                                       SWITCH      52
```

```
                                                                         SXTP         2
      SUBROUTINE SXTP(M,V,PAR)                                           SXTP         3
C     SX    SXTP       KB  L     D2B          BRHO          EXACT        SXTP         4
C ---------------------------------------------------------------       SXTP         5
C     SX IS A SEXTUPOLE THAT ACTS AT A POINT.  IT REPRESENTS THE         SXTP         6
C  EFFECT OF A SEXTUPOLE OF LENGTH L, WITH D2B = D2B/DX2.                SXTP         7
C  IF L=0 ON INPUT, L IS SET =1.                                         SXTP         8
C     EXACT IS AN OPTION TO CALL FOR AN EXACT CALCULATION, USING         SXTP         9
C  ELLIPTIC FUNCTIONS ACCORDING TO A METHOD OF B. AUTIN.  IT IS          SXTP        10
C  EXERCISED BY SETTING EXACT TO A NON-ZERO VALUE, BUT IS ONLY           SXTP        11
C  OBEYED IF THE RAY IS IN THE HORIZONTAL PLANE AND L IS NOT ZERO.       SXTP        12
C ----------------------------------------------------------------------SXTP        13
C                                                                        SXTP        14
C     IF KB NOT 0, SETS R(2,6) = R(4,6) = 0.                             SXTP        15
C                                                                        SXTP        16
C  IF KA.NE.0 SEXTUPOLE IS REPLACED BY GENERALIZED MULTIPOLE             SXTMPOL      1
C OF ORDER KA, SKEW IF KB=1 (SEE NPOL WRITEUP)                           SXTMPOL      2
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL    DIM          2
                                                                         DIM          3
                                                                         BLANK        2
      LEVEL 2, STORE,INFF,IWORK                                          BLANK        3
      COMMON STORE(48000),IWORK(10)                                      86MARSIZ     1
      DIMENSION INFF(24,2000)                                            86MARSIZ     2
      EQUIVALENCE (INFF,STORE)                                           86MARSIZ     3
                                                                         BLANK        5
      COMMON /MP/ ML    $         LEVEL 2, ML                            SXLL         1
                                                                         SXLL         2
      DIMENSION V(1),PAR(1),W(4),P(4),R(7,7),RW(3),ML(3),MAG(6)          SXTP        19
      EQUIVALENCE (X,W(1)),(PX,W(2)),(Y,W(3)),(PY,W(4)),                 SXTP        20
     1            (L,P(1)),(D2B,P(2)),(BRHO,P(3))                        SXTP        21
     2         ,(LSX,MAG(1)),(K,MAG(2)),(ONE,MAG(3)),(CURV,MAG(4)),      SXTP        22
     3             (EPS0,MAG(5)),(EPS1,MAG(6))                           SXTP        23
      REAL L,LSX,LH,K,KP,MAG                                             SXTP        24
      DATA ONE,S3,RADEG/1.0,1.732050808,0.0174532925/                    SXTP        25
      DATA S32,SS3,S13,S23/.433012702,1.44224957,.33333333,3.464101616/  SXTP        26
C                                                                        SXTP        27
      IF(INFF(4,M).EQ.0) GO TO 15                                        SXTMPOL      3
      CALL NPOL(M,V,PAR)                                                 SXTMPOL      4
      RETURN                                                             SXTMPOL      5
   15 CONTINUE                                                           SXTMPOL      6
      DO 1 I=1,4                                                         SXTP        28
      W(I)=V(I)                                                          SXTP        29
    1 P(I)=PAR(I)                                                        SXTP        30
      DPP = V(6)                                                         SXTP        31
      FAC = 1. + DPP                                                     SXTP        32
      BR = FAC * BRHO                                                    SXTP        33
      F = D2B / BR                                                       SXTP        34
      KP=F                                                               SXTP        35
C  IF KB"0,TURN OFF COLUMN 6                                             SXTP        36
      KB = INFF(5,M)                                                     SXTP        37
                                                                         SXTP        38
      IF(X.EQ.0..AND.PX.EQ.0..OR.KP.EQ.0.) GO TO 11                      SXTP        39
      IF(P(4).NE.0..AND.L.NE.0..AND.Y.EQ.0..AND.PY.EQ.0.)  GO TO 5       SXTP        40
   11 IF (L.EQ.0.) GO TO 2                                               SXTP        41
      LH = L/2.                                                          SXTP        42
      X = X + PX * LH                                                    SXTP        43
      Y = Y + PY * LH                                                    SXTP        44
```

```
      F = L * F

    2 CONTINUE
      DPX=-(F/2.)*(X*X-Y*Y)
      DPY=F*X*Y
      V(2) = V(2) + DPX
      V(4) = V(4) + DPY
      IF (L.EQ.0) GO TO 3
      V(1) = X + V(2)*LH
      V(3) = Y + V(4)*LH

    3 IF (M.EQ.0) RETURN
      CALL RTRV7(MUNIT,R,RW)
      T21 = - F * X
      T23 =   F * Y
      R(2,1) = T21
      R(2,3) = T23
      R(4,1) = T23
      R(4,3) = - T21
      R(2,6)=-DPX
      R(2,7)=-DPX
      R(4,6)=-DPY
      R(4,7)=-DPY
      R(5,1)=DPX
      R(5,3)=DPY
      IF (KB.EQ.0)   GO TO 7
      R(2,6) = 0.
      R(4,6) = 0.
    7 RW(2) = -DPX

      IF (L.EQ.0.) GO TO 4
      ML(1)=MEND-3
      ML(2)=M
      ML(3)=MEND-3
      CALL DRIFT(MEND-3,LH)
      CALL STOR7(M,R,RW)
      CALL MMM(M,3,ML)
      RETURN

    4 CONTINUE
      CALL STOR7(M,R,RW)
      RETURN

C  EXACT CALCULATION FOR THICK SEXTUPOLE, H-PLANE ONLY. --- B.AUTIN
    5 CONTINUE
C
C     CONDITIONS INITIALES C
C
      SK=SIGN(1.,KP)
      AK=(ABS(KP))**S13
      U=KP*PX
      SU=-SIGN(1.,U)
      C=X*X*X*KP/3.+PX*PX
      IF(C.EQ.0.) GO TO 99
      AC=ABS(C)**S13
      SC=SIGN(1.,C)
      AXM=AC*SS3/AK
```

```
      XM=SC*SK*AXM
      TM=SQRT(S3*AC)*AK/SS3
C
C     MODULE K2
C
      XK2=.5+SC*S32
      XK=SQRT(XK2)
C
C     LIGNES TRIGONOMETRIQUES DE PHI0
C
      T=ABS(1.-X/XM)/S3
      D=1.+T
      CT=(1.-T)/D
      ST=SU*2.*SQRT(T)/D
      ST2=ST*ST
      DT=SQRT(1.-XK2*ST2)
C
C     FONCTIONS DE JACOBI
C
      CALL QFUNCT(XK,XKPRIM,Q)
      XK=SQRT(XK)
      XKP=SQRT(XKPRIM)
      Z=TM*L
      TH3=THETA3(0.,Q)
      U=Z/(TH3*TH3)
      TH4=THETA4(U,Q)
      XT=XK*TH4
      SN=THETA1(U,Q)/XT
      SN2=SN*SN
      CN=XKP*THETA2(U,Q)/XT
      DN=XKP*THETA3(U,Q)/TH4
C
C     X1
C
      C1=(CT*CN-ST*SN*DT*DN)/(1.-XK2*ST2*SN2)
      XX=SC*S3*(1.-C1)/(1.+C1)
      X1M=1.-XX
      X1=X1M*XM
C
C     XPRIME1
C
      S1=ST*CN*DN+CT*SN*DT
      U=KP*S1
      SU=-SIGN(1.,U)
      CX=ABS(C*XX)
      PX1=SU*SQRT(CX*(1.+X1M+X1M*X1M))
      GO TO 100
C
C     CAS PARTICULIER C=0
C
   99 XM=-SK/AK
      TM=AK/S23
      T=SU*SQRT(X/XM)
      T1=T/(1.-TM*T*L)
      T12=T1*T1
      X1=XM*T12
      PX1=-SK*T12*T1/S3
```

```
  100 V(1)=X1
      V(2)=PX1

      IF(M.EQ.0) RETURN
C  MAG(I) = LSX,K,1.0,CURV,EPS0,EPS1 ---- INPUT TO S/R MAGNET
      LSX=L
      THETA=PX-PX1
      XAV=(X+X1)/2. + THETA*L/12.
      K=KP*XAV
      CURV=THETA/L
      EPS0=PX/RADEG
      EPS1=-PX1/RADEG
      CALL MAGNET(M,MAG)

      RETURN
      END
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
44	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
63	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
63	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
64	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
64	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
66	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
67	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
67	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
173	I	V	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

```
                                                                                SXTP       175
            SUBROUTINE TABULAT(M,TABSW)                                         TABUL        2
                                                                                BLANK        2
            LEVEL 2, STORE,INFF,IWORK                                           BLANK        3
            COMMON STORE(48000),IWORK(10)                                       86MARSIZ     1
            DIMENSION INFF(24,2000)                                             86MARSIZ     2
            EQUIVALENCE (INFF,STORE)                                            86MARSIZ     3
                                                                                BLANK        5
            COMMON/SWTCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER     SWTCH        2
            LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS                            SWTCH        3
                                                                                SWTCH        4
            EQUIVALENCE(IVAR,FVAR)                                              TABUL        5
            LOGICAL TABSW                                                       TABUL        6
            DIMENSION INDX(10),NAM(10),JND(10),F(10)                            TABUL        7
            INTEGER ELNUM,BDAT                                                  TABUL        8
                                                                                TABUL        9
            KA = INFF(4,M)                                                      TABUL       10
             KB = INFF(5,M)                                                     TABUL       11
            KS = KA*60                                                          TABUL       12
      C  IF KB.GT.0, SET IPR=KB, WHICH GOES TO INNF(6) OF CYX                   TABUL       13
      C INSTRUCTION TO SUPPRESS PRINT.                                          TABUL       14
             IPR = 0                                                            TABUL       15
            IF (KB.GT.0) IPR = KB                                               TABUL       16
            IF (TABSW) GO TO 120                                                TABUL       17
      C    FETCH BCD LIST                                                       TABUL       18
      C  SET UP TABLE OF INDICES                                                TABUL       19
            CALL DATA(M,2,1,KA,NAM)                                             TABUL       20
            DO 100 I=1,KA                                                       TABUL       21
            NM = NAM(I)                                                         TABUL       22
            INDX(I) = ELNUM(NM)                                                 TABUL       23
            JND(I) = 1                                                          TABUL       24
            LM = INDX(I)                                                        TABUL       25
            IF (LM.NE.0) INFF(6,LM) = IPR                                       TABUL       26
            IF (.NOT.VCSW) GO TO 100                                            TABUL       27
      C  WHEN VCSW IS ON, ELEMENT IS ONE OF A VECTOR                            TABUL       28
      C  SHIFT OFF LETTER NAME.  SHIFT RIGHT FOR INTEGER NUMBER                 TABUL       29
            LNUM= NM.AND.00777777777777777777                                   TABUL       30
            LNUM= SHIFT(LNUM,-48)                                               TABUL       31
            JND(I) = LNUM                                                       TABUL       32
        100 CONTINUE                                                            TABUL       33
            CALL STDAT(M,7,1,KA,INDX)                                           TABUL       34
            CALL STDAT(M,8,1,KA,JND)                                            TABUL       35
            RETURN                                                              TABUL       36
        120 CONTINUE                                                            TABUL       37
            LQ = INFF(10,M)                                                     TABUL       38
      C  GET INDECES                                                            TABUL       39
            CALL DATA(M,7,1,KA,INDX)                                            TABUL       40
            CALL DATA(M,8,1,KA,JND)                                             TABUL       41
      C  GET CURSOR FROM LQ                                                     TABUL       42
            FVAR=STORE(LQ)                                                      TABUL       43
            IC=IVAR                                                             TABUL       44
            IUP = IC + KA                                                       TABUL       45
            IF (IUP.GT.KS) GO TO 130                                            TABUL       46
            DO 125 I=1,KA                                                       TABUL       47
      C  INDEX FOR VECTOR WILL NOT YET BE SET IF ORIGINALLY DEFINED IN SUB       TABUL       48
            IF (INDX(I).NE.0) GO TO 124                                         TABUL       49
            NM = BDAT(M,I)                                                      TABUL       50
```

```
          INDX(I) = ELNUM(NM)                                 TABUL   51
          CALL REPLQ(M,2,I,INDX(I))                           TABUL   52
          LM = INDX(I)                                        TABUL   53
          INFF(6,LM) = IPR                                    TABUL   54
          IF (.NOT.VCSW) GO TO 124                            TABUL   55
          LNUM= SHIFT(NM,6)                                   TABUL   56
          DECODE (10,1,LNUM) JND(I)                           TABUL   57
    1     FORMAT (I1,9X)                                      TABUL   58
          CALL REPLQ(M,3,I,JND(I))                            TABUL   59
  124     IN = INDX(I)                                        TABUL   60
          JN = JND(I)                                         TABUL   61
C   FETCH VALUE                                               TABUL   62
          CALL FLQ(IN,2,JN,F(I))                              TABUL   63
  125     CONTINUE                                            TABUL   64
C   STORE                                                     TABUL   65
          CALL STDAT(M,5,IC,KA,F)                             TABUL   66
C   RESET CURSOR                                              TABUL   67
          IVAR=IUP                                            TABUL   68
          STORE(LQ)=FVAR                                      TABUL   69
          RETURN                                              TABUL   70
C   STORAGE FILLED.  PRINT TABLE AND CONTINUE                 TABUL   71
  130     CALL PRNTAB(M)                                      TABUL   72
          GO TO 120                                           TABUL   73
          END                                                 TABUL   74
```

```
      SUBROUTINE SYCONJ(T,TSC,TXTSC)                                    SYCONJ   2
      RETURN                                                            SYCONJ   3
      END                                                               SYCONJ   4
```

```
                  SUBROUTINE TEST(M)                                         TEST        2
                  RETURN                                                     TEST        3
                  END                                                        TEST        4
```

```
      FUNCTION THETA1(U,Q)                                    THETA1   2
      A=SIN(U)                                                THETA1   3
      GO TO 1                                                 THETA1   4
      ENTRY THETA2                                            THETA1   5
      A=COS(U)                                                THETA1   6
    1 QN=2.*Q**.25                                            THETA1   7
      THETA1=QN*A                                             THETA1   8
      IF(THETA1.EQ.0.) RETURN                                 THETA1   9
      C=(A+A)**2-2.                                           THETA1  10
      B=A                                                     THETA1  11
      QM=1.                                                   THETA1  12
      GO TO 3                                                 THETA1  13
      ENTRY THETA3                                            THETA1  14
      A=COS(U+U)                                              THETA1  15
      GO TO 2                                                 THETA1  16
      ENTRY THETA4                                            THETA1  17
      A=-COS(U+U)                                             THETA1  18
    2 C=A+A                                                   THETA1  19
      B=C*A-1.                                                THETA1  20
      QM=Q**3                                                 THETA1  21
      QN=QM*(Q+Q)                                             THETA1  22
      THETA1=QN*B+Q*C+1.                                      THETA1  23
      IF(Q.EQ.0.) RETURN                                      THETA1  24
    3 QQ=AMIN1(Q**2,0.81)                                     THETA1  25
    4 A=C*B-A                                                 THETA1  26
      B=C*A-B                                                 THETA1  27
      QM=QQ*QM                                                THETA1  28
      QN=QM*QN                                                THETA1  29
      THETA1=QN*A+THETA1                                      THETA1  30
      QM=QQ*QM                                                THETA1  31
      QN=QM*QN                                                THETA1  32
      THETA1=QN*B+THETA1                                      THETA1  33
      IF(QN.GE.1.E-14) GO TO 4                                THETA1  34
      RETURN                                                  THETA1  35
      END                                                     THETA1  36
```

```
      SUBROUTINE TINV7(TINV,T)
C  INVERSE OF 7X7 SYMPLETIC MATRIX
C  6X6 MATRIX   INVERSE OF S =   SC = - U X STR X U     K=(T(I,7),I=1,6)
C                                (SC    -SC X K)
C  7X7 MATRIX   INVERSE OF T =
C                               (0        1)
      DIMENSION TINV(7,7),T(7,7), STR(6,6),U(6,6),SC(6,6),VK(6)
C
C  STR IS TRANSPOSE OF 6X6 OF  T
      DO 10  I = 1,6
      DO 10  J = 1,6
      STR(I,J) = T(J,I)
   10 CONTINUE
C  SET UP  U ARRAY
      DO 15  J = 1,6
      DO 15  I = 1,6
      U(I,J)= 0.0
   15 CONTINUE
      DO 20  I =1,5,2
      U(I,I+1) = -1.
      U(I+1,I) = +1.
   20 CONTINUE
      CALL MXM6 (STR,U,SC)
      CALL MXM6 (U,SC,SC)
      DO  25  J = 1,6
      DO  25  I = 1,6
      TINV(I,J) = - SC(I,J)
   25 CONTINUE
C  SET UP VECTOR K
      DO  30  I = 1,6
      VK(I) = T(I,7)
   30 CONTINUE
      CALL MXV6(SC,VK,VK)
      DO  35  I = 1,6
      TINV(I,7) = VK(I)
      TINV(7,I) = 0.0
   35 CONTINUE
      TINV(7,7) = 1.0
      RETURN
      END
```

```
            SUBROUTINE TRNSPOS                                                TRNSPOS    2
                                                                              BLANK      2
            LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
            COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
            DIMENSION INFF(24,2000)                                           86MARSIZ   2
            EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                              BLANK      5
            COMMON/INSTR/OPNAME,NAME,OP,KA,KB,OBJA,OBJB,NXTM,TRA,LQ,NQ,LIN,   INSTR      2
           1             NIN,LFL,NFL,LBC,NBC,NTOT,IX,KIND,TYPE,ROWS,COLS,EXTR, INSTR      3
           2             M,NF,NB,NI,MSUBR,ISIGN                               INSTR      4
            INTEGER OPNAME,OP,OBJA,OBJB,TRA,TYPE,ROWS,COLS,EXTR               INSTR      5
                                                                              INSTR      6
      C                                                                       TRNSPOS    5
      C  MATRIX STORED BY ROWS AT LFL TRANSPOSED AND STORED BY COLUMNS AT LQ  TRNSPOS    6
      C                                                                       TRNSPOS    7
            IL = LQ                                                           TRNSPOS    8
            IRR = LFL - 1                                                     TRNSPOS    9
      C                                                                       TRNSPOS   10
            DO 1 J=1,KB                                                       TRNSPOS   11
            IRR = IRR + 1                                                     TRNSPOS   12
            IR = IRR                                                          TRNSPOS   13
            DO 1 I=1,KA                                                       TRNSPOS   14
            STORE(IL) = STORE(IR)                                             TRNSPOS   15
            IL = IL + 1                                                       TRNSPOS   16
        1   IR = IR + KB                                                      TRNSPOS   17
            RETURN                                                            TRNSPOS   18
            END                                                               TRNSPOS   19
```

```
      SUBROUTINE TRK(M)                                                 TRK        2
                                                                        BLANK      2
      LEVEL 2, STORE,INFF,IWORK                                         BLANK      3
      COMMON STORE(48000),IWORK(10)                                     86MARSIZ   1
      DIMENSION INFF(24,2000)                                           86MARSIZ   2
      EQUIVALENCE (INFF,STORE)                                          86MARSIZ   3
                                                                        BLANK      5
      COMMON/BMI/MI(5000)                                               BMIL       1
      COMMON/NELS/NELS                                                  BMIL       2
      LEVEL 2,MI                                                        BMIL       3
                                                                        BMIL       4
                                                                        BMI        3
      COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMPL   DIM        2
                                                                        DIM        3
      COMMON/TRKINT/G,EX,EY,OMSQ,SEND,VMX,VMN,DINT,BEG,DPR,JPR,         TRKINT     2
     1   THET,PLT,NPT,NZ,PMAX,NSIZ,ITITLE(6)                            TRKINT     3
      LOGICAL BEG,DPR,PLT                                               TRKINT     4
      COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,    CONTRL     2
     1          LDFLG,FIN                                               CONTRL     3
      LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN      CONTRL     4
                                                                        CONTRL     5
      DIMENSION V(7),V0(7),U(56),BX(9),BY(9),VV(7)                      TRK        8
      LOGICAL DPRSAV                                                    TRK        9
      DIMENSION R7(7,7),RW(3),U0(7),UP(7),UM(7),UF(7)                   TRK       10
      EQUIVALENCE (AXCOS,V(1)), (AYCOS,V(3)), (DPP,U(6))                TRK       11
      INTEGER BDAT                                                      TRK       12
      COMMON/CDERIV/IFLAG,MR,NR,DP,NV(11),DAT(20),UZ,STOT,              CDERIV     2
     1  DAT2(7),MP2FLG,NF,H0,LOCALPH,NUMALPH                            CDERIV     3
      DIMENSION SCALE(4)                                                MY3TRK     1
      DATA MASK/77000 00000 00000 00000B/                               TRK       14
      DATA IBLNK,(RW(I),I=1,3)/1H ,3*0.0/                               TRK       15
      DATA DELP/1.E-08/                                                 TRK       16
                                                                        TRK       17
C REVISED FEB. 1976                                                     TRK       18
C INSTRUCTION TRK----TRACK PARTICLES  THROUGH LINEAR OR                 TRK       19
C NON-LINEAR TRANSFORMATIONS                                            TRK       20
                                                                        TRK       21
C-----------------------------------------------------------------------TRK       22
C T   TRK  NP IMP//V    BLIN/MAT  V0  /   NR IOF/IVSAV IMODE/IPR  IPL/  MY3TRK     2
C                //XMAX     /DXMAX    /YMAX     /DYMAX                  MY3TRK     3
C-----------------------------------------------------------------------TRK       24
                                                                        TRK       25
C NP=NUMBER PARTICLES.  THESE ARE STORED IN V, A PREVIOUS PVEC.         TRK       26
C NR = NUMBER OF TRANSITS OF BLIN TO BE MADE BY EACH PARTICLE.          TRK       27
C V = NAME OF PVEC INSTRUCTION CONTAINING INITIAL COORDINATES OF RAYS.  TRK       28
C V0 = NAME OF REFERENCE PARTICLE,ALSO DEFINED BY A PVEC.               TRK       29
C U = V0 + V IS THE PARTICLE VECTOR BEING TRACKED                       TRK       30
C BLIN = NAME OF THE BEAM LINE, DEFINED BY A BML INSTRUCTION.           TRK       31
C    IF BLANK, BEAM LINE CURRENTLY STORED IN MI ARRAY IS USED.          TRK       32
C MAT = NAME OF MATRIX THAT GIVES BETAS AT PRINT POSITIONS.             TRK       33
C IOF = OUTPUT FREQUENCY =                                              TRK       34
C                     = 00001, EACH POSITION                            TRK       35
C                     = 0R00P, EVERY R?TH TRANSIT AT P?TH POSITION..    TRK       36
                                                                        TRK       37
C  IMP = INPUT OPTION                                                   TRK       38
C  IMP=0  V=X,DX,Y,DY  (METERS,RADIANS)                                 TRK       39
C  IMP=1  V=EX(M-MRAD),PSIX(DEG),EY,PSIY                                TRK       40
```

```
      C  IF V0 IS LEFT BLANK, V STANDS FOR TOTAL VECTOR U, WITH 6 COMPONENTS  TRK      41
      C      U = X,DX,Y,DY,-DS,DP/P  OR  U = EX,PSIX,EY,PSIY,-DS,DP/P          TRK      42
      C   IVSAV = 1  SAVE TRACKED VECTORS. (REPLACE ORIGINAL VECTOR WITH       TRK      43
      C               NEW ONE IN ITS OWN STORAGE.)                             TRK      44
                                                                               TRK      45
      C   STORED VALUES WILL BE V AFTER TRACKING.                              TRK      46
      C   STORED IN CARTESIAN COORDINATES, X,DX,Y,DY,...                       TRK      47
      C   FOR SUBSEQUENT RUNS USING V, SET IMP = 0.                            TRK      48
      C     OTHERWISE, INPUT VECTOR STORAGE REMAINS UNCHANGED.                 TRK      49
      C    IMODE = 0,1  TRACK WITH STORED MATRICES AND MAPS                    TRK      50
      C          =2     TRACK USING REVMAT AND SAVE CUMULATIVE MATRIX IN       TRK      51
      C                 STORAGE OF TRK.                                        TRK      52
      C          =5     CALL REVMAT, DO NOT SAVE MATRIX                        TRK      53
      C                 (TRANSMITED TO TRKCN AS IOP)                           TRK      54
                                                                               TRK      55
      C   IPR = -1  NO PRINT                                                   TRK      56
      C   IPR =  0  DO PRINT                                                   TRK      57
      C   IPR =  2  PRINT IF TAGGED " ONLY (4-8 PUNCH)                         TRK      58
      C   IPL = -1,0  NO PLOT                                                  TRK      59
      C   IPL = 1     PLOT                                                     TRK      60
      C                                                                        MY3TRK    4
      C   XMAX, DXMAX, YMAX, DYMAX = SCALES FOR PHASE PLOTS                    MY3TRK    5
      C   (CARD MUST ALWAYS BE INCLUDED)                                       MY3TRK    6
                                                                               TRK      61
            PIO2 = ACOS(0.)                                                    TRK      62
            RADEG = PIO2/90.                                                   TRK      63
            DEGRAD = 1./RADEG                                                  TRK      64
            NPARTS=INFF(4,M)                                                   TRK      65
            NRR = IDAT(M,1)                                                    MY3TRK    7
            MVECS=MDAT(M,1)                                                    TRK      67
            NVECS=BDAT(M,1)                                                    TRK      68
            NBML=BDAT(M,2)                                                     TRK      69
            CALL DATA (M,1,1,4,SCALE)                                          MY3TRK    8
            NREF=2                                                             TRK      70
            MR = 4                                                             TRK      71
            NR = 1                                                             TRK      72
            IFLAG = 1                                                          TRK      73
            NV(1) = 1                                                          TRK      74
            NVEC0=BDAT(M,4)                                                    TRK      75
            IF (NVEC0.NE.5H     ) GO TO 2                                      TRK      76
            WRITE (3,1000) NBML                                                TRK      77
            NREF=1                                                             TRK      78
        1   DO 3 J=1,7                                                         TRK      79
            U(J)=0.                                                            TRK      80
        3   V0(J)=0.                                                           TRK      81
            GO TO 4                                                            TRK      82
        2   MVEC0=MDAT(M,4)                                                    TRK      83
            CALL DATA(MVEC0,1,1,7,V0)                                          TRK      84
            DO 21 J=1,5                                                        TRK      85
       21   U(J) = 1.E3*V0(J)                                                  TRK      86
            U(6) = 100.*V0(6)                                                  TRK      87
            U(7)=V0(7)                                                         TRK      88
            WRITE (3,1006) NVEC0,NVEC0,(U(J),J=1,7)                            TRK      89
        4   CONTINUE                                                           TRK      90
            IMP = INFF(5,M)                                                    MY3TRK    9
            IOF = IDAT(M,2)                                                    TRK      92
            IVSAV = IDAT(M,3)                                                  TRK      93
```

```
      IMODE = IDAT(M,4)
      IPR = IDAT(M,5)
      IPL = IDAT(M,6)
      IF (IMODE.EQ.0)  IMODE = 1
      IF (IMODE.NE.2)  GO TO 30
      NPARTS = 1
      CALL RTRV7(MUNIT,R7,RW)
      CALL STOR7(M,R7,RW)
   30 IF (NBML.NE.IBLNK)  GO TO 402
      NELS = IABS(NELS)
      GO TO 401
  402 CONTINUE
      MBML=MDAT(M,2)
C  PUT BEAM LIST IN MI ARRAY
      NDAT=INFF(17,MBML)
      CALL MIFILL(MBML,1,NDAT,NELS,MI)
      IF(ERROR) RETURN
C  FIND BETAS FOR REFERENCE MATRIX MAT
  401 NMAT=BDAT(M,3)
      IF (NMAT.EQ.5H     ) GO TO 403
      MEL=MDAT(M,3)
      CALL BET(MEL,BX,BY,0)
  403 CONTINUE
      PLT=.FALSE.
      DPR=.FALSE.
      IF (IPR.LT.0)  DPR = .TRUE.
      IF (IPL.GT.0)  PLT = .TRUE.
      KREV = IOF/1000
      KPOS = MOD(IOF,1000)
      IF (PLT) WRITE(5) NPARTS,NRR,NVECS,NBML,NMAT,KPOS,NVEC0,
     1                  (U(J),J=1,7)

      DO 250 IP=1,NPARTS
      BEG=.TRUE.
      IPS=IP*7-6
      CALL DATA(MVECS,1,IPS,7,V)
      IGO = 0
      IF (IMODE.EQ.2)  IGO = 1
   10 IF (IMP.EQ.0)  GO TO 8
      EX=V(1)*1.E-6
      EY=V(3)*1.E-6
      PHIX=V(2)*RADEG
      PHIY=V(4)*RADEG
      AX=SQRT(EX*BX(2))
      AY=SQRT(EY*BY(2))
      C=COS(PHIX)
      S=SIN(PHIX)
      V(1)=AX*C
      V(2)=-AX*(S+BX(3)*C)/BX(2)
      C=COS(PHIY)
      S=SIN(PHIY)
      V(3)=AY*C
      V(4)=-AY*(S+BY(3)*C)/BY(2)
    8 IF (NREF.EQ.1) GO TO 81
      DO 41 I=5,7
   41 V(I) = 0.
   81 DO 82 J=1,7
```

```
      82  VV(J) = V(J)                                                  TRK      151
      83  DO 5 J=1,7                                                    TRK      152
          U(J) = V0(J) + V(J)                                           TRK      153
       5  U0(J) = U(J)                                                  TRK      154
C         NM = INFF(2,M)                                                TRK      155
          IF(IOF.EQ.0) IOF = 1                                          TRK      156
          IF (NR.EQ.0) NR=1                                             TRK      157
C  TRACK ONE PARTICLE THROUGH THE BEAM SYSTEM.                          TRK      158
C  PRINT HEADING AND INITIAL CONDITIONS                                 TRK      159
          IF(DPR) GO TO 51                                              TRK      160
          IF (IOF.NE.1) GO TO 551                                       TRK      161
          WRITE(3,1001)                                                 TRK      162
          GO TO 51                                                      TRK      163
     551  IF(NREF.EQ.1) WRITE(3,1021)                                   TRK      164
    1021 FORMAT(*0PARTICLE NAME   REV   POSITION*9X*X*8X*DX*9X*Y*8X*DY*  TRK      165
        16X*DS*6X*DP/P  EX(MM-MR) EY(MM-MR)*)                           TRK      166
          IF(NREF.EQ.2) WRITE(3,1011)                                   TRK      167
      51  CONTINUE                                                      TRK      168
          IRV = 1                                                       TRK      169
          IPOS = 1                                                      TRK      170
C  FIND NAME OF PARTICLE                                                TRK      171
          IF(DPR) GO TO 52                                              TRK      172
          IF(NREF.EQ.2) WRITE(3,1012) NVECS,IP                          TRK      173
          IF(NREF.EQ.1) WRITE(3,1002) NVECS,IP                          TRK      174
      52  CONTINUE                                                      TRK      175
C  LOOP OVER REVOLUTIONS                                                TRK      176
      55  DO 213 IRV = 1,NRR                                            TRK      177
          IRM=IRV-1                                                     TRK      178
          IF(.NOT.DPR.AND.IOF.EQ.1) WRITE(3,1005) IRM                   TRK      179
C  LOOP THROUGH BEAM SYSTEM                                             TRK      180
          NELF=NELS                                                     TRK      181
          IF (IRV.EQ.NRR) NELF = NELF + 1                               TRK      182
C  RESET RANDOM SEED SO ERRORS REPEAT ON SUCCESSIVE TURNS               TRK      183
          CALL RANGET(NRN)                                              TRK      184
          DO 212 IBS=1,NELF                                             TRK      185
          NEL=MI(IBS)                                                   TRKAPR86   1
          IPOS=IBS-1                                                    TRK      186
C  IF INTEGRATION STEPS ARE BEING PRINTED, OMIT TRK PRINT.              TRK      187
          IF(IOF.NE.01000)  GO TO 61                                    TRK      188
          IF (IBS.NE.1)  GO TO 211                                      TRK      189
          IF (NMAT.EQ.IBLNK)  GO TO 214                                 TRK      190
          GO TO 62                                                      TRK      191
      61  CONTINUE                                                      TRK      192
      62  DO 6 J=1,4                                                    MY3TRK    10
       6  V(J) = 1.E3 * ( U(J) - V0(J) )                                MY3TRK    11
          IF(IOF.EQ.1) GO TO 214                                        TRK      193
          IF (NMAT.EQ.5H     ) GO TO 214                                TRK      194
          IF (IPOS.NE.KPOS) GO TO 211                                   TRK      195
          IF (MOD(IRV-1,KREV) .NE. 0)  GO TO 211                        TRK      196
          WX = BX(2)*V(2)*V(2) + 2.*BX(3)*V(1)*V(2) + BX(4)*V(1)*V(1)   TRK      199
          WY = BY(2)*V(4)*V(4) + 2.*BY(3)*V(3)*V(4) + BY(4)*V(3)*V(3)   TRK      200
C  TIME FOR OUTPUT                                                      TRK      201
          IF (NREF.NE.1) GO TO 7                                        TRK      202
          IF(DPR) GO TO 215                                             TRK      203
          WRITE(3,1005) IRM,IPOS,(U(I),I=1,6),WX,WY                     TRK      204
          GO TO 215                                                     TRK      205
       7  CONTINUE                                                      TRK      206
```

```
      IF (WX.NE.0.) GO TO 71                                     TRK       207
      PHIX = 0.                                                  TRK       208
      GO TO 72                                                   TRK       209
   71 AXSIN = - BX(2)*V(2) - BX(3)*V(1)                          TRK       210
      PHIX = ATAN2(AXSIN,AXCOS)*DEGRAD                           TRK       211
   72 IF (WY.NE.0.) GO TO 73                                     TRK       212
      PHIY = 0.                                                  TRK       213
      GO TO 74                                                   TRK       214
   73 AYSIN = - BY(2)*V(4) - BY(3)*V(3)                          TRK       215
      PHIY = ATAN2(AYSIN,AYCOS)*DEGRAD                           TRK       216
   74 CONTINUE                                                   TRK       217
      IF(DPR) GO TO 215                                          TRK       218
      WRITE(3,1014) IRM,IPOS,DPP,WX,PHIX,WY,PHIY,(V(J),J=1,4)    TRK       219
      GO TO 215                                                  TRK       220
  214 CONTINUE                                                   TRK       221
      IF (IPR.NE.2)  GO TO 216                                   TRK       222
      MATI=INFF(2,NEL)                                           TRK       223
      IF ((MASK.AND.MATI).NE.1L") GO TO 215                      TRK       224
  216 CONTINUE                                                   TRK       225
      IF (DPR) GO TO 215                                         TRK       226
      WRITE (3,1004) IRM,IPOS,(U(J),J=1,6)                       MY3TRK     12
  215 CONTINUE                                                   TRK       228
      IF(.NOT.PLT) GO TO 211                                     TRK       229
      IF(IBS.GT.1) GO TO 211                                     TRKAPR86    2
      WRITE(5) (U(J),J=1,4),(V(J),J=1,4),IP,IRM,IPOS             MY3TRK     13
      WRITE(6) (U(J),J=1,4),(V(J),J=1,4),IP,IRM,IPOS             MY3TRK     14
C ADVANCE THROUGH BEAM SYSTEM                                    TRK       231
  211 CONTINUE                                                   TRK       232
      IF (IBS.GT.NELS) GO TO 212                                 TRK       233
      NEL = MI(IBS)                                              TRK       234
      CALL TRKCN(M,NEL,U,0,IMODE)                                TRK       235
  212 CONTINUE                                                   TRK       236
      CALL RANSET(NRN)                                           TRK       237
  213 CONTINUE                                                   TRK       238
      IF (IGO.EQ.0)  GO TO 245                                   TRK       239
      IF (IGO-2) 220,225,240                                     TRK       240
  220 CALL RTRV7(M,R7,RW)                                        TRK       241
      DPRSAV=DPR                                                 TRK       242
      DPR=.TRUE.                                                 TRK       243
      DLP = -DELP                                                TRK       244
      DO 222 I=1,7                                               TRK       245
  222 UF(I) = U(I)                                               TRK       246
      GO TO 230                                                  TRK       247
  225 DLP = + DELP                                               TRK       248
      DO 227 I=1,7                                               TRK       249
  227 UM(I) = U(I)                                               TRK       250
  230 DO 231 J=1,7                                               TRK       251
  231 U(J) = V0(J) + VV(J)                                       TRK       252
      U(6) = U(6) + DLP                                          TRK       253
      IGO = IGO + 1                                              TRK       254
      GO TO 55                                                   TRK       255
  240 DO 242 I=1,7                                               TRK       256
  242 UP(I) = U(I)                                               TRK       257
      DO 243 I=1,6                                               TRK       258
  243 R7(I,6) = 0.5*(UP(I) - UM(I))/DELP                         TRK       259
      CALL MXV7(R7,U0,U0)                                        TRK       260
      DO 244 I=1,5                                               TRK       261
```

```
  244 R7(I,7) = UF(I) - U0(I)                                        TRK       262
      CALL STOR7(M,R7,RW)                                            TRK       263
      CALL MXV7(R7,V0,V0)                                            TRK       264
      DO 246 J=1,7                                                   TRK       265
  246 V(J) = UF(J) - V0(J)                                           TRK       266
      DPR=DPRSAV                                                     TRK       267
      GO TO 248                                                      TRK       268
  245 CONTINUE                                                       TRK       269
      DO 247 J=1,4                                                   TRK       270
  247 V(J) = U(J) - V0(J)                                            TRK       271
  248 IF (IVSAV.EQ.1)  CALL STDAT(MVECS,1,IPS,7,V)                   TRK       272
  250 CONTINUE                                                       TRK       273
                                                                     TRK       274
      CALL HED                                                       TRK       275
      IF (.NOT.PLT) RETURN                                           TRK       276
      END FILE 5                                                     TRK       277
      CALL TRKPLT(SCALE)                                             MY3TRK     15
      RETURN                                                         TRK       279
 1000 FORMAT (27H PARTICLE TRACKING THROUGH A5,1H.)                  TRK       280
 1001 FORMAT (106H0PARTICLE NAME   REV.   POSITION        X        DX TRK       281
     1                Y          DY           DS          DP/P)      TRK       282
 1002 FORMAT(1H0,3X,A5,I3)                                           TRK       283
 1004 FORMAT (16X,I3,6X,I3,5X,6(1X,F12.8))                           MY3TRK     16
 1005 FORMAT (16X,I3,6X,I3,5X,8(F10.7))                              TRK       285
 1006 FORMAT (10X,*TRACKING RELATIVE TO REFERENCE RAY *A5*, ( UNITS(X,DX TRK       286
     1,Y,DY,DS,DP/P)  = (MM,MRAD,MM,MRAD,MM,0/0) )* /20X,A5* = *7F10.5)  TRK       287
 1011 FORMAT(1H0,*      RAY      REV. POS.  DP/P   EX(MM-MR)     PHIX(DEG TRK       288
     1) EY(MM-MR)     PHIY(DEG)  X(MM)       DX(MR)      Y(MM)       DY( TRK       289
     2MR)* )                                                         TRK       290
 1012 FORMAT (1H0,3X,A5,I2)                                          TRK       291
 1014 FORMAT (10X,2I5,F11.3,8F12.5)                                  TRK       292
      END                                                            TRK       293
```

```
      SUBROUTINE TRKPLT(SCALE)                                          MY3TKPL    1
      DIMENSION U(7),V(7),T(3),SCALE(1)                                 MY3TKPL    2
      DATA T/1HX,1HY,1H./                                               MY3TKPL    3
      REWIND 5                                                          MY3TKPL    4
      REWIND 6                                                          MY3TKPL    5
      XR1 = SCALE(1)                                                    MY3TKPL    6
      YU1 = SCALE(2)                                                    MY3TKPL    7
      XR2 = SCALE(3)                                                    MY3TKPL    8
      YU2 = SCALE(4)                                                    MY3TKPL    9
      READ(5) NPARTS,NR,NVECS,NBML,NMAT,KPOS,NVEC0,U                    MY3TKPL   10
      DO 1 K1=1,NPARTS                                                  MY3TKPL   11
      NERR=0                                                            MY3TKPL   12
      NDX=0                                                             MY3TKPL   13
      NDY=0                                                             MY3TKPL   14
      DO 2 K2=1,NR                                                      MY3TKPL   15
      READ(5) (U(J),J=1,4),(V(J),J=1,4),IP,IRM,IPOS                     MY3TKPL   16
      IF(NERR.EQ.1) GO TO 2                                             MY3TKPL   17
      IF(ABS(V(1)).LT.1.E04.AND.ABS(V(2)).LT.1.E04) GO TO 4433          MY3TKPL   18
      K22=K2-1                                                          MY3TKPL   19
      WRITE(3,777)K1,K22                                                MY3TKPL   20
  777 FORMAT(1H0,*PARTICLE NUMBER*,I6,*  WENT UNSTABLE ON REV*,I6)      MY3TKPL   21
      NERR=1                                                            MY3TKPL   22
      GO TO 2                                                           MY3TKPL   23
 4433 CONTINUE                                                          MY3TKPL   24
      IF(K2.GT.1) GO TO 20                                              MY3TKPL   25
      WRITE(3,100)                                                      MY3TKPL   26
  100 FORMAT(1H1,5X,*U1*,7X,*U2*,7X,*U3*,7X,*U4*,17X,*V1*,7X,*V2*,7X,   MY3TKPL   27
     .       *V3*,7X,*V4*,7X,*IPX  IRM  IPOS*)                          MY3TKPL   28
      WRITE(3,101) (U(J),J=1,4),(V(J),J=1,4),IP,IRM,IPOS                MY3TKPL   29
  101 FORMAT(1H0,4X,4F9.6,6X,4F9.3,3I5)                                 MY3TKPL   30
      XL1=-XR1                                                          MY3TKPL   31
      YL1=-YU1                                                          MY3TKPL   32
      CALL SETP(XL1,XR1,YL1,YU1)                                        MY3TKPL   33
   20 CONTINUE                                                          MY3TKPL   34
      CALL PLOTT(1,T(1),V(1),V(2))                                      TRKP683    1
      IF(K2.EQ.NR) CALL PAGE(0,K1)                                      MY3TKPL   36
      IF(K2.EQ.NR) CALL PAGTRKP(0,K1)                                   86TRKPL    1
      IF(ABS(V(1)).GT.XR1.OR.ABS(V(2)).GT.YU1) NDX=NDX+1                86TRKPL    2
      IF(K2.EQ.NR) WRITE(3,102) NDX                                     86TRKPL    3
  102 FORMAT(1H0,I6,"  PATICLES OUTSIDE DIAGRAM")                       86TRKPL    4
    2 CONTINUE                                                          86TRKPL    5
      DO 3 K2=1,NR                                                      86TRKPL    6
      READ(6) (U(J),J=1,4),(V(J),J=1,4),IP,IRM,IPOS                     86TRKPL    7
      IF(NERR.EQ.1) GO TO 3                                             86TRKPL    8
      IF(ABS(V(3)).LT.1.E4.AND.ABS(V(4)).LT.1.E4) GO TO 4434            86TRKPL    9
      K22=K2-1                                                          86TRKPL   10
      WRITE(3,777)K1,K22                                                86TRKPL   11
      NERR=1                                                            86TRKPL   12
      GO TO 3                                                           86TRKPL   13
 4434 CONTINUE                                                          86TRKPL   14
      IF(K2.GT.1) GO TO 30                                              86TRKPL   15
      WRITE(3,100)                                                      86TRKPL   16
      WRITE(3,101) (U(J),J=1,4),(V(J),J=1,4),IP,IRM,IPOS                86TRKPL   17
      XL2=-XR2                                                          86TRKPL   18
      YL2=-YU2                                                          86TRKPL   19
      CALL SETP(XL2,XR2,YL2,YU2)                                        86TRKPL   20
   30 CONTINUE                                                          86TRKPL   21
```

```
              CALL PLOTT(1,T(2),V(3),V(4))                          86TRKPL   22
              IF(K2.EQ.NR) CALL PAGTRKP(0,K1)                       86TRKPL   23
              IF(ABS(V(3)).GT.XR1.OR.ABS(V(4)).GT.YU2) NDY=NDY+1    86TRKPL   24
              IF(K2.EQ.NR) WRITE(3,102) NDY                         MY3TKPL   60
        3     CONTINUE                                              MY3TKPL   61
        1     CONTINUE                                              MY3TKPL   62
              REWIND 5                                              MY3TKPL   63
              REWIND 6                                              MY3TKPL   64
              RETURN                                                MY3TKPL   65
              END                                                   MY3TKPL   66
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
7	I	SCALE	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
8	I	SCALE	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
9	I	SCALE	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

```
      SUBROUTINE SETP(XL,XTOP,YL,YTOP)                      MY3TKPL   67
      DIMENSION A(112,51),Z(6),X(1),Y(1)                    MY3TKPL   68
      COMMON/SAML/A,Z,XLOW,XRANGE,YLOW,YRANGE               MY3TKPL   69
      DATA BLANK/4H    /                                    MY3TKPL   70
      XR=XTOP-XL                                            MY3TKPL   71
      YR=YTOP-YL                                            MY3TKPL   72
      XLOW=XL                                               MY3TKPL   73
      XRANGE=XR                                             MY3TKPL   74
      YLOW=YL                                               MY3TKPL   75
      YRANGE=YR                                             MY3TKPL   76
      DO 10 I=1,112                                         MY3TKPL   77
      DO 10 J=1,51                                          MY3TKPL   78
      A(I,J)=BLANK                                          MY3TKPL   79
   10 CONTINUE                                              MY3TKPL   80
      RETURN                                                MY3TKPL   81
      END                                                   MY3TKPL   82
```