



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)

Notice: This technical note has been authored by employees of Brookhaven Science Associates, LLC under Contract No.DE-AC02-76CH00016 with the U.S. Department of Energy. The publisher by accepting the technical note for publication acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this technical note, or allow others to do so, for United States Government purposes.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or any third party's use or the results of such use of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

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. Introduction1

II. Directory Listings1-2

> [PARSA1.SYNCH]READ.ME 2

> HOW TO RUN THE CDC - VERSION OF SYNCH.....2

> CDCDEMO.DAT3

> VAX VERSION INFO.....4

> HOW TO USE THE VAX - VERSION OF SYNCH4

> HOW TO RUN THE VAX VERSION OF SYNCH5

> WORK FILES5-6

> CDCDEMO.DAT 7

> CDCSYNCH.DAT.....9

III. [PARSA1.SYNCH]CDCSYNCH.FOR.....9

III.Paged (1-500)



1



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	FORO11.DAT;1
FORO12.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]
COPYSF.
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 following 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 SYNI.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 SYNIIII.FOR for synch with plot)
SYNCHINC.TLB
```

Compile MAINS,SYNI,SYNII,SYNIIII. 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 plotting) 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 plotting 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.										


```

          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.
COPYS.
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].



1	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
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
10		MAIN	11
	C	MAIN	12
	COMMON/CLC/REG(109)	CLC	2
		BLANK	2
	LEVEL 2, STORE,INFF,IWORK	BLANK	3
15	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
20	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
25	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
30	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
35	* 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
40	COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMP	DIM	2
		DIM	3
	COMMON/BMI/MI(5000)	BMIL	1
	COMMON/NELS/NELS	BMIL	2
	LEVEL 2,MI	BMIL	3
45		BMIL	4
		BMI	3
	COMMON/BMI1/MI1(16000)	NV3BMI1	1
	COMMON/NELS1/NELS1	BMI1L	2
	LEVEL 2,MI1	BMI1L	3
50		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
55	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/SWCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	SWCH	2
60	LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS	SWCH	3
		SWCH	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
65	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
70		OPLIST	3
	COMMON/SVNAM/NAMRUN	SVNAM	2
	COMMON/COPY/CPYSW	COPY	2
	LOGICAL CPYSW	COPY	3
		COPY	4
75	COMMON/CORB/TH(84),TV(84),NMON,NNH,NNV,IH,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
80	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
85	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
90	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
95	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
100	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
105	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
110	C	MAIN	63
	CALL SETOPS	MAIN	64
		MAIN	65
	9 FIN = .FALSE.	MAIN	66
	MINFLG=1	MAIN	67

115	NONU = .FALSE.	MAIN	68
	NOPR = .FALSE.	MAIN	69
	MIFLG = .FALSE.	MAIN	70
	MATFLG = .FALSE.	MAIN	71
	GLOBAL = .FALSE.	MAIN	72
120	LFILE = 1	MAIN	73
	LDFLG=.FALSE.	MAIN	74
	KNFLAG = .FALSE.	MAIN	75
	EMPTY = .FALSE.	MAIN	76
	IERR=.FALSE.	MAIN	77
125	SMINCAL=.FALSE.	MAIN	78
	CPYSW = .TRUE.	MAIN	79
	DO 1 I=1,7	MAIN	80
	DO 1 J=1,10	MAIN	81
	1 DWORD7(I,J)=0.00000001	MAIN	82
130	MINCOM=3	MAIN	83
	NOPR=.FALSE.	MAIN	84
	NONU=.FALSE.	MAIN	85
	LEND=LMAX	MAIN	86
	MEND=MAX	MAIN	87
135	MIN=MAX	MAIN	88
	M7END = MEND - ISAV	MAIN	89
	* FILL BLANK PORTION OF ARRAY WITH ZEROES.	MAIN	90
	* IFL IS USED MAINLY TO DISTINGUISH SYMBOLIC FL.PT. FROM REAL FL.PT.	MAIN	91
	DO 8 J=1,15	MAIN	92
140	IFL(J) = MASK1.AND.IFM(J)	MAIN	93
	8 CONTINUE	MAIN	94
	2 CALL RDINST	MAIN	95
		MAIN	96
	4 CALL SWITCH	MAIN	97
145	IF (FIN) GO TO 9	MAIN	98
	IF(ERROR) GO TO 6	MAIN	99
	C CHECK FOR REDEFINITION OF ELEMENT BY NAME	MAIN	100
	IF (MODE.EQ.1.OR.MODE.EQ.2) CALL NAMCHK(M)	MAIN	101
	IF(MODE.EQ.3) GO TO 3	MAIN	102
150	IF (.NOT.MSSW) GO TO 7	MAIN	103
	CALL RDINF(IC,M)	MAIN	104
	CALL MESH(M,KA,OPNAME)	MAIN	105
	IF (MSSW) GO TO 3	MAIN	106
	C IF PAGE OR COMMENT OUTSIDE SUBROUTINE, REUSE INFF SPACE.	MAIN	107
155	7 IF ((OPNAME.EQ.HPAGE.OR.OPNAME.EQ.HREM).AND.(MODE.EQ.1)) GO TO 2	MAIN	108
	MIN = MIN - 1	MAIN	109
	NXTM=MIN	MAIN	110
	CALL INFW(MIN,8,M)	MAIN	111
	M=MIN	MAIN	112
160	GO TO 2	MAIN	113
	3 NXTM = INFF(8,M)	MAIN	114
	M=NXTM	MAIN	115
	IF(.NOT.TRASW) GO TO 5	MAIN	116
	M=TRA	MAIN	117
165	TRASW=.FALSE.	MAIN	118
	5 CALL RDINF(IC,M)	MAIN	119
	GO TO 4	MAIN	120
	6 CALL ERRSR(M)	MAIN	121
	GO TO 9	MAIN	122
170	END	MAIN	123

1	SUBROUTINE SETPLT	PLT	7
	INTEGER FR80(2),ITEK(4),C835(2)	PLT	8
		BLANK	2
	LEVEL 2, STORE,INFF,IWORK	BLANK	3
5	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
10	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
15	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
20	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
25	CALL CALINIT	PLT	19
	RETURN	PLT	20
	END	PLT	21

1

SUBROUTINE FINPL
CALL FIN
RETURN
END

PLT 22
PLT 23
PLT 24
PLT 25

1	SUBROUTINE CCLTR(X,Y,KOR,KS,KBCD,NC)	PLT	26
	DIMENSION KBCD(7)	PLT	27
	NCH=10	PLT	28
	CALL NOARG(NA)	PLT	29
5	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
10	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
15	IF(NA.EQ.6) NCH=NC	PLT	39
	CALL SYMBOL(XZ,YZ,HT,KBCD,ANG,NCH)	PLT	40
	RETURN	PLT	41
	END	PLT	42

1	SUBROUTINE CCPLLOT(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
5	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)	RENAMB3	1
10	COMMON/BPLTCOM/MN,KW,BXX,BYX,NPLT	RENAMB3	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
15	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
20	X(N+2)=DX	PLT	51
	Y(N+2)=DY	PLT	52
	ISYM=3	PLT	53
	J=0	PLT	54
	K=1	PLT	55
25	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
30	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
35	CALL LINE(X,Y,N,1,0,3)	PRCH	4
	RETURN	PL2	4
	END	PLT	64

1	SUBROUTINE CCNEXT	PLT	65
		BLANK	2
	LEVEL 2, STORE, INFF, IWORK	BLANK	3
	COMMON STORE(48000), IWORK(10)	86MARSIZ	1
5	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
10	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
15	CALL PLOT(XN,0.,-3)	PLT	69
	RETURN	PLT	70
	END	PLT	71

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

0	1617012207000000001	VFD	42/0LNOARG,18/1
1	0000000000000000000	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	512000001 +	SA2	NOARG EQ00AAAAA000000000
	21236	AX2	30 ALIGN ADDRESS OF CALL+1
3	523277776	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

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
VFDSKAN	10	PROGRAM*	9/46 L	9/48			

1	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
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
10	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
15	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
20		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
25	COMMON /DIM/ LEND, MEND, ISAV, ISAV7, M7END, KADD, KADDR, MUNIT, MSYMP	DIM	2
		DIM	3
	COMMON/BMI/MI(5000)	BMIL	1
	COMMON/NELS/NELS	BMIL	2
	LEVEL 2, MI	BMIL	3
30		BMIL	4
		BMI	3
	COMMON/BMI1/MI1(16000)	NV3BMI1	1
	COMMON/NELS1/NELS1	BMI1L	2
	LEVEL 2, MI1	BMI1L	3
35		BMI1L	4
	C	BMI1	3
	COMMON/SWCH/BEND, MSIZE, CYCSWT, VCSW, PV, MSSW, VPR, MHS, MODS, MINZER	SWCH	2
	LOGICAL CYCSWT, MSSW, VCSW, PV, VPR, MHS, MODS	SWCH	3
		SWCH	4
40	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
45	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
50	COMMON/CCFACT/FACTOR	PL6683	2
		BETPTS	5
	COMMON/SVNAM/NAMRUN	SVNAM	2
	COMMON/COPY/CPYSW	COPY	2
	LOGICAL CPYSW	COPY	3
55		COPY	4
		SW1	21
	INTEGER BDAT, PVECS	SW1	22

	DIMENSION PAR(100),KREM(100),MATNM(20)	SW1	23
	INTEGER R7,ROT,PRD	SW1	24
60	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
65	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
70	* 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
75	* **, 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
80	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
85	* 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
90	RETURN	SW1	54
	* RUN ENTRY	SW1	55
	3 CALL SRUN(1)	SW1	56
	RETURN	SW1	57
	* SUB ENTRY	SW1	58
95	4 CALL SUBR(1)	SW1	59
	RETURN	SW1	60
	* END ENTRY	SW1	61
	5 CALL SUBR(2)	SW1	62
	RETURN	SW1	63
100	* 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
105	. 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
110	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

115	10	IF (MODE.EQ.3) GO TO 101	SW1	79
		CALL DRFRSV(M,NAME,KA,KB,0)	SW1	80
		IF (KB.EQ.0) CALL LOAD(M,1,1,0,0,0,0,0)	SW1	81
		IF (KB.GT.0) CALL LOAD(M,2,1,KB,2,0,0,0)	SW1	82
		IF (KA.EQ.0) GO TO 102	SW1	83
120	*	SET UP MATRIX SPACE FOR KA NUMBER OF DRIFTS, WHOSE LENGTHS WILL	SW1	84
	*	BE ELEMENTS OF INPUT VECTOR.	SW1	85
		CALL RSVMAT	SW1	86
		MIN = MIN - KA	SW1	87
	102	IF (XSW) RETURN	SW1	88
125		IF (ERROR) RETURN	SW1	89
	101	IF (KA.GT.0) GO TO 103	SW1	90
		EL = FDAT(M,1)	SW1	91
		IF (KB.EQ.0) GO TO 106	SW1	92
	C	VARIABLE DRIFT LENGTH	SW1	93
130		FLTOT = 0.	SW1	94
		DO 104 K=1,KB	SW1	95
		MM = MDAT(M,K)	SW1	96
	104	FLTOT = FLTOT + FDAT(MM,1)	SW1	97
		FLSEP = FDAT(M,2)	SW1	98
135		IF (FLSEP.NE.0.) GO TO 105	SW1	99
		FLSEP = EL + FLTOT	SW1	100
		CALL REPFLT (M,2,FLSEP)	SW1	101
		GO TO 106	SW1	102
	105	EL = FLSEP - FLTOT	SW1	103
140		CALL REPFLT(M,1,EL)	SW1	104
	106	IF (MODE.EQ.2) RETURN	SW1	105
		CALL DRIFT(M,EL)	SW1	106
		RETURN	SW1	107
	103	IF (MODE.EQ.2) RETURN	SW1	108
145		CALL DRFMATS(M,KA)	SW1	109
		RETURN	SW1	110
	C		SW1	111
	C	EQUATE ENTRY	SW1	112
	C	PROCESSES EQU DATA. EQUATES MATRIX M TO N IF IN EXECUTION MODE.	SW1	113
150	C	MATRICES COULD HAVE VARIABLE DIMENSIONS.	SW1	114
	11	CALL INP(M,0,0,	SW1	115
		.1,2,0,0,0,0,0,0	SW1	116
		.)	SW1	117
		IF (MODE.EQ.2) RETURN	SW1	118
155		IF (XSW) RETURN	SW1	119
		N=MDAT(M,1)	SW1	120
		CALL EQU(M,N)	SW1	121
		RETURN	SW1	122
	C		SW1	123
160	C	INVERT ENTRY	SW1	124
	C	PROCESSES INVERT DATA. INVERTS IF IN EXECUTION MODE.	SW1	125
	12	KN = IBLNK	SW1	126
		IF (MSIZE.EQ.7) KN=R7	SW1	127
		IF (MSIZE.EQ.3) KN=MXPMY	SW1	128
165		CALL INP(M,KN,MINZER,	SW1	129
		. 1,2,0,0,0,0,0,0	SW1	130
		.)	SW1	131
		IF (MODE.EQ.2) RETURN	SW1	132
		IF (XSW) RETURN	SW1	133
170		N=MDAT(M,1)	SW1	134
	C	INVERT MATRIX AT INDEX N AND STORE IN INDEX M	SW1	135

	CALL INV(M,N)	SW1	136
	RETURN	SW1	137
	C	SW1	138
175	C MAGNET ENTRY	SW1	139
	C PROCESSES MAG DATA. SETS UP MAGNET MATRICES IF IN EXEC. MODE.	SW1	140
	13 IF (MODE.EQ.3) GO TO 131	SW1	141
	CALL MAGRSV(M,NAME,KA,KB,0)	SW1	142
	CALL LOAD(M,	SW1	143
180	. 6,1,0,0,0,0,0,0	SW1	144
	.)	SW1	145
	IF (KA.EQ.0) GO TO 132	SW1	146
	* SET UP MATRIX SPACE FOR KA NUMBER OF MAGNETS. SOME OF THE	SW1	147
	* INPUT PARAMETERS DERIVE FROM PREVIOUSLY DEFINED VECTOR.	SW1	148
185	CALL RSVMAT	SW1	149
	MIN = MIN - KA	SW1	150
	132 IF (MODE.EQ.2) RETURN	SW1	151
	IF (XSW) RETURN	SW1	152
	IF (ERROR) RETURN	SW1	153
190	131 IF (KA.GT.0) GO TO 133	SW1	154
	CALL DATA (M,1,1,6,PAR)	SW1	155
	CALL MAGNET(M,PAR)	SW1	156
	RETURN	SW1	157
	133 CALL MAGMATS(M,KA)	SW1	158
195	RETURN	SW1	159
	C	SW1	160
	C KICK ENTRY	SW1	161
	45 IF (MODE.EQ.3) GO TO 451	SW1	162
	CALL INP(M,R7,MINZER,1,2,3,1,0,0,0,0)	SW1	163
200	IF ((MODE.EQ.2).OR.(XSW).OR.(ERROR)) RETURN	SW1	164
	451 CALL KICK(M)	SW1	165
	RETURN	SW1	166
		SW1	167
	C	SW1	168
205	C MMM ENTRY	SW1	169
	C PROCESSES MMM DATA. MULTIPLIES MATRICES IF IN EXEC. MODE	SW1	170
	14 IF (MODE.EQ.3) GO TO 141	SW1	171
	IF (MSIZE.NE.3) KN = R7	SW1	172
	CALL INP(M,KN,MINZER,1,-2,0,0,0,0,0,0)	SW1	173
210	IF (MODE.EQ.2) RETURN	SW1	174
	IF (XSW) RETURN	SW1	175
	141 NDAT=INFF(17,M)	SW1	176
	CALL MIFILL(M,1,NDAT,NELS,MI)	SW1	177
	KB = INFF(5,M)	SW1	178
215	IF (KB.NE.0.AND.KB.NE.1) GO TO 142	SW1	179
	CALL MMM(M,NELS,MI)	SW1	180
	RETURN	SW1	181
	142 CALL EXECMM(M,NELS,KB,MI)	SW1	182
	RETURN	SW1	183
220	C	SW1	184
	C WMA ENTRY	SW1	185
	C PROCESSES WMA DATA. WRITES MATRICES IF IN EXEC.MODE.	SW1	186
	15 CALL INP(M,0,0,	SW1	187
	. KA,2,0,0,0,0,0,0	SW1	188
225	.)	SW1	189
	IF(MODE.EQ.2) RETURN	SW1	190
	IF (XSW) RETURN	SW1	191
	CALL MLIST(M,1,KA,MI)	SW1	192

	IF(ERROR) RETURN	SW1	193
230	CALL WMA(KA)	SW1	194
	RETURN	SW1	195
	C	SW1	196
	C SHIFT ENTRY	SW1	197
	C PROCESSES SHF DATA. SETS UP SHIFT MATRIX IF IN EXEC. MODE	SW1	198
235	16 CALL INP(M,MXPMY,MINZER,	SW1	199
	. 4,1,0,0,0,0,0,0	SW1	200
	.)	SW1	201
	IF (MODE.EQ.2) RETURN	SW1	202
	IF (XSW) RETURN	SW1	203
240	CALL DATA(M,1,1,4,PAR)	SW1	204
	CALL SHF(M,PAR)	SW1	205
	RETURN	SW1	206
	C	SW1	207
	C REFLECT ENTRY	SW1	208
245	C PROCESSES REF DATA. COMPUTES MATRIX REFLECTION IF IN EXEC.MODE	SW1	209
	18 KN = IBLNK	SW1	210
	IF (MSIZE.EQ.7) KN = R7	SW1	211
	IF (MSIZE.EQ.3) KN = MXPMY	SW1	212
	CALL INP(M,KN,MINZER,	SW1	213
250	. 1,2,0,0,0,0,0,0	SW1	214
	.)	SW1	215
	IF (MODE.EQ.2) RETURN	SW1	216
	IF (XSW) RETURN	SW1	217
	N=MDAT(M,1)	SW1	218
255	CALL REF(M,N)	SW1	219
	RETURN	SW1	220
	C	SW1	221
	C INCR ENTRY	SW1	222
	* DETERMINE TYPE TO BE INCREMENTED— FL.PT., INTEGER, KA OR KB.	SW1	223
260	19 IF (MODE.EQ.3) GO TO 192	SW1	224
	DECODE (10,201,ICARD(3)) KT	SW1	225
	KE = 0	SW1	226
	IF (KT.EQ.1H) KE = 1	SW1	227
	IF (KT.EQ.1HF) KE = 1	SW1	228
265	IF (KT.EQ.1HI) KE = 3	SW1	229
	IF (KT.EQ.2HKA) KE = 3	SW1	230
	IF (KT.EQ.2HKB) KE = 3	SW1	231
	IF (KE.NE.0) GO TO 191	SW1	232
	* ILLEGITIMATE TYPE	SW1	233
270	WRITE (3,190) KT	SW1	234
	190 FORMAT (5X,1H*,*ERROR*,2H* ,A5,* IS AN ILLEGITIMATE TYPE OF INCREM	SW1	235
	1ENT.*)	SW1	236
	ERROR = .TRUE.	SW1	237
	RETURN	SW1	238
275	191 CONTINUE	SW1	239
	CALL RESRV(M,0,0,	SW1	240
	. 2,2,2,KE,0,0,0,0	SW1	241
	.)	SW1	242
	CALL LOAD(M,	SW1	243
280	. 2,2,1,KE,0,0,0,0	SW1	244
	.)	SW1	245
	IF (MODE.EQ.2) GO TO 193	SW1	246
	192 IF (XSW) RETURN	SW1	247
	KO = KA	SW1	248
285	CALL INCR(M,KO)	SW1	249

	RETURN	SW1	250
	193 CALL INFV(MSUBR,19,M)	SW1	251
	RETURN	SW1	252
	C	SW1	253
290	C REPL ENTRY	SW1	254
	20 CONTINUE	SW1	255
	IF (MODE.EQ.3) GO TO 204	SW1	256
	* DETERMINE TYPE OF REPLACEMENT — FL. PT., INTEGER, BCD OR KA,KB	SW1	257
	DECODE (10,201,ICARD(3)) KT	SW1	258
295	201 FORMAT (5X,A5)	SW1	259
	KE = 0	SW1	260
	* IF BLANK, DEFAULT IS F	SW1	261
	IF (KT.EQ.1H) KE =1	SW1	262
	IF (KT.EQ.1HF) KE = 1	SW1	263
300	IF (KT.EQ.2HSF) KE = 1	SW1	264
	IF (KT.EQ.1HQ) KE = 1	SW1	265
	IF (KT.EQ.1HH) KE = 2	SW1	266
	IF (KT.EQ.1HI) KE = 3	SW1	267
	IF (KT.EQ.2HKA) KE = 3	SW1	268
305	IF (KT.EQ.2HKB) KE = 3	SW1	269
	IF (KE.NE.0) GO TO 202	SW1	270
	* ILLEGITIMATE TYPE	SW1	271
	WRITE (3,203) KT	SW1	272
	203 FORMAT (5X,1H*,*ERROR*,2H* ,A5,* IS AN ILLEGITIMATE TYPE OF REPLAC	SW1	273
310	1EMENT.*)	SW1	274
	ERROR = .TRUE.	SW1	275
	RETURN	SW1	276
	202 CONTINUE	SW1	277
	CALL INP(M,0,0,	SW1	278
315	. 2,2,1,KE,0,0,0,0	SW1	279
	.)	SW1	280
	IF (MODE.EQ.2) RETURN	SW1	281
	IF (XSW) RETURN	SW1	282
	204 CONTINUE	SW1	283
320	CALL REPL(M)	SW1	284
		SW1	285
	RETURN	SW1	286
	C	SW1	287
	C STAR ENTRY	SW1	288
325	C PROCESSES STAR DATA. COMPUTES IF IN EXEC. MODE	SW1	289
	21 KN = IBLNK	SW1	290
	IF (MSIZE.EQ.7) KN=R7	SW1	291
	IF (MSIZE.EQ.3) KN=MXPMY	SW1	292
	CALL INP(M,KN,MINZER,	SW1	293
330	. 1,2,0,0,0,0,0,0	SW1	294
	.)	SW1	295
	IF (MODE.EQ.2) RETURN	SW1	296
	IF (XSW) RETURN	SW1	297
	CALL STAR(M,KA)	SW1	298
335	RETURN	SW1	299
	C	SW1	300
	C	SW1	301
	C CYA AND CYC ENTRY	SW1	302
	* CYA, CYC, BEST ENTRY	SW1	303
340	23 IF (MODE.EQ.3) GO TO 543	SW1	304
	C MAKE RESERVATION AND STORE INPUT DATA	SW1	305
	KN = MXPMY	SW1	306

	CALL INP(M,KN,MINZER,	SW1	307
	. 4,2,2,1,0,0,0,0)	NOV3SW1	1
345	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.3HMX) KN = MXPMY	SW1	313
	IF (OPNAME.NE.3HCYA.AND.KA.NE.21) GO TO 521	SW1	314
350	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
355	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
360	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
365	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
370	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
375	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
380	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
385	.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
390	240 CONTINUE	SW1	355
	CALL SETCYB(M,KA)	SW1	356
	IF (ERROR) RETURN	SW1	357
	245 CALL CYX(M)	SW1	358
	RETURN	SW1	359
395	C	SW1	360
	C FITQ,FITB,FITR,FITV ENTRY	SW1	361
	25 CALL SETFIT	SW1	362
	RETURN	SW1	363
		SW1	364

400	C		SW1	365
	C	WBE ENTRY	SW1	366
	C	PROCESSES WBE DATA. CALLS WBE TO WRITE BETATRON FUNCTIONS	SW1	367
	C	IF IN EXECUTION MODE	SW1	368
	26	CALL INP(M,0,0,	SW1	369
405		. KA,2,0,0,0,0,0,0	SW1	370
		.)	SW1	371
		IF(MODE.EQ.2) RETURN	SW1	372
		IF (XSW) RETURN	SW1	373
		CALL MLIST(M,1,KA,MI)	SW1	374
410		IF(ERROR) RETURN	SW1	375
		CALL WBE(KA,KB)	SW1	376
		RETURN	SW1	377
	C		SW1	378
	C	= ENTRY	SW1	379
415	27	IF (MODE.EQ.3) GO TO 275	SW1	380
		KN = IBLNK	SW1	381
		DECODE (10,5000,ICARD(4)) IOP	SW1	382
	5000	FORMAT (A5,5X)	SW1	383
		IF (IOP.NE.IBLNK) GO TO 271	SW1	384
420	C	KIND AND TYPE ARE SCALAR AND SINGLE	SW1	385
		KN = ISCL	SW1	386
		CALL INP(M,KN,0,	SW1	387
		. 1,1,1,2,0,0,0,0	SW1	388
		.)	SW1	389
425		RETURN	SW1	390
	C	LOOK AT FIRST INPUT PARAMETER	SW1	391
	271	CALL CHKD(ICARD(3),KN,ITP)	SW1	392
		IF (KN.NE.4HSCAL) GO TO 272	SW1	393
		IQ = 1	SW1	394
430		KN = ISCL	SW1	395
		IF (ITP.NE.4HSNGL) IQ = 2	SW1	396
		CALL INP(M,KN,IQ,	SW1	397
		. 1,1,1,2,2,1,0,0	SW1	398
		.)	SW1	399
435		F = FDAT(M,3)	SW1	400
		CALL REPLQ(M,1,1,F)	SW1	401
		IF (MODE.EQ.2.OR.XSW) RETURN	SW1	402
		GO TO 276	SW1	403
	C	MATRIX. INPUT IS BCD,BUT IN 10 COLUMN FIELDS.	SW1	404
440	272	IF (KN.EQ.3HMX) KN=1	SW1	405
		IF (KN.EQ.2HR7) KN = 5	SW1	406
		CALL INP(N,KN,MINZER,	SW1	407
		. 3,2,0,0,0,0,0,0	SW1	408
		.)	SW1	409
445		IF (MODE.EQ.2.OR.XSW) RETURN	SW1	410
		GO TO 277	SW1	411
	275	IF (XSW) RETURN	SW1	412
		KN = INFF(20,M)	SW1	413
		IF (KN.NE.4HSCAL) GO TO 277	SW1	414
450		IOP = BDAT(M,1)	SW1	415
	C	RETRIEVE OPERATION SYMBOL. IF KIND NOT SCALAR,IT IS 2ND BCD INPUT	SW1	416
	C	IF SINGLE SYMBOLIC ARGUMENT, GET ITS VALUE	SW1	417
		IF (IOP.EQ.IBLNK) RETURN	SW1	418
		ITP = INFF(21,M)	SW1	419
455	276	IQ = 2	SW1	420
		IF (ITP.NE.4HSNGL) IQ = 4	SW1	421

	CALL DATA (M,1,1,IQ,PAR)	SW1	422
	CALL AROP(M,PAR,IOP)	SW1	423
	RETURN	SW1	424
460	277 KREM(1) = BDAT(M,1)	SW1	425
	KREM(2) = BDAT(M,3)	SW1	426
	IOP = BDAT(M,2)	SW1	427
	CALL AROP(M,KREM,IOP)	SW1	428
	RETURN	SW1	429
465	C	SW1	430
	C CELL ENTRY	SW1	431
	29 IF(MODE.EQ.3) GO TO 293	SW1	432
	C MAKE RESERVATIONS FOR FIRST CYCLED MATRIX AND STORE INPUT DATA	SW1	433
	C RESERVE SPACE IN DATA FOR THE MI LIST	SW1	434
470	CALL RESRV (M,MXPMY,MINZER,	SW1	435
	. 7,1,12,3,0,0,0,0	SW1	436
	.)	SW1	437
	CALL LOAD(M,	SW1	438
	. 7,1,0,0,0,0,0,0	SW1	439
475	.)	SW1	440
	C SET UP SPACE FOR CYCLED MATRICES	SW1	441
	DECODE (1,520,NAME) NM	SW1	442
	CALL DEFSET (M,NM,12,MATNM,4,MXPMY)	SW1	443
	IF (MODE.EQ.2) GO TO 294	SW1	444
480	IF (XSW) RETURN	SW1	445
	293 CALL DATA(M,1,1,7,PAR)	SW1	446
	CALL CELL(M,PAR)	SW1	447
	IF (MODE.EQ.3) RETURN	SW1	448
	294 MIN=MIN-16	SW1	449
485	RETURN	SW1	450
	C	SW1	451
	C ALTC ENTRY	SW1	452
	30 IF (MODE.EQ.3) GO TO 301	SW1	453
	C MAKE RESERVATIONS FOR FIRST CYCLED MATRIX AND STORE INPUT	SW1	454
490	C RESERVE SPACE IN DATA FOR MI LIST	SW1	455
	CALL RESRV(M,MXPMY,MINZER,	SW1	456
	. 8,1,12,3,0,0,0,0	SW1	457
	.)	SW1	458
	CALL LOAD(M,	SW1	459
495	.8,1,0,0,0,0,0,0	SW1	460
	.)	SW1	461
	DECODE (1,520,NAME) NM	SW1	462
	C SET UP SPACE FOR CYCLED MATRICES AND QF,QD,QS,QT	SW1	463
	CALL DEFSET (M,NM,12,MATNM,4,MXPMY)	SW1	464
500	IF (MODE.EQ.2) GO TO 302	SW1	465
	IF (XSW) RETURN	SW1	466
	301 CALL DATA(M,1,1,8,PAR)	SW1	467
	CALL ALTC(M,KA,PAR)	SW1	468
	IF(MODE.EQ.3) RETURN	SW1	469
505	302 MIN=MIN-16	SW1	470
	RETURN	SW1	471
	C	SW1	472
	* FIT ENTRY	SW1	473
	31 IF (MODE.EQ.3) GO TO 311	SW1	474
510	K1 = KA*2	SW1	475
	K2 = KB*2	SW1	476
	K = K1 + K2 + 2	SW1	477
	CALL RESRV(M,0,0,	SW1	478

515	. K,2,1,3,0,0,0,0	SW1	479
	.)	SW1	480
	CALL LOAD(M,	SW1	481
	. 2,2,1,3,0,0,0,0	SW1	482
	.)	SW1	483
	CALL LOAD(M,	SW1	484
520	. K1,2,0,0,0,0,0,0	SW1	485
	.)	SW1	486
	CALL LOAD(M,	SW1	487
	. K2,2,0,0,0,0,0,0	SW1	488
	.)	SW1	489
525	IF (MODE.EQ.2) RETURN	SW1	490
	IF (XSW) RETURN	SW1	491
311	CALL DOFIT(M)	SW1	492
	RETURN	SW1	493
		SW1	494
530	C COPY ENTRY	SW1	495
32	IF (MODE.EQ.3) GO TO 320	SW1	496
	CALL INP(M,0,0,0,0,0,0,0,0,0)	SW1	497
320	IF (MODE.EQ.2) RETURN	SW1	498
	IF (XSW) RETURN	SW1	499
535	CPYSW = .TRUE.	SW1	500
	RETURN	SW1	501
	C NCPY ENTRY	SW1	502
33	IF (MODE.EQ.3) GO TO 330	SW1	503
	CALL INP(M,0,0,0,0,0,0,0,0,0)	SW1	504
540	330 IF (MODE.EQ.2) RETURN	SW1	505
	IF (XSW) RETURN	SW1	506
	CPYSW = .FALSE.	SW1	507
	RETURN	SW1	508
	C PAGE ENTRY	SW1	509
545	C RESTORES PAGE IF IN EXECUTION MODE	SW1	510
34	IF (MODE.EQ.3) GO TO 341	SW1	511
*	STORE DATA FIELD TO PRINT AS A COMMENT	SW1	512
	CALL INP(M,0,0,	SW1	513
	. 12,2,0,0,0,0,0,0	SW1	514
550	.)	SW1	515
	IF (MODE.EQ.2) RETURN	SW1	516
	IF (XSW) RETURN	SW1	517
341	K0 = INFF(17,M)	SW1	518
	CALL DATA (M,2,1,K0,KREM)	SW1	519
555	IF (CPYSW) WRITE (3,342) (KREM(I),I=1,K0)	SW1	520
342	FORMAT (1H1/8X,16A5)	SW1	521
	RETURN	SW1	522
	C	SW1	523
	C REM ENTRY	SW1	524
560	C PRINTS REMARKS	SW1	525
35	IF (MODE.EQ.3) GO TO 352	SW1	526
	CALL INP(M,0,0,	SW1	527
	. 12,2,0,0,0,0,0,0	SW1	528
	.)	SW1	529
565	352 IF (XSW) RETURN	SW1	530
	K0 = INFF(17,M)	SW1	531
	CALL DATA (M,2,1,K0,KREM)	SW1	532
	IF (CPYSW) WRITE (3,353) (KREM(I),I=1,K0)	SW1	533
353	FORMAT (/8X,16A5)	SW1	534
570	RETURN	SW1	535

	C		SW1	536
	*	MAGV ENTRY	SW1	537
	42	CONTINUE	SW1	538
		IF (MODE.EQ.3) GO TO 421	SW1	539
575		CALL INP(M,R7,MINZER,	SW1	540
		. 6,1,0,0,0,0,0,0	SW1	541
		.)	SW1	542
		IF (MODE.EQ.2) RETURN	SW1	543
		IF (XSW) RETURN	SW1	544
580	421	CALL DATA (M,1,1,6,PAR)	SW1	545
		CALL MAGV(M,PAR)	SW1	546
		RETURN	SW1	547
			SW1	548
	*	BETP ENTRY — PLOT BETATRON FUNCTIONS	SW1	549
585	43	IF (MODE.EQ.3) GO TO 431	SW1	550
		CALL INP(M,0,0,	SW1	551
		. 1,2,2,3,3,1,0,0	SW1	552
		.)	SW1	553
		IF (MODE.EQ.2) RETURN	SW1	554
590		IF (XSW) RETURN	SW1	555
	431	CALL PLOTBET(M)	SW1	556
		RETURN	SW1	557
	C		SW1	558
	C	SIZE ENTRY, SWITCH DEFINING MATRIX SIZE	SW1	559
595	61	IF (MODE.EQ.3) RETURN	SW1	560
	C	SET MSZ TO KA. IF KA = 7 FOR 7X7, 3 FOR 3X3, 37 IF NOT KNOWN	SW1	561
		MSIZE = KA	SW1	562
		IF (CPYSW) WRITE (3,610) NAME,OPNAME,KA,KB	SW1	563
600	610	FORMAT (6H ***,1X,A5,2X,A5,1X,I3,1X,I3)	SW1	564
		RETURN	SW1	565
	*	MESH AND VPAR ENTRY	SW1	566
	74	CONTINUE	SW1	567
		IF (MODE.EQ.3) GO TO 741	SW1	568
		KL = KA + 1	SW1	569
605		KM = 3*KA	SW1	570
		CALL RESRV(M,0,0,	SW1	571
		. KL,2,KA,3,KM,1,0,0	SW1	572
		.)	SW1	573
		CALL LOAD(M,	SW1	574
610		. 1,2,0,0,0,0,0,0	SW1	575
		.)	SW1	576
		DO 740 IJ=1,KA	SW1	577
		CALL LOAD(M,	SW1	578
		. 1,3,1,2,3,1,0,0	SW1	579
615		.)	SW1	580
	740	CONTINUE	SW1	581
		IF (MODE.EQ.2) RETURN	SW1	582
		IF (XSW) RETURN	SW1	583
	741	CONTINUE	SW1	584
620		CALL MESH(M,KA,OPNAME)	SW1	585
		RETURN	SW1	586
	C	TAB ENTRY	SW1	587
	81	IF (MODE.EQ.3) GO TO 810	SW1	588
		IF (KA.GT.10) KA = 10	SW1	589
625		KS = KA*60+1	SW1	590
		CALL INP(M,0,KS,	SW1	591
		. KA,2,0,0,0,0,0,0	SW1	592

	.)	SW1	593
	C ZERO ARRAY AT LQ	SW1	594
630	DO 811 I=1,60	SW1	595
	811 PAR(I) = 0.	SW1	596
	IK = 2	SW1	597
	DO 812 I=1,KA	SW1	598
	CALL STDAT(M,5,IK,60,PAR)	SW1	599
635	IK = IK + 60	SW1	600
	812 CONTINUE	SW1	601
	C SET CURSOR = 2 AT LQ	SW1	602
	CALL REPLQ(M,1,1,2)	SW1	603
	TABSW = .FALSE.	SW1	604
640	C SET UP INDICES TABLE	SW1	605
	CALL TABULAT(M,TABSW)	SW1	606
	IF (MODE.EQ.2) RETURN	SW1	607
	810 TABSW = .TRUE.	SW1	608
	CALL TABULAT(M,TABSW)	SW1	609
645	RETURN	SW1	610
	C PTAB ENTRY	SW1	611
	82 IF (MODE.EQ.3) GO TO 820	SW1	612
	CALL INP(M,0,0,	SW1	613
	. KA,2,0,0,0,0,0,0	SW1	614
650	.)	SW1	615
	IF (MODE.EQ.2) RETURN	SW1	616
	820 CALL PRNTAB(M)	SW1	617
	RETURN	SW1	618
	* BML ENTRY — BEAM LINE	SW1	619
655	86 IF (MODE.EQ.3) RETURN	SW1	620
	CALL INP(M,0,0,	SW1	621
	. 1,-2,0,0,0,0,0,0	SW1	622
	.)	SW1	623
	RETURN	SW1	624
660		SW1	625
	C BETA ENTRY	SW1	626
	44 IF(MODE.EQ.3) GO TO 441	SW1	627
	CALL INP(M,ISCL,MINZER,	SW1	628
	. 1,2,1,1,0,0,0,0	SW1	629
665	.)	SW1	630
	IF(MODE.EQ.2) RETURN	SW1	631
	441 MM=MDAT(M,1)	SW1	632
	CALL BET(MM,BX(1),BX(11),0)	SW1	633
	X=BX(KA)	SW1	634
670	CALL REPFLT(M,0,X)	SW1	635
	RETURN	SW1	636
	C RAND ENTRY	SW1	637
	C	SW1	638
	46 IF(MODE.NE.3) CALL INP(M,ISCL,0,3,1,0,0,0,0,0,0)	SW1	639
675	IF(MODE.EQ.2) RETURN	SW1	640
	RNS=FDAT(M,3)	SW1	641
	IF(RNS.GT.0..AND.RNS.LT.1) CALL RANSET(RNS)	SW1	642
	KB=MAX0(KB,1)	SW1	643
	IF(KA.LE.0) RETURN	SW1	644
680	C RUN RANDOM GENERATOR KA*KB TIMES	SW1	645
	KB = MAX0(KB,1)	SW1	646
	DO461 I=1,KA	SW1	647
	DO461 J=1,KB	SW1	648
	461 X=RANF(D)	SW1	649

685	RETURN	SW1	650
		SW1	651
	C SHF7 ENTRY 7X7 SHIFT MATRIX	SW1	652
	47 IF (MODE.EQ.3) GO TO 471	SW1	653
	CALL INP (M,R7,MINZER,	SW1	654
690	. 6,1,0,0,0,0,0,0	SW1	655
	.)	SW1	656
	IF (MODE.EQ.2) RETURN	SW1	657
	471 CALL DATA (M,1,1,6,PAR)	SW1	658
	CALL SHF7(M,PAR)	SW1	659
695	RETURN	SW1	660
		SW1	661
	C SOLENOID ENTRY	SW1	662
	48 IF (MODE.EQ.3) GO TO 481	SW1	663
	CALL INP (M,R7,MINZER,	SW1	664
700	. 4,1,0,0,0,0,0,0	SW1	665
	.)	SW1	666
	IF (MODE.EQ.2) RETURN	SW1	667
	481 CALL DATA (M,1,1,4,PAR)	SW1	668
	CALL SOL(M,PAR)	SW1	669
705	RETURN	SW1	670
		SW1	671
	C VAR ENTRY	SW1	672
	C DEFINES VARIABLE BY INSTRUCTION NAME AND POSITION.	SW1	673
	C.....	SW1	674
	C VNAME VAR KA NAME	SW1	675
710	C.....	SW1	676
	C EXAMPLE 1	SW1	677
	C QF MAG 1.5 0.06 1.	SW1	678
	C QFH VAR 1 QF	SW1	679
	C QFH MAG 0.75 QFH 1.	SW1	680
715	C-----	SW1	681
	C.....	SW1	682
	C	SW1	683
	C IN THIS EXAMPLE QFH HAS HALF THE LENGTH OF QF, BUT THE SAME	SW1	684
	C GRADIENT, 0.06.	SW1	685
720	C	SW1	686
	C EXAMPLE 2 — A MORE USEFUL EXAMPLE	SW1	687
	C L DRF 2.	SW1	688
	C SR SUB	SW1	689
	C QF MAG 1. 0.05 1.	SW1	690
725	C QD MAG 1. -0.05 1.	SW1	691
	C C MMM QF L QD L	SW1	692
	C END	SW1	693
	C FITQ SR C QF QD 2 2 .25 .25	SW1	694
	C	SW1	695
730	C GF VAR 2 QF	SW1	696
	C GD VAR 1 QD	SW1	697
	C QFH MAG .75 GF 1.	SW1	698
	C QDH MAG .75 GD 1.	SW1	699
	C CYC -1 QFH L QDH	SW1	700
735	C-----	SW1	701
	C	SW1	702
	49 IF (MODE.EQ.3) GO TO 491	SW1	703
	CALL INP (M,0,0,1,2,0,0,0,0,0)	SW1	704
	IF (MODE.EQ.2) RETURN	SW1	705
740	491 FVAR = FLDAT(M,1)	SW1	706
	NAM=MNAME(M)	SW1	

	WRITE (3,490) NAM,FVAR	SW1	707
	490 FORMAT (30X,A5,* VALUE — *,E15.8)	SW1	708
	C	SW1	709
745	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)	SW1	713
	IF (MODE.EQ.2) RETURN	SW1	714
750	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

1	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
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
10	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
15	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
20		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
25	COMMON /DIM/ LEND, MEND, ISAV, ISAV7, M7END, KADD, KADDR, MUNIT, MSYMP	DIM	2
		DIM	3
	COMMON/BMI/MI(5000)	BMIL	1
	COMMON/NELS/NELS	BMIL	2
	LEVEL 2, MI	BMIL	3
30		BMIL	4
		BMI	3
	COMMON/SWTC/BEND, MSIZE, CYCSWT, VCSW, PV, MSSW, VPR, MHS, MODS, MINZER	SWTCH	2
	LOGICAL CYCSWT, MSSW, VCSW, PV, VPR, MHS, MODS	SWTCH	3
		SWTCH	4
35	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
40	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
45	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
50	* 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
55	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	SW3	36
	*	STR2,STR4,STRP,STRN	SW3	37
60			SW3	38
	17	CALL WFLSR	SW3	39
		RETURN	SW3	40
	C		SW3	41
	22	CALL TEST(M)	SW3	42
65		RETURN	SW3	43
	C		SW3	44
	C		SW3	45
	C	MAGS ENTRY	SW3	46
	C	PROCESSES MAGS DATA.CALCULATES TRANSFER MATRIX FOR NEW MAGNET IF IN	SW3	47
70	C	EXECUTION MODE	SW3	48
	28	IF (MODE.EQ.3) GO TO 281	SW3	49
		CALL MAGSRV(M,NAME,KA,KB,0)	SW3	50
		IF(MODE.EQ.2) RETURN	SW3	51
		IF (XSW) RETURN	SW3	52
75	281	CALL DATA (M,1,1,5,PAR)	SW3	53
		N=MDAT(M,1)	SW3	54
		CALL MAGS(M,N,PAR)	SW3	55
		RETURN	SW3	56
	C	SUM ENTRY	SW3	57
80	36	IF(MODE.EQ.3) GO TO 361	SW3	58
		CALL INP(M,0,1,	SW3	59
		. KA,1,0,0,0,0,0,0	SW3	60
		.)	SW3	61
		IF(MODE.EQ.2) RETURN	SW3	62
85		IF (XSW) RETURN	SW3	63
	361	CALL DATA (M,1,1,KA,PAR)	SW3	64
		CALL SUM(KA,PAR,RES)	SW3	65
		CALL STDAT(M,5,1,1,RES)	SW3	66
		RETURN	SW3	67
90	C		SW3	68
	C	CELC ENTRY	SW3	69
	37	IF (MODE.EQ.3) GO TO 371	SW3	70
	C	MAKE RESERVATIONS FOR FIRST CYCLED MATRIX AND STORE INPUT DATA	SW3	71
	C	RESERVE SPACE IN DATA FOR THE MI LIST	SW3	72
95		CALL RESRV(M,MXPMY,MINZER,	SW3	73
		. 8,1,12,3,0,0,0,0	SW3	74
		.)	SW3	75
		CALL LOAD(M,	SW3	76
		. 7,1,0,0,0,0,0,0	SW3	77
100		.)	SW3	78
		DECODE(1,520,NAME) NM	SW3	79
	C	SET UP SPACE FOR CYCLED MATRICES AND QF,QD,QS,QT,QE	SW3	80
		CALL DEFSET(M,NM,12,MATNM,5,MXPMY)	SW3	81
		IF (MODE.EQ.2) GO TO 372	SW3	82
105		IF (XSW) RETURN	SW3	83
	371	CALL DATA (M,1,1,8,PAR)	SW3	84
		CALL CELC(M,PAR)	SW3	85
		IF(MODE.EQ.3) RETURN	SW3	86
	372	MIN=MIN-17	SW3	87
110		RETURN	SW3	88
	C		SW3	89
	C	CFD ENTRY	SW3	90
	38	IF (MODE.EQ.3) GO TO 381	SW3	91
	C	MAKE RESERVATIONS FOR FIRST CYCLED MATRIX AND STORE INPUT DATA	SW3	92

115	C RESERVE SPACE IN DATA FOR THE MILIST	SW3	93
	CALL RESRV(M,MXPMY,MINZER,	SW3	94
	. 8,1,12,3,0,0,0,0	SW3	95
	.)	SW3	96
	CALL LOAD(M,	SW3	97
120	. 8,1,0,0,0,0,0,0	SW3	98
	.)	SW3	99
	DECODE(1,520,NAME) NM	SW3	100
	C SET UP SPACE FOR CYCLED MATRICES AND QF,QD,QS,QT	SW3	101
	CALL DEFSET(M,NM,12,MATNM,4,MXPMY)	SW3	102
125	IF (MODE.EQ.2) GO TO 382	SW3	103
	IF (XSW) RETURN	SW3	104
	381 CALL DATA(M,1,1,8,PAR)	SW3	105
	CALL CFD(M,PAR)	SW3	106
	IF (MODE.EQ.3) RETURN	SW3	107
130	382 MIN=MIN-16	SW3	108
	RETURN	SW3	109
	C	SW3	110
	C DCFD ENTRY	SW3	111
	39 IF(MODE.EQ.3) GO TO 391	SW3	112
135	CALL RESRV(M,MXPMY,MINZER,	SW3	113
	. 9,1,12,3,0,0,0,0	SW3	114
	.)	SW3	115
	CALL LOAD(M,	SW3	116
	. 9,1,0,0,0,0,0,0	SW3	117
140	.)	SW3	118
	DECODE(1,520,NAME) NM	SW3	119
	C SET UP SPACE FOR CYCLED MATRICES AND QF,QD,QS,QT	SW3	120
	CALL DEFSET(M,NM,12,MATNM,4,MXPMY)	SW3	121
	IF (MODE.EQ.2) GO TO 392	SW3	122
145	IF (XSW) RETURN	SW3	123
	391 CALL DATA(M,1,1,9,PAR)	SW3	124
	CALL DCFD(M,PAR)	SW3	125
	IF (MODE.EQ.3) RETURN	SW3	126
	392 MIN = MIN-16	SW3	127
150	RETURN	SW3	128
	C STR4 ENTRY. PROCESSES STR4 DATA	SW3	129
	44 IF (MODE.EQ.3) GO TO 442	SW3	130
	CALL INP(M,MXPMY,MINZER,	SW3	131
	. 3,2,4,1,0,0,0,0	SW3	132
155	.)	SW3	133
	M1 = M - 1	SW3	134
	DO 441 I=1,4	SW3	135
	CALL MAGRSV (-M1,JMAG(I),0,0,NAME)	SW3	136
	441 M1 = M1 - 1	SW3	137
160	DO 440 I=5,7	SW3	138
	CALL DRFRSV(-M1,JMAG(I),0,0,NAME)	SW3	139
	440 M1 = M1 - 1	SW3	140
	IF (MODE.EQ.2) GO TO 443	SW3	141
	IF (XSW) RETURN	SW3	142
165	442 CALL DATA(M,2,1,3,NCELL)	SW3	143
	CALL DATA(M,1,1,4,PAR)	SW3	144
	CALL STR4(M,NCELL,PAR)	SW3	145
	IF (MODE.EQ.3) RETURN	SW3	146
	443 MIN = MIN - 7	SW3	147
170	RETURN	SW3	148
	C	SW3	149

	C		SW3	150
	C	STRN ENTRY.	SW3	151
175	46	IF (MODE.EQ.3) GO TO 462	SW3	152
		IF (KA.NE.2) KA=1	SW3	153
		KF = 4	SW3	154
		IF(KA.EQ.2) KF=8	SW3	155
		CALL RESRV(M,MXPMY,MINZER,	SW3	156
		. 2,2,KF,1,0,0,0,0	SW3	157
180		.)	SW3	158
		CALL LOAD (M,	SW3	159
		. 2,2,4,1,0,0,0,0	SW3	160
		.)	SW3	161
185		IF (KA.EQ.2) CALL LOAD (M,	SW3	162
		. 4,1,0,0,0,0,0,0	SW3	163
		.)	SW3	164
		DECODE (1,520,NAME) NM	SW3	165
		DO 461 I = 1,6	SW3	166
190	461	MXLIST(I) = MATNM(I + 6)	SW3	167
		MXLIST(7)= MATNM(5)	SW3	168
		MXLIST(8) = MATNM(3)	SW3	169
		M1 = M - 1	SW3	170
		DO 460 I=1,6	SW3	171
		CALL MAGRSV(-M1,MXLIST(I),0,0,NAME)	SW3	172
195	460	M1 = M1 - 1	SW3	173
		DO 464 I=7,8	SW3	174
		CALL DRFRSV(-M1,MXLIST(I),0,0,NAME)	SW3	175
	464	M1 = M1 - 1	SW3	176
		IF (MODE.EQ.2) GO TO 463	SW3	177
200		IF (XSW) RETURN	SW3	178
	462	CALL DATA(M,2,1,2,NCELL)	SW3	179
		KF = 4	SW3	180
		IF (KA.EQ.2) KF = 8	SW3	181
		CALL DATA(M,1,1,KF,PAR)	SW3	182
205		CALL STRN(M,NCELL,PAR)	SW3	183
		IF (MODE.EQ.3) RETURN	SW3	184
	463	MIN = MIN - 8	SW3	185
		RETURN	SW3	186
	C	STR2 ENTRY. DESIGN COLLINS STRAIGHT SECTION FOR INSERTION.	SW3	187
210	47	IF (MODE.EQ.3) GO TO 502	SW3	188
		CALL RESRV(M,MXPMY,MINZER,	SW3	189
		. 1,2,4,1,6,3,0,0	SW3	190
		.)	SW3	191
		CALL LOAD(M,	SW3	192
215		. 1,2,4,1,0,0,0,0	SW3	193
		.)	SW3	194
		DO 501 I = 1,3	SW3	195
	501	MXLIST(I) = MATNM(I)	SW3	196
		MXLIST(4) = MATNM(5)	SW3	197
220		DECODE (1,520,NAME) NM	SW3	198
		CALL DEFSET(M,NM,0,MXLIST,4,MXPMY)	SW3	199
		M1 = BDAT(M,1)	SW3	200
		M2 = M - 5	SW3	201
		ENCODE (10,504,INFF(2,M2)) M1,MATNM(6)	SW3	202
225	504	FORMAT (A1,A4,5X)	SW3	203
		CALL RESRV(M-5,MXPMY,MINZER,	SW3	204
		. 13,3,0,0,0,0,0,0	SW3	205
		.)	SW3	206

	IF (MODE.EQ.2) GO TO 503	SW3	207
230	IF (XSW) RETURN	SW3	208
	502 CALL STR2(M)	SW3	209
	IF (MODE.EQ.3) RETURN	SW3	210
	503 MIN = MIN - 5	SW3	211
	RETURN	SW3	212
235	C STRPI ENTRY. DESIGN PI-TYPE STRAIGHT SECTION FOR INSERTION.	SW3	213
	48 IF (MODE.EQ.3) GO TO 512	SW3	214
	CALL RESRV(M,MXPMY,MINZER,	SW3	215
	. 1,2,5,1,10,3,0,0	SW3	216
	.)	SW3	217
240	CALL LOAD(M,	SW3	218
	. 1,2,5,1,0,0,0,0	SW3	219
	.)	SW3	220
	DECODE (1,520,NAME) NM	SW3	221
	CALL DEFSET(M,NM,0,MATNM,4,MXPMY)	SW3	222
245	IF (MODE.EQ.2) GO TO 513	SW3	223
	IF (XSW) RETURN	SW3	224
	512 CALL STRPI(M)	SW3	225
	IF (MODE.EQ.3) RETURN	SW3	226
	513 MIN = MIN - 4	SW3	227
250	RETURN	SW3	228
	C SCOPE ENTRY	SW3	229
	49 CALL INP(M,0,0,	SW3	230
	. 0,0,0,0,0,0,0,0	SW3	231
	.)	SW3	232
255	IF (MODE.EQ.2) RETURN	SW3	233
	IF (XSW) RETURN	SW3	234
	CALL SCOPCN	SW3	235
	RETURN	SW3	236
	C ACT OR DELE ENTRY	SW3	237
260	80 CONTINUE	SW3	238
	IF (MODE.EQ.3) GO TO 800	SW3	239
	C IF KB=0, INPUT CONTAINS KA ELEMENTS	SW3	240
	C IF KB=1, KA ELEMENTS SHOULD BE CHANGED BEGINNING WITH ONE INPUT	SW3	241
	IF (KB.NE.0) CALL INP(M,0,0,1,2,0,0,0,0,0,0)	SW3	242
265	IF (KB.EQ.0) CALL INP(M,0,0,1,-2,0,0,0,0,0,0)	SW3	243
	IF (MODE.EQ.2) RETURN	SW3	244
	IF(XSW) RETURN	SW3	245
	800 CALL XEQCON(M)	SW3	246
	RETURN	SW3	247
270	520 FORMAT(A1)	SW3	248
	END	SW3	249

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.

1	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
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
10	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
15	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
20		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
25	COMMON /DIM/ LEND, MEND, ISAV, ISAV7, M7END, KADD, KADDR, MUNIT, MSYMP	DIM	2
		DIM	3
	COMMON/BMI/MI(5000)	BMIL	1
	COMMON/NELS/NELS	BMIL	2
	LEVEL 2, MI.	BMIL	3
30		BMIL	4
		BMI	3
	COMMON/SWCH/BEND, MSIZE, CYCSWT, VCSW, PV, MSSW, VPR, MHS, MODS, MINZER	SWCH	2
	LOGICAL CYCSWT, MSSW, VCSW, PV, VPR, MHS, MODS	SWCH	3
		SWCH	4
35	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
40	LOGICAL TOG	MOVTOG	4
	C	MOVTOG	5
		SW4	16
	DIMENSION PAR(100), KREM(100)	SW4	17
	DIMENSION T(49), RW(3)	SW4	18
45	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
50	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
55		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
60	1	50,55,56,57) NOP	SW4683	2
			SW4	35
	8	CALL BMIS	SW4	36
	10	RETURN	SW4	37
	9	CALL EMIS	SW4	38
65		RETURN	SW4	39
	C		SW4	40
	C		SW4	41
	*	SXTP ENTRY	SW4	42
	33	IF (MODE.EQ.3) GO TO 331	SW4	43
70		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
75	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
80		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
85		. KA,2,0,0,0,0,0,0	SW4	59
		.)	SW4	60
		RETURN	SW4	61
			SW4	62
	C	TRK ENTRY	SW4	63
90	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
95		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
100		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
105		.)	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
110		IF (XSW) RETURN	SW4	84
	401	CALL FXPT(M)	SW4	85
	402	RETURN	SW4	86
			SW4	87
	C	ROTZ ENTRY	SW4	88

115	41	IF(MODE.EQ.3) RETURN	SW4	89
		CALL INP(M,ROT,MINZER,	SW4	90
		. 1,1,0,0,0,0,0,0	SW4	91
		.)	SW4	92
		RETURN	SW4	93
120	C		SW4	94
	C	ROT ENTRY——ROTATE MATRIX	SW4	95
	42	IF (MODE.EQ.3) GO TO 420	SW4	96
		CALL INP(M,R7,MINZER,	SW4	97
		. 1,2,1,1,0,0,0,0	SW4	98
125		.)	SW4	99
		IF (MODE.EQ.2) RETURN	SW4	100
		IF (XSW) RETURN	SW4	101
	420	MB = MDAT(M,1)	SW4	102
		THETA = FDAT(M,1)	SW4	103
130		THETA = THETA*RADEG	SW4	104
		CALL ROTM(M,MB,THETA)	SW4	105
		RETURN	SW4	106
			SW4	107
	*	INV2 ENTRY ROTATE MATRIX 180 DEGREES AND REFLECT	SW4	108
135	43	IF (MODE.EQ.3) GO TO 430	SW4	109
		CALL INP(M,R7,MINZER,	SW4	110
		. 1,2,0,0,0,0,0,0	SW4	111
		.)	SW4	112
		IF (MODE.EQ.2) RETURN	SW4	113
140		IF (XSW) RETURN	SW4	114
	430	MB = MDAT(M,1)	SW4	115
		CALL INV2(M,MB)	SW4	116
		RETURN	SW4	117
			SW4	118
145	C	ELQ ENTRY	SW4	119
	44	IF (MODE.EQ.3) RETURN	SW4	120
		CALL INP (M,MAP,MINZER,	SW4	121
		. 1,2,5,1,0,0,0,0)	SW4	122
		RETURN	SW4	123
150	C		SW4	124
	C	MAP ENTRY	SW4	125
	1000	IF (MODE.EQ.3) RETURN	SW4	126
		CALL INP(M,MAP,MINZER,	SW4	127
		. KA,1,0,0,0,0,0,0	SW4	128
155		.)	SW4	129
		RETURN	SW4	130
	C		SW4	131
	C	MOVE ENTRY	SW4	132
	39	IF (MODE.EQ.3) GO TO 390	SW4	133
160		CALL RESRV(M,R7,MINZER,3,2,10,1,0,0,0,0)	SW4	134
		CALL LOAD(M,3,2,4,1,0,0,0,0)	SW4	135
		MQ = MDAT(M,1)	SW4	136
		NOPQ = INFF(1,MQ)	SW4	137
		IF (NOPQ.EQ.HSXTP) INFF(20,M) = HMAP	SW4	138
165		IF (MODE.EQ.2) RETURN	SW4	139
		IF (XSW) RETURN	SW4	140
	390	DO 391 I=1,7	SW4	141
	391	V0(I)=0.	SW4	142
		CALL RANGET(NRN)	MY3SW4	3
170		CALL MOVE(M,M,V0)	SW4	143
		TOG = .TRUE.	MY3SW4	4

	CALL RANSET(NRN)	MY3SW4	5
	RETURN	SW4	144
		SW4	145
175	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 WHICH IS THE TRANSFER	SW4	149
	C MATRIX FOR RAYS IN THE NEIGHBORHOOD OF A REFERENCE TRAJECTORY.	SW4	150
180	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
185	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)	SW4	158
	IF (MODE.EQ.2) RETURN	SW4	159
	IF (XSW) RETURN	SW4	160
190	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
195	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
200	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
205	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
210	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
215	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
220	IF(MODE.EQ.2) RETURN	SW4683	8
	IF(XSW) RETURN	SW4683	9
	571 CALL ORBIT(M)	SW4683	10
	RETURN	SW4683	11
	C	SW4683	12
225	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.

1	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
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
10	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
15	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
20		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
25	COMMON /DIM/ LEND, MEND, ISAV, ISAV7, M7END, KADD, KADDR, MUNIT, MSYMP	DIM	2
		DIM	3
	COMMON/BMI1/MI1(16000)	NV3BMI1	1
	COMMON/NELS1/NELS1	BMI1L	2
	LEVEL 2, MI1	BMI1L	3
30		BMI1L	4
	C	BMI1	3
	COMMON/SWCH/BEND, MSIZE, CYCSWT, VCSW, PV, MSSW, VPR, MHS, MODS, MINZER	SWCH	2
	LOGICAL CYCSWT, MSSW, VCSW, PV, VPR, MHS, MODS	SWCH	3
		SWCH	4
35	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
40	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
45	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
50	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
55	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		86SW7	6
	C	TRKB ENTRY	86SW7	7
60	36	IF (MODE.EQ.3) GO TO 360	86SW7	8
		CALL RESRV(M,MXPMY,MINZER,	86SW7	9
		. 4,2,6,3,2,1,0,0	86SW7	10
		.)	86SW7	11
		CALL LOAD(M,	86SW7	12
65		. 4,2,4,3,2,1,0,0)	86SW7	13
		CALL STDAT(M,3,6,1,ZZ)	86SW7	14
	C	6TH INTEGER IS A FLAG TO TELL TRKB IF IT HAS BEEN CALLED YET.	86SW7	15
		IP1 = IDAT(M,1)	86SW7	16
		IP2 = IDAT(M,2)	86SW7	17
70		NAMVEC = BDAT(M,3)	86SW7	18
		NAMOP = INFF(1,M)	86SW7	19
		NCOL = 20	86SW7	20
		IF(NAMVEC.EQ.4H) NCOL = 14	86SW7	21
		CALL STDAT(M,3,5,1,NCOL)	86SW7	22
75		MBML = MDAT(M,1)	86SW7	23
		NDAT = INFF(17,MBML)	86SW7	24
		CALL MIFILL(MBML,1,NDAT,NEL,MI1)	86SW7	25
		IF (KA.NE.0.OR.KB.NE.0) GO TO 361	86SW7	26
		NQ3 = NCOL*(NEL+1)	86SW7	27
80		KB = NEL	86SW7	28
		GO TO 362	86SW7	29
	361	IF (KA.LE.-1) GO TO 363	86SW7	30
		NQ3 = NCOL*(KB-KA+1)	86SW7	31
	362	LQ3 = LFILE + 1	86SW7	32
85		LFILE = LQ3 + NQ3	86SW7	33
		NTOT = NTOT + NQ3	86SW7	34
		INFF(5,M) = NEL	86SW7	35
		INFF(18,M) = NTOT	86SW7	36
		INFF(24,M) = LQ3	86SW7	37
90	363	IF (IP1.NE.0.OR.IP2.NE.0) GO TO 364	86SW7	38
		CALL REPINT(M,1,0)	86SW7	39
		CALL REPINT(M,2,NEL)	86SW7	40
	364	IF (MODE.EQ.2) RETURN	86SW7	41
		IF (XSW) RETURN	86SW7	42
95	360	CALL TRKB(M)	86SW7	43
		RETURN	86SW7	44
	C		86SW7	45
	C	TRKE ENTRY	86SW7	46
	39	IF (MODE.EQ.3) GO TO 390	86SW7	47
100		CALL RESRV(M,MXPMY,MINZER,	86SW7	48
		. 4,2,6,3,2,1,0,0	86SW7	49
		.)	86SW7	50
		CALL LOAD(M,	86SW7	51
		. 4,2,4,3,2,1,0,0)	86SW7	52
105		CALL STDAT(M,3,6,1,ZZ)	86SW7	53
	C	6TH INTEGER IS A FLAG TO TELL TRKE IF IT HAS BEEN CALLED YET.	86SW7	54
		IP1 = IDAT(M,1)	86SW7	55
		IP2 = IDAT(M,2)	86SW7	56
		NAMVEC = BDAT(M,3)	86SW7	57
110		NAMOP = INFF(1,M)	86SW7	58
		NCOL = 20	86SW7	59
		IF(NAMVEC.EQ.4H) NCOL = 14	86SW7	60
		CALL STDAT(M,3,5,1,NCOL)	86SW7	61
		MBML = MDAT(M,1)	86SW7	62

115	NDAT = INFF(17,MBML)	86SW7	63
	CALL MIFILL(MBML,1,NDAT,NEL,MI1)	86SW7	64
	IF (KA.NE.0.OR.KB.NE.0) GO TO 391	86SW7	65
	NQ3 = NCOL*(NEL+1)	86SW7	66
	KB = NEL	86SW7	67
120	GO TO 392	86SW7	68
	391 IF (KA.LE.-1) GO TO 393	86SW7	69
	NQ3 = NCOL*(KB-KA+1)	86SW7	70
	392 LQ3 = LFILE + 1	86SW7	71
	LFILE = LQ3 + NQ3	86SW7	72
125	NTOT = NTOT + NQ3	86SW7	73
	INFF(5,M) = NEL	86SW7	74
	INFF(18,M) = NTOT	86SW7	75
	INFF(24,M) = LQ3	86SW7	76
	393 IF (IP1.NE.0.OR.IP2.NE.0) GO TO 394	86SW7	77
130	CALL REPINT(M,1,0)	86SW7	78
	CALL REPINT(M,2,NEL)	86SW7	79
	394 IF (MODE.EQ.2) RETURN	86SW7	80
	IF (XSW) RETURN	86SW7	81
	390 CALL TRKE(M)	86SW7	82
135	RETURN	86SW7	83
	C	86SW7	84
	C TRKM ENTRY	86SW7	85
	40 IF (MODE.EQ.3) GO TO 400	86SW7	86
	CALL RESRV(M,MXPMY,MINZER,	86SW7	87
140	. 4,2,6,3,2,1,0,0	86SW7	88
	.)	86SW7	89
	CALL LOAD(M,	86SW7	90
	. 4,2,4,3,2,1,0,0)	86SW7	91
	CALL STDAT(M,3,6,1,ZZ)	86SW7	92
145	C 6TH INTEGER IS A FLAG TO TELL TRKM IF IT HAS BEEN CALLED YET.	86SW7	93
	IP1 = IDAT(M,1)	86SW7	94
	IP2 = IDAT(M,2)	86SW7	95
	NAMVEC = BDAT(M,3)	86SW7	96
	NAMOP = INFF(1,M)	86SW7	97
150	NCOL = 20	86SW7	98
	IF(NAMVEC.EQ.4H) NCOL = 14	86SW7	99
	CALL STDAT(M,3,5,1,NCOL)	86SW7	100
	MBML = MDAT(M,1)	86SW7	101
	NDAT = INFF(17,MBML)	86SW7	102
155	CALL MIFILL(MBML,1,NDAT,NEL,MI1)	86SW7	103
	IF (KA.NE.0.OR.KB.NE.0) GO TO 401	86SW7	104
	NQ3 = NCOL*(NEL+1)	86SW7	105
	KB = NEL	86SW7	106
	GO TO 402	86SW7	107
160	401 IF (KA.LE.-1) GO TO 403	86SW7	108
	NQ3 = NCOL*(KB-KA+1)	86SW7	109
	402 LQ3 = LFILE + 1	86SW7	110
	LFILE = LQ3 + NQ3	86SW7	111
	NTOT = NTOT + NQ3	86SW7	112
165	INFF(5,M) = NEL	86SW7	113
	INFF(18,M) = NTOT	86SW7	114
	INFF(24,M) = LQ3	86SW7	115
	403 IF (IP1.NE.0.OR.IP2.NE.0) GO TO 404	86SW7	116
	CALL REPINT(M,1,0)	86SW7	117
170	CALL REPINT(M,2,NEL)	86SW7	118
	404 IF (MODE.EQ.2) RETURN	86SW7	119

	IF (XSW) RETURN	86SW7	120
	400 CALL TRKM(M)	86SW7	121
	RETURN	86SW7	122
175	C	86SW7	123
	C IBET ENTRY	86SW7	124
	41 IF (MODE.EQ.3) RETURN	86SW7	125
	CALL INP(M,0,0,12,1,0,0,0,0,0)	86SW7	126
	RETURN	86SW7	127
180	C SOLV ENTRY	86SW7	128
	C	SW7	71
	C SOLV ENTRY	SW7	72
	37 IF (MODE.EQ.3) GO TO 370	SW7	73
	NF1 = 1 + 2*KA + 3*KB	SW7	74
185	C NB1 = 4 + 2*KA + 6*KB	SW7	75
	C NI1 = 5 + 6*KA	SW7	76
	NB1 = 4 + 8*KA + 6*KB	SW7	77
	NI1 = 5	SW7	78
	CALL RESRV(M,0,0,	SW7	79
190	. NF1,1,NB1,2,NI1,3,0,0	SW7	80
	.)	SW7	81
	CALL LOAD(M,4,2,5,3,1,1,0,0)	SW7	82
	DO 371 I=1,KA	SW7	83
	CALL LOAD(M,	SW7	84
195	.8,2,2,1,0,0,0,0	SW7	85
	.)	SW7	86
	371 CONTINUE	SW7	87
	372 DO 373 J=1,KB	SW7	88
	CALL LOAD(M,	SW7	89
200	. 6,2,3,1,0,0,0,0	SW7	90
	.)	SW7	91
	373 CONTINUE	SW7	92
	IF (MODE.EQ.2) RETURN	SW7	93
	IF (XSW) RETURN	SW7	94
205	370 CALL SOLV(M)	SW7	95
	RETURN	SW7	96
		SW7	97
	C SMIN ENTRY	SW7	98
	38 IF (MODE.EQ.3) GO TO 380	SW7	99
210	NF1 = 5*KA	SW7	100
	NB1 = 2*KA	SW7	101
	CALL RESRV(M,0,0,	SW7	102
	. NF1,1,NB1,2,0,0,0,0	SW7	103
	.)	SW7	104
215	DO 381 I=1,KA	SW7	105
	CALL LOAD(M,	SW7	106
	. 2,2,5,1,0,0,0,0	SW7	107
	.)	SW7	108
	381 CONTINUE	SW7	109
220	IF (XSW) RETURN	SW7	110
	380 CALL SMIN(M)	SW7	111
	RETURN	SW7	112
	END	SW7	113

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.

1	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
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
10	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
15	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
20		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
25	COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMP	DIM	2
		DIM	3
	COMMON/BMI/MI(5000)	BMIL	1
	COMMON/NELS/NELS	BMIL	2
	LEVEL 2,MI	BMIL	3
30		BMIL	4
		BMI	3
	COMMON/SWTC/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	SWTC	2
	LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS	SWTC	3
		SWTC	4
35	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
40	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
45	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
50	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
55	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
60		CALL INP(M,0,0,1,-2,0,0,0,0,0)	MY3SW5	6
		IF (MODE.EQ.2) RETURN	MY3SW5	7
	401	IF (XSW) RETURN	MY3SW5	8
		CALL PBML(M)	MY3SW5	9
		RETURN	MY3SW5	10
65	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
70		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
75		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
80	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
85	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
90	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
95		.)	SW5	65
		IF (MODE.EQ.2) RETURN	SW5	66
		IF (XSW) RETURN	SW5	67
	620	CALL PRNTV7(M,KA)	SW5	68
		RETURN	SW5	69
100			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
105		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
110		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

115	.)	SW5	85
	NF = NF + 1	SW5	86
	NPL = 7*I	SW5	87
	CALL REPFLT(M,NPL,V7)	SW5	88
640	CONTINUE	SW5	89
120	RETURN	SW5	90
	C MXV ENTRY. MATRIX TIMES VECTOR	SW5	91
65	IF(MODE.EQ.3) GO TO 650	SW5	92
	ROWS = 7	SW5	93
	COLS = 1	SW5	94
125	CALL INP(M,PVECS,7,2,2,0,0,0,0,0,0)	SW5	95
	IF (MODE.EQ.2) RETURN	SW5	96
	IF (XSW) RETURN	SW5	97
	C RETRIEVE ELEMENTNUMBER OF INPUT MATRIX AND VECTOR.FIND LQ OF VECTORS	SW5	98
650	M1 = MDAT(M,1)	SW5	99
130	NM = BDAT(M,2)	SW5	100
	M2 = ELNUM(NM)	SW5	101
	LOC = 1	SW5	102
	IF(.NOT.VCSW) GO TO 651	SW5	103
	DECODE (10,652,NM) NUM	SW5	104
135	652 FORMAT (1X,I1,8X)	SW5	105
	LOC = (NUM-1)*7+1	SW5	106
	651 CALL DATA (M2,5,LOC,7,PAR(8))	SW5	107
	CALL RTRV7(M1,T,RW)	SW5	108
	CALL MXV7(T,PAR(8),PAR(1))	SW5	109
140	CALL STDAT(M,5,1,7,PAR)	SW5	110
	RETURN	SW5	111
	C EQIL ENTRY	SW5	112
66	IF (MODE.EQ.3) GO TO 660	SW5	113
	KA = 3	SW5	114
145	CALL INP(M,PVECS,MINZER,	SW5	115
	. 2,2,0,0,0,0,0,0	SW5	116
	.)	SW5	117
	C SET KA FOR NUMBER OF COLUMN VECTORS	SW5	118
	IF (MODE.EQ.2) RETURN	SW5	119
150	IF (XSW) RETURN	SW5	120
	660 CALL EQIL(M)	SW5	121
	RETURN	SW5	122
	C PRINT VECTORS ANY SIZE	SW5	123
67	IF(MODE.EQ.3) GO TO 670	SW5	124
155	CALL INP(M,0,0,	SW5	125
	. KA,2,0,0,0,0,0,0	SW5	126
	.)	SW5	127
	IF (MODE.EQ.2) RETURN	SW5	128
	IF (XSW) RETURN	SW5	129
160	670 CALL PRINTV(M,KA)	SW5	130
	RETURN	SW5	131
	C EVEC ENTRY	SW5	132
68	IF (MODE.EQ.3) GO TO 680	SW5	133
	CALL INP(M,0,112,	SW5	134
165	. 2,2,0,0,0,0,0,0	SW5	135
	.)	SW5	136
	IF (MODE.EQ.2) RETURN	SW5	137
	IF (XSW) RETURN	SW5	138
	680 CALL EVEC(M)	SW5	139
170	RETURN	SW5	140
		SW5	141

	C CPLX ENTRY ——A COMPLEX NUMBER	SW5	142
	71 IF (MODE.EQ.3) RETURN	SW5	143
	TYPE = 4HCPLX	SW5	144
175	KN = ISCL	SW5	145
	CALL INP(M,KN,0,	SW5	146
	. 2,1,0,0,0,0,0,0	SW5	147
	.)	SW5	148
	RETURN	SW5	149
180	C CVEC ENTRY ——COMPLEX PARTICLE VECTOR,KA, IN NUMBER	SW5	150
	C IF KB=BLANK OR 1, ROW 7 IS 1. IF KB = 0, ROW7 IS 0.	SW5	151
	72 IF (MODE.EQ.3) RETURN	SW5	152
	V7R = 0.	SW5	153
	V7I=0.	SW5	154
185	IF (KB.EQ.1.OR.KB.EQ.1H) V7R = 1.	SW5	155
	IF(KA.EQ.0) KA=1	SW5	156
	KC = 14*KA	SW5	157
	ROWS = 7	SW5	158
	COLS = KA	SW5	159
190	TYPE = 4HCPLX	SW5	160
	CALL RESRV(M,PVECS,0,	SW5	161
	. KC,1,0,0,0,0,0,0	SW5	162
	.)	SW5	163
	DO 721 I=1,KA	SW5	164
195	CALL LOAD(M,	SW5	165
	. 12,1,0,0,0,0,0,0	SW5	166
	.)	SW5	167
	NF = NF + 2	SW5	168
	NPL = 14*(I-1) + 13	SW5	169
200	CALL REPFLT(M,NPL,V7R)	SW5	170
	NPL = NPL + 1	SW5	171
	CALL REPFLT (M,NPL,V7I)	SW5	172
	721 CONTINUE	SW5	173
	RETURN	SW5	174
205	C LIST ENTRY	SW5	175
	C	SW5	176
	C A LIST KA A1 A2 A3 A4 A5	SW5	177
	C	SW5	178
	C THE LIST INSTRUCTION SPECIFIES A SET OF ELEMENTS THAT ARE TO BE	SW5	179
210	C USED SUCCESSIVELY. THE NAME APPEARS IN A BML INSTRUCTION. EACH	SW5	180
	C TIME THE SAME LIST NAME IS ENCOUNTERED, THE NEXT AI IS USED.	SW5	181
	C	SW5	182
	C EXAMPLE	SW5	183
	C	SW5	184
215	C QQ LIST Q1 Q2 Q3 Q4 Q5	SW5	185
	C L DRF 3.	SW5	186
	C .S BML 5(Q L)	SW5	187
	C	SW5	188
	C .S IS EQUIVALENT TO	SW5	189
220	C .S BML Q1 L Q2 L Q3 L Q4 L Q5 L	SW5	190
	C	SW5	191
	74 IF (MODE.EQ.3) GO TO 741	SW5	192
	CALL INP(M,0,0,1,-2,0,0,0,0,0)	SW5	193
	741 IF (KB.NE.0) RETURN	SW5	194
225	KB=1	SW5	195
	INFF(5,M)=1	SW5	196
	RETURN	SW5	197
		SW5	198

	C SCALAR FUNCTIONS	SW5	199
230	75 IF (MODE.EQ.3) GO TO 750	SW5	200
	KN = ISCL	SW5	201
	CALL INP(M,KN,1,1,1,0,0,0,0,0)	SW5	202
	IF (MODE.EQ.2.OR.XSW) RETURN	SW5	203
	750 XX = FDAT(M,1)	SW5	204
235	NNOP = OP - 111	SW5	205
	GO TO (76,77,78,79,80,81,82,83,84,85,86) NNOP	SW5	206
	76 YY = SIN(XX)	SW5	207
	GO TO 99	SW5	208
	77 YY = COS(XX)	SW5	209
240	GO TO 99	SW5	210
	78 YY = EXP(XX)	SW5	211
	GO TO 99	SW5	212
	79 YY = ALOG(XX)	SW5	213
	GO TO 99	SW5	214
245	80 YY = XX*XX	SW5	215
	GO TO 99	SW5	216
	81 YY = SQRT(XX)	SW5	217
	GO TO 99	SW5	218
	82 YY = TAN(XX)	SW5	219
250	GO TO 99	SW5	220
	83 YY = ASIN(XX)	SW5	221
	GO TO 99	SW5	222
	84 YY = ACOS(XX)	SW5	223
	GO TO 99	SW5	224
255	85 YY = ATAN(XX)	SW5	225
	GO TO 99	SW5	226
	86 YY = ABS(XX)	SW5	227
	99 CALL STDAT(M,5,1,1,YY)	SW5	228
	RETURN	SW5	229
260		SW5	230
	C CALC ENTRY	SW5	231
	123 IF (MODE.EQ.3) GO TO 1231	SW5	232
	CALL INP(M,0,5,1,-2,0)	SW5	233
	IF (MODE.EQ.2.OR.XSW) RETURN	SW5	234
265	1231 CALL CALC(M)	SW5	235
	RETURN	SW5	236
	END	SW5	237

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.

1	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
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
10	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
15	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
20		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
25	COMMON /DIM/ LEND, MEND, ISAV, ISAV7, M7END, KADD, KADDR, MUNIT, MSYMP	DIM	2
		DIM	3
	COMMON/BMI/MI(5000)	BMIL	1
	COMMON/NELS/NELS	BMIL	2
	LEVEL 2, MI	BMIL	3
30		BMIL	4
		BMI	3
	COMMON/SWCH/BEND, MSIZE, CYCSWT, VCSW, PV, MSSW, VPR, MHS, MODS, MINZER	SWTCH	2
	LOGICAL CYCSWT, MSSW, VCSW, PV, VPR, MHS, MODS	SWTCH	3
		SWTCH	4
35	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
40	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
45	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
50	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
55	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

	.)	SW6	36
	C LOAD SECOND CARD	SW6	37
60	CALL LOAD(M,	SW6	38
	. 4,1,0,0,0,0,0,0	SW6	39
	.)	SW6	40
	IF (MODE.EQ.2) RETURN	SW6	41
	IF (XSW) RETURN	SW6	42
65	761 CALL BEAM(M)	SW6	43
	RETURN	SW6	44
	C DKE ENTRY. ENERGY GAIN	SW6	45
	77 CONTINUE	SW6	46
	IF (MODE.EQ.3) RETURN	SW6	47
70	CALL INP(M,0,0,	SW6	48
	. 1,1,0,0,0,0,0,0	SW6	49
	.)	SW6	50
	RETURN	SW6	51
	C EDRF ENTRY.	SW6	52
75	78 CONTINUE	SW6	53
	IF (MODE.EQ.3) RETURN	SW6	54
	CALL INP(M,0,0,	SW6	55
	. 1,1,0,0,0,0,0,0	SW6	56
	.)	SW6	57
80	RETURN	SW6	58
	C SOL ENTRY	SW6	59
	79 CONTINUE	SW6	60
	IF (MODE.EQ.3) RETURN	SW6	61
	CALL INP(M,0,0,	SW6	62
85	. 3,1,0,0,0,0,0,0	SW6	63
	.)	SW6	64
	RETURN	SW6	65
	C	SW6	66
		SW6	67
90	* CYEM ENTRY	SW6	68
	83 IF (MODE.EQ.3) GO TO 831	SW6	69
	CALL INP(M,0,4,	SW6	70
	. 1,2,5,1,0,0,0,0	SW6	71
	.)	SW6	72
95	IF (MODE.EQ.2) RETURN	SW6	73
	IF (XSW) RETURN	SW6	74
	831 CALL CYEM(M)	SW6	75
	RETURN	SW6	76
	C BVAL ENTRY	SW6	77
100	84 IF (MODE.EQ.3) GO TO 841	SW6	78
	CALL INP(M,0,4,6,1,0,0,0,0,0,0)	SW6	79
	IF (MODE.EQ.2) RETURN	SW6	80
	IF (XSW) RETURN	SW6	81
	841 CALL BVAL(M)	SW6	82
105	RETURN	SW6	83
		SW6	84
	* CYAE ENTRY	SW6	85
	85 IF (MODE.EQ.3) GO TO 851	SW6	86
	CALL INP(M,0,0,	SW6	87
110	. 2,2,3,1,0,0,0,0	SW6	88
	.)	SW6	89
	IF (MODE.EQ.2) RETURN	SW6	90
	IF (XSW) RETURN	SW6	91
	851 CALL CYAE(M)	SW6	92

115

RETURN
ENDSW6
SW693
94

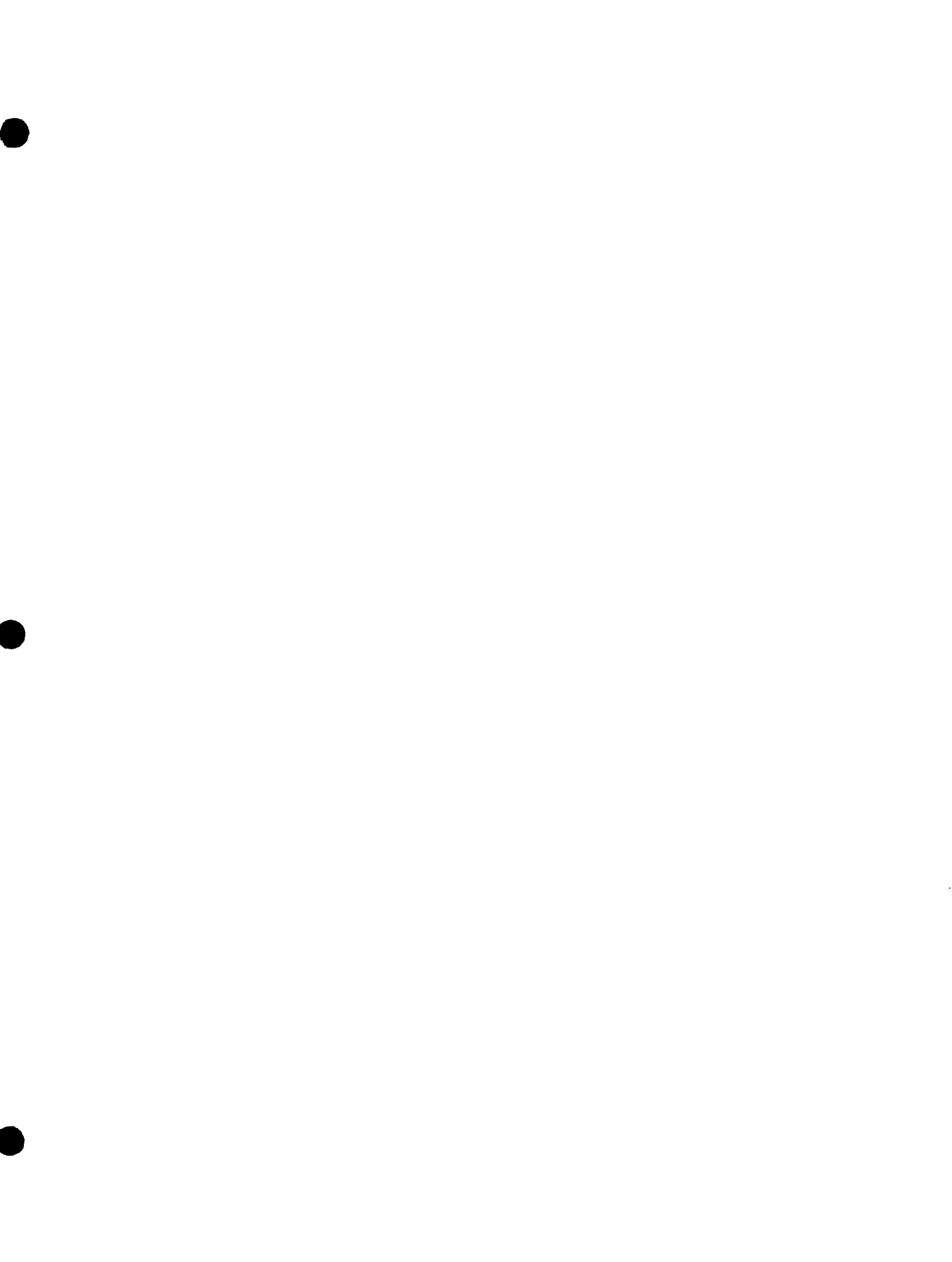
CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

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 ***



1	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
5	F=F+DELTA	AFD	6
	CALL REPFLT(M,J,F)	AFD	7
	RETURN	AFD	8
	END	AFD	9

1	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
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/SWTC/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	SWTCH	2
10	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
15		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
20	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
25	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
30	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/ 13X,6(3X,F12.6))	ALTC	29
	RETURN	ALTC	31
35	END	ALTC	32

1	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
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
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
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
20	ND(4)=N4	ARGINP	21
	I(4)=I4	ARGINP	22
	8 RETURN	ARGINP	23
	END	ARGINP	24

1	SUBROUTINE AROP(M, INAM, IOP)	AROP	2
		BLANK	2
	LEVEL 2, STORE, INFF, IWORK	BLANK	3
	COMMON STORE(48000), IWORK(10)	86MARSIZ	1
5	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
10	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
15	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
20	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
25	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
30	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
35	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
40	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
45	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
50	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
55	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	AROP	51
		DB = - DB	AROP	52
60	9	IF (MINUS.OR.PLUS) DC = DA + DB	AROP	53
		IF (MULT) DC=DA*DB	AROP	54
		IF(DIV) DC = DA/DB	AROP	55
		GO TO 11	AROP	56
	C	OPERATE ON MATRICES. CONSIDER ONLY +,-,* FOR PRESENT	AROP	57
65	10	MA = ELMUM(INAM(1))	AROP	58
		MB = ELMUM(INAM(2))	AROP	59
		IF (.NOT.MULT) GO TO 15	AROP	60
		MI(1) = MA	AROP	61
		MI (2) = MB	AROP	62
70		CALL MMM(M,2,MI)	AROP	63
		RETURN	AROP	64
	15	IF (KN.NE.3HMX) GO TO 30	AROP	65
		CALL RXY(MA,RX,RY,RW)	AROP	66
		CALL RXY(MB,RX1,RY1,RW1)	AROP	67
75		IF (.NOT.MINUS) GO TO 20	AROP	68
		DO 17 J=1,3	AROP	69
		DO 16 I=1,2	AROP	70
		RX1(I,J) = -RX1(I,J)	AROP	71
		RY1(I,J) = - RY1(I,J)	AROP	72
80	16	CONTINUE	AROP	73
		RW1(J) = -RW1(J)	AROP	74
	17	CONTINUE	AROP	75
	20	DO 25 J=1,3	AROP	76
		DO 22 I=1,2	AROP	77
85		RX(I,J) = RX(I,J) + RX1(I,J)	AROP	78
		RY(I,J) = RY(I,J) + RY1(I,J)	AROP	79
	22	CONTINUE	AROP	80
		RW(J) = RW(J) + RW1(J)	AROP	81
	25	CONTINUE	AROP	82
90		CALL STXY(M,RX,RY,RW)	AROP	83
		RETURN	AROP	84
	30	CALL RTRV7(MA,T,RW)	AROP	85
		CALL RTRV7(MB,T1,RW1)	AROP	86
		IF (.NOT.MINUS) GO TO 35	AROP	87
95		DO 31 I=1,49	AROP	88
	31	T1(I)= - T1(I)	AROP	89
		DO 32 I=1,3	AROP	90
	32	RW1(I) = -RW1(I)	AROP	91
	35	DO 37 I=1,49	AROP	92
100	37	T(I) = T(I) + T1(I)	AROP	93
		DO 38 I=1,3	AROP	94
	38	RW(I) = RW(I) + RW1(I)	AROP	95
		CALL STOR7(M,T,RW)	AROP	96
		RETURN	AROP	97
105		END	AROP	98

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

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

1	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
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/MAGL/P(10),DESMU,K	AUXC	6
10	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
15	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

1

SUBROUTINE AUXD(Y,FMUY)
RETURN
END

AUXD 2
AUXD 3
AUXD 4

1 SUBROUTINE AUXF(X,FMUX)
 RETURN
 END

AUXF 2
AUXF 3
AUXF 4

1	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
5	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
10	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
15	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
20	BDAT=IDATA	BDAT	13
	RETURN	BDAT	14
	END	BDAT	15

1 SUBROUTINE BEAM(M)
 RETURN
 END

BEAM 2
BEAM 3
BEAM 4

1								BEND	2
								BEND	3
								BEND	4
								BEND	5
5								BEND	6
								BEND	7
								BEND	8
								BEND	9
								BEND	10
10								BEND	11
								BEND	12
	C B	BEND	L	0.	BRHO	BZ	\$	BEND	13
								BEND	14
								BEND	15
15								DIM	2
								DIM	3
								BENDLL	1
								BEND	17
								BEND	18
20								BEND	19
								BEND	20
								BEND	21
								BEND	22
								BEND	23
25								BEND	24
								BEND	25
								BEND	26
								BEND	27
								BEND	28
30								BEND	29
								BEND	30
								BEND	31
								BEND	32
								BEND	33
35								BEND	34
								BEND	35
								BEND	36
								BEND	37
								BEND	38
40	3	CONTINUE						BEND	39
								BEND	40
								BEND	41
								BEND	42
								BEND	43
45								BEND	44
								BEND	45
								BEND	46
								BEND	47
								BEND	48
50								BEND	49
								BEND	50
								BEND	51
								BEND	52
								BEND	53
55								BEND	54
								BEND	55
								BEND	56

	ARG3 = PIO2 - GAM0	BEND	57
	R(4) = RHO * CEXP(XI*ARG3)	BEND	58
60		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
65	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
70	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
75		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
80	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
85	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
90	V(1) = -X0F	BEND	88
	V(2) = -AL0F	BEND	89
	4 RETURN	BEND	90
		BEND	91
	2 CALL MAGNET(M,PAR)	BEND	92
95	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.

1		BEND	95
	SUBROUTINE BASIS(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
5	*	BESIS	5
	DIMENSION FI(1), PI(600)	BESIS	6
	NMAX = MAX0((NN+1),2)	BESIS	7
	KMAX=NMAX	BESIS	8
	SUM=0.	BESIS	9
10	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
15	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
20	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
25	12 FI(N)=A*PI(N)	BESIS	25
	RETURN	BESIS	26
	END	BESIS	27

1	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
5		BET	6
	COMMON/SWTC/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	SWTC	2
	LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS	SWTC	3
		SWTC	4
	DIMENSION RX(2,3),RY(2,3),BETX(1),BETY(1),RXI(2,3),RYI(2,3)	BET	8
10	DIMENSION RW(3),RWI(3)	BET	9
	DATA PI /3.141592654/	BET	10
		BET	11
	TWOPI = 2.*PI	BET	12
	QINV = 0.	BET	13
15	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
20	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
25	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
30	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
35	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
40	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
45	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
50	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
55	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.

1	SUBROUTINE BMIS	BMIS	2
	C BEGIN MISALIGNMENT	BMIS	3
	COMMON/SWCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	SWCH	2
	LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS	SWCH	3
5		SWCH	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
10		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
15		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
20	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
25	RETURN	BMIS	17
	END	BMIS	18

1	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
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
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
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
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
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
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
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
40		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
45	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
50	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
55	AMPC2R = 1./AMPC2	BVAL	50
	IF (KA.NE.0) GO TO 15	BVAL	51
	C 1ST PARAM IS P	BVAL	52

	P = PAR(1)	BVAL	53
	BETGAM = P*AMPC2R	BVAL	54
50	GAM = SQRT(BETGAM*BETGAM + 1.)	BVAL	55
	TK = AMPC2*(GAM-1.)	BVAL	56
	GO TO 20	BVAL	57
	C 1ST PARAM IS T	BVAL	58
	15 TK = PAR(1)	BVAL	59
65	GAM = TK*AMPC2R + 1.	BVAL	60
	BETGAM = SQRT(GAM*GAM - 1.)	BVAL	61
	P = AMPC2*BETGAM	BVAL	62
	20 BETT = BETGAM/GAM	BVAL	63
	BRHO = P*CONK	BVAL	64
70	IF (KB.NE.0) KNFLAG = .TRUE.	BVAL	65
	WRITE (3,101)	BVAL	66
	CALL DASH	BVAL	67
	WRITE (3,100) P,TK,A,BRHO,BETT,GAM,BETGAM	BVAL	68
	IF ((PAR(3).EQ.0.).AND.(PAR(4).EQ.0.).AND.(PAR(5).EQ.0.).AND.	BVAL	69
75	1 (PAR(6).EQ.0.)) RETURN	BVAL	70
	BETGAMR = 1./BETGAM	BVAL	71
	EPSX = PAR(3)*BETGAMR	BVAL	72
	EPSY = PAR(4)*BETGAMR	BVAL	73
	EPSL = PAR(5)*BETGAMR	BVAL	74
80	GO TO 25	BVAL	75
	C ALREADY UNNORMALIZED CASE	BVAL	76
	22 EPSX = PAR(3)	BVAL	77
	EPSY = PAR(4)	BVAL	78
	EPSL = PAR(5)	BVAL	79
85	25 SIGL = PAR(6)	BVAL	80
	SIGP = EPSL/SIGL	BVAL	81
	CALL STDAT (M,5,1,4,EM)	BVAL	82
	WRITE (3,105) EPSX,EPSY,EPSL,SIGL,SIGP	BVAL	83
	CALL DASH	BVAL	84
90	C	BVAL	85
	100 FORMAT (8X,*P = *,F10.5,* GEV/C*,6X,*T = *,	BVAL	86
	1 F10.5,* GEV*,12X,*A = *,F10.5,6X,*BRHO = *,F10.5, /	BVAL	87
	2 5X,*BETA = *,E15.8,3X,*GAMMA = *,E15.8,3X,	BVAL	88
	3 *BETAGAMMA = *,E15.8)	BVAL	89
95	101 FORMAT (1H)	BVAL	90
	105 FORMAT (/5X,*EMITTANCES UNNORMALIZED (MM-MRAD)* /	BVAL	91
	1 8X,*EPSX = *,F10.6,4X,*EPSY = *,F10.6,4X,*EPSL = *,	BVAL	92
	2 F10.6,10X,*SIGL = *,F10.6,1X,*MM*,4X,*SIGP = *,F10.6,	BVAL	93
	3 * (0/00)*)	BVAL	94
100	RETURN	BVAL	95
	END	BVAL	96

1 SUBROUTINE CADJ6(R,S,DET)
 RETURN
 END

CADJ6 2
CADJ6 3
CADJ6 4

1	SUBROUTINE STOREC(M,I,NREG,NSY)	STOREC	2
		BLANK	2
	LEVEL 2, STORE,INFF,IWORK	BLANK	3
	COMMON STORE(48000),IWORK(10)	86MARSIZ	1
5	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
10	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
15	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
20	END	STOREC	16

1 SUBROUTINE STKLFT(X,Y,Z,T)
 T = Z \$ Z = Y \$ Y = X
 RETURN
 END

STKLFT 2
STKLFT 3
STKLFT 4
STKLFT 5

1	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
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
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
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
20	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
25	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
30	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
35	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
40	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
45	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
50	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
55	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
60     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
65     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
70     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
75     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
80     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
85     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
90     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
95     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
100    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
105    7     GO TO ( 1,12,13, 1,15, 21,22,23,24,25, 1, 1,33,34,35,
          . 41,42,43,44, 1, 51, 1, 1, 1, 1, 61, 1,63,64, 1,
          . 71, 1, 1, 1, 1, 81,82,83, 1, 1, 1,92,93, 1,95,
          . 1,102,103,104,105, 1, 1, 1, 1, 1, 1, 1,1,124, 1,
          . 131, 1, 1, 1, 1, 141, 1, 1, 1, 1, 151, 1, 1, 1, 1,
110    . 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

```

115	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
120	* 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
125	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
130	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
135	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
140	* 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
145	* 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
150	* 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
155	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
160	* *	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
165	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
170	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
175	*	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
180	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
185		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
190	*	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
195	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
200	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
205		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
210		END		CALC	192

1 SUBROUTINE CALFUN(NC,NV,F,X,INIT)
RETURN
END

CALFUN 2
CALFUN 3
CALFUN 4

1 SUBROUTINE CDET5(C,DET)
 RETURN
 END

CDET5 2
CDET5 3
CDET5 4

1

SUBROUTINE CELC(M,PAR)
RETURN
END

CELC 2
CELC 3
CELC 4

1	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
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
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
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
20	COMMON/NELS/NELS	BMIL	2
	LEVEL 2,MI	BMIL	3
		BMIL	4
		BMI	3
	COMMON/SWTC/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	SWTC	2
25	LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS	SWTC	3
		SWTC	4
	DIMENSION MMI(12)	CELLL	1
	DIMENSION PAR(7),P(7)	CELL	22
		CELL	23
30	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
35	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
40	* 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
45	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
50	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
55	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
60	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
65	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
70	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
75	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
80	END	CELL	72

1	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
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
10	C THIS SUBROUTINE GENERATES 17 MATRICES AS FOLLOWS...	CFD	11
	C QF = THE FOCUSING MAGNET = MATRIX M - 13	CFD	12
	C QD = THE DEFOCUSING 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
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
20	C	CFD	21
	COMMON/BMI/MI(5000)	BMIL	1
	COMMON/NELS/NELS	BMIL	2
	LEVEL 2,MI	BMIL	3
		BMIL	4
25		BMI	3
	COMMON/SWTC/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
30	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
35	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
40	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
45	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
50	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
55	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
60	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
65	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
70	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
75	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
80	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
85	CALL STDAT(M,3,1,12,MMI)	CFDL	3
		CFDL	4
	RETURN	CFD	79
	END	CFD	80

1	SUBROUTINE CHKD(NFT,KN,ITP)	CHKD	2
		BLANK	2
	LEVEL 2, STORE,INFF,IWORK	BLANK	3
	COMMON STORE(48000),IWORK(10)	86MARSIZ	1
5	DIMENSION INFF(24,2000)	86MARSIZ	2
	EQUIVALENCE (INFF,STORE)	86MARSIZ	3
		BLANK	5
	COMMON/FLTIN/IFL(15)	FLTIN	2
		FLTIN	3
10	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
15	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
20	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
25	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
30	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
35	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
40	RETURN	CHKD	31
	C FLOATING POINT. SCALAR AND SINGLE	CHKD	32
	25 KN = 4HSCAL	CHKD	33
	ITP = 4HSNGL	CHKD	34
	RETURN	CHKD	35
45	END	CHKD	36

1	SUBROUTINE CHRМ(GK,RI,SS,TAN1, TAN2,BETX,BETY,BXP,BYP,CRX,CRY)	CHRM	2
	DIMENSION BETX(6),BETY(6),BXP(6),BYP(6),	CHRM	3
	1 BX1(6),BY1(6),BX2(6),BY2(6)	CHRM	4
	CRX=0.	CHRM	5
5	CRY=0.	CHRM	6
	IF(GK.EQ.0..AND.RI.EQ.0.) RETURN	CHRM	7
	IF(RI.NE.0.) GO TO 11	CHRM	8
	CRX = (BETX(4)+GK*BETX(2))*SS + BETX(3) - BXP(3)	CHRM	9
	CRY = (BETY(4)-GK*BETY(2))*SS + BETY(3) - BYP(3)	CHRM	10
10	RETURN	CHRM	11
	11 CONTINUE	CHRM	12
	DO 1 I=1,6	CHRM	13
	BX1(I)=BXP(I)	CHRM	14
	BY1(I)=BYP(I)	CHRM	15
15	BX2(I)=BETX(I)	CHRM	16
	1 BY2(I)=BETY(I)	CHRM	17
	BX1(3)=BX1(3)-TAN1*RI*BX1(2)	CHRM	18
	BX2(3)=BX2(3)+TAN2*RI*BX2(2)	CHRM	19
	BY1(3)=BY1(3)+TAN1*RI*BY1(2)	CHRM	20
20	BY2(3)=BY2(3)-TAN2*RI*BY2(2)	CHRM	21
	BX1(6)=BX1(6)+TAN1*RI*BX1(5)	CHRM	22
	BX2(6)=BX2(6)-TAN2*RI*BX2(5)	CHRM	23
	RI2=RI**2	CHRM	24
	GKX=GK+RI2	CHRM	25
25	BKGX = GKX*BX1(2) + (1.+BX1(3)**2)/BX1(2)	CHRM	26
	BKGY = -GK*BY1(2) + (1.+BY1(3)**2)/BY1(2)	CHRM	27
	F2 = (SS*BKGY+BY2(3)-BY1(3))/2.	CHRM	28
	IF (GKX.NE.0.) GO TO 12	CHRM	29
	F5 = SS*(BX1(5)+BX2(5)-SS**2*RI/6.)/2.	CHRM	30
30	F1 = SS*(BX1(2)+BX2(2)+SS*(BX2(3)-BX1(3))/3.) /2.	CHRM	31
	F3 = (BX1(2)+BX2(2))*(BX1(5)+BX2(5))	CHRM	32
	1 + BX1(2)*BX2(5) + BX2(2)*BX1(5))/3.	CHRM	33
	2 + (BX2(3)-BX1(3))*(5.*(BX1(5)+BX2(5))-RI*SS**2)*SS/30.	CHRM	34
	3 - (BX1(2)+BX2(2))*RI*SS**2/12.	CHRM	35
35	F3 = F3*SS/2.	CHRM	36
	GO TO 13	CHRM	37
	12 F1 = (SS*BKGX+BX2(3)-BX1(3))/GKX/2.	CHRM	38
	F5 = (RI*SS-BX2(6)+BX1(6))/GKX	CHRM	39
	F3=(2.*BKGX*F5-RI*F1 + BX2(6)*BX2(2) + 2.*BX2(3)*BX2(5)	CHRM	40
40	1 - BX1(6)*BX1(2) - 2.*BX1(3)*BX1(5))/3./GKX	CHRM	41
	13 CRX = - (GKX+RI2)*F1 + RI*(GK+GK)*F3 + RI*BKGX*F5	CHRM	42
	1 - RI*(BXP(2)*BXP(6)-BETX(2)*BETX(6))	CHRM	43
	CRY = - F2 + RI*BKGY*F5 + RI*(BYP(2)*BXP(6)-BETY(2)*BETX(6))	CHRM	44
	IF (TAN1.EQ.0..AND.TAN2.EQ.0.) GO TO 14	CHRM	45
45	T1 = RI*TAN1	CHRM	46
	T2 = RI*TAN2	CHRM	47
	CRX = CRX + BXP(2)* (T1-TAN1*((GK+GKX)*BXP (5)-T1*BXP (6)))	CHRM	48
	1 + BETX(2)* (T2-TAN2*((GK+GKX)*BETX(5)+T2*BETX(6)))	CHRM	49
	2 +(BXP(3)+BX1(3))*BXP(5)*T1*TAN1 -(BETX(3)+BX2(3))*BETX(5)*T2*TAN2	CHRM	50
50	CRY = CRY + BYP (2)*(-T1+TAN1*(2.*GK*BXP (5)+T1*BXP (6)))	CHRM	51
	1 + BETY(2)*(-T2+TAN2*(2.*GK*BETX(5)-T2*BETX(6)))	CHRM	52
	2 -(BYP(3)+BY1(3))*BXP(5)*T1*TAN1+(BETY(3)+BY2(3))*BETX(5)*T2*TAN2	CHRM	53
	14 CONTINUE	CHRM	54
	CRX= -CRX - CRX	CHRM	55
55	CRY = -CRY - CRY	CHRM	56
	RETURN	CHRM	57
	END	CHRM	58

1 SUBROUTINE CMATX6(A,B,C)
RETURN
END

CMATX6 2
CMATX6 3
CMATX6 4

1	SUBROUTINE CMXDIV(V, IN, IM, DET, IDIM)	CMXDIV	2
	C REPLACES B BY (A INVERSE)*B. DESTROYS A.	CMXDIV	3
	C A IS N BY N, B IS N BY M. A AND B ARE STORED IN V(N,N+M).	CMXDIV	4
	COMPLEX SAVE, PIVOT, DETERM, DET, V, CABS	CMXDIV	5
5	DIMENSION V(IDIM,1)	CMXDIV	6
	INTEGER P	CMXDIV	7
	N=IN	CMXDIV	8
	M=IM	CMXDIV	9
	C	CMXDIV	10
10	NM1=N-1	CMXDIV	11
	NP1=N+1	CMXDIV	12
	NPM=N+M	CMXDIV	13
	DETERM=(1.0E0,0.0E0)	CMXDIV	14
	P=1	CMXDIV	15
15	202 CONTINUE	CMXDIV	16
	IMIN=P+1	CMXDIV	17
	C FIND PIVOT	CMXDIV	18
	RPIVOT=0.0E0	CMXDIV	19
	DO 209 I=P,N	CMXDIV	20
20	RSAVE=CABS(V(I,P))	CMXDIV	21
	IF (RSAVE .LE. RPIVOT) GO TO 209	CMXDIV	22
	RPIVOT=RSAVE	CMXDIV	23
	IBIG=I	CMXDIV	24
	209 CONTINUE	CMXDIV	25
25	IF (RPIVOT .GT. 0.0E0) GO TO 210	CMXDIV	26
	DETERM=(0.0E0,0.0E0)	CMXDIV	27
	GO TO 290	CMXDIV	28
	C ROW EXCHANGE AND ROW REDUCTION	CMXDIV	29
	210 PIVOT=V(IBIG,P)	CMXDIV	30
30	DETERM=DETERM*PIVOT	CMXDIV	31
	DO 219 J=P,NPM	CMXDIV	32
	SAVE=V(IBIG,J)/PIVOT	CMXDIV	33
	V(IBIG,J)=V(P,J)	CMXDIV	34
	V(P,J)=SAVE	CMXDIV	35
35	219 CONTINUE	CMXDIV	36
	IF (IBIG .NE. P) DETERM=-DETERM	CMXDIV	37
	IF (P .GE. N) GO TO 250	CMXDIV	38
	V(P,P) =(1.0E0,0.0E0)	CMXDIV	39
	C BLOCK REDUCTION	CMXDIV	40
40	DO 239 I=IMIN,N	CMXDIV	41
	DO 238 J=IMIN,NPM	CMXDIV	42
	V(I,J)=V(I,J)-V(I,P)*V(P,J)	CMXDIV	43
	238 CONTINUE	CMXDIV	44
	V(I,P)=(0.0E0,0.0E0)	CMXDIV	45
45	239 CONTINUE	CMXDIV	46
	P=P+1	CMXDIV	47
	249 GO TO 202	CMXDIV	48
	C BACK SUBSTITUTION	CMXDIV	49
	250 CONTINUE	CMXDIV	50
50	IF (M .LE. 0) GO TO 290	CMXDIV	51
	DO 259 K=NP1,NPM	CMXDIV	52
	DO 253 P=1,NM1	CMXDIV	53
	I=N-P	CMXDIV	54
	DO 252 J=I,NM1	CMXDIV	55
55	V(I,K)=V(I,K)-V(J+1,K)*V(I,J+1)	CMXDIV	56
	252 CONTINUE	CMXDIV	57
	253 CONTINUE	CMXDIV	58

	259	CONTINUE	CMXDIV	59
	C		CMXDIV	60
60	290	CONTINUE	CMXDIV	61
		DET=DETERM	CMXDIV	62
		RETURN	CMXDIV	63
		END	CMXDIV	64

1

SUBROUTINE CONJ(A,ABAR)
RETURN
END

CONJ 2
CONJ 3
CONJ 4

1

SUBROUTINE CONTYP(K,NB,ICTYP)
RETURN
END

CONTYP 2
CONTYP 3
CONTYP 4

1	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
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
10	EQUIVALENCE(T1,T2)	CONVMAT	8
		CONVMAT	9
	DO 1 I=1,49	CONVMAT	10
	T1(I) = T(I)	CONVMAT	11
	1 CONTINUE	CONVMAT	12
15	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
20	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
25	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
30	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
35	* EXTRACT P1,P2,Q1,Q2	CONVMAT	33
	DO 10 I=1,4	CONVMAT	34
40	10 RPQ(I) = T2(5,I)	CONVMAT	35
	RETURN	CONVMAT	36
	* CONVERT RX,RY OF 3X3 TO T OF 7X7	CONVMAT	37
45	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
50	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
55	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.	CONVMAT	56
	* SET P1,P2,Q1,Q2	CONVMAT	57
60	DO 22 I=1,2	CONVMAT	58
	T2(5,I) = T2(1,6)*T2(2,I) - T2(2,6)*T2(1,I)	CONVMAT	59
	T2(5,I+2) = T2(3,6)*T2(4,I+2) - T2(4,6)*T2(3,I+2)	CONVMAT	60
	RPQ(I) = T2(5,I)	CONVMAT	61
22	RPQ(I+2) = T2(5,I+2)	CONVMAT	62
65	DO 26 I=1,49	CONVMAT	63
	T(I) = T1(I)	CONVMAT	64
26	CONTINUE	CONVMAT	65
	RETURN	CONVMAT	66
		CONVMAT	67
70	* ERROR IF KN IS NOT MXY OR R7	CONVMAT	68
30	WRITE (3,100) KN	CONVMAT	69
100	FORMAT (5H *** ,*ERROR. CANNOT CONVERT MATRIX OF KIND *,I5)	CONVMAT	70
	ERROR = .TRUE.	CONVMAT	71
	RETURN	CONVMAT	72
75	END	CONVMAT	73

1 SUBROUTINE CUBSOL(PP,QQ,RR,Y)
 RETURN
 END

CUBSOL 2
CUBSOL 3
CUBSOL 4

1	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
5		CYAE	6
	C.....	CYAE	7
	* NAME CYAE KA /BL BEAM KSQ EPXCO EPYCO	CYAE	8
	C.....	CYAE	9
	* BL = BEAMLINE NAME	CYAE	10
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 DISPLACEMENT	CYAE	15
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
20	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
25	COMMON/BMI/MI(5000)	BMIL	1
	COMMON/NELS/NELS	BMIL	2
	LEVEL 2, MI	BMIL	3
		BMIL	4
		BMI	3
30	COMMON /DIM/ LEND, MEND, ISAV, ISAV7, M7END, KADD, KADDR, MUNIT, MSYMP	DIM	2
		DIM	3
		CYAE	23
	COMMON /MC/ ML \$ LEVEL 2, ML	CYAE	1
		CYAE	2
35	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
40	DATA IBLNK/1H /	CYAE	29
		CYAE	30
		CYAE	31
	* GET DATA EPX, EPY AND DP/P	CYAE	32
	MD = MDAT(M, 2)	CYAE	33
45	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
50	* 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
55	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)	CYAE	47
	I = 0	CYAE	48
60	EL = 0.	CYAE	49
	FL = 0.	CYAE	50
	MAT = IBLNK	CYAE	51
		CYAE	52
	XKSQ=FDAT(M,1)	CYAE	53
65	RK=SQRT(XKSQ)	CYAE	54
	EPXCO=FDAT(M,2)	CYAE	55
	EPYCO=FDAT(M,3)	CYAE	56
	RX=SQRT(EPXCO/EPX)	CYAE	57
	RY=SQRT(EPYCO/EPY)	CYAE	58
70		CYAE	59
	* PRINT TABLE HEADING AND SIGX AND SIGY, AS DEFINED BY FACTK	CYAE	60
	CALL DASH	CYAE	61
	WRITE(3,1) P(1),EPXCO,P(3),P(2),EPYCO,P(4)	CYAE	62
	IG = 1	CYAE	63
75	IF (KA.EQ.0) IG = 2	CYAE	64
	GO TO (104,105) IG	CYAE	65
	104 WRITE (3,6)	CYAE	66
	GO TO 106	CYAE	67
	105 WRITE (3,8)	CYAE	68
80	106 WRITE(3,2) RK	CYAE	69
	WRITE(3,11)	CYAE	70
		CYAE	71
	115 CONTINUE	CYAE	72
	CALL BET(MS,BETX,BETY,1)	CYAE	73
85	* COMPUTE AND PRINT BEAM ENVELOPES	CYAE	74
	X=SQRT(EPX*BX)*RK	CYAE	75
	Y=SQRT(EPY*BY)*RK	CYAE	76
	XP=DX*DPP	CYAE	77
	YP=DY*DPP	CYAE	78
90	XCO=RX*X	CYAE	79
	YCO=RY*Y	CYAE	80
	XPR=SQRT(EPX*GX)*RK	CYAE	81
	YPR=SQRT(EPY*GY)*RK	CYAE	82
	XPPR=DDX*DPP*RK	CYAE	83
95	YPPR=DDY*DPP*RK	CYAE	84
	XCOPR=RX*XPR	CYAE	85
	YCOPR=RY*YPR	CYAE	86
	GO TO (116,117) IG	CYAE	87
	116 XTOT = X + ABS(XP)	CYAE	88
100	XPRTOT = XPR + ABS(XPPR)	CYAE	89
	YTOT = Y + ABS(YP)	CYAE	90
	YPRTOT = YPR + ABS(YPPR)	CYAE	91
	GO TO 118	CYAE	92
	117 CONTINUE	CYAE	93
105	XTOT = SQRT(X * X + XP * XP)	CYAE	94
	XPRTOT = SQRT(XPR*XPR + XPPR*XPPR)	CYAE	95
	YTOT = SQRT(Y*Y + YP*YP)	CYAE	96
	YPRTOT = SQRT(YPR*YPR + YPPR*YPPR)	CYAE	97
	118 CONTINUE	CYAE	98
110	XTOT=XTOT+XCO	CYAE	99
	YTOT=YTOT+YCO	CYAE	100
	XPRTOT=XPRTOT+XCOPR	CYAE	101
	YPRTOT=YPRTOT+YCOPR	CYAE	102
	EL = EL + FL	CYAE	103

115	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
120	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
125	CALL MMM(MS,3,ML)	CYAE	114
	GO TO 115	CYAE	115
		CYAE	116
	125 CALL DASH	CYAE	117
	RETURN	CYAE	118
130		CYAE	119
	*****	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
135	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
140	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
145	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
150		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

1		SUBROUTINE CYCLE(M,K,MI,L,WS)	CYCLE	2
			CYCLE	3
		COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMP	DIM	2
			DIM	3
5		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
10			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
15	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
20		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
25	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
30	2	CONTINUE	CYCLE	28
		RETURN	CYCLE	29
		END	CYCLE	30

```

1          SUBROUTINE CYEM(M)                                CYEM    2
                                                    CYEM    3
*   CALCULATES ELECTRON INTEGRALS, RF OUTPUT, ELECTRON EMITTANCE FACTORS CYEM    4
                                                    CYEM    5
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
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
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
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
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
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
35 C _____ 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
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
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
50 BLANK 5
COMMON/BMI/MI(5000) BMIL 1
COMMON/NELS/NELS BMIL 2
LEVEL 2,MI BMIL 3
BMIL 4
55 BMI 3
COMMON /DIM/ LEND, MEND, ISAV, ISAV7, M7END, KADD, KADDR, MUNIT, MSYML DIM 2
DIM 3

```

	C		MOVTOG	2
		COMMON/MOVTOG/TOG	MOVTOG	3
60		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
65		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
70		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
75		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
80		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,KN,N,LMTOT,K3R,JE,JX,JZ	CYEM	67
85		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
90		DATA RW/3*0.0/	CYEM	73
			CYEM	74
		CALL RANGET(NRN)	MY3CYEM	4
			CYEM	75
		XM2I = (0.,-2.)	CYEM	76
95	*	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
100		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
105		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
110		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

115		GO TO (51,52) NGO	CYEM	97
			CYEM	98
	52	MFY=MB	CYEM	99
		MB=MDAT(MFY,2)	CYEM	100
		NP=MDAT(MFY,1)	CYEM	101
120		KA = INFF(4,M)	CYEM	102
		SIGFLG = .FALSE.	CYEM	103
		IF (KA.GE.0) GO TO 55	CYEM	104
		SIGFLG = .TRUE.	CYEM	105
		KA = -KA	CYEM	106
125	55	KAF = INFF(4,MFY)	CYEM	107
		LONG = .FALSE.	CYEM	108
		IF(KA.GT.1) LONG = .TRUE.	CYEM	109
		KB = INFF(5,MFY)	CYEM	110
		MS7 = M7END - 3	CYEM	111
130			CYEM	112
	53	CONTINUE	CYEM	113
		CALL DATA(MFY,7,1,49,VZ)	CYEM	114
		DO 60 I=1,4	CYEM	115
		EV(I,1) = CMLPX(VZ(I,2),VZ(I,3))	CYEM	116
135		EV(I,3) = CMLPX(VZ(I,4),VZ(I,5))	CYEM	117
		EV(I,2) = CONJG(EV(I,1))	CYEM	118
	60	EV(I,4) = CONJG(EV(I,3))	CYEM	119
		DETX = EV(1,1)*EV(2,2) - EV(2,1)*EV(1,2)	CYEM	120
		1 + EV(3,1)*EV(4,2) - EV(4,1)*EV(3,2)	CYEM	121
140		DETY = EV(1,3)*EV(2,4) - EV(2,3)*EV(1,4)	CYEM	122
		1 + EV(3,3)*EV(4,4) - EV(4,3)*EV(3,4)	CYEM	123
		CALL STOR7(MS7,VZ,RW)	CYEM	124
		CALL DATA(MFY,7,1,7,VW)	CYEM	125
		CALL RXY(MFY,RX,RY,RW)	CYEM	126
145		CALL STXY(MS,RX,RY,RW)	CYEM	127
		IF (.NOT.FLAG) GO TO 54	CYEM	128
		II = 0	CYEM	129
		CALL DASH	CYEM	130
		IF (KA.EQ.1.OR.KA.EQ.3) GO TO 56	CYEM	131
150		WRITE (3,19)	CYEM	132
		GO TO 57	CYEM	133
	56	WRITE (3,20)	CYEM	134
	57	IF (SIGFLG) GO TO 58	CYEM	135
		WRITE (3,16)	CYEM	136
155		GO TO 503	CYEM	137
	58	WRITE (3,18)	CYEM	138
		GO TO 503	CYEM	139
	54	WRITE (3,14)	CYEM	140
			CYEM	141
160	51	CONTINUE	CYEM	142
		I = INFF(17,MB)	CYEM	143
		CALL MIFILL (MB,1,I,NELS,MI)	CYEM	144
		IF(NGO.EQ.2) GO TO 59	CYEM	145
		CALL MMM(MS,NELS,MI)	CYEM	146
165		IF (KB.GE.0) GO TO 59	CYEM	147
		NELS=NELS+1	CYEM	148
		MI(NELS)=MEND-6	CYEM	149
		CALL REF(MEND-6,MS)	CYEM	150
		ML(1)=MS	CYEM	151
170		ML(2)=MEND-6	CYEM	152
		CALL MMM(MS,2,ML)	CYEM	153

	59	IF(KB.LT.0) KB=-2*KB	CYEM	154
		KW = NELS + 1	CYEM	155
		MI(KW) = MI(1)	CYEM	156
175		BSW = .FALSE.	CYEM	157
		MOVSW = .FALSE.	CYEM	158
		I1 = 0.	CYEM	159
		I2 = 0.	CYEM	160
		I3 = 0.	CYEM	161
180		I4 = 0.	CYEM	162
		I5 = 0.	CYEM	163
		I41 = 0.	CYEM	164
		I43 = 0.	CYEM	165
		I51 = 0.	CYEM	166
185		I53 = 0.	CYEM	167
		II = 0	CYEM	168
		*****	CYEM	169
		*****	CYEM	170
		*****	CYEM	171
190	100	CALL BET(MS,BETX,BETY,1)	CYEM	172
		IF (II.EQ.0) BX0=BETX(2)	CYEM	173
		IF (NGO.EQ.2) GO TO 503	CYEM	174
	101	II = II + 1	CYEM	175
	*	IF I = KW, END OF BEAM LINE	CYEM	176
195		IF (II.GT.KW) GO TO 400	CYEM	177
		NI = MI(II)	CYEM	178
		IF (FLAG) GO TO 121	CYEM	179
	*	IF NOT AT BEGINNING OR END OF A BENDING MAGNET, CONTINUE	CYEM	180
	110	IOP = INFF(1,NI)	CYEM	181
200		IF (BSW) GO TO 300	CYEM	182
	111	IF (IOP.NE.4HMOVE) GO TO 115	MY3CYEM	5
		DO 105 IV=1,7	MY3CYEM	6
	105	VWW(IV) = VW(IV)	MY3CYEM	7
		TOG1 = TOG	MY3CYEM	8
205		CALL RANGET(NRN1)	MY3CYEM	9
		NN1 = M7END - 5	MY3CYEM	10
		CALL MOVE(NI,NN1,VWW)	MY3CYEM	11
		CALL RANSET(NRN1)	MY3CYEM	12
		TOG = TOG1	MY3CYEM	13
210		CALL DATA(NI,1,7,4,HMOV)	CYEM	184
		MOVSW = .TRUE.	CYEM	185
		MQ = MDAT(NI,1)	CYEM	186
		IOP = INFF(1,MQ)	CYEM	187
		IF (IOP.NE.4HMAG) GO TO 120	CYEM	188
215		CALL DATA (MQ,1,1,6,P)	CYEM	189
		GO TO 116	CYEM	190
	115	IF (IOP.NE.4HMAG) GO TO 120	CYEM	191
	*	GET MAGNET DATA. L,K,RHO,OMEGA,GAMMA,DELTA	CYEM	192
		CALL DATA (NI,1,1,6,P)	CYEM	193
220	116	IF ((BZ.NE.0.) .OR. (NGO.EQ.2)) GO TO 200	CYEM	194
	120	CONTINUE	CYEM	195
			CYEM	196
		GO TO (122,121) NGO	CYEM	197
225	121	NF = M7END - 5	CYEM	198
		CALL TRKCN(MS7,NI,VW,NP,0)	CYEM	199
		NI = NF	CYEM	200
		IF (KAF.EQ.0) GO TO 122	CYEM	201
			CYEM	202

	C PRINT AIJ, BIB	CYEM	203
230	900 CONTINUE	CYEM	204
	IPOS = II-1	CYEM	205
	IF(IPOS.NE.0) GO TO 901	CYEM	206
	MATI=1H	CYEM	207
	IF(FLAG) GO TO 903	CYEM	208
235	GO TO 902	CYEM	209
	901 IF (.NOT.LONG) GO TO 122	CYEM	210
	MK=MI(IPOS)	CYEM	211
	MATI=INFF(2,MK)	CYEM	212
	IF(FLAG) GO TO 903	CYEM	213
240	IF(KA.NE.4) GO TO 122	CYEM	214
	902 WRITE(3,15) IPOS,MATI,	CYEM	215
	1 AA(1,1),AA(1,2),AA(2,2),AA(1,3),AA(1,4),	CYEM	216
	2 AA(2,3),AA(2,4),AA(3,3),AA(3,4),AA(4,4),	CYEM	217
	3 BB(1,1),BB(1,2),BB(2,2),BB(1,3),BB(1,4),	CYEM	218
245	4 BB(2,3),BB(2,4),BB(3,3),BB(3,4),BB(4,4)	CYEM	219
	GO TO 122	CYEM	220
	903 WRITE (3,17) IPOS,MATI,	CYEM	221
	1 CC(1,1),CC(1,2),CC(2,2),CC(1,3),CC(1,4),	CYEM	222
	2 CC(2,3),CC(2,4),CC(3,3),CC(3,4),CC(4,4)	CYEM	223
250	IF (.NOT.LONG.OR.KA.EQ.4) GO TO 400	CYEM	224
	GO TO 503	CYEM	225
		CYEM	226
	122 CONTINUE	CYEM	227
	ML(1) = -NI	CYEM	228
255	ML(2) = MS	CYEM	229
	ML(3) = NI	CYEM	230
	CALL MMM(MS,3,ML)	CYEM	231
	GO TO 100	CYEM	232
		CYEM	233
260	*****	CYEM	234
		CYEM	235
	C CALCULATE BILINEAR FORMS — GENERALIZED ELLIPSE MATRICES	CYEM	236
	503 CALL RTRV7(MS7,VZ,RW)	CYEM	237
	C H = GAM*E*E + 2.*A*E*F + B*F*F	CYEM	238
265	ASSIGN 504 TO IRET	CYEM	239
	500 DO 501 I=1,4	CYEM	240
	EV(I,1) = CMPLX(VZ(I,2),VZ(I,3))	CYEM	241
	EV(I,3) = CMPLX(VZ(I,4),VZ(I,5))	CYEM	242
	EV(I,2) = CONJG(EV(I,1))	CYEM	243
270	EV(I,4) = CONJG(EV(I,3))	CYEM	244
	DO 502 J=5,8	CYEM	245
	502 EV(I,J) = CMPLX(0.,0.)	CYEM	246
	501 EV(I,I+4) = CMPLX(1.,0.)	CYEM	247
	600 DO 601 J=1,2	CYEM	248
275	I=2*J-1	CYEM	249
	601 EI(J) = CONJG(VZ(1,6)*EV(2,I) - VZ(2,6)*EV(1,I)	CYEM	250
	1 + VZ(3,6)*EV(4,I) - VZ(4,6)*EV(3,I))	CYEM	251
	EJ1 = EI(1)*EV(1,1)*XM2I/DETX	CYEM	252
	EJ3 = EI(1)*EV(3,1)*XM2I/DETX	CYEM	253
280	EK1 = EI(2)*EV(1,3)*XM2I/DETY	CYEM	254
	EK3 = EI(2)*EV(3,3)*XM2I/DETY	CYEM	255
	CALL CMXDIV(EV,4,4,DET,4)	CYEM	256
	GO TO IRET,(53,504)	CYEM	257
	504 DO 505 I=1,4	CYEM	258
285	DO 505 J=1,4	CYEM	259

	AA(I,J) = 4.0 * REAL(CONJG(EV(1,I+4)) * EV(1,J+4))	CYEM	260
	BB(I,J) = 4.0 * REAL(CONJG(EV(3,I+4)) * EV(3,J+4))	CYEM	261
	IF (.NOT.FLAG) GO TO 505	CYEM	262
	CC(I,J) = REP1*AA(I,J) + REP3*BB(I,J)	CYEM	263
290	UU = 0.	CYEM	264
	IF (J.EQ.I) UU = 1.	CYEM	265
	CC(I,J+4) = UU	CYEM	266
	505 CONTINUE	CYEM	267
		CYEM	268
295	IF (.NOT.FLAG) GO TO 101	CYEM	269
	CALL MXDIV(CC,4,4,DCC,4)	CYEM	270
	IF (.NOT.SIGFLG) GO TO 510	CYEM	271
	DO 511 I=1,4	CYEM	272
	DO 511 J=1,4	CYEM	273
300	511 CC(I,J) = CC(I,J+4)	CYEM	274
	GO TO 101	CYEM	275
	510 DO 506 I=1,4	CYEM	276
	DO 506 J=1,4	CYEM	277
	SPII = 0.	CYEM	278
305	SPJJ = 0.	CYEM	279
	SPIJ = 0.	CYEM	280
	IF(KA.NE.0.AND.KA.NE.3) GO TO 509	CYEM	281
	SPII = VZ(I,6)*VZ(I,6)*SIGPSQ	CYEM	282
	SPJJ = VZ(J,6)*VZ(J,6)*SIGPSQ	CYEM	283
310	SPIJ = VZ(I,6)*VZ(J,6)*SIGPSQ	CYEM	284
	509 CONTINUE	CYEM	285
	SII = CC(I,I+4) + SPII	CYEM	286
	IF (J.EQ.I) GO TO 507	CYEM	287
	SIJ = CC(I,J+4) + SPIJ	CYEM	288
315	SJJ = CC(J,J+4) + SPJJ	CYEM	289
	YY = 2.*SIJ	CYEM	290
	XX = SII - SJJ	CYEM	291
	TIJ = 0.	CYEM	292
	IF (YY.EQ.0..AND.XX.EQ.0.) GO TO 508	CYEM	293
320	TIJ = ATAN2(YY,XX) / 2.	CYEM	294
	508 CC(I,J) = TIJ	CYEM	295
	GO TO 506	CYEM	296
	507 CC(I,I) = SQRT(SII)	CYEM	297
	506 CONTINUE	CYEM	298
325	GO TO 101	CYEM	299
		CYEM	300
	C		
	MAGNET — ENTRANCE	CYEM	301
200	LH = L/2.	CYEM	302
	BR = P(3)	CYEM	303
330	IF (NGO.EQ.2) BR = BR*(1.+VW(6))	CYEM	304
	IF (BZ.EQ.0.) GO TO 262	CYEM	305
	G = BZ/BR	CYEM	306
	RHO = 1./G	CYEM	307
	KX = P(2) + OM*G	CYEM	308
335	KK = SQRT(ABS(KX/P(3)))	CYEM	309
	PHI = KK*L	CYEM	310
	N = - P(2) * P(3) / (OM*OM)	CYEM	311
	G2 = G*G	CYEM	312
	G3 = G*G*G	CYEM	313
340	AG3 = ABS(G3)	CYEM	314
	AG3L = AG3*LH	CYEM	315
	DFAC = G3*(1.-2.*N)	CYEM	316

	EL2G = DFAC*L*L*L*G/24.	CYEM	317
	Q10= DFAC*LH	CYEM	318
345	Q3 = 0.	CYEM	319
	THET = L * ABS(G)	CYEM	320
		CYEM	321
	* E0,E1 ARE ENTRANCE EXIT ANGLES	CYEM	322
	ANG0=P(5)	CYEM	323
350	ANG1=P(6)	CYEM	324
	IF ((IANG0.EQ.1H\$).OR.(IANG1.EQ.1H\$)) GO TO 201	CYEM	325
	E0 = P(5)*RADEG	CYEM	326
	E1 = P(6)*RADEG	CYEM	327
	GO TO 203	CYEM	328
355	201 E0 = .5*THET	CYEM	329
	E1=E0	CYEM	330
	203 CONTINUE	CYEM	331
	TE0 = TAN(E0) \$ TE1 = TAN(E1)	CYEM	332
	GAM =(1. + A*A)/ B	CYEM	333
360		CYEM	334
	IF (E0.EQ.0.) GO TO 202	CYEM	335
	F = F + E*TE0/RHO	CYEM	336
	A = A - B*TE0/RHO	CYEM	337
	GAM = (1. + A*A)/B	CYEM	338
365	202 K3R = 1. / (KK*KK*KK*RHO)	CYEM	339
		CYEM	340
	IF (KX) 230,210,220	CYEM	341
		CYEM	342
	* COMPUTE V. DO U AT END OF MAGNET	CYEM	343
370		CYEM	344
	* FOR KX = 0	CYEM	345
	210 CONTINUE	CYEM	346
	RHO2 = RHO*RHO	CYEM	347
	XL2=L*L	CYEM	348
375	XL3=L*XL2	CYEM	349
	XL4=L*XL3	CYEM	350
	XL5=L*XL4	CYEM	351
	V = (GAM*E*E + 2.*A*E*F + B*F*F) * L	CYEM	352
	1 + (A*E + B*F) * XL2 / RHO	CYEM	353
380	2 + (B/RHO - GAM*E - A*F) * XL3 / (3.*RHO)	CYEM	354
	4 - A * XL4 / (4.*RHO2) + GAM * XL5 / (20.*RHO2)	CYEM	355
	GO TO 250	CYEM	356
		CYEM	357
	* FOR KX.GT.0, F MAGNET	CYEM	358
385	220 CONTINUE	CYEM	359
	T = TAN(.5*PHI)	CYEM	360
	CS = COS(PHI)	CYEM	361
	S = SIN(PHI)	CYEM	362
	SC = S*CS	CYEM	363
390	VC1 = 1. - CS	CYEM	364
	VC2 = PHI - S	CYEM	365
	VC3 = 2.*K3R	CYEM	366
	VC4 = 1./((KK*B)	CYEM	367
	V1 = L* (GAM*E*E + 2.*A*E*F + B*F*F)	CYEM	368
395	V2 = VC3*E* (KK*A*VC1 - GAM*VC2)	CYEM	369
	V3 = VC3*F* (KK*B*VC1 - A*VC2)	CYEM	370
	VC5 = .5*(PHI-SC) -2.*A*VC4*(VC1 - .5*S*S)	CYEM	371
	VC6 = 1.5*PHI - 2.*S + .5*SC	CYEM	372
	V4 = (B*K3R / RHO)* (VC5 + GAM*VC4*VC6/KK)	CYEM	373

400	V = V1 + V2 + V3 + V4	CYEM	374
	GO TO 250	CYEM	375
	* FOR KX.LT.0 D MAGNET	CYEM	376
	230 CONTINUE	CYEM	377
405	EX = EXP(PHI)	CYEM	378
	EXM = EXP(-PHI)	CYEM	379
	SH = .5*(EX - EXM)	CYEM	380
	CH = .5 * (EX + EXM)	CYEM	381
	SC = SH*CH	CYEM	382
410	VC1 = 1. - CH	CYEM	383
	VC2 = PHI - SH	CYEM	384
	VC3 = 2.*K3R	CYEM	385
	VC4 = 1./(KK*B)	CYEM	386
	VC5 = PHI - SC	CYEM	387
415	VC6 = VC1 + .5*SH*SH	CYEM	388
	VC7 = 1.5*PHI - 2.*SH + .5*SC	CYEM	389
	V1 = L* (GAM*E*E + 2.*A*E*F + B*F*F)	CYEM	390
	V2 = VC3*E*(KK*A*VC1 - GAM*VC2)	CYEM	391
	V3 = VC3*F* (KK*B*VC1 - A*VC2)	CYEM	392
420	V4 = (B*K3R/RHO) * (.5*VC5 + 2.*A*VC4*VC6 - GAM*VC4*VC7/KK)	CYEM	393
	V = V1 - V2 - V3 - V4	CYEM	394
	GO TO 250	CYEM	395
	C QUADRUPOLE	CYEM	396
425	262 K = DB/BR	CYEM	397
	EL2G = 0.	CYEM	398
	KSQ = K*K	CYEM	399
	XK2L = KSQ*LH	CYEM	400
	AK3L = ABS(K*KSQ*LH)	CYEM	401
430	TKSQL = 2.*KSQ*LH	CYEM	402
	IF (.NOT.MOVSW) GO TO 265	CYEM	403
	X = HMOV(1)	CYEM	404
	Y = HMOV(2)	CYEM	405
	GO TO 266	CYEM	406
435	250 GO TO (251,260) NGO	CYEM	407
	260 IF (BZ.NE.0.) GO TO 264	CYEM	408
	IF (.NOT.MOVSW) GO TO 265	CYEM	409
	X = HMOV(3)	CYEM	410
440	Y = HMOV(4)	CYEM	411
	GO TO 266	CYEM	412
	265 X = VW(1)	CYEM	413
	Y = VW(3)	CYEM	414
	266 RSQ = X*X + Y*Y	CYEM	415
445	RQ = SQRT(RSQ)	CYEM	416
	G2L = RSQ*XK2L	CYEM	417
	AG3L = RQ*RSQ*AK3L	CYEM	418
	I2 = I2 + G2L	CYEM	419
	I3 = I3 + AG3L	CYEM	420
450	Q1 = TKSQ*Y	CYEM	421
	Q3 = TKSQ*X	CYEM	422
	GO TO 263	CYEM	423
	264 Q1 = Q10 - TE0*G2	CYEM	424
	IF (BSW) Q1 = Q10 - TE1*G2	CYEM	425
455	263 I41 = I41 - Q1*AIMAG(EJ1) - Q3*AIMAG(EJ3) - EL2G	CYEM	426
	I43 = I43 - Q1*AIMAG(EK1) - Q3*AIMAG(EK3)	CYEM	427

	DO 261 I=1,4	CYEM	431
	DO 261 J=1,4	CYEM	432
	I51 = I51 + AA(I,J)*VZ(I,6)*VZ(J,6)*AG3L	CYEM	433
460	261 I53=I53 + BB(I,J)*VZ(I,6)*VZ(J,6)*AG3L	CYEM	434
	IF (BSW) GO TO 302	CYEM	435
		CYEM	436
	251 CONTINUE	CYEM	437
	BSW = .TRUE.	CYEM	438
465	* SAVE E TO COMPUTE U AT END OF MAGNET	CYEM	439
	ESAV = E	CYEM	440
	GO TO 120	CYEM	441
		CYEM	442
	* BENDING MAGNET EXIT. CALCULATE U AND DO SUMMING.	CYEM	443
470	300 CONTINUE	CYEM	444
	GO TO (301, 260) NGO	CYEM	445
	302 IF (OM.EQ.0.) GO TO 550	CYEM	446
	301 IF (KX) 330,310,320	CYEM	447
	310 U = (ESAV + E) * .5*L - (L*L*L)/(12.*RHO)	CYEM	448
475	GO TO 340	CYEM	449
	320 T = TAN(.5*PHI)	CYEM	450
	U = (ESAV + E) *T/KK + K3R*(PHI - 2.*T)	CYEM	451
	GO TO 340	CYEM	452
	330 T = TANH(.5*PHI)	CYEM	453
480	U=(ESAV + E)*T/KK - K3R* (PHI - 2.*T)	CYEM	454
	340 CONTINUE	CYEM	455
	I1 = I1 + G*U	CYEM	456
	I2 = I2 + G*G*L	CYEM	457
	I3 = I3 + AG3*L	CYEM	458
485	TR = (ESAV*TAN(E0) + E*TAN(E1))/(RHO*RHO)	CYEM	459
	I4 = I4 + DFAC*U - TR	CYEM	460
	I5 = I5 + AG3*V	CYEM	461
	550 CONTINUE	CYEM	462
	MOVSW = .FALSE.	CYEM	463
490	BSW = .FALSE.	CYEM	464
	* GO BACK TO FIND OUT IF THIS ALSO BEGINS A BENDING MAGNET.	CYEM	465
	GO TO 111	MY3CYEM	14
		CYEM	467
	*****	CYEM	468
495		CYEM	469
	* RF OUTPUT	CYEM	470
	400 CONTINUE	CYEM	471
	CALL RANSET(NRN)	MY3CYEM	15
	IF (NGO.EQ.1) GO TO 561	CYEM	472
500	IF (FLAG) GO TO 563	CYEM	473
	PC = PC*(1.+VW(6))	CYEM	474
	FLAG = .TRUE.	CYEM	475
	561 CONTINUE	CYEM	476
	I1 = I1 * KB	CYEM	477
505	I2 = I2 * KB	CYEM	478
	I3 = I3 * KB	CYEM	479
	I4 = I4 * KB	CYEM	480
	I5 = I5 * KB	CYEM	481
	GO TO (552,553) NGO	CYEM	482
510	553 I51 = I51 * KB	CYEM	483
	I53 = I53 * KB	CYEM	484
	I41 = I41*KB	CYEM	485
	I43 = I43*KB	CYEM	486

	D1 = I41/I2	CYEM	487
515	D3 = I43/I2	CYEM	488
	J1 = 1.0 - D1	CYEM	489
	J3 = 1.0 - D3	CYEM	490
	J6 = 2.0 + D1 + D3	CYEM	491
552	CONTINUE	CYEM	492
520	PI2 = 2.*PI	CYEM	493
	LMTOT = PI2*RHO	CYEM	494
	EL = PI2*R	CYEM	495
	PC4 = PC**4	CYEM	496
	ALPHA = I1/EL	CYEM	497
525	GAMT = (1./ALPHA)	CYEM	498
	GAMT = CSQRT(GAMT)	CYEM	499
	U0 = CGAM*PC4*I2/PI2	CYEM	500
	UM = U0*1.E03	CYEM	501
	D = I4 /I2	CYEM	502
530	JX = 1.0 - D	CYEM	503
	JE = 2.0 + D	CYEM	504
	T0 = EL/C	CYEM	505
	TAU0 = 4.*PI *T0 /(CGAM*PC**3)	CYEM	506
	TAUY = TAU0/I2	CYEM	507
535	TAUX = TAU0 / (I2 - I4)	CYEM	508
	TAUE = TAU0 /(2.*I2 + I4)	CYEM	509
	SQSIGE= ABS(CQP*PC4*I3/(JE*I2))	CYEM	510
	SIGE = SQRT(SQSIGE)	CYEM	511
	FK = FREQ*T0	CYEM	512
540	IF (V0.LT.U0) V0 = U0/.86603	CYEM	513
	CPHIS = U0/V0	CYEM	514
	PHIS = ACOS(CPHIS)	CYEM	515
	TE = SQRT(PI2*PC*T0*T0 / (ALPHA*FK*V0*SIN(PHIS)))	CYEM	516
	SQE = 2.*U0*(TAN(PHIS) - PHIS)/(PI*FK*ALPHA*PC)	CYEM	517
545	ASQE = ABS(SQE)	CYEM	518
	EPSM = PC*SQRT(ASQE)	CYEM	519
	PSI = (EPSM/SIGE)**2/2.	CYEM	520
	IF (PSI.LE.740.3) GO TO 556	CYEM	521
	PSITRU=PSI	CYEM	522
550	PSI=740.3	CYEM	523
	WRITE (3,1) PSITRU,PSI	CYEM	524
556	CONTINUE	CYEM	525
	TAUQ =(TAUE/2.)*EXP(PSI) /PSI	CYEM	526
	SIGPHI = SIGE*SQRT (PI2*FK*ALPHA / (V0*SIN(PHIS)*PC))	CYEM	527
555	SIGL = SIGPHI*R/FK	CYEM	528
	PHISD = PHIS*180./PI	CYEM	529
		CYEM	530
*	COMPUTE EPSILONS FOR EMITTANCE AND STORE	CYEM	531
	BETX(2)=BX0	CYEM	532
560	IF (JX.GT.0.) GO TO 574	CYEM	533
	SIGXOB = 0.	CYEM	534
	WRITE (3,21)	CYEM	535
	GO TO 575	CYEM	536
574	CONTINUE	CYEM	537
565	SIGXOB = SQRT (CQP*PC*PC*I5 / (JX*I2))	CYEM	538
575	CONTINUE	CYEM	539
	SIGX = SIGXOB * SQRT(BETX(2))	CYEM	540
	EPX0 = SIGX*SIGX/BETX(2)	CYEM	541
	IF (FLAG.AND.FKAP.EQ.0.0) FKAP = 1.E-08	CYEM	542
570	EPX = (1. - FKAP*FKAP)*EPX0	CYEM	543

	EPY = FKAP*FKAP*EPX0	CYEM	544
	GO TO (554,555)NGO	CYEM	545
555	CK = CQP*PC*PC/I2	CYEM	546
	EP1 = CK*I51/J1	CYEM	547
575	EP3 = CK*I53/J3	CYEM	548
	EP3S = EP3	CYEM	549
	IF(EP3.EQ.0.)EP3S=EPY	CYEM	550
	REP1=1./EP1	CYEM	551
	REP3=1./EP3S	CYEM	552
580	554 CONTINUE	CYEM	553
	SIGBG = SIGE/RME	CYEM	554
	EPLN = SIGBG * SIGL	CYEM	555
	BETGAM = PC/RME	CYEM	556
	EPL = EPLN/BETGAM	CYEM	557
585	C SIGY = SQRT(EPY*BETY(2))	CYEM	558
	SIGDP = SIGE/PC	CYEM	559
	SIGPSQ = SIGDP*SIGDP	CYEM	560
	C DPP = SIGDP	CYEM	561
	EPXN = EPX * BETGAM	CYEM	562
590	EPYN = EPY * BETGAM	CYEM	563
		CYEM	564
	C SCALE AND STORE	CYEM	565
	EPSX = EPX*1.E06	CYEM	566
	EPSY = EPY*1.E06	CYEM	567
595	EPSL = EPL*1.E06	CYEM	568
	SIGP = SIGDP*1.E03	CYEM	569
	CALL STDAT(M,5,1,4,EM)	CYEM	570
	IF (NGO.EQ.1) GO TO 562	CYEM	571
	IF ((J1.GE.0.).AND.(J3.GE.0.).AND.(J6.GE.0.)) GO TO 562	CYEM	572
600	WRITE (3,21)	CYEM	573
	FLAG = .FALSE.	CYEM	574
		CYEM	575
	562 IF (FLAG) GO TO 53	CYEM	576
	563 CONTINUE	CYEM	577
605		CYEM	578
	CALL DASH	CYEM	579
	WRITE (3,2) PC,GAMT,R,UM	CYEM	580
	WRITE (3,4) FMHZ,T0,V0MV,TAUX	CYEM	581
	WRITE (3,6) FKAP,TAUY,TAUE	CYEM	582
610	WRITE (3,3) D,SIGX,PHISD,TAUQ	CYEM	583
	WRITE (3,5) TE,SIGPHI,EPSM,SIGL	CYEM	584
	WRITE (3,7) XI,SIGDP	CYEM	585
	WRITE (3,11) SIGE,SIGBG	CYEM	586
	WRITE (3,10) FK,BETGAM,ALPHA,BX0	CYEM	587
615	WRITE(3,9)I1,I2,I3,I4,I41,I43,I5,I51,I53	CYEM	588
	WRITE (3,12) JX,JZ,JE	CYEM	589
	IF (NGO.EQ.2) WRITE (3,22) J1,J3,J6	CYEM	590
	WRITE (3,8) EPXN,EPYN,EPLN,EPX,EPY,EPL,EPX0	CYEM	591
	IF (NGO.EQ.2) WRITE (3,23) EP1,EP3	CYEM	592
620	CALL DASH	CYEM	593
		CYEM	594
	*****	CYEM	595
	* FORMATS	CYEM	596
	1 FORMAT (1X,*PSI = *E13.5* — TAUQ IS GREATER THAN VALUE PRINTED*	CYEM	597
625	1 * BELOW, WHICH CORRESPONDS TO PSI = *E13.5)	CYEM	598
	2 FORMAT (* MOMENTUM*,25X,*P=*,E12.5,* GEV/C*,11X,*TRANSITION *,	CYEM	599
	1*GAMMA*,10X,*GAMT=(*,E12.5,*,*,E12.5,*)*/	CYEM	600

```

      . * AVERAGE RADIUS*,17X,* R=*,E12.5, CYEM 601
      2 * METERS*,9X,* SYNCHROTRON RADIATION*,7X,*U0=*,E12.5,* MEV/TURN*) CYEM 602
630 3  FORMAT (* DAMPING PARTITION*16X*D=*E12.5,16X, CYEM 603
      1 * SIGMAX(0-COUPILING)*,7X,*SIGX0=*,E12.5,* METERS*, CYEM 604
      A/,* PHASE ANGLE*,19X, CYEM 605
      2*PHIS=*,E12.5,* DEGREES*,8X,* QUANTUM LIFETIME*,10X,*TAUQ=*,E12.5, CYEM 606
      3 * SECONDS*) CYEM 607
635 4  FORMAT (* R F FREQUENCY*,18X,* F=*,E12.5,* MHZ*,12X,* CIRCULATION CYEM 608
      1PERIOD*,10X,*T0=*,E12.5,* SECONDS*,/,* R F PEAK VOLTS*,18X,*V0=*, CYEM 609
      2E12.5,* MV*,14X,*HORIZONTAL DAMPING*,8X,*TAUX=*,E12.5,* SECONDS*) CYEM 610
640 5  FORMAT (* ENERGY OSCILLATION PERIOD*,7X,*TE=*,E12.5,* SECONDS*, CYEM 611
      1 9X,*PHASE WIDTH*,13X,*SIGPHI=*,E12.5,* RADIANS*/ CYEM 612
      2* ENERGY APERTURE*,15X,*EPSM=*,E12.5,* GEV *,10X, CYEM 613
      3* BUNCH LENGTH*,14X,*SIGL=*,E12.5,* METERS*) CYEM 614
645 6  FORMAT (* COUPLING CONSTANT*,16X,*K=*,E12.5,17X,*VERTICAL * CYEM 615
      1 *DAMPING*,10X,*TAUY=*,E12.5,* SECONDS*/ CYEM 616
      2 65X,*ENERGY DAMPING*,12X,*TAUE=*,E12.5,* SECONDS*) CYEM 617
645 7  FORMAT(* (EPSM*,2H**,*/SIGE*,2H**,*/2)/2*,11X,* XI=*,E12.5,17X, CYEM 618
      2 *SIGMA(DP/P)*,14X,*SIGDP=*,E12.5) CYEM 619
650 9  FORMAT(/* I1=*,E12.5,/,* I2=*,E12.5,/,* I3=*,E12.5,/,* I4=*,E12.5, CYEM 620
      A 14X*I41 = *E12.5,14X,*I43 = *E12.5, CYEM 621
      1 /,*, I5=*,E12.5,14X,*I51 = *,E12.5,14X,*I53 = *E12.5 ) CYEM 622
      11 FORMAT (* ENERGY SPREAD*,17X,*SIGE=*,E12.5,* GEV*,13X,*MOMENTUM *, CYEM 623
      1*SPREAD*,8X,*SIGBGAM=*,E12.5) CYEM 624
      12 FORMAT (/* JX = *,F12.5,14X,*JY = *,F12.5,14X,*JE = *,F12.5) CYEM 625
      22 FORMAT ( CYEM 626
      1 * J1 = *F12.5,14X,*J3 = *F12.5,14X,*J6 = *F12.5) CYEM 627
655 10 FORMAT (* HARMONIC NUMBER*,18X,*H=*,E12.5,17X,*MOMENTUM*,13X, CYEM 628
      1 *BETAGAMMA=*,E12.5/ CYEM 629
      2 * MOMENTUM COMPACTION*,10X,*ALPHA=*,E12.5,17X,*BETA FUNCTION*, CYEM 630
      3 11X,*BETAX0=*,E12.5) CYEM 631
660 8  FORMAT (/10X,*EMITTANCES (METER RADIANS)*10X,*EPXN = *,E12.5, CYEM 632
      1 7X,*EPYN = *,E12.5,7X,*EPLN = *,E12.5/10X,* EPX = *,E12.5,7X, CYEM 633
      2 * EPY = *,E12.5,8X,*EPL = *,E12.5,7X,*EPX0 = *,E12.5) CYEM 634
      23 FORMAT (11X,*EP1 = *,E12.5,7X,* EP3 = *,E12.5/) CYEM 635
      13 FORMAT (1X,*CYEM*/) CYEM 636
665 14 FORMAT(2X*POS*11X*A11*9X*A12*9X*A22*9X*A13*9X*A14*9X*A23*9X*A24* CYEM 637
      1 9X*A33*9X*A34*9X*A44*/ CYEM 638
      2 16X*B11*9X*B12*9X*B22*9X*B13*9X*B14*9X*B23*9X*B24*9X*B33*9X*B34* CYEM 639
      3 9X*B44*/ ) CYEM 640
      15 FORMAT(/1X,I4,1X,A5,1X,10F12.7/ 12X,10F12.7) CYEM 641
670 16 FORMAT (1X*BEAM ELLIPSE MATRIX — SIJ = SIGMA(I,J),* CYEM 642
      1* RII = SQRT(SII), TIJ = ATAN( 2.SIJ / (SII-SJJ) ) / 2 * / CYEM 643
      2 2X*POS*11X*R11*9X*T12*9X*R22*9X*T13* CYEM 644
      3 9X*T14*9X*T23*9X*T24*9X*R33*9X*T34*9X*R44*/ ) CYEM 645
      18 FORMAT (1X*BEAM ELLIPSE MATRIX — SIJ = SIGMA(I,J) * / CYEM 646
      1 2X*POS*11X*S11*9X*S12*9X*S22*9X*S13* CYEM 647
      2 9X*S14*9X*S23*9X*S24*9X*S33*9X*S34*9X*S44*/ ) CYEM 648
675 17 FORMAT (1X,I4,1X,A5,1X,10E12.4) CYEM 649
      19 FORMAT(1X* (BETATRON PART OF SIGMA MATRIX) * ) CYEM 650
      20 FORMAT(1X* (TOTAL SIGMA MATRIX) * ) CYEM 651
      21 FORMAT (1X,*NEGATIVE DAMPING CONSTANT*) CYEM 652
680 RETURN CYEM 653
      END CYEM 654

```


1	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
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 TAPES OPTION IF GT 0.	CYX	9
		CYX	10
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
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
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
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
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
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
40	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
45	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
50	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
55	COMMON/NELS1/NELS1	BMI1L	2
	LEVEL 2,MI1	BMI1L	3
		BMI1L	4

	C		BMI1	3
		DIMENSION MI(1)	CYX	42
60		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
65		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
70		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
75		EQUIVALENCE (GL,TAB)	CYX	51
		EQUIVALENCE (IX1,XI1),(IX2,XI2)	CYX	52
		COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYML	DIM	2
			DIM	3
		COMMON/SWTC/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	SWTC	2
80		LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS	SWTC	3
			SWTC	4
		COMMON/KINET/KNFLAG,TK,P,GAM,BETT,BETGAM,BRHO	KINET	2
		LOGICAL KNFLAG	KINET	3
		COMPLEX TGAM,TGAM2	CYX	56
85		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
90		EQUIVALENCE (GL(4),XNUX),(GL(5),YNUY),(GL(8),CHRX),	CYX	61
	1	(GL(9),CHRY),(GL(10),ALPHA),(GL(1),BXMN),	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
95	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
100			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
105			CYX	75
		PI2=8.*ATAN(1.)	CYX	76
		RADEG=PI2/360.	CYX	77
		FLG = .TRUE.	CYX	78
		SKIP = .FALSE.	CYX	79
110		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

115		MVF = .FALSE.	CYX	83
		BTABL = .FALSE.	CYX	84
		IF(NELS.GT.44) PFLAG = .TRUE.	JAN86CYX	1
		ITP=0	CYX	86
		KS=0	CYXMAY85	2
120		ITAPE=0	NOV3CYX	6
		CALL RANGET(NRN)	CYX	87
		LNUM=0	CYX	88
			CYX	89
		IF(ERROR) RETURN	CYX	90
125	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
130	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
135	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
140		IA=0	CYX	106
		EL=0.	CYX	107
		THETH = 0.	CYX	108
		THETV = 0.	CYX	109
		KK=K	CYX	110
145		ISK=0	CYX	111
		NK=0	CYX	112
		ETAST = 0.	CYX	113
	C	FKK = KK	CYX	114
		PR = PROD	CYX	115
150		NGO = 1	CYX	116
		YCO = DXCO = DYCO = 0.	CYX620	1
		XCO = 0.	CYX	117
		NMVAR = HZEQ	CYX	118
		IF (KNFLAG) NMVAR = HTEQ	CYX	119
155		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
160		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
165		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
170		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
175	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
180	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
185	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
190	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
195	* 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
200	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
205	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
210	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
215	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
220	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
225	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
230	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
235	EXP0=VP(2)	CYX	192
	EY0=VP(3)	CYX	193
	EYP0=VP(4)	CYX	194
	MN = MNAME(PROD)	CYX	195
	ITP = 0	CYX	196
240	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
245	DELX = 0.	CYX	202
	DELY = 0.	CYX	203
	I = 1	CYX	204
	SLENI=0.	CYX	205
	THETI=0.	CYX	206
250	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
255	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
260	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
265	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
270	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
275	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
280	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
285		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
290		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
295		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
300		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
305	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
310	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
315		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
320		DYCO = -YP0*FAC	CYX	272
		EX=EX0	CYX	273
		EXP=-EXP0	CYX	274
		EY=EY0	CYX	275
		EYP=-EYP0	CYX	276
325		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
330		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
335		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
340	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
345	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
350	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
355	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
360	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
365	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
370	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
375	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
380		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
385	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
390	IF (.NOT.BTABL) GO TO 573	CYX	347
	STORE(LINE) = PSIX	CYX	348
	STORE(LINE+6) = PSII	CYX	349
	STORE(LINE+12) = EL	CYX	350
	STORE(LINE+13) = THETH	CYX	351
395	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

400	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
405	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
410	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
415	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
420	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
425	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
430	CALL MAXMIN (BXXM,BXXN,BETX(2),MBX,IPOS)	CYX	373
	CALL MAXMIN (BYMX,BYMN,BETX(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
435	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
440		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
445	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
450	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
455	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
460	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
465	WRITE (5) KK, IPOS, MATI, SLENI, ZE, ZE, THETI, KV, DKV, ZE,	CYX	408
	1 BETY(2), BETX(2), BETY(3), BETX(3), PSII, 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
470	WRITE(3,2) KK, IPOS, MATI, BETY(2), BETX(2), BETY(3), BETX(3),	CYXJUL84	8
	1 PSII, 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
475	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
480	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
485	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
490	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
495	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
500	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
505	ZZ=0.	CYX	447
	IF (NGO.EQ.2) ZZ = YCO/FAC	CYX	448
	WRITE (11) ELL, PSII, BETY(2), RHOI, GK, XI1, XI2, ZZ	CYX	449
	* COMPUTE LENGTH	CYX	450
55	EL = EL + RW(1)	CYX	451
510	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
515	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
520	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
525	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
530	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
535	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
540		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
545	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
550	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
555	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
560	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
565	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
570	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
575	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
580	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
585	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
590	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
595	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
600	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
605	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
610	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
615	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
620	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
625	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
630		PR=PR-1	CYX	535
		GO TO 50	CYX	536
	*	COMPLETES LOOP FROM STATEMENTS 50 TO 90	CYX	537
		*****	CYX	538
635	90	CONTINUE	CYX	539
		CALL RANSET(NRN)	CYX	540
		IF (SKIP) GO TO 210	CYX	541
		IF(.NOT.PLOT) GO TO 91	CYX	542
		WRITE(12) (XX(J),Y1(J),Y2(J),Y3(J),Y4(J),J=1,KW)	CYX	543
640		WRITE(12) BXX, BYX	CYX583	2
		BXX = BXX	CYX583	3
		BYX = BYX	CYX	546
	91	CONTINUE	CYX	547
		TKB = IABS(KB)	CYX	548
645		IF(KB.EQ.0) TKB=1.	CYX	549
		XNUX=TKB*PSIX	CYX	550
		YNUY=TKB*PSIY	CYX	551
		CALL RXY(IR,RX,RY,RW)	CYX	552
		THETH = THETH * TKB	CYX	553
650		THETV=THETH*TKB	CYX	554
		RLEN=ELL/PI2	CYX	555
		RLEN = RLEN * TKB	CYX	556
		CIRC = ELL*TKB	CYX	557
		STOTI = 1./ELL	CYX	558
655		ETASTOT = ETAST*STOTI	CYX	559
		ALPHA = -VP(5)*STOTI	CYX	560
	C	IPP.GT.0 PRINTS ONLY TABULAR VALUES.	CYX	561
		IF (IPP.GT.0) GO TO 100	CYX	562
		IF (.NOT.NOPR) WRITE (3,1006)	CYX	563
660		CHM =-TKB*0.25/PI2	CYX	564
		IF (KB.LT.0) CHM = 2.*CHM	CYX	565
		CHRX = CHM*CHX	CYX	566
		CHRY = CHM*CHY	CYX	567
		IF(.NOT.CHFLG) GO TO 100	CYX	568
665	C	CALCULATE CHROM CORRECTIONS	NOV3CYX	26
		DT=ABX(1,1)*ABX(2,2)-ABX(1,2)*ABX(2,1)	NOV3CYX	27
		ABXI(1,1)=ABX(2,2)/DT \$ ABXI(1,2)=-ABX(1,2)/DT	NOV3CYX	28
		ABXI(2,2)=ABX(1,1)/DT \$ ABXI(2,1)=-ABX(2,1)/DT	NOV3CYX	29
		SPN=IABS(KB)	NOV3CYX	30
670		IF (KB.EQ.0)SPN=1.	NOV3CYX	31
		IF(KB.LT.0) SPN=SPN+SPN	NOV3CYX	32
		CON=2.*PI2/SPN	NOV3CYX	33
		B(1)=CON*(XI(1)-CHRX)	NOV3CYX	34
		B(2)=CON*(CHRY-XI(2))	NOV3CYX	35
675		DO 99 I=1,2	NOV3CYX	36
		S(I)=0.	NOV3CYX	37
		DO 99 J=1,2	NOV3CYX	38
	99	S(I)=S(I)+ABXI(I,J)*B(J)	NOV3CYX	39
	C		NOV3CYX	40
680	C	APPLY CHROM CORRECTION TO SECOND SEXT PARAMETER	NOV3CYX	41
		DO 95 I=1,2	NOV3CYX	42
		MSFD(I)=ELNUM(ISFD(I))	NOV3CYX	43
		MSS=MSFD(I) \$ FF=FDAT(MSS,1)	NOV3CYX	44
		IF (FF.EQ.0.) FF= 1.	APR84CYX	1
			APR84CYX	2

685	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
690	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
695	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
700	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
705	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
710	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),BXXM,MBY(1),BYMX,MXQ(1),XQMX,MYQ(1),	CYX	595
715	1 YQMX	CYX	596
	LABEL = HMIN	CYX	597
	WRITE (3,1007) LABEL,MBX(2),BXXM,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
720	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
725	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
730	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
735	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
740	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
745	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
750	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
755	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
760	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
765	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
770	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
775	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
780	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
785	WRITE (3,1010) EL	CYX	629
	GO TO 210	CYX	630
		CYX	631
	*****	CYX	632
	* FORMATS	CYX	633
790	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
795	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(1X14,1XA5,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
800		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
805		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*THEY = *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
810		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
815		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
820			2(MR)*)		CYXF 6
		1021	FORMAT(1X I4,1X A5,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
825		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
830		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.

1

 SUBROUTINE DASH

CYX 680

 C WRITES ONE LINE OF DASHES

DASH 2

 WRITE (3,2)

DASH 3

5

 2 FORMAT (1X,131(1H--))

DASH 4

 RETURN

DASH 5

 END

DASH 6

DASH 7

1	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
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
10	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
15		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
20	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
25	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
30	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
35	END	DATA	30

1		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
5		LOGICAL IERR,BSW,VSW,RSW,QSW,XSW	GRR	3
			GRR	4
		COMMON/SWCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	SWCH	2
		LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS	SWCH	3
			SWCH	4
10	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
15	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
20		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
25		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
30		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
35		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

1	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
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
10	LEVEL 2,MI	BMIL	3
		BMIL	4
		BMI	3
	COMMON/SWTC/H/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	SWTCH	2
	LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS	SWTCH	3
15		SWTCH	4
	DIMENSION LIST (100)	DEFSETL	1
	DIMENSION MXLIST(1)	DEFSET	13
		DEFSET	14
	IF (KC.EQ.0) GO TO 536	DEFSET	15
20	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
25	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
30	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
35	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
40	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
45	292 CONTINUE	DEFSET	39
	IF (IOP.EQ.I1) RETURN	DEFSET	40
		DEFSET	41
	* RESERVE FOR DRIFTS	DEFSET	42
	I1 = I1 + 1	DEFSET	43
50	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
55	END	DEFSET	49

1	SUBROUTINE DELQ(S,W,DW)	DELQ	2
		BLANK	2
	LEVEL 2, STORE, INFF, IWORK	BLANK	3
	COMMON STORE(48000), IWORK(10)	86MARSIZ	1
5	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
10	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
15	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
20	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
25	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
30	RETURN	DELQ	25
	END	DELQ	26

1		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
5		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
10			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
15		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
20		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
25	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
30		DPY = DW(4)	DER1	30
		UY=0.	DER1	31
		UZ = A0*Z	DER1	32
			DER1	33
		DO 3 N=1,NH	DER1	34
35		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
40		EI=1./E	DER1	40
		CH=(E+EI)/2.	DER1	41
		SH=(E-EI)/2.	DER1	42
		UZ = UZ + AB*CH/XK	DER1	43
45	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
50		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
55		DW(1) = DX	DER1	55
		DW(3) = DY	DER1	56
		DW(4) = DPY	DER1	57

DER1	58
DER1	59
DER1	60

60

RETURN
END

1		DER1	61
	SUBROUTINE DER2(S,W,DW)	DER2	2
	C	DER2	3
	C ENVELOPES AND DISPERSIONS OF IMPLoding BUNCH 11/79	DER2	4
5	C L0 = TOTAL LENGTH OF SYSTEM	DER2	5
	C ZT = BUNCH HALF LENGTH AT TARGET	DER2	6
	C DELT = (DP/P) HALF WIDTH OF TARGET	DER2	7
	C QT = 4*ZSQ*RP*N/(A*BETASQ*GAMACUBE) AT TARGET	DER2	8
	C EPS = TRANSVERSE EMITTANCE/PI — NOT NORMALIZED	DER2	9
10	C E = ETA = DISPERSION FOR BUNCH CENTER	DER2	10
	C E0 = ETA = DISPERSION FOR BUNCH END	DER2	11
	C	DER2	12
	COMMON/CDERIV/IFLAG,MR,NR,DP,NV(11),DAT(20),UZ,STOT,	DER2	13
	1 DAT2(7),MP2FLG	DER2	14
15	DIMENSION W(8,1),DW(8,1)	DER2	15
	EQUIVALENCE (LMAG,DAT(1)),(GRAD,DAT(2)),(BRHO,DAT(3)),	DER2	16
	1 (BZ, DAT(4))	DER2	17
	EQUIVALENCE (EPS, DAT2(1)),(QT, DAT2(2)),(ZT, DAT2(3)),	DER2	18
	1 (DELT,DAT2(4)),(GAM, DAT2(5)),(L0, DAT2(6))	DER2	19
20	REAL L,L0,K,KBAR,LMAG	DER2	20
	C	DER2	21
	IF (IFLAG) 1,1,2	DER2	22
	1 K = GRAD/BRHO	DER2	23
	RHOI = BZ/BRHO	DER2	24
25	ZTG2 = ZT*GAM*GAM	DER2	25
	QTDT = QT/DELT	DER2	26
	EPS2 = EPS*EPS	DER2	27
	RETURN	DER2	28
	C	DER2	29
30	2 AX = W(1,1)	DER2	30
	AY = W(3,1)	DER2	31
	E = W(5,1)	DER2	32
	E0 = W(7,1)	DER2	33
	L = L0 - S - STOT	DER2	34
35	IF (L.LT.ZT.OR.L0.LT.0.) L = ZT	DER2	35
	DEL0 = ZTG2/L	DER2	36
	Q = QTDT*DEL0	DER2	37
	AP = E*DEL0	DER2	38
	AAX = AX + AP	DER2	39
40	AX3 = AX*AX*AX	DER2	40
	AY3 = AY*AY*AY	DER2	41
	A = AAX + AY	DER2	42
	QA = Q/A	DER2	43
	QAXA = QA/AX	DER2	44
45	EPSAX = EPS2/AX3	DER2	45
	EPSAY = EPS2/AY3	DER2	46
	KBAR = K - QAXA	DER2	47
	C	DER2	48
	DW(1,1) = W(2,1)	DER2	49
50	DW(3,1) = W(4,1)	DER2	50
	DW(5,1) = W(6,1)	DER2	51
	DW(7,1) = W(8,1)	DER2	52
	DW(2,1) = -KBAR*AX + EPSAX	DER2	53
	DW(4,1) = K*AY + QA + EPSAY	DER2	54
55	DW(6,1) = -KBAR*E + RHOI	DER2	55
	DW(8,1) = -K*E0 + RHOI	DER2	56
	RETURN	DER2	57

END

DER2

58

1			DER2	59
		SUBROUTINE DER3(S,W,DW)	DER3	2
	C	INTEGRATE TRANSVERSE AND LONGITUDINAL ENVELOPES.	DER3	3
		COMMON/CDERIV/IFLAG,MR,NR,DP,NV(11),DAT(20),UZ,STOT,	DER3	4
5	1	DAT2(7),MP2FLG	DER3	5
		EQUIVALENCE (L,DAT(1)), (GRAD,DAT(2)), (BRHO,DAT(3)),	DER3	6
	1	(EPS,DAT(4)), (Q1,DAT(5)), (H0,DAT(6)),	DER3	7
	2	(KZ,DAT(8)), (EPSL,DAT(10)),(G,DAT(11))	DER3	8
		DIMENSION W(7,1),DW(7,1)	DER3	9
10		REAL K,KZ,L	DER3	10
	C		DER3	11
		IF (IFLAG) 1,1,2	DER3	12
	1	BRHO=BRHO*(1.+ DP)	DER3	13
		K = GRAD/BRHO	DER3	14
15		EPS2 = EPS*EPS	DER3	15
		EPSL2 = EPSL*EPSL	DER3	16
		Q1G = Q1*G	DER3	17
		RETURN	DER3	18
	C		DER3	19
20	2	AX = W(1,1)	DER3	20
		AY = W(3,1)	DER3	21
		AZ = W(5,1)	DER3	22
		AX3 = AX*AX*AX	DER3	23
		AY3 = AY*AY*AY	DER3	24
25		AZ3 = AZ*AZ*AZ	DER3	25
		AXYZ = AZ*(AX+AY)	DER3	26
		AZ2 = 2.*AZ*AZ	DER3	27
		EAX3 = EPS2/AX3	DER3	28
		EAY3 = EPS2/AY3	DER3	29
30		EAZ3 = EPSL2/AZ3	DER3	30
		QAXYZ = Q1/AXYZ	DER3	31
		QAZ = Q1G/AZ2	DER3	32
	C		DER3	33
		DW(1,1) = W(2,1)	DER3	34
35		DW(3,1) = W(4,1)	DER3	35
		DW(5,1) = W(6,1)	DER3	36
	C		DER3	37
		DW(2,1) = -K*AX + EAX3 + QAXYZ	DER3	38
		DW(4,1) = K*AY + EAY3 + QAXYZ	DER3	39
40		DW(6,1) = KZ*AZ + EAZ3 + QAZ	DER3	40
	C		DER3	41
		RETURN	DER3	42
		END	DER3	43

1	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
5	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
10	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
15		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
20	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
25		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
30		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
35	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
40	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
45	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
50	QAY=QA/AY	DER4	50
	KX=-K+QAX	DER4	51
	KY= K+QAY	DER4	52
		DER4	53
	DO 6 J=2,NR	DER4	54
55	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)

DER4 58

DER4 59

DER4 60

DER4 61

60

RETURN

END

1		SUBROUTINE DER5(S,W,DW)	DER5	2
	C		DER5	3
	C	DIFFERENTIAL EQUATION OF THE MOTION IN A SEXTUPOLE	DER5	4
	C		DER5	5
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
10		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
15	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
20		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
25		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

1

SUBROUTINE DER6(S,W,DW)
RETURN
END

DER6 2
DER6 3
DER6 4

1

SUBROUTINE DER7(S,W,DW)
RETURN
END

DER7 2
DER7 3
DER7 4

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.

1	SUBROUTINE DET(C, X)	DET	2
	C CALCULATE THE DETERMINANT OF A 3X3 COMPLEX MATRIX.	DET	3
	COMPLEX C, X	DET	4
	DIMENSION C(3,3)	DET	5
5	X = C(1,1)*(C(2,2)*C(3,3) - C(2,3)*C(3,2)) -	DET	6
	1 C(1,2)*(C(2,1)*C(3,3) - C(2,3)*C(3,1)) +	DET	7
	2 C(1,3)*(C(2,1)*C(3,2) - C(2,2)*C(3,1))	DET	8
	RETURN	DET	9
	END	DET	10

1	SUBROUTINE DIFEQ(NN,V,DERI)	DIFEQ	2
		BLANK	2
	LEVEL 2, STORE,INFF,IWORK	BLANK	3
	COMMON STORE(48000),IWORK(10)	86MARSIZ	1
5	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
10	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
15	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
20	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
25	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
30		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
35	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
40	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
45	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
50	JV1=NV(1)	DIFEQ	45
	DP = V(6,JV1)	DIFEQ	46
		DIFEQ	47
	S = 0.	DIFEQ	48
	ISTEP=0	DIFEQ	49
55	IOSTP=0	DIFEQ	50
	NSTP=SMAX/H0+.000001	DIFEQ	51
	H=SMAX/FLOAT(NSTP)	DIFEQ	52

	OUT=.FALSE.	DIFEQ	53
	IF (NPR.GT.0) OUT=.TRUE.	DIFEQ	54
60		DIFEQ	55
	IFLAG=-IFLAG	DIFEQ	56
	UZ = 0.	DIFEQ	57
	CALL DERI(S,W,DW)	DIFEQ	58
	W(2) = W(2) + UZ	DIFEQ	59
65	IFLAG = -IFLAG	DIFEQ	60
	CALL ZAM0(MN,S,DERI,W,DW,T,H)	DIFEQ	61
		DIFEQ	62
	IF (.NOT.OUT) GO TO 3	DIFEQ	63
	PRINT 101,ISTEP,S, (W(I),I=1,MN)	DIFEQ	64
70		DIFEQ	65
	3 CALL ZAM(S,DERI,W,DW,T,SWPR)	DIFEQ	66
		DIFEQ	67
	ISTEP=ISTEP+1	DIFEQ	68
	IF (SWPR) GO TO 4	DIFEQ	69
75	GO TO 5	DIFEQ	70
		DIFEQ	71
	4 IF (.NOT.OUT) GO TO 5	DIFEQ	72
	IOSTP=IOSTP+1	DIFEQ	73
	IF (IOSTP.LT.NPR) GO TO 5	DIFEQ	74
80	PRINT 101,ISTEP,S, (W(I),I=1,MN)	DIFEQ	75
	IOSTP = 0	DIFEQ	76
		DIFEQ	77
	5 IF (S.LT.SMAX-.0000000001) GO TO 3	DIFEQ	78
	W(2) = W(2) - UZ	DIFEQ	79
85	IF (.NOT.OUT) GO TO 7	DIFEQ	80
	PRINT 101,ISTEP,S, (W(I),I=1,MN)	DIFEQ	81
		DIFEQ	82
	7 DO 6 J=1,NR	DIFEQ	83
	JV=NV(J)	DIFEQ	84
90	JM = MR*(J-1)	DIFEQ	85
	DO 6 I=1,MR	DIFEQ	86
	6 V(I,JV) = W(I+JM)	DIFEQ	87
		DIFEQ	88
	RETURN	DIFEQ	89
95	101 FORMAT(5X,I5,10F11.6/(10X,10F11.6))	DIFEQ	90
	END	DIFEQ	91

1

```
SUBROUTINE DKE(NBSY,V,NBP)
RETURN
END
```

DIFEQ	92
DKE	2
DKE	3
DKE	4

1

SUBROUTINE DOFIT(M)
RETURN
END

DOFIT 2
DOFIT 3
DOFIT 4

1	SUBROUTINE DOIT(MSR)	DOIT	2
	INTEGER OPNAME,BDAT	DOIT	3
		BLANK	2
	LEVEL 2, STORE,INFF,IWORK	BLANK	3
5	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
10	COMMON/NELS/NELS	BMIL	2
	LEVEL 2,MI	BMIL	3
		BMIL	4
		BMI	3
	COMMON/BMI1/MI1(16000)	NV3BMI1	1
15	COMMON/NELS1/NELS1	BMI1L	2
	LEVEL 2,MI1	BMI1L	3
		BMI1L	4
		BMI1	3
	C COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,	CONTRL	2
20	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
25	C COMMON/MOVTOG/TOG	MOVTOG	2
	LOGICAL TOG	MOVTOG	3
		MOVTOG	4
	C EQUIVALENCE(INDEF,IINDEF)	MOVTOG	5
		DOIT	9
30	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
35	COMMON/TSW/TRSW,MCY(20),JM,LOCC	NOV3TSW	1
	LOGICAL TRSW	TSW	3
	LOGICAL TABSW	NOV30DOIT	2
		DOIT	15
	COMMON/SWCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	SWCH	2
40	LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS	SWCH	3
		SWCH	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
45	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
50		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
55	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	DOIT	33
60	C	GO TO (END,=,CRD,BML,DRF,MAG,MAGV,EQU,INV,SHF,	DOIT	34
	C	REF,**,MMM,CYA,CYB,CYC,SUM,TRKB,VEC,MXV,	DOIT	35
	C	SHF7,MOVE,TAB,CALC)	DOIT	36
			DOIT	37
			DOIT	38
		GO TO (5,11,200,200,14,15,16,200,18,19,	DOIT	39
65		1 20,21,22,201,201,23,51,75,200,104,	DOIT	40
		2 47,78,810,77) NSTAT	DOIT	41
			DOIT	42
	5	RETURN	DOIT	43
			DOIT	44
70			DOIT	45
	C	ERROR ENCOUNTERED. OPERATION CALLED IS NOT ALLOWED.	DOIT	46
	201	WRITE(3,1) OPNAME	DOIT	47
		ERROR = .TRUE.	DOIT	48
	1	FORMAT (13H ***ERROR*** A5,57H IS AN OPERATION THAT CANNOT BE CALL	DOIT	49
75		IED BY SUBROUTINE DOIT.)	DOIT	50
		RETURN	DOIT	51
			DOIT	52
	C	DRIFT ENTRY	DOIT	53
	14	KB = INFF(5,MS)	DOIT	54
80		IF (KB.EQ.0) GO TO 1106	DOIT	55
		FLTOT = 0.	DOIT	56
		DO 1104 K=1,KB	DOIT	57
		MM = MDAT(MS,K)	DOIT	58
	1104	FLTOT = FLTOT + FDAT(MM,1)	DOIT	59
85		FLSEP = FDAT(MS,2)	DOIT	60
		EL = FLSEP - FLTOT	DOIT	61
		CALL REPFLT(MS,1,EL)	DOIT	62
		GO TO 1107	DOIT	63
	1106	EL = FDAT(MS,1)	DOIT	64
90		1107 CALL DRIFT(MS,EL)	DOIT	65
		GO TO 200	DOIT	66
			DOIT	67
	C	MAGNET ENTRY	DOIT	68
	15	KA = INFF(4,MS)	DOIT	69
95		IF (KA.GT.0) GO TO 130	DOIT	70
		CALL DATA (MS,1,1,6,PAR)	DOIT	71
		CALL MAGNET (MS,PAR)	DOIT	72
		GO TO 200	DOIT	73
	130	CALL MAGMATS(MS,KA)	DOIT	74
100		GO TO 200	DOIT	75
			DOIT	76
	C	MAGV ENTRY	DOIT	77
	16	CALL DATA (MS,1,1,6,PAR)	DOIT	78
		CALL MAGV(MS,PAR)	DOIT	79
105		GO TO 200	DOIT	80
	C		DOIT	81
	C	MMM ENTRY	DOIT	82
	22	NDAT = INFF(17,MS)	DOIT	83
		CALL MIFILL (MS,1,NDAT,NELM,MIM)	DOIT	84
110		KB = INFF(5,MS)	DOIT	85
		IF (KB.NE.0.AND.KB.NE.1) GO TO 142	DOIT	86
		CALL MMM(MS,NELM,MIM)	DOIT	87
		GO TO 200	DOIT	88
	142	CALL EXECMM(MS,NELM,KB,MIM)	DOIT	89

115	GO TO 200	DOIT	90
	C MOVE ENTRY	DOIT	91
	78 DO 781 I=1,7	DOIT	92
	781 V0(I) = 0.	DOIT	93
120	CALL MOVE(MS,MS,V0)	DOIT	94
	TOG = .TRUE.	DOIT	95
	GO TO 200	MY3DOIT	2
	C	DOIT	96
	C MXV ENTRY	DOIT	97
125	104 M1 = MDAT(MS,1)	DOIT	98
	M2 = MDAT(MS,2)	DOIT	99
	LOC = 1	DOIT	100
	IF (.NOT.VCSW) GO TO 105	DOIT	101
	DECODE (10,106,M2) NUM	DOIT	102
130	106 FORMAT (1X,I1,8X)	DOIT	103
	LOC = (NUM-1)*7 + 1	DOIT	104
	105 CALL DATA (M2,5,LOC,7,PAR(8))	DOIT	105
	C = ENTRY	DOIT	106
	CALL RTRV7(M1,T,RW)	DOIT	107
135	CALL MXV7(T,PAR(8),PAR(1))	DOIT	108
	CALL STDAT(MS,5,1,7,PAR)	DOIT	109
	GO TO 200	DOIT	110
	C REF ENTRY	DOIT	111
	20 NJ = MDAT(MS,1)	DOIT	112
140	CALL REF(MS,NJ)	DOIT	113
	GO TO 200	DOIT	114
	C SHF ENTRY	DOIT	115
	19 CALL DATA(MS,1,1,4,PAR)	DOIT	116
	CALL SHF(MS,PAR)	DOIT	117
145	GO TO 200	DOIT	118
	C SHF7 ENTRY	DOIT	119
	47 CALL DATA (MS,1,1,6,PAR)	DOIT	120
	CALL SHF7 (MS,PAR)	DOIT	121
	GO TO 200	DOIT	122
150	C	DOIT	123
	C INV ENTRY	DOIT	124
	18 NJ = MDAT(MS,1)	DOIT	125
	CALL INV(MS,NJ)	DOIT	126
	GO TO 200	DOIT	127
155	C STAR ENTRY	DOIT	128
	21 KS = INFF(4,MS)	DOIT	129
	CALL STAR(MS,KS)	DOIT	130
	GO TO 200	DOIT	131
	C SUM ENTRY	DOIT	132
160	51 KS = INFF(4,MS)	DOIT	133
	CALL DATA(MS,1,1,KS,PAR)	DOIT	134
	CALL SUM(KS,PAR,RES)	DOIT	135
	CALL STDAT(MS,5,1,1,RES)	DOIT	136
	GO TO 200	DOIT	137
165	C	DOIT	138
	C = ENTRY	DOIT	139
	11 KN = INFF(20,MS)	DOIT	140
	IF (KN.NE.4HSCAL) GO TO 266	DOIT	141
	IOP = BDAT(MS,1)	DOIT	142
170	IF (IOP.EQ.1H) GO TO 200	DOIT	143
	IQ = 2	DOIT	144
		DOIT	145

	ITP = INFF(21,MS)	DOIT	146
	IF (ITP.NE.4HSNGL) IQ = 4	DOIT	147
	CALL DATA(MS,1,1,IQ,PAR)	DOIT	148
175	CALL AROP(MS,PAR,IOP)	DOIT	149
	GO TO 200	DOIT	150
	C SCALAR OPERATIONS	DOIT	151
	76 XX = FDAT(MS,1)	DOIT	152
	NOPMS = IOPMS - 111	DOIT	153
180	GO TO (86,87,88,89,90,91,92,93,94,95,96) NOPMS	DOIT	154
	86 YY = SIN(XX) \$ GO TO 99	DOIT	155
	87 YY = COS(XX) \$ GO TO 99	DOIT	156
	88 YY = EXP(XX) \$ GO TO 99	DOIT	157
	89 YY = ALOG(XX) \$ GO TO 99	DOIT	158
185	90 YY = XX*XX \$ GO TO 99	DOIT	159
	91 YY = SQRT(XX) \$ GO TO 99	DOIT	160
	92 YY = TAN(XX) \$ GO TO 99	DOIT	161
	93 YY = ASIN(XX) \$ GO TO 99	DOIT	162
	94 YY = ACOS(XX) \$ GO TO 99	DOIT	163
190	95 YY = ATAN(XX) \$ GO TO 99	DOIT	164
	96 YY = ABS(XX) \$ GO TO 99	DOIT	165
	99 CALL STDAT(MS,5,1,1,YY)	DOIT	166
	GO TO 200	DOIT	167
		DOIT	168
195	266 KREM(1)=BDAT(MS,1)	DOIT	169
	KREM(2)=BDAT(MS,3)	DOIT	170
	IOP = BDAT(MS,2)	DOIT	171
	CALL AROP(MS,KREM,IOP)	DOIT	172
	GO TO 200	DOIT	173
200		DOIT	174
	* CYA AND CYC ENTRIES	DOIT	175
	23 CONTINUE	DOIT	176
	25 NDAT = INFF(17,MS)	DOIT	177
	KA = INFF(4,MS)	DOIT	178
205	IF (MINFLG.NE.4.OR.MIFLG) CALL MIFILL(MS,1,NDAT,NELS1,MI1)	DOIT	179
	CALL CYX(MS)	DOIT	180
	C GO TO 280	DOIT	181
	GO TO 200	DOIT	182
	* CYB ENTRY	DOIT	183
210	C 24 CALL SETCYB(MS)	DOIT	184
	C CALL CYX(MS)	DOIT	185
	C 280 JMT = JMT + 1	DOIT	186
	* IF TRSW IS ON, ANGLE FIT WILL BE REPLACED BY TRACE FIT	DOIT	187
	C 122 IF (TRSW) RETURN	DOIT	188
215	C GO TO 200	DOIT	189
		DOIT	190
	C TRKB ENTRY	DOIT	191
	75 CALL TRKB(MS)	DOIT	192
	GO TO 200	DOIT	193
220		DOIT	194
	C TAB ENTRY	DOIT	195
	810 TABSW = .TRUE.	DOIT	196
	CALL TABULAT(MS,TABSW)	DOIT	197
	GO TO 200	DOIT	198
225		DOIT	199
	C CALC ENTRY	DOIT	200
	77 CALL CALC(MS)	DOIT	201
	GO TO 200	DOIT	202

230 200 NXTM = INFF(8,MS)
MS = NXTM
GO TO 100
END

DOIT 203
DOIT 204
DOIT 205
DOIT 206

1	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
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
10		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
15	EQUIVALENCE(IVAR,FVAR)	DRFMATS	8
	INTEGER ELNUM	DRFMATS	9
	DATA MASKL/7777777777B/	DRFMATS	10
	DATA MASK1/777777777000000000B/	DRFMATS	11
		DRFMATS	12
20	* 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
25	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
30	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
35	1 FORMAT (5X,3H***,1X,*THE FOLOWING OCTAL NUMBER IS NOT ACCEPTABLE	DRFMATS	28
	1AS A VECTOR,*/9X,*WHOSE ELEMENTS ARE NEEDED FOR DEFINING MULTIPLE	DRFMATS	29
	2 DRIFTS.*2X,020)	DRFMATS	30
		DRFMATS	31
	* GET VECTOR NAME, INDEX AND OPNAME.	DRFMATS	32
40	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
45	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
50	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
55	M1 = M1 - 1	DRFMATS	48
	25 CONTINUE	DRFMATS	49
		DRFMATS	50

	RETURN	DRFMATS	51
40	WRITE (3,3) NV	DRFMATS	52
60	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
65	1*ARE NEEDED TO DEFINE CURRENT DRIFT MATRICES.*)	DRFMATS	58
50	ERROR = .TRUE.	DRFMATS	59
	RETURN	DRFMATS	60
	END	DRFMATS	61

1	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
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/SWTC/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	SWTC	2
	LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS	SWTC	3
10		SWTC	4
		DRFRSV	10
	DATA MXP MY/1/	DRFRSV	11
	CALL STINFO(M0,NAME0,SHDRF ,KA0,KB0,NAME1)	DRFRSV	12
	IF (KB0.EQ.0) CALL RESRV(M0,MXP MY,MINZER,1,1,0,0,0,0,0)	DRFRSV	13
15	IF (KB0.GT.0) CALL RESRV(M0,MXP MY,MINZER,2,1,KB0,2,0,0,0)	DRFRSV	14
	RETURN	DRFRSV	15
	END	DRFRSV	16

```

1      SUBROUTINE DRIFT(M,EL)                                DRIFT 2
      DIMENSION RX(2,3),RW(3)                                DRIFT 3
      DATA RW(2),RW(3)/2*0.0/                                DRIFT 4
5      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
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
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
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
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
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
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
40     CALL STXY(M,RX,RX,RW) DRIFT 41
      RETURN DRIFT 42
      END DRIFT 43

```

1

FUNCTION DT(A)
DT= RINDEF
RETURN
END

DT 2
DT 3
DT 4
DT 5

1

SUBROUTINE EDRF(NBSY,V,NBP)
RETURN
END

EDRF 2
EDRF 3
EDRF 4

1 SUBROUTINE EIGSIX(R7,LMBD,UE,TRC,BLMBD)
RETURN
END

EIGSIX 2
EIGSIX 3
EIGSIX 4

1	INTEGER FUNCTION ELNUM(NAME)	ELNUM	2
	C SEARCHES FOR INDEX OF ELEMENT NAMED NAME	ELNUM	3
	LEVEL 2, STORE, INFF, IWORK	BLANK	2
5	COMMON STORE(48000), IWORK(10)	BLANK	3
	DIMENSION INFF(24, 2000)	86MARSIZ	1
	EQUIVALENCE (INFF, STORE)	86MARSIZ	2
		86MARSIZ	3
		BLANK	5
	COMMON/STORE/LMAX, LINF, LFILE, MAX, MIN, ICARD(11)	STORE	2
10	* 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
15		CONTRL	5
	COMMON/SWTCB/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
20		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
25	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
30	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
35	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
40	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
45	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
50	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
55	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	ELNUM	36
	22 IF(VCSW) GO TO 23	ELNUM	37
60	C IF SECOND LETTER OF NAME IS NOT A NUMBER, STOP SEARCHING.	ELNUM	38
	NNAM= SHIFT(NAM,6)	ELNUM	39
	DECODE(10,4,NNAM) INAM	ELNUM	40
	4 FORMAT(R1,9X)	ELNUM	41
65	DO 21 I=1,10	ELNUM	42
	IF(INAM.EQ.INB(I)) GO TO 24	ELNUM	43
	21 CONTINUE	ELNUM	44
	GO TO 23	ELNUM	45
	24 CONTINUE	ELNUM	46
70	C SEARCH FOR ELEMENT WHOSE WHOLE NAME IS FIRST LETTER OF THIS ONE.	ELNUM	47
	VCSW = .TRUE.	ELNUM	48
	DECODE (10,2,NAM) NM	ELNUM	49
	2 FORMAT (A1,9X)	ELNUM	50
	GO TO 17	ELNUM	51
75	C FAILED TO MATCH NAMES	ELNUM	52
	23 ERROR = .TRUE.	ELNUM	53
	WRITE(3,11) NAME	ELNUM	54
	ELNUM = -0	ELNUM	55
	RETURN	ELNUM	56
80	11 FORMAT(15H *** ERROR *** ,A5,30H IS AN UNDEFINED ELEMENT NAME.)	ELNUM	57
	25 II=I	ELNUM	58
	IF (.NOT.VCSW) GO TO 26	ELNUM	59
	NOP=INFF(1,II)	ELNUM	60
	IF ((NOP.EQ.3HCYC).OR.(NOP.EQ.3HCYA)) GO TO 26	ELNUM	61
85	IF ((NOP.NE.4HPVEC).AND.(NOP.NE.4HCVEC)) GO TO 23	ELNUM	62
	26 ELNUM = II	ELNUM	63
	IF (MODE.EQ.2) RETURN	ELNUM	64
	IF(.NOT.MINV) RETURN	ELNUM	65
	ELNUM = - ELNUM	ELNUM	66
90	RETURN	ELNUM	67
	30 ELNUM=-0	ELNUM	68
	IF (ERPR) WRITE(3,31) NAME	ELNUM	69
	31 FORMAT (15H *** ERROR *** ,A5,* IS NOT A LEFT ADJUSTED ELEMENT NAM	ELNUM	70
	1E*)	ELNUM	71
95	ERPR=.TRUE.	ELNUM	72
	RETURN	ELNUM	73
	END	ELNUM	74
		ELNUM	75

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

1	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
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/SWITCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	SWTCH	2
10	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
15		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
20	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
25	RETURN	EMIS	17
	END	EMIS	18

1	SUBROUTINE ENVTRAC	ENVTRAC	2
		BLANK	2
	LEVEL 2, STORE, INFF, IWORK	BLANK	3
	COMMON STORE(48000), IWORK(10)	86MARSIZ	1
5	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
10	LOGICAL BEG, DPR, PLT	TRKINT	4
	COMMON/BMI/MI(5000)	BMIL	1
	COMMON/NELS/NELS	BMIL	2
	LEVEL 2, MI	BMIL	3
		BMIL	4
15		BMI	3
	RETURN	ENVTRAC	6
	END	ENVTRAC	7

1	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
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
10	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
15	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
20	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
25	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
30	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
35	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
40	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
45	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
50	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
55	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      56
        CALL STDAT(M,5,1,21,V)                   EQIL      57
60      RETURN                                    EQIL      58
100     FORMAT (/20X,15H CLOSED ORBITS A6, 12H DEFINED BY A6/29X,1HX,10X, EQIL      59
1       5HDX/DS,8X,1HY,10X,5HDY/DS,7X,4HS-S0,8X,4HDP/P,7X,6HMISVAR//) EQIL      60
105     FORMAT (25H SYNCHRONOUS C.O. VS ,7(F10.5,1X) /25H DP/P = 0. EQIL      61
1       C.O. VO ,7(F10.5,1X)/25H DP/P = 1. C.O. VP ,7(F10.5,1X)/ EQIL      62
65     2/)                                       EQIL      63
110     FORMAT (25H VALUES AFTER VS ,7(F10.5,1X)/25H MULTIPLYIN EQIL      64
1       VO ,7(F10.5,1X))                         EQIL      65
115     FORMAT (5X,3H BY,1X,A4,7X,2HVP,3X,7(F10.5,1X)) EQIL      66
        END                                       EQIL      67
```


1	SUBROUTINE EQU(M,N)	EQU	2
		EQU	3
	C EQUATE COMPUTED DATA STORAGE OF M TO THAT OF N.	EQU	4
	C 1H* IS A FLAG TO LEAVE THE PRESENT VALUES OF M UNTOUCHED,	EQU	5
5	C I.E. KEEP THE ORIGINALS SET FOR M. OTHERS WILL BE CHANGED TO	EQU	6
	C THOSE READ IN FROM N OR RESET.	EQU	7
		EQU	8
		EQU	9
	DIMENSION INF(24)	EQU	10
10	CALL RDINF(INF,N)	EQU	11
	DO 1 I=1,9	EQU	12
	1 INF(I) = 1H*	EQU	13
	C USE LQ AND NQ OF N. COMPUTE NEW NTOT	EQU	14
	DO 2 I=12,17	EQU	15
15	2 INF(I) = 1H*	EQU	16
	INF(18) = INF(11) + INF(17)	EQU	17
	C USE KIND, ROWS, ETC OF N	EQU	18
	C REWRITE INFO OF M.	EQU	19
	CALL WRTINF(INF,M)	EQU	20
20	RETURN	EQU	21
	END	EQU	22

1	SUBROUTINE ERRSR(M)	ERRSR	2
	C ERROR HAS BEEN ENCOUNTERED. EXECUTION OF RUN HALTED.	ERRSR	3
	C READS CARDS TO FIND END OF RUN	ERRSR	4
	C RETURNS TO MAIN FOR NEXT RUN	ERRSR	5
5	INTEGER OPNAME	ERRSR	6
	DIMENSION ICARD(8)	ERRSR	7
	CALL WFLSR	ERRSR	8
	1 READ (2,5) (ICARD(IW),IW=1,8)	ERRSR	9
	5 FORMAT(8A10)	ERRSR	10
10	DECODE (12,6,ICARD) OPNAME	ERRSR	11
	6 FORMAT (7X,A5)	ERRSR	12
	IF (OPNAME.NE.3HFIN) GO TO 1	ERRSR	13
	WRITE (3,20)	ERRSR	14
	20 FORMAT (30H *** ERROR *** RUN TERMINATED./1X,119(1H.))	ERRSR	15
15	RETURN	ERRSR	16
	END	ERRSR	17

1

SUBROUTINE EVEC(M)
RETURN
END

EVEC

2

EVEC

3

EVEC

4

1	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
5	CALL STXY(N,RY,RX,RW)	EXCHM	6
	RETURN	EXCHM	7
	END	EXCHM	8

1	SUBROUTINE EXECMM(M,NELS,KB,MI)	EXECMM	2
		EXECMM	3
	C EXECUTES MMM (MATRIX MULTIPLICATION) INSTRUCTION	EXECMM	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
		EXECMM	7
	C SYNCH INSTRUCT MMM	EXECMM	8
	C X MMM KB // A B C . . .	EXECMM	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
	C FORM A LIST OF MATRIC INDICES, STORED IN MI(I). THESE ARE	EXECMM	12
	C MULTIPLIED TOGETHER TO FORM MATRIX NAMED X.	EXECMM	13
		EXECMM	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
	C SYMMETRIC SUPERPERIOD NEED BE SPECIFIED.	EXECMM	17
		EXECMM	18
	C THE ABS(KB) IS THE NUMBER OF SUPER PERIODS. IF ABS(KB) .GT. 1, THE	EXECMM	19
	C MATRIX IS RAISED TO THAT POWER.	EXECMM	20
20		EXECMM	21
	COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYML	DIM	2
		DIM	3
	DIMENSION MI(1)	EXECMM	23
	LEVEL 2,MI	EXMML	1
25		EXMML	2
		EXECMM	24
	CALL MMM(M,NELS,MI)	EXECMM	25
	IF (KB.GT.1) GO TO 5	EXECMM	26
	CALL REF(M7END,M)	EXECMM	27
30	MI(1) = M	EXECMM	28
	MI(2) = M7END	EXECMM	29
	CALL MMM(M,2,MI)	EXECMM	30
5	KK = IABS(KB)	EXECMM	31
	IF (KK.LE.1) RETURN	EXECMM	32
35	DO 10 I=1,KK	EXECMM	33
	10 MI(I) = M	EXECMM	34
	CALL MMM(M,MI,MI)	EXECMM	35
	RETURN	EXECMM	36
	END	EXECMM	37

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

31 I MI ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

1	SUBROUTINE FCN(NPAR,G,F,X,IFLAG)	FCN	2
		FCN	3
		BLANK	2
	LEVEL 2, STORE, INFF, IWORK	BLANK	3
5	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
10	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
15	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
20		FCN	11
	C EVALUATION OF F, IE CHISQUARE, FOR MINUIT	FCN	12
		FCN	13
	MINFLG=IFLAG	FCN	14
	NGO=2	FCN	15
25	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
30	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
35	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
40	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
45	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
50	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
55	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	FCN	49
		IF (ICONK.EQ.34) GO TO 22	FCN	50
60		IF (MATFLG.OR.ICONK.EQ.35) GO TO 3	FCN	51
		IF (BXBYFLG) GO TO 21	FCN	52
		LTRTBK = LTRTAB - 1 + ICONK	FCN	53
		GO TO 28	FCN	54
	21	LBXK = LTRTAB + 1	FCN	55
65		LBYK = LTRTAB + 7	FCN	56
	28	CONTINUE	FCN	57
			FCN	58
		IF(NPOSK.EQ.1) GO TO 17	FCN	59
		FIK = 0.0	FCN	60
70			FCN	61
		DO 6 L=1,NPOSK	FCN	62
		IPLK = 14*IPOS(L,K)	FCN	63
		IF (.NOT.BXBYFLG) GO TO 25	FCN	64
		NIKL1 = LBXK + IPLK	FCN	65
75		NIKL2 = LBYK + IPLK	FCN	66
		BX = STORE(NIKL1)	FCN	67
		BY = STORE(NIKL2)	FCN	68
		BETKL = BX/BY	FCN	69
		GO TO 26	FCN	70
80	25	NIKL = LTRTBK + IPLK	FCN	71
		BETKL = STORE(NIKL)	FCN	72
	26	DEL = BETKL - BFITK	FCN	73
		FIK = FIK + DEL*DEL	FCN	74
		GO TO (11,6) NGO	FCN	75
85	11	NTYP=5H	FCN	76
		N=N+1	FCN	77
		WRITE(3,101) N,NAMBFC(ICONK),NTYP,IPRPOS(L,K),BETKL,BFITK,DEL,	FCN	78
	1	SIG(K)	FCN	79
	6	CONTINUE	FCN	80
90			FCN	81
		GO TO 5	FCN	82
			FCN	83
	17	CONTINUE	FCN	84
		IPLK = 14*IPOS(1,K)	FCN	85
95		IF (.NOT.BXBYFLG) GO TO 27	FCN	86
		NIKL1 = LBXK + IPLK	FCN	87
		NIKL2 = LBYK + IPLK	FCN	88
		BX = STORE(NIKL1)	FCN	89
		BY = STORE(NIKL2)	FCN	90
100		BET1K = BX/BY	FCN	91
		GO TO 23	FCN	92
	27	NIK1 = LTRTBK + IPLK	FCN	93
		BET1K=STORE(NIK1)	FCN	94
			FCN	95
105	23	CONTINUE	FCN	96
		DEL=BET1K-BFITK	FCN	97
		FIK = DEL*DEL	FCN	98
		GO TO (12,5) NGO	FCN	99
	12	NTYP=5H	FCN	100
110		N=N+1	FCN	101
		IF (NPOSK.EQ.1) GO TO 36	FCN	102
		WRITE(3,104) N,NAMBFC(ICONK),NTYP,(IPRPOS(L,K),L=1,NN),BET1K,	FCN	103
	1	BFITK,DEL,SIG(K)	FCN	104
		GO TO 5	FCN	105

115	36	WRITE(3,101) N,NAMBFC(ICONK),NTYP,IPRPOS(1,K),BET1K,BFITK,DEL,	FCN	106
	1	SIG(K)	FCN	107
		GO TO 5	FCN	108
			FCN	109
	10	CONTINUE	FCN	110
120		FIK=0.0	FCN	111
		DO 14 L=1,NPOSK	FCN	112
		NIKL=LRTAB+14*IPOS(L,K)-1+ICONK	FCN	113
		BETKL=STORE(NIKL)	FCN	114
		IF (L.EQ.1) GO TO 140	FCN	115
125		DBKJ = BETKL - BETKJ	FCN	116
		DEL = DBKJ - BFITK	FCN	117
		FIK = FIK + DEL*DEL	FCN	118
	140	IF (ITYPK.EQ.2) BETKL = -BETKL	FCN	119
		BETKJ = BETKL	FCN	120
130		IF (L.EQ.1) GO TO 14	FCN	121
		GO TO (13,14) NGO	FCN	122
	13	NTYP=5HDIF	FCN	123
		IF (ITYPK.EQ.2) NTYP=5HSUM	FCN	124
		N=N+1	FCN	125
135		WRITE(3,101) N,NAMBFC(ICONK),NTYP,IPRPOS(L,K),DBKJ,BFITK,	FCN	126
	1	DEL,SIG(K)	FCN	127
	14	CONTINUE	FCN	128
			FCN	129
		GO TO 5	FCN	130
140			FCN	131
	3	CONTINUE	FCN	132
		IF (BETFLG) GO TO 33	FCN	133
		CALL BET(MTR,BETX,BETY,0)	FCN	134
		BETFLG = .TRUE.	FCN	135
145	33	IF (ICONK.EQ.35) GO TO 24	FCN	136
		IF (BXBYFLG) GO TO 29	FCN	137
		IF (ICONK.GT.6) GO TO 31	FCN	138
		BET1K = BETX(ICONK)	FCN	139
		GO TO 32	FCN	140
150	31	BET1K = BETY(ICONK-6)	FCN	141
	32	IF (ICONK.EQ.1.OR.ICONK.EQ.7) BET1K = RTPI * BET1K	FCN	142
		GO TO 23	FCN	143
			FCN	144
	29	BX = BETX(2)	FCN	145
155		BY = BETY(2)	FCN	146
		BET1K = BX/BY	FCN	147
		GO TO 23	FCN	148
			FCN	149
	24	CONTINUE	FCN	150
160		APQN = 0.5*(IPOS(1,K)*BETX(1) + IPOS(2,K)*BETY(1)) * IPOS(3,K)	FCN	151
	1	/ IPOS(4,K)	FCN	152
		BET1K = 1./ (ABS(SIN(APQN)) + .000000001)	FCN	153
		GO TO 23	FCN	154
			FCN	155
165	22	CONTINUE	FCN	156
		CALL RTRV7(MTR,T,RW)	FCN	157
		II = IPOS(1,K)	FCN	158
		JJ = IPOS(2,K)	FCN	159
		BET1K = T(II,JJ)	FCN	160
170		GO TO 23	FCN	161
			FCN	162

	5	F = F + FIK * WTK*WTK	FCN	163
			FCN	164
175	2	CONTINUE	FCN	165
			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.

1	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
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
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
15	COMMON STORE(48000),IWORK(10)	86MARSIZ	1
	DIMENSION INFF(24,2000)	86MARSIZ	2
	EQUIVALENCE (INFF,STORE)	86MARSIZ	3
		BLANK	5
	C	FDAT	15
20	LOC = INFF(14,M) + J - 1	FDAT	16
	FDAT = REALNUM(LOC)	FDAT	17
	C	FDAT	18
	RETURN	FDAT	19
	END	FDAT	20

1	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
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
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
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
20	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
25	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
30		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),BXY(20),RW(3),Q(2)	FIT	24
	LOGICAL IND,INF	FIT	25
35	LOGICAL INDEFF,INFINN	FIT	26
	INTEGER P1,P2,ELNUM	FIT	27
		FIT	28
	LOGICAL SINGL	FIT	29
	SINGL=.FALSE.	FIT	30
40	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
45	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
50	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
55	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	FIT	49
	21 FORMAT (3I1)	FIT	50
60	K1 = I1*6 + (I3-1) *2 +I2	FIT	51
	K2 = J1*6 +(J3-1)*2 + J2	FIT	52
	GO TO 16	FIT	53
60	K1 = KA	FIT	54
	K2 = KB	FIT	55
65	16 M1 = ELNUM(MV1(1))	FIT	56
	M2 = ELNUM(MV1(2))	FIT	57
	P1 = IV1(1)	FIT	58
	P2 = IV1(2)	FIT	59
	C LOOK UP INPUT GUESS FOR VARIABLES	FIT	60
70	VAR(1) = FDAT(M1,P1)	FIT	61
	VAR(2) = FDAT(M2,P2)	FIT	62
	C SAVE ORIGINAL INPUT VARIABLES	FIT	63
	VR1=VAR(1)	FIT	64
	VR2=VAR(2)	FIT	65
75	IF (.NOT.QSW) GO TO 50	FIT	66
	C FIT Q CASE	FIT	67
	PHI(1) = COS(6.283185307 * PHI(1))	FIT	68
	PHI(2) = COS(6.283185307 * PHI(2))	FIT	69
	GO TO 55	FIT	70
80	* OTHER FITS	FIT	71
	50 PHI(1) = PHI(1)/100.	FIT	72
	PHI(2) = PHI(2) /100.	FIT	73
	55 K=0	FIT	74
	C SET MAXIMUM ITERATIONS	FIT	75
85	KI = 100	FIT	76
	75 DO 100 I=1,2	FIT	77
	U(I)=VAR(I)	FIT	78
	V(I)=VAR(I)	FIT	79
	100 W(I)=VAR(I)	FIT	80
90	V(1) = V(1) + HSTEP	FIT	81
	IF(SINGL) V(2)=V(1)	FIT	82
	W(2) = W(2) + HSTEP	FIT	83
	DO 115 I=1,3	FIT	84
	GO TO (101,102,103),I	FIT	85
95	101 VC1 = V(1)	FIT	86
	VC2 = V(2)	FIT	87
	GO TO 104	FIT	88
	102 VC1 = W(1)	FIT	89
	VC2 = W(2)	FIT	90
100	GO TO 104	FIT	91
	103 VC1 = U(1)	FIT	92
	VC2 = U(2)	FIT	93
	104 CALL REPFLT(M1,P1,VC1)	FIT	94
	CALL REPFLT(M2,P2,VC2)	FIT	95
105	CALL DOIT(MSR)	FIT	96
	IF (ERROR) RETURN	FIT	97
	IF(BSW) GO TO 105	FIT	98
	IF (VSW) GO TO 110	FIT	99
	IF (RSW) GO TO 112	FIT	100
110	C FITQ	FIT	101
	CALL RXY(MC,RX,RY,RW)	FIT	102
	FC(1) = (RX(1,1) + RX(2,2)) * .5	FIT	103
	FC(2) = (RY(1,1) + RY(2,2)) * .5	FIT	104
	GO TO 106	FIT	105

115	105	CALL 'BET(MC,BXBY(1),BXBY(11),0)	FIT	106
		FC(1)=BXBY(K1)/100.	FIT	107
		FC(2) = BXBY(K2)/100.	FIT	108
		GO TO 106	FIT	109
	110	CALL DATA(MC,5,1,7,VC)	FIT	110
120		FC(1) = VC(K1)/100.	FIT	111
		FC(2)=VC(K2)/100.	FIT	112
		GO TO 106	FIT	113
	112	CALL RXY(MC,RXRY(1),RXRY(7),RW)	FIT	114
		FC(1)=RXRY(K1)/100.	FIT	115
125		FC(2)=RXRY(K2)/100.	FIT	116
	106	GO TO (107,108,109),I	FIT	117
	107	G(1) = FC(1)	FIT	118
		G(2) = FC(2)	FIT	119
		GO TO 115	FIT	120
130	108	H(1) = FC(1)	FIT	121
		H(2) = FC(2)	FIT	122
		GO TO 115	FIT	123
	109	F(1) = FC(1)	FIT	124
		F(2)= FC(2)	FIT	125
135		IND= INDEFF(F(1))	FIT	126
		INF= INFINN(F(1))	FIT	127
		IF(IND.OR.INF) GO TO 130	FIT	128
		IND = INDEFF(F(2))	FIT	129
		INF= INFINN(F(2))	FIT	130
140		IF(IND.OR.INF) GO TO 130	FIT	131
	115	CONTINUE	FIT	132
		D(1) = PHI(1) - F(1)	FIT	133
		D(2) = PHI(2) - F(2)	FIT	134
		IF (ABS(D(1)).LE.DEL.AND.ABS(D(2)).LE.DEL) GO TO 120	FIT	135
145	C	FIT NOT CLOSE ENOUGH. ITERATE AGAIN UNLES LIMIT HAS BEEN REACHED	FIT	136
		K = K + 1	FIT	137
		IF (K.EQ.KI) GO TO 125	FIT	138
	C	SET UP NEXT GUESS	FIT	139
		HD = 1./HSTEP	FIT	140
150		A(1,1) = (G(1) - F(1))*HD	FIT	141
		A(1,2) = (H(1) - F(1))*HD	FIT	142
		A(2,1) = (G(2) - F(2))*HD	FIT	143
		A(2,2) = (H(2) - F(2))*HD	FIT	144
		IF(SINGL) GO TO 999	FIT	145
155		DET = (A(1,1)*A(2,2)) - (A(1,2) * A(2,1))	FIT	146
		B(1,1) = A(2,2)/DET	FIT	147
		B(1,2) = -A(1,2)/DET	FIT	148
		B(2,1) = -A(2,1) /DET	FIT	149
		B(2,2) = A(1,1) /DET	FIT	150
160		S(1) = B(1,1)*D(1) + B(1,2)*D(2)	FIT	151
		S(2)=B(2,1)*D(1) + B(2,2) *D(2)	FIT	152
		VAR(1) = VAR(1) + (S(1)*XLAM)	FIT	153
		VAR(2) = VAR(2) + (S(2)*XLAM)	FIT	154
		GO TO 75	FIT	155
165	999	VAR(1)=VAR(1)+XLAM*D(1)/A(1,1)	FIT	156
		VAR(2)=VAR(1)	FIT	157
		GO TO 75	FIT	158
			FIT	159
	C	FIT SATISFIED. LEAVE NEW VARIABLES IN STORAGE	FIT	160
170	120	WRITE (3,1) P1,MV1(1),VAR(1),P2,MV1(2),VAR(2)	FIT	161
		RETURN	FIT	162

	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.

1		SUBROUTINE FITE(MSR,MC,MV1,IV1,Q,KA,KB)	FITE	2
		LEVEL 2, STORE,INFF,IWORK	BLANK	2
		COMMON STORE(48000),IWORK(10)	BLANK	3
5		DIMENSION INFF(24,2000)	86MARSIZ	1
		EQUIVALENCE (INFF,STORE)	86MARSIZ	2
			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
10		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
15		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
20	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
25		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
30		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
35		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
40		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
45	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,15,1X,	FITE	42
		1 4H OF ,A5,3H = ,E16.9,7X,I5,1X,4H OF ,A5,3H = ,E16.9)	FITE	43
50		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
55		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.

1	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
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
10	DIMENSION INFF(24,2000)	86MARSIZ	2
	EQUIVALENCE (INFF,STORE)	86MARSIZ	3
		BLANK	5
	INTEGER HVAR	FLDAT	9
	DATA HVAR/3HVAR/	FLDAT	10
15	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
20	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
25	END	FLDAT	21

1	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
5	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
10	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
15	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
20	15 F = REALNUM(LOC)	FLQ	16
	RETURN	FLQ	17
	END	FLQ	18

1	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
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
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
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
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
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
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
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
40	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 3X0 MATRICES	FXPT	42
	C OUT = OUTPUT OPTION IN BCD	FXPT	43
		FXPT	44
45	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
50	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
55	COMMON STORE(48000), IWORK(10)	86MARSIZ	1
	DIMENSION INFF(24,2000)	86MARSIZ	2
	EQUIVALENCE (INFF,STORE)	86MARSIZ	3

		BLANK	5
		NV3BMI1	1
60	COMMON/BMI1/MI1(16000)	BMI1L	2
	COMMON/NELS1/NELS1	BMI1L	3
	LEVEL 2,MI1	BMI1L	4
		BMI1	3
	C	FXPTL	1
65	COMMON/ MH/ML \$ LEVEL 2,ML	FXPT	55
	DIMENSION MI(1)	FXPT	56
	EQUIVALENCE (MI,MI1),(NELS,NELS1)	CONTRL	2
	COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,	CONTRL	3
	1 LDFLG,FIN	CONTRL	4
70	LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN	CONTRL	5
	COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMP	DIM	2
		DIM	3
	COMMON/CDERIV/IFLAG,MR,NR,DP,NV(11),DAT(20),UZ,STOT,	CDERIV	2
75	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
80	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
85	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)/ 6H1LMD1 , 6H1/LMD1, 6H1LMD3 ,	FXPT	71
	2 6H1/LMD3 /	FXPT	72
90	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
95	DATA IBLNK/1H /	FXPT	78
		FXPT	79
	CALL RANGET(NRN)	FXPT	80
	IIH = 1	FXPT683	4
	IIV = 1	FXPT683	5
100	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
105	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
110	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

115	EPST = P(2)	FXPT	95
	DELW = P(3)	FXPT	96
	DELT = P(4)	FXPT	97
	DELP = P(5)	FXPT	98
	IF (EPSW.EQ.0.) EPSW=1.E-08	FXPT	99
120	IF (EPST.EQ.0.) EPST=1.E-08	FXPT	100
	IF (DELP.EQ.0) DELP = 1.E-08	FXPT	101
	NGO = 2	FXPT	102
	IF (IFLAG.EQ.0) GO TO 2	FXPT	103
	NGO=1	FXPT	104
125	IF (DELW.EQ.0.) DELW=1.E-08	FXPT	105
	IF (DELT.EQ.0.) DELT=1.E-08	FXPT	106
	2 ML(1)=M7END-3	FXPT	107
	ML(2)=M7END-4	FXPT	108
	C SET UP BEAM SYSTEM ARRAY MI	FXPT	109
130	MBML = MDAT(M,2)	FXPT	110
	NDAT = INFF(17,MBML)	FXPT	111
	CALL MIFILL(MBML,1,NDAT,NELS,MI)	FXPT	112
	IF (ERROR) RETURN	FXPT	113
	C OPTIONS ON ITERATIONS AND PRINT	FXPT	114
135	C SET UP ARRAY FOR PREVIOUSLY DEFINED PARTICLE	FXPT	115
	C SET UP INITIAL REFERENCE RAY	FXPT	116
	C ZERO VW ARRAY	FXPT	117
	DO 3 I=1,112	FXPT	118
	VW(I) = 0.0	FXPT	119
140	3 CONTINUE	FXPT	120
	NP = MDAT(M,1)	FXPT	121
	CALL DATA (NP,1,1,7,VW)	FXPT	122
	NV(1) = 1	FXPT	123
	33 MB=1	FXPT	124
145	ME = NELS	FXPT	125
	NITS = 0	FXPT	126
	IF(ITERS.EQ.0)ITERS = 25	FXPT	127
	C	FXPT	128
	E(1) = EPSW	FXPT	129
150	E(2) = EPST	FXPT	130
	E(3) = EPSW	FXPT	131
	E(4) = EPST	FXPT	132
	F(1) = DELW	FXPT	133
	F(2) = DELT	FXPT	134
155	F(3) = DELW	FXPT	135
	F(4) = DELT	FXPT	136
	CALL HED	FXPT	137
	NPM = INFF(2,NP)	FXPT	138
	NAM = INFF(2,M)	FXPT	139
160	WRITE(3,1018)	FXPT	140
	WRITE(3,1000) NAM,NPM,(VW(I),I=1,7)	FXPT	141
	MM = 1	FXPT	142
	IOP = IFLAG	FXPT	143
	GO TO (31,32) NGO	FXPT	144
165	32 MN = 1	FXPT	145
	391 CALL RTRV7(MUNIT,R7,RW)	FXPT	146
	CALL STOR7(M,R7,RW)	FXPT	147
	GO TO 4	FXPT	148
	31 MN=9	FXPT	149
170	MR = 4	FXPT	150
	NR = 1	FXPT	151

	ASSIGN 4 TO NRS	FXPT	152
	IF (IFLAG.EQ.1) GO TO 800	FXPT	153
175	NR=5	FXPT	154
	MN=1	FXPT	155
	GO TO 802	FXPT	156
	39 IF (IFLAG.NE.1) GO TO 802	FXPT	157
	GO TO 800	FXPT	158
	C SAVE INITIAL CONDITIONS OF THE REFERENCE RAY.	FXPT	159
180	4 DO 5 I=1,7	FXPT	160
	5 VV(I) = VW(I)	FXPT	161
	IF (IFLAG.EQ.4) CALL DATA(M,7,1,7,V(1,8))	FXPT	162
	C TRACK PARTICLES MM TO MN THROUGH BEAMLINE ELEMENTS MB TO ME.	FXPT	163
	ASSIGN 510 TO TRKR	FXPT	164
185	DO 1 ISTR=1,NSTR	FXPT	165
	CALL RANSET(NRN)	FXPT	166
	GO TO 900	FXPT	167
	510 CONTINUE	FXPT	168
	1 CONTINUE	FXPT	169
190	GO TO (51,52) NGO	FXPT	170
	C CALCULATE THE 4X4 MATRIX, M.	FXPT	171
	51 IF (IFLAG.EQ.1) GO TO 511	FXPT	172
	DO 512 J=1,4	FXPT	173
	DO 512 I=1,4	FXPT	174
195	512 RM(I,J) = V(I,J+1)	FXPT	175
	GO TO 513	FXPT	176
	511 DO 55 J = 1 , 4	FXPT	177
	N = J + 1	FXPT	178
	DN=.5/F(J)	FXPT	179
200	DO 55 I=1,4	FXPT	180
	55 RM(I,J) = (V(I,N) - V(I,N+4)) * DN	FXPT	181
	513 IF (KB.GE.0) GO TO 56	FXPT	182
	RX(1,1) = RM(2,1)	FXPT	183
	RX(2,1) = RM(4,1)	FXPT	184
205	RX(1,2) = RM(2,3)	FXPT	185
	RX(2,2) = RM(4,3)	FXPT	186
	RX(1,3) = -V(2,1)	FXPT	187
	RX(2,3) = -V(4,1)	FXPT	188
	GO TO 56	FXPT	189
210	52 CALL RTRV7(M,R7,RW)	FXPT	190
	IF (KB.GE.0) GO TO 57	FXPT	191
	RX(1,1)=R7(2,1)	FXPT	192
	RX(2,1)=R7(4,1)	FXPT	193
	RX(1,2)=R7(2,3)	FXPT	194
215	RX(2,2)=R7(4,3)	FXPT	195
	RX(1,3)=- V(2,1)	FXPT	196
	RX(2,3)= -V(4,1)	FXPT	197
	57 DO 53 J=1,4	FXPT	198
	DO 53 I=1,4	FXPT	199
220	53 RM(I,J) = R7(I,J)	FXPT	200
	56 CONTINUE	FXPT	201
	IF (KB.GE.0) GO TO 58	FXPT	202
	CALL MXDIV(RX,2,1,D,2)	FXPT	203
	V(1,1)=RX(1,3)+VV(1)	FXPT	204
225	V(3,1)=RX(2,3)+VV(3)	FXPT	205
	DX=V(2,1)	FXPT	206
	DY=V(4,1)	FXPT	207
	V(2,1)=0.	FXPT	208

	V(4,1)=0.	FXPT	209
230	NITS = NITS + 1	FXPT	210
	IF ((ABS(DX).LT.E(2)).AND.(ABS(DY).LT.E(2))) GO TO 146	FXPT	211
	IF (NITS.GT.ITER) GO TO 146	FXPT	212
	GO TO (39,391) NGO	FXPT	213
	146 GO TO (141,142) NGO	FXPT	214
235	58 CONTINUE	FXPT	215
	C SET UP N - I AND REF RAY JUMP (RHS).	FXPT	216
	DO 7 I = 1, 4	FXPT	217
	DO 6 J = 1, 4	FXPT	218
	6 RMI(I,J) = RM(I,J)	FXPT	219
240	7 RMI(I,I) = RMI(I,I) - 1.0	FXPT	220
	DO 61 I=1,4	FXPT	221
	61 RMI(I,5) = VV(I)-V(I,1)	FXPT	222
	C SOLVE FOR EQ.-ORBIT-INITIAL CONDITIONS (IN TERMS OF REF. RAY).	FXPT	223
	CALL MXDIV(RMI, 4, 1, D, 4)	FXPT	224
245	IF(D.NE.0.) GO TO 75	FXPT	225
	WRITE(3, 1006)	FXPT	226
	WRITE (3,1001) ((RM(I,J),J=1,4),I=1,4)	FXPT	227
	1001 FORMAT (/10X,4F16.8)	FXPT	228
	CALL HED	FXPT	229
250	RETURN	FXPT	230
	C CALCULATE EQ.-ORBIT-INITIAL-CONDITIONS.	FXPT	231
	75 DO 8 I=1,4	FXPT	232
	8 V(I,1) = RMI(I,5) + VV(I)	FXPT	233
	C CHECK FOR CONVERGENCE.	FXPT	234
255	NITS = NITS + 1	FXPT	235
	IF (KA.LT.2) GO TO 14	FXPT	236
	WRITE (3,1013) NITS,(V(I,1),I=1,4)	FXPT	237
	14 CONTINUE	FXPT	238
	DO 9 I = 1, 4	FXPT	239
260	IF(ABS(RMI(I,5)).LT.E(I)) GO TO 9	FXPT	240
	IF (NITS.GT.ITER) GO TO 9	FXPT	241
	GO TO (39,391) NGO	FXPT	242
	9 CONTINUE	FXPT	243
	GO TO (141,142) NGO	FXPT	244
265	141 CALL RTRV7(MUNIT,R7,RW)	FXPT	245
	IOP = 1	FXPT	246
	DO 200 J=1,4	FXPT	247
	DO 200 I=1,4	FXPT	248
	200 R7(I,J)=RM(I,J)	FXPT	249
270	MM = 10	FXPT	250
	MN = 11	FXPT	251
	NR=1	FXPT	252
	IF (IFLAG.EQ.1) GO TO 138	FXPT	253
	IF (IFLAG.EQ.4) GO TO 142	FXPT	254
275	MM = 6	FXPT	255
	MN = 7	FXPT	256
	138 DO 143 I=1,7	FXPT	257
	V(I,MM) = VW(I)	FXPT	258
	143 V(I,MN) = VW(I)	FXPT	259
280	V(6,MM) = VW(6) + DELP	FXPT	260
	V(6,MN) = VW(6) - DELP	FXPT	261
	ASSIGN 144 TO TRKR	FXPT	262
	GO TO 900	FXPT	263
	144 DO 145 I=1,5	FXPT	264
285	145 R7(I,6) = 0.5*(V(I,MM)-V(I,MN))/DELP	FXPT	265

	ASSIGN 142 TO R5J	FXPT	266
	GO TO 805	FXPT	267
	142 IOP = IFLAG	FXPT	268
	CALL ROW5(R7)	FXPT	269
290	IF (KB.GE.0) GO TO 149	FXPT	270
	ML1=ML(1) \$ ML2=ML(2)	FXPTL	2
	CALL STOR7(ML1,R7,RW) \$ CALL REF(ML2,ML1)	FXPTL	3
	CALL MMM(ML1,2,ML) \$ CALL RTRV7(ML1,R7,RW)	FXPTL	4
	DO 139 J=1,4	FXPT	275
295	DO 139 I=1,4	FXPT	276
	139 RM(I,J) = R7(I,J)	FXPT	277
	149 IF(NGO.EQ.2) GO TO 1400	FXPT	278
	DO 151 I=1,7	FXPT	279
	151 VV(I) = VW(I)	FXPT	280
300	CALL MXV7(R7,VV,VV)	FXPT	281
	DO 152 I=1,5	FXPT	282
	152 R7(I,7) = VW(I) - VV(I)	FXPT	283
	1400 CONTINUE	FXPT	284
	ML1 = ML(1) \$ CALL STOR7(ML1,R7,RW)	FXPTL	5
305	C CALCULATE OFF-MOMENTUM ORBIT VP AND STORE AT LQ3	FXPT	286
	140 DO 148 I=1,4	FXPT	287
	RMI(I,5)=-R7(I,6)	FXPT	288
	DO 147 J=1,4	FXPT	289
	147 RMI(I,J)=R7(I,J)	FXPT	290
310	148 RMI(I,I)=RMI(I,I)-1.0	FXPT	291
	CALL MXDIV(RMI,4,1,D,4)	FXPT	292
	DO 150 I=1,4	FXPT	293
	150 VP(I)=RMI(I,5)	FXPT	294
	CALL STDAT(M,8,1,7,VP)	FXPT	295
315	C REPLACE INITIAL CONDITIONS OF PARTICLE WITH COMPUTED VALUES	FXPT	296
	IF ((IVSV.NE.0).AND.(IFLAG.NE.4))	FXPT	297
	1 CALL STDAT(NP,1,1,4,VW)	FXPT	298
	C STORE CLOSED ORBIT VECTOR AT LQ2	FXPT	299
	VW(5)=0.	FXPT	300
320	IF (IFLAG.NE.4) CALL STDAT(M,7,1,7,VW)	FXPT	301
	WRITE(3, 1003) NITS	FXPT	302
	WRITE (3,1007) NAM, ((R7(I,J),J=1,7) , I = 1,7)	FXPT	303
	C CALCULATE THE EIGENVALUES OF M.	FXPT	304
	TX = RM(1,1) + RM(2,2)	FXPT	305
325	TY = RM(3,3) + RM(4,4)	FXPT	306
	CTR = CMLPX(TX + TY, 0.)	FXPT	307
	CSQ = CSQRT(CMLPX((TX - TY)**2 + 4.*((RM(1,3) + RM(4,2))	FXPT	308
	1 *(RM(2,4) + RM(3,1)) - (RM(1,4) - RM(3,2))	FXPT	309
	2 *(RM(2,3) - RM(4,1))), 0.0))	FXPT	310
330	BLAM(1) = 0.5*(CTR + CSQ)	FXPT	311
	BLAM(2) = 0.5*(CTR - CSQ)	FXPT	312
	IF(TX.GE.TY) GO TO 41	FXPT	313
	C SWITCH BLAM(1) AND BLAM(2) IF TX.LT.TY .	FXPT	314
	LAM(1) = BLAM(1)	FXPT	315
335	BLAM(1) = BLAM(2)	FXPT	316
	BLAM(2) = LAM(1)	FXPT	317
	41 LAM(1) = 0.5*(BLAM(1) + CSQRT(BLAM(1)*BLAM(1) - (4.0, 0.0)))	FXPT	318
	LAM(2) = 1./LAM(1)	FXPT	319
	LAM(3) = 0.5*(BLAM(2) + CSQRT(BLAM(2)*BLAM(2) - (4.0, 0.0)))	FXPT	320
340	LAM(4) = 1./LAM(3)	FXPT	321
	C WRITE OUT THE EIGENVALUES...BOTH IN CARTESIAN AND POLAR FORMS.	FXPT	322
	DO 42 I = 1, 4	FXPT	323

	C(I) = REAL(CABS(LAM(I)))	FXPT	324
	MU(I) = ATAN2(AIMAG(LAM(I)), REAL(LAM(I)))	FXPT	325
345	Q(I) = SPER*MU(I)/TPI	FXPT	326
	Q(I) = AMOD(Q(I),1.)	FXPT	327
	IF (Q(I).LT.0.) Q(I)=Q(I)+1.	FXPT	328
	42 CONTINUE	FXPT	329
	C STORE EIGENVALUES AT LQ3+7	FXPT	330
350	CALL STDAT(M,8,8,8,LAM)	FXPT	331
	WRITE(3, 1004)	FXPT	332
	43 WRITE(3, 1005) (LABELA(I), LABELB(I), LAM(I), I, C(I), I,	FXPT	333
	1 MU(I), I, Q(I), I = 1, 4)	FXPT	334
	IF (KA.EQ.0) GO TO 93	FXPT	335
355		FXPT	336
	C COMPUTE EIGENVECTORS.	FXPT	337
	C FIRST AND SECOND EIGENVECTORS.	FXPT	338
	C FIRST COMPONENTS.	FXPT	339
	35 DO 45 L=1,3	FXPT	340
360	DO 44 J = 1, 3	FXPT	341
	RT = RM(L+1,J+1)	FXPT	342
	44 CM(L,J) = CMPLX(RT,0.0)	FXPT	343
	45 CM(L,L) = CM(L,L) - LAM(1)	FXPT	344
	CALL DET(CM, X(1,1))	FXPT	345
365	C SECOND THROUGH FOURTH COMPONENTS.	FXPT	346
	D = -1.0	FXPT	347
	DO 50 L = 2, 4	FXPT	348
	I = L - 1	FXPT	349
	DO 46 J = 1, 3	FXPT	350
370	RT = RM(J+1,I)	FXPT	351
	46 CM(J,I) = CMPLX(RT,0.0)	FXPT	352
	IF(L.EQ.3) CM(1,2) = CM(1,2) - LAM(1)	FXPT	353
	IF(L.EQ.4) CM(2,3) = CM(2,3) - LAM(1)	FXPT	354
	CALL DET(CM, X(L,1))	FXPT	355
375	X(L,1) = D*X(L,1)	FXPT	356
	D = -D	FXPT	357
	50 CONTINUE	FXPT	358
	C1 = CABS(X(1,1))	FXPT	359
	EX1 = X(1,1)/C1	FXPT	360
380	DO 536 I=1,4	FXPT	361
	536 X(I,1) = X(I,1)/EX1	FXPT	362
	RHO1 = AIMAG(CONJG(X(1,1))*X(2,1) + CONJG(X(3,1))*X(4,1))	FXPT	363
	FAC = 1.	FXPT	364
	IF (RHO1.NE.0.) FAC = 1./SQRT(ABS(RHO1))	FXPT	365
385	DO 537 I=1,4	FXPT	366
	X(I,1) = FAC * X(I,1)	FXPT	367
	537 X(I,2) = CONJG(X(I,1))	FXPT	368
	L = 1	FXPT	369
	ASSIGN 467 TO IWR	FXPT	370
390	GO TO 998	FXPT	371
	467 L = 2	FXPT	372
	ASSIGN 468 TO IWR	FXPT	373
	GO TO 998	FXPT	374
	C THIRD AND FOURTH EIGENVECTORS.	FXPT	375
395	C FIRST COMPONENTS.	FXPT	376
	468 DO 531 I = 1, 2	FXPT	377
	DO 531 J = 2, 4	FXPT	378
	RT = RM(I,J)	FXPT	379
	531 CM(I,J-1) = CMPLX(RT,0.0)	FXPT	380

400	DO 532 J = 2, 4	FXPT	381
	RT = RM(4,J)	FXPT	382
	532 CM(3,J-1) = CMPLX(RT,0.0)	FXPT	383
	CM(2,1) = CM(2,1) - LAM(3)	FXPT	384
	CM(3,3) = CM(3,3) - LAM(3)	FXPT	385
405	CALL DET(CM, X(1,3))	FXPT	386
	X(1,4) = CONJG(X(1,3))	FXPT	387
	C SECOND THROUGH FOURTH COMPONENTS.	FXPT	388
	D = -1.0	FXPT	389
	DO 533 L = 2, 4	FXPT	390
410	I = L - 1	FXPT	391
	CM(1,I) = CMPLX(RM(1,I), 0.0)	FXPT	392
	CM(2,I) = CMPLX(RM(2,I), 0.0)	FXPT	393
	CM(3,I) = CMPLX(RM(4,I), 0.0)	FXPT	394
	IF(L.EQ.2) CM(1,1) = CM(1,1) - LAM(3)	FXPT	395
415	IF(L.EQ.3) CM(2,2) = CM(2,2) - LAM(3)	FXPT	396
	CALL DET(CM, X(L,3))	FXPT	397
	X(L,3) = D*X(L,3)	FXPT	398
	D = -D	FXPT	399
	533 CONTINUE	FXPT	400
420	C3 = CABS(X(3,3))	FXPT	401
	EX3 = X(3,3)/C3	FXPT	402
	DO 546 I=1,4	FXPT	403
	546 X(I,3) = X(I,3)/EX3	FXPT	404
	RHO3 = AIMAG(CONJG(X(1,3))*X(2,3) + CONJG(X(3,3))*X(4,3))	FXPT	405
425	FAC = 1.	FXPT	406
	IF (RHO3.NE.0.) FAC = 1./SQRT(ABS(RHO3))	FXPT	407
	DO 547 I=1,4	FXPT	408
	X(I,3) = FAC * X(I,3)	FXPT	409
	547 X(I,4) = CONJG(X(I,3))	FXPT	410
430	L = 3	FXPT	411
	ASSIGN 534 TO IWR	FXPT	412
	GO TO 998	FXPT	413
	534 L = 4	FXPT	414
	ASSIGN 535 TO IWR	FXPT	415
435	GO TO 998	FXPT	416
	C 1. TRACE EQUILIBRIUM ORBIT AND EIGENVECTOR PARTS THROUGH SYSTEM,	FXPT	417
	C 2. LINEARIZE INDIVIDUAL NLT"S.	FXPT	418
	535 CONTINUE	FXPT	419
	C SET UP EIGENVECTOR INITIAL CONDITIONS.	FXPT	420
440	DO 91 I = 1, 3, 2	FXPT	421
	DO 91 J=1,4	FXPT	422
	VZ(J,I) = REAL(X(J,I))	FXPT	423
	VZ(J,I+1) = AIMAG(X(J,I))	FXPT	424
	91 CONTINUE	FXPT	425
445		FXPT	426
	C STORE 4 EIGENVECTORS AT LQ2+7, 7 COMPONENTS EACH.	FXPT	427
	C ORDER— REAL X1, IMAG X1, REAL X3, IMAG X3.	FXPT	428
		FXPT	429
	DO 92 I=1,4	FXPT	430
450	DO 92 J=5,7	FXPT	431
	92 VZ(J,I) = 0.	FXPT	432
	DO 97 I=5,7	FXPT	433
	DO 97 J=1,7	FXPT	434
	97 VZ(J,I) = 0.	FXPT	435
455	IQ=8	FXPT	436
	DO 90 I=1,4	FXPT	437

	CALL STDAT(M,7,IQ,7,VZ(1,I))	FXPT	438
	IQ=IQ+7	FXPT	439
	90 CONTINUE	FXPT	440
460	C WRITE CLOSED AND OFF-MOMENTUM ORBITS AT POSITION 0	FXPT	441
	93 WRITE(3,1017)	FXPT	442
	WRITE(3,1014) LBL(5),(V(I,1),I=1,7)	FXPT	443
	WRITE(3,1014) LBL(6),(VP(I),I=1,7)	FXPT	444
	IF (KA.EQ.0) GO TO 94	FXPT	445
465	WRITE(3, 1002)	FXPT	446
	C WRITE INITIAL CONDITIONS.	FXPT	447
	L=0	FXPT	448
	MATL = IBLNK	FXPT	449
	ASSIGN 600 TO IBET	FXPT	450
470	GO TO 599	FXPT	451
	600 IF (IFLAG.EQ.0) CALL STOR7(M,VZ,RW)	FXPT	452
	IF (IFLAG.EQ.4) CALL DATA(M,7,1,7,V(1,8))	FXPT	453
	94 II = 0	FXPT	454
	CALL RANSET(NRN)	FXPT	455
475	100 II = II + 1	FXPT	456
	L = L + 1	FXPT	457
	MB = II	FXPT	458
	ME = II	FXPT	459
	C SAVE INITIAL CONDITIONS.	FXPT	460
480	DO 101 J=1,7	FXPT	461
	101 VV(J) = VW(J)	FXPT	462
	IF (IFLAG.NE.4) GO TO 103	FXPT	463
	DO 104 J=1,7	FXPT	464
	104 VP(J) = V(J,8)	FXPT	465
485	103 ASSIGN 102 TO TRKR	FXPT	466
	MM = 1	FXPT	467
	MN = 1	FXPT	468
	NR=1	FXPT	469
	GO TO 900	FXPT	470
490	102 CONTINUE	FXPT	471
	IF (KA.LT.3.OR.IFLAG.NE.0) GO TO 604	FXPT	472
	CALL RTRV7(M,VZ,RW)	FXPT	473
	ASSIGN 604 TO IBET	FXPT	474
	GO TO 599	FXPT	475
495	604 CONTINUE	FXPT	476
	GO TO (95,40) NGO	FXPT	477
		FXPT	478
	95 IF(NL.EQ.0) GO TO 40	FXPT	479
	C LINEARIZE NON-LIN. TRANS	FXPT	480
500	C FORM 3X3 MATRICES	FXPT	481
	C RESTORE INITIAL CONDITIONS.	FXPT	482
	DO 111 J=1,7	FXPT	483
	F(J) = VW(J)	FXPT	484
	111 VW(J) = VV(J)	FXPT	485
505	IF (IFLAG.NE.4) GO TO 105	FXPT	486
	DO 106 J=1,7	FXPT	487
	106 V(J,8) = VP(J)	FXPT	488
	105 ASSIGN 112 TO NRS	FXPT	489
	ASSIGN 116 TO TRKR	FXPT	490
510	MM=1	FXPT	491
	MN=11	FXPT	492
	JP=10	FXPT	493
	JQ=11	FXPT	494

	IF (IFLAG.EQ.1) GO TO 800	FXPT	495
515	MN=1	FXPT	496
	JP=6	FXPT	497
	JQ=7	FXPT	498
	NR=5	FXPT	499
	IF (IFLAG.NE.4) ASSIGN 114 TO TRKR	FXPT	500
520	GO TO 802	FXPT	501
	112 DO 113 I=1,7	FXPT	502
	V(I,JP) = VW(I)	FXPT	503
	113 V(I,JQ) = VW(I)	FXPT	504
	V(6,JP) = VW(6) + DELP	FXPT	505
525	V(6,JQ) = VW(6) - DELP	FXPT	506
	GO TO 900	FXPT	507
	114 MM=6	FXPT	508
	MN=7	FXPT	509
	NR=1	FXPT	510
530	IOP = 1	FXPT	511
	ASSIGN 115 TO TRKR	FXPT	512
	GO TO 900	FXPT	513
	115 IOP = IFLAG	FXPT	514
	C PUT LINEARIZED 3X3 MATRIX REPRESENTING AN EXT IN ITS STORAGE	FXPT	515
535	116 CALL RTRV7(MUNIT,R7,RW)	FXPT	516
	IF (IFLAG.EQ.1) GO TO 15	FXPT	517
	DO 16 J=1,4	FXPT	518
	DO 16 I=1,4	FXPT	519
	16 R7(I,J) = V(I,J+1)	FXPT	520
540	GO TO (15,17,17,18) IFLAG	FXPT	521
	15 D=DELTW	FXPT	522
	DO 11 J=1,4	FXPT	523
	D=DELTW*2.	FXPT	524
	IF (J.EQ.2.OR.J.EQ.4) D=DELT*2.	FXPT	525
545	DO 11 I=1,4	FXPT	526
	11 R7(I,J) = (V(I,J+1) - V(I,J+5)) / D	FXPT	527
	17 DO 12 I=1,4	FXPT	528
	12 R7(I,6) = 0.5 * (V(I,JP) - V(I,JQ)) / DELP	FXPT	529
	ASSIGN 18 TO R5J	FXPT	530
550	GO TO 805	FXPT	531
	18 CALL MXV7(R7,VV,VV)	FXPT	532
	DO 13 I=1,5	FXPT	533
	13 R7(I,7) = VW(I) - VV(I)	FXPT	534
	DO 19 I=1,7	FXPT	535
555	19 VW(I) = F(I)	FXPT	536
	RW(1)=FDAT(NL,1)	FXPT	537
	CALL STOR7(NL,R7,RW)	FXPT	538
	40 IF (II.LT.NELS) GO TO 100	FXPT	539
		FXPT	540
560	402 CONTINUE	FXPT	541
		FXPT	542
	C TRACK BETA FUNCTIONS, ETA AND CLOSED ORBITS THROUGH BEAM SYSTEM.	FXPT	543
	C TRACKING DONE IN CYX, WHICH CALLS TRKCN FOR ORBIT TRACKING.	FXPT	544
	C BETA FUNCTIONS FORMED FROM 2X2 SUBMATRICES ARE ONLY APPROXIMATE	FXPT	545
565	C IF THERE IS COUPLING.	FXPT	546
		FXPT	547
	C TEST TRACE	FXPT	548
	TRX = RM(1,1) + RM(2,2)	FXPT	549
	TRY = RM(3,3) + RM(4,4)	FXPT	550
570	IF ((ABS(TRX).LT.2.) .AND. (ABS(TRY).LT.2.)) GO TO 405	FXPT	551

	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
575	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
580	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
585	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
590	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
595		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
600	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
605	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
610	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
615	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
620	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
625	KIND = INFF(20,NL)	FXPT	605
	IF (KIND.EQ.3HPRD.OR.KIND.EQ.3HMAP) NLF = NL	FXPT	606
	950 CONTINUE	FXPT	607

	NL = NLF	FXPT	608
	GO TO TRKR, (102, 114, 115, 116, 144, 510)	FXPT	609
630	C WRITE OUT AN EIGENVECTOR.	FXPT	610
	998 IF (IOUT.NE.5H) WRITE (3,1008) LAM(L), (X(J,L), J=1,4)	FXPT	611
	GO TO IWR, (467, 468, 534, 535)	FXPT	612
	C	FXPT	613
	C PRINT EIGENFUNCTIONS IN POLAR FORM	FXPT	614
635	599 CONTINUE	FXPT	615
	DO 601 K=1,2	FXPT	616
	DO 601 I=1,4	FXPT	617
	J = 2 * K - 1	FXPT	618
	IF ((VZ(I,J).NE.0.) .OR. (VZ(I,J+1).NE.0.)) GO TO 602	FXPT	619
640	EMOD(I,K) = CMLPX(0.,0.)	FXPT	620
	EPHI(I,K) = CMLPX(0.,0.)	FXPT	621
	GO TO 601	FXPT	622
	602 EMOD(I,K) = SQRT (VZ(I,J)*VZ(I,J) + VZ(I,J+1)*VZ(I,J+1))	FXPT	623
	EPHI(I,K) = ATAN2 (VZ(I,J+1),VZ(I,J))	FXPT	624
645	601 CONTINUE	FXPT	625
	WRITE (3,1003) L,MATL, (EMOD(I,1),EPHI(I,1),I=1,4) ,	FXPT	626
	1 (EMOD(I,2),EPHI(I,2),I=1,4)	FXPT	627
	GO TO IBET,(600,604)	FXPT	628
		FXPT	629
650	1000 FORMAT(64H0CALCULATION OF THE EQUILIBRIUM ORBIT AND BETATRON FUNCT	FXPT	630
	1IONS OF A5,1H./34H INITIAL REFERENCE RAY DEFINED BY A5/7H X = ,	FXPT	631
	2 F11.8,3X,*DX = *,F11.8,3X,*Y = *,F11.8,3X,*DY = *,F11.8,	FXPT	632
	3 5X,*DS = *,F11.8,3X,*DP/P = *,F11.8,3X,F11.8)	FXPT	633
	1002 FORMAT (1H1,*EIGENVECTORS 1 AND 3 IN POLAR COORDINATES*/	FXPT	634
655	1 1X*POS*23X*X1 *22X*DX1*22X*Y1 *22X*DY1*/	FXPT	635
	2 27X*X3 *22X*DX3*22X*Y3 *22X*DY3*/)	FXPT	636
	1003 FORMAT (1X,I4,1X,A5,1X, 4(2F11.6,3X) / 12X, 4(2F11.6,3X) /)	FXPT	637
	1004 FORMAT (*0EIGENVALUES OF THE 4X4 SUBMATRIX*)	FXPT	638
	1005 FORMAT(2A6,4H = (,2F12.8,5X,6H), C(I1,4H) = F12.8,5H, MU(I1,	FXPT	639
660	1 4H) = F12.8,12H RAD, Q(I1,4H) = F12.8)	FXPT	640
	1006 FORMAT(21H0SINGULAR 4X4 MATRIX.)	FXPT	641
	1007 FORMAT (*07X7 MATRIX FOR *,A5/(/10X,7F16.8))	FXPT	642
	1008 FORMAT(16H0EIGENVALUE = (F16.8,2H, F16.8,21H), EIGENVECTOR = (FXPT	643
	1F16.8,2H, F16.8,2H)/(69X,2H(F16.8,2H, F16.8,2H)))	FXPT	644
665	1013 FORMAT(12H ITERATION =,I4,4H XO=,F16.8,5H DXO=,F16.8,4H YO=,F16.8,	FXPT	645
	15H DYO=,F16.8)	FXPT	646
	1014 FORMAT (/1X,A10,7F15.8)	FXPT	647
	1018 FORMAT (1H1)	FXPT	648
	1017 FORMAT(/15X,7X,* X*,13X,*DX*,13X,* Y*,13X,*DY*,13X,*DS*,13X,*DP/P*	FXPT	649
670	1)	FXPT	650
	C	FXPT	651
	END	FXPT	652

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 LETTER 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.

1	SUBROUTINE GRT(N, C, IN, AUX)	GRT	2
		GRT	3
	COMMON/CONTRL/ERROR,MODE;RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,	CONTRL	2
1	LDLFLG,FIN	CONTRL	3
5	LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDLFLG,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
10		GRR	4
	LOGICAL IND,INF	GRT	7
	LOGICAL INDEFF,INFINN	GRT	8
	DATA FC /1.0E-10/	GRT	9
		GRT	10
15	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
20	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
25	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
30	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
35	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
40	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
45	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
50	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
55	53 DN=BI+DEN	GRT	51
	DM=BI-DEN	GRT	52
	IF (ABS(DN)-ABS(DM)) 57,57,56	GRT	53

	56	DEN=DN	GRT	54
		GO TO 58	GRT	55
60	57	DEN=DM	GRT	56
	58	IF (DEN)55,54,55	GRT	57
	54	DEN=1.	GRT	58
	55	DI=(-2.*X2*DD)/DEN	GRT	59
		H=DI*H	GRT	60
65		RT=RT+H	GRT	61
		IF (ABS(H/RT)-1.0E-8) 75,75,60	GRT	62
60		ASSIGN 7 TO NN	GRT	63
		GO TO 80	GRT	64
	7	IF (ABS(FPRT)-ABS(X2*10.)) 62,61,61	GRT	65
70	61	DI=DI*.5	GRT	66
		H=H*.5	GRT	67
		RT=RT-H	GRT	68
		GO TO 80	GRT	69
	62	X0=X1	GRT	70
75		X1=X2	GRT	71
		X2=FPRT	GRT	72
		D=DI	GRT	73
		GO TO 49	GRT	74
	75	CALL AUX (RT,FRT)	GRT	75
80		IF (ERROR) RETURN	GRT	76
		IF(JK.LT.150) GO TO 76	GRT	77
		IERR=.TRUE.	GRT	78
	74	WRITE (3,28) RT	GRT	79
	28	FORMAT (55H GRT UNABLE TO FIND ROOT IN 150 ITERATIONS.LAST ROOT IS	GRT	80
85		1, E20.8)	GRT	81
	76	IND= INDEFF(RT)	GRT	82
		INF= INFINN(RT)	GRT	83
		IF(IND.OR.INF) IERR=.TRUE.	GRT	84
	78	IF(IERR) WRITE (3,29)	GRT	85
90	29	FORMAT (32H ROOT IS INDEFINITE OR INFINITE.//)	GRT	86
		C(L)=RT	GRT	87
		IF (IN)100,77,100	GRT	88
	77	WRITE (3,26) L,RT,FRT	GRT	89
	100	CONTINUE	GRT	90
95	33	RETURN	GRT	91
	80	JK=JK+1	GRT	92
		IF(150-JK) 75,75,86	GRT	93
	86	CALL AUX (RT,FRT)	GRT	94
		IF (ERROR) RETURN	GRT	95
100		IND= INDEFF(RT)	GRT	96
		INF= INFINN(RT)	GRT	97
		IF(IND.OR.INF) IERR=.TRUE.	GRT	98
		IF(IERR) GO TO 78	GRT	99
		FPRT=FRT	GRT	100
105		IF (L-1)81,91,81	GRT	101
	81	DO 82 I=2,L	GRT	102
		TEM=RT-C(I-1)	GRT	103
		IF (ABS(TEM)-FC) 85,82,82	GRT	104
	82	FPRT=FPRT/TEM	GRT	105
110	91	IF (IN) 89,83,89	GRT	106
		83 WRITE (3,25) L,RT,FRT,FPRT	GRT	107
		89 IF (ABS(FRT)-FC) 90,84,84	GRT	108
		90 IF (ABS(FPRT)-FC) 76,84,84	GRT	109
	84	GO TO NN,(1,2,3,4,5,6,7)	GRT	110

115	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
120	26	FORMAT (I3,2E20.8/1H)	GRT	116
	27	FORMAT (//)	GRT	117
		END	GRT	118

1 SUBROUTINE GRTB(N,C,IN,AUX)
 RETURN
 END

GRTB 2
GRTB 3
GRTB 4

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

```
      HED 2  
      HED 3  
      HED 4  
      HED 5  
      HED 6  
      HED 7
```

1	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
5	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
10	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
15	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
20	IDAT=IDATA	IDAT	13
	RETURN	IDAT	14
	END	IDAT	15

1	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
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
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
15		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
20	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
25	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
30	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
35	* 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
40	* 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
45	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
50	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
55	33 JI = 4	INCR	48
	GO TO 35	INCR	49
34	J1 = 5	INCR	50

	35	IDF = IDAT(M,1)	INCR	51
		INDAT = INFF(JI,N)	INCR	52
60		INEW = IDF + INDAT	INCR	53
		INFF(JI,N) = INEW	INCR	54
	40	KO = KA	INCR	55
		IF (IPP.GT.0) RETURN	INCR	56
		GO TO (41,42,43,44) IGO	INCR	57
65	41	WRITE (3,1) KO,NM,DF,F	INCR	58
		RETURN	INCR	59
	42	WRITE (3,2) KO,NM,IDF,INEW	INCR	60
		RETURN	INCR	61
	43	WRITE (3,3) NM,IDF,INEW	INCR	62
70		RETURN	INCR	63
	44	WRITE (3,4) NM,IDF,INEW	INCR	64
		RETURN	INCR	65
	*	SAVE ORIGINAL PARAMETER	INCR	66
	50	GO TO (51,52,53,54) IGO	INCR	67
75	51	LOC = INFF(14,N) + KA - 1	INCR	68
		SF = STORE(LOC)	INCR	69
		CALL REPFLT(M,2,SF)	INCR	70
		GO TO 20	INCR	71
	52	ISF = IDAT(N,KA)	INCR	72
80		GO TO 55	INCR	73
	53	ISF = INFF(4,N)	INCR	74
		GO TO 55	INCR	75
	54	ISF = INFF(5,N)	INCR	76
	55	CALL REPINT(M,2,ISF)	INCR	77
85		GO TO 20	INCR	78
	*	ITERATIONS COMPLETED. RESTORE ORIGINAL PARAMETER.	INCR	79
	60	GO TO (61,62,62,62) IGO	INCR	80
	61	LOC = INFF(14,M) + 2 - 1	INCR	81
		SF = STORE(LOC)	INCR	82
90		CALL REPFLT(N,KA,SF)	INCR	83
		F = FDAT(M,2)	INCR	84
		DF = 0.	INCR	85
		GO TO 40	INCR	86
	62	ISF = IDAT(M,2)	INCR	87
95		GO TO (63,63,64,65) IGO	INCR	88
	*	SHOULD NOT GET HERE	INCR	89
	63	WRITE (3,5) IGO	INCR	90
		ERROR = .TRUE.	INCR	91
		RETURN	INCR	92
100	64	JI = 4	INCR	93
		GO TO 66	INCR	94
	65	JI = 5	INCR	95
	66	INFF(JI,N) = ISF	INCR	96
		INEW = ISF	INCR	97
105		IDF = 0	INCR	98
		GO TO 40	INCR	99
		*****FORMATS*****	INCR	100
	1	FORMAT (6H *** ,8X,5HINCR ,1X,I3,4X,5H // ,A5,5X,F10.6,5X,	INCR	101
		1 9H VALUE = ,F14.6)	INCR	102
110	2	FORMAT (6H *** ,8X,5HINCR ,1X,I3,4X,5H // ,A5,2X,I10,5X,	INCR	103
		1 9H VALUE = ,I10)	INCR	104
	3	FORMAT (6H *** ,8X,5HINCR ,2X,2HKA,4X,5H // ,A5,5X,I10,5X,	INCR	105
		1 9H VALUE = ,I10)	INCR	106
	4	FORMAT (6H *** ,8X,5HINCR ,2X,2HKB,4X,5H // ,A5,5X,I10,5X,	INCR	107

115

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

ADDRESS	LENGTH	BINARY CONTROL CARDS.
0	5	IDENT INDEFF
5		END

ENTRY POINTS.

INDEFF 1+

ADDRESS	LENGTH	ENT	INDEFF	FUNCTION
0	11160405060600000001			IDENT INDEFF FUNCTION INDEFF(X) ENTRY INDEFF 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
7600-TYPE CPU ASSEMBLY

13 STATEMENTS
0.010 SECONDS

2 SYMBOLS
5 REFERENCES

SYMBOLIC REFERENCE TABLE.

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

ADDRESS	LENGTH	BINARY CONTROL CARDS.
0	5	IDENT INFINN
5		END

ENTRY POINTS.

INFINN 1+

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

16100B LCM 31300B SCM STORAGE USED
7600-TYPE CPU ASSEMBLY

13 STATEMENTS
0.010 SECONDS

2 SYMBOLS
5 REFERENCES

SYMBOLIC REFERENCE TABLE.

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

1	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
5	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
10	RETURN	INFW	6
	END	INFW	7

1	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
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
10	. 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
15	.)	INP	13
	RETURN	INP	14
	END	INP	15

1	SUBROUTINE INTCON(N,NA)	INTCON	2
		INTCON	3
	* NA IS AN ARRAY OF LENGTH N. IT CONTAINS NO MORE THAN 4 BCD	INTCON	4
	* CHARACTERS, REPRESENTING AN INTEGER WITHIN PARENTHESES.	INTCON	5
5	* ROUTINE EXTRACTS THE INTEGER AND RETURNS IT IN THE ARRAY IN	INTCON	6
	* INTEGER FORM.	INTCON	7
		INTCON	8
	DIMENSION NA(1)	INTCON	9
		INTCON	10
10	DO 100 I=1,N	INTCON	11
	DECODE (10,1,NA(I)) NB	INTCON	12
1	FORMAT (3X,A1,6X)	INTCON	13
	IF (NB.EQ.1H) GO TO 50	INTCON	14
	* ASSUMES 1 DIGIT INTEGER	INTCON	15
15	DECODE (10,2,NA(I)) NB	INTCON	16
2	FORMAT (1X,I1,8X)	INTCON	17
	GO TO 70	INTCON	18
	* ASSUMES 2 DIGIT INTEGER	INTCON	19
50	CONTINUE	INTCON	20
20	DECODE (10,3,NA(I)) NB	INTCON	21
3	FORMAT (1X,I2,7X)	INTCON	22
70	NA(I) = NB	INTCON	23
100	CONTINUE	INTCON	24
	RETURN	INTCON	25
25	END	INTCON	26

1	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
5	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
10	COMMON/SWTC/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	SWTC	2
	LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS	SWTC	3
		SWTC	4
	DIMENSION T(49),RW(3),RX(2,3),RY(2,3),RPQ(4)	INV	7
	K1 = INFF(20,M)	INV	8
15	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
20	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
25	15 CALL STOR7(M,T,RW)	INV	19
	RETURN	INV	20
	END	INV	21

1	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
5		INV2	6
	COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMP	DIM	2
		DIM	3
	DATA PI/3.1415926536/	INV2	8
		INV2	9
10	CALL ROTM(M7END,MB,PI)	INV2	10
	NN = M7END	INV2	11
	CALL REF(M,NN)	INV2	12
	RETURN	INV2	13
	END	INV2	14

	IF ((B0.EQ.0.) .OR. (K.EQ.1)) GO TO 2	KICK	54
	DBB=BK/B0	KICK	55
60	T(1,7)=-T(1,6)*DBB	KICK	56
	T(2,7)=-T(2,6)*DBB	KICK	57
4	CALL STOR7(M,T,RW)	KICK	58
	BK=BKK	KICK	59
	RETURN	KICK	60
65	END	KICK	61

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,*E0 =*,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

1	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
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
10	END	KINEM	11

1

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

```
LINMIN 2
LINMIN 3
LINMIN 4
```

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

	NB = 0	LOAD	41
	NF = 0	LOAD	42
60	EMPTY = .FALSE.	LOAD	43
	LDFLG=.TRUE.	LOAD	44
	GO TO 3	LOAD	45
	C	LOAD	46
	1 READ(2,100) (ICARD(IW),IW=1,8)	LOAD	47
65	100 FORMAT(8A10)	LOAD	48
	IF(ICARD(1).EQ.IBLANK) GO TO 201	LOAD	49
	3 DECODE (20,101,ICARD(1)) ISGN,NM,OPN,KAM,KBM	LOAD	50
	101 FORMAT (A1,A5,1X,A5,A3,1X,A3,1X)	LOAD	51
	IF (.NOT.CPYSW) GO TO 501	LOAD	52
70	IF (IPRNT.NE.0) GO TO 501	LOAD	53
	WRITE (3,106) ISGN,NM,OPN,KAM,KBM,(ICARD(IW),IW=3,8)	LOAD	54
	106 FORMAT (6H *** ,A1,A5,2X,A5,1X,A3,1X,A3,5H // ,6A10)	LOAD	55
	IPRNT=IPRNT+1	LOAD	56
	501 IF (.NOT.(NS.EQ.NSETS.AND.INDEF)) GO TO 202	LOAD	57
75	GO TO 74	LOAD	58
	201 IF (.NOT.CPYSW) GO TO 202	LOAD	59
	WRITE (3,107) (ICARD(IW),IW=3,8)	LOAD	60
	107 FORMAT (6H * ,21X,5H // ,6A10)	LOAD	61
	202 IF (NSETS.EQ.0) RETURN	LOAD	62
80	DECODE (60,102,IF(1)) (IB(J),J=1,12)	LOAD	63
	102 FORMAT (12A5)	LOAD	64
	DECODE(60,105,IF(1)) (II(J),J=1,12)	LOAD	65
	105 FORMAT (12R5)	LOAD	66
	NI0 = 0	LOAD	67
85	NB0 = 0	LOAD	68
	NF0 = 0	LOAD	69
	I = 1	LOAD	70
	J = 1	LOAD	71
	GO TO RET (2,5,6,7,8,9)	LOAD	72
90	C.....	LOAD	73
	2 DO 4 NS=1,NSETS	LOAD	74
	ASSIGN 5 TO RET	LOAD	75
	IF (J.GT.12) GO TO 1	LOAD	76
	5 K = 1	LOAD	77
95	ITNS=IT(NS)	LOAD	78
	GO TO (6,7,8,9) ITNS	LOAD	79
	C	LOAD	80
	C FLOATING POINT	LOAD	81
	C TEST FOR SYMBOLIC FLOATING POINT	LOAD	82
100	6 MSK=MASKF.AND.IF(I)	LOAD	83
	C TEST FOR - AND +	LOAD	84
	IF (MSK.NE.IFL(1).AND.MSK.NE.IFL(12)) GO TO 12	LOAD	85
	MSK= SHIFT(IF(I),6)	LOAD	86
	MSK = MASKF.AND.MSK	LOAD	87
105	12 DO 10 L=2,14	LOAD	88
	IF (MSK.EQ.IFL(L)) GO TO 120	LOAD	89
	10 CONTINUE	LOAD	90
	C SYMBOLIC. CIRCULAR LEFT SHIFT 5 CHARACTERS, FLAG BY FILLING FIRST	LOAD	91
	C 2 CHARACTERS WITH ZEROES.	LOAD	92
110	IFCD(NF0+1)= SHIFT(IF(I),30)	LOAD	93
	IFCD(NF0+1)=IFCD(NF0+1).AND.MASKL	LOAD	94
	GO TO 110	LOAD	95
	C FLOATING POINT NUMBER	LOAD	96
	120 DECODE (10,103,IF(I)) FCD(NF0+1)	LOAD	97

115	110	K = K + 1	LOAD	98
	103	FORMAT(F10.5)	LOAD	99
		I=I+1	LOAD	100
		J=J+2	LOAD	101
		NF0=NF0+1	LOAD	102
120		IF(K.GT.ND(NS)) GO TO 61	LOAD	103
		IF(I.LE.6) GO TO 6	LOAD	104
		CALL STDAT(M,1,NF + 1,NF0, FCD)	LOAD	105
		NF = NF + NF0	LOAD	106
		ASSIGN 6 TO RET	LOAD	107
125		GO TO 1	LOAD	108
	61	CALL STDAT(M,1,NF+1,NF0, FCD)	LOAD	109
		NF=NF+NF0	LOAD	110
		NF0=0	LOAD	111
		GO TO 4	LOAD	112
130	C		LOAD	113
	C	BCD	LOAD	114
	7	BCD(NB0+1) = IB(J)	LOAD	115
		K = K + 1	LOAD	116
	72	J = J + 1	LOAD	117
135		I = J/2 + 1	LOAD	118
		NB0 = NB0 + 1	LOAD	119
		IF(K.GT.ND(NS)) GO TO 73	LOAD	120
		IF(J.LE.12) GO TO 7	LOAD	121
		CALL STDAT(M,2,NB + 1,NB0,BCD)	LOAD	122
140		NB = NB + NB0	LOAD	123
		ASSIGN 7 TO RET	LOAD	124
		IF (NS.NE.NSETS) GO TO 1	LOAD	125
		IF (INDEF) ND(NS)=ND(NS)+12	LOAD	126
		GO TO 1	LOAD	127
145	73	J = 2*(J/2) + 1	LOAD	128
		CALL STDAT(M,2,NB+1,NB0,BCD)	LOAD	129
		NB=NB+NB0	LOAD	130
		NB0=0	LOAD	131
		GO TO 4	LOAD	132
150			LOAD	133
	C	LAST SET WAS BCD OF INDEFINITE LENGTH	LOAD	134
	C	WARNING—AS PRESENTLY CODED(7/1975) , NO INTEGER SETS MAY BE	LOAD	135
	C	INCLUDED. TO BE SAFE, THERE SHOULD ONLY BE A SINGLE SET LOADED.	LOAD	136
	C	E.G. CALL INP(M,MXPMY,-0,1,-2,0)	LOAD	137
155			LOAD	138
	74	EMPTY=.TRUE.	LOAD	139
		INDEF=.FALSE.	LOAD	140
		NBC=NB	LOAD	141
		LFMX=LBC+NBC	LOAD	142
160		NTOT=LFMX-LFILE	LOAD	143
		INFF(17,M)=NBC	LOAD	144
		INFF(18,M)=NTOT	LOAD	145
		LFILE=LFMX	LOAD	146
		IF(LFILE.LE.LMAX) RETURN	LOAD	147
165		WRITE(3,75)	LOAD	148
		NSTO = LMAX - LFILE	LOAD	149
		NINFF = MAX - MIN	LOAD	150
		WRITE (3,76) LMAX,LFILE,NSTO,MAX,NINFF,M	LOAD	151
	75	FORMAT (6H0*****,*STORE OVERFLOWED.*	LOAD	152
170	1 *	INCREASE STORE ARRAY AND LMAX.*)	LOAD	153
	76	FORMAT(/1X*CORE USE SUMMARY*,27X,*MAXIMUM*,17X,*USED*,8X,*UNUSED*/	LOAD	154

	1 34X,*STORE (ELEMENT STORAGE)*,14X,I10,1X,* (LMAX)*,4X,I10,5X,I10/	LOAD	155
	2 34X,*INFF (ELEMENT DEFINITIONS)*,10X,I10,1X,* (MAX)*,5X,I10,5X,	LOAD	156
	3 I10)	LOAD	157
175	ERROR=.TRUE.	LOAD	158
	RETURN	LOAD	159
	C	LOAD	160
	C INTEGER	LOAD	161
180	8 DECODE (10,104,II(J)) ICD(NI0+1)	LOAD	162
	104 FORMAT (5X,I5)	LOAD	163
	K = K + 1	LOAD	164
	82 J = J + 1	LOAD	165
	I = J/2 + 1	LOAD	166
185	NI0 = NI0 + 1	LOAD	167
	IF(K.GT.ND(NS)) GO TO 83	LOAD	168
	IF (J .LE.12) GO TO 8	LOAD	169
	CALL STDAT(M,3,NI + 1,NI0,ICD)	LOAD	170
	NI = NI + NI0	LOAD	171
190	ASSIGN 8 TO RET	LOAD	172
	GO TO 1	LOAD	173
	83 J = 2*(J/2) + 1	LOAD	174
	CALL STDAT(M,3,NI+1,NI0,ICD)	LOAD	175
	NI=NI+NI0	LOAD	176
195	NI0=0	LOAD	177
	GO TO 4	LOAD	178
	C	LOAD	179
	C SKIP ND(NS) WORDS	LOAD	180
200	9 K = K + 1	LOAD	181
	I = I + 1	LOAD	182
	J = J + 2	LOAD	183
	IF(K.GT.ND(NS)) GO TO 4	LOAD	184
	IF (I.LE.6) GO TO 9	LOAD	185
	ASSIGN 9 TO RET	LOAD	186
205	GO TO 1	LOAD	187
	4 CONTINUE	LOAD	188
	C.....	LOAD	189
	IF (OPNAME.EQ.4HEVEC.OR.OPNAME.EQ.4HEQIL) GO TO 310	LOAD	190
	IF (KIND.NE.4H) RETURN	LOAD	191
210	C KIND HAS TO BE DETERMINED BY BCD INPUT.	LOAD	192
	IJ = 1	LOAD	193
	NCD=IABS(MDAT(M,1))	LOAD	194
	IF (NCD.NE.0) KND=INFF(20,NCD)	LOAD	195
	IF (OPNAME.EQ.1H=) IJ = 2	LOAD	196
215	IF (OPNAME.EQ.1H=.AND.KND.EQ.4HSCAL) GO TO 315	LOAD	197
	DO 305 J=1,NBC,IJ	LOAD	198
	NCD=IABS(MDAT(M,J))	LOAD	199
	C IF ELEMENT NO.=0,IT HAS NOT YET BEEN DEFINED	LOAD	200
	IF (NCD.EQ.0) GO TO 320	LOAD	201
220	307 KND = INFF(20,NCD)	LOAD	202
	IF (KND.EQ.2HR7.OR.KND.EQ.3HROT) GO TO 320	LOAD	203
	305 CONTINUE	LOAD	204
	C MAKE ELEMENT A MXY KIND	LOAD	205
	INFF(20,M) = 3HMX	LOAD	206
225	ITE = 1	LOAD	207
	ITYP = INFF(21,NCD)	LOAD	208
	INFF(21,M)=ITYP	LOAD	209
	IF(ITYP.NE.4HSNGL) ITE=2	LOAD	210
		LOAD	211

230	IF (OPNAME.NE.3HCYB) NQ = 12*ITE + KADDR	LOAD	212
	INFF(22,M) = 2	LOAD	213
	GO TO 325	LOAD	214
310	ICD(2) = BDAT(M,2)	LOAD	215
	IF (ICD(2).NE.4HNOSV) RETURN	LOAD	216
	ICD(1) = BDAT(M,1)	LOAD	217
235	LBC = LQ	LOAD	218
	INFF(16,M) = LBC	LOAD	219
	CALL STDAT(M,2,1,2,ICD)	LOAD	220
	NQ = 0	LOAD	221
	LFILE = LBC + NBC - 1	LOAD	222
240	GO TO 330	LOAD	223
	C KIND IS SCALAR	LOAD	224
315	INFF(20,M) = KND	LOAD	225
	NQ = 1	LOAD	226
	TYPE = INFF(21,NCD)	LOAD	227
245	INFF(21,M) = TYPE	LOAD	228
	IF (TYPE.NE.4HSNGL) NQ = 2	LOAD	229
	GO TO 326	LOAD	230
	C SET KIND TO A 7X7	LOAD	231
320	INFF(20,M) = 2HR7	LOAD	232
250	ITE = 1	LOAD	233
	C IF ELEMENT UNDEFINED, LEAVE IT TYPE SINGLE	LOAD	234
	IF(NCD.EQ.0) GO TO 324	LOAD	235
	ITYP = INFF(21,NCD)	LOAD	236
	INFF(21,M) = ITYP	LOAD	237
255	IF (ITYP.NE.4HSNGL) ITE=2	LOAD	238
324	IF(OPNAME.NE.3HCYB) NQ=49*ITE+KADD	LOAD	239
	INFF(22,M)=7	LOAD	240
325	INFF(23,M)=7	LOAD	241
326	LQ = LFILE + 1	LOAD	242
260	LFILE = LQ + NQ - 1	LOAD	243
	INFF(10,M) = LQ	LOAD	244
330	INFF(11,M) = NQ	LOAD	245
	NTOT = NQ + NFL + NIN + NBC	LOAD	246
	INFF(18,M) = NTOT	LOAD	247
265	RETURN	LOAD	248
	END	LOAD	249

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

266 I 91 NON-INNER LOOP BEGINNING AT THIS CARD IS ENTERED FROM OUTSIDE ITS RANGE.

1	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
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
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
15	C NLOC = 3	LOC	16
	C	LOC	17
	DIMENSION LIST(1),LOCA(1)	LOC	18
	LEVEL 2, LIST	LOC	19
	DATA MAX/100/	LOC	20
20	C	LOC	21
	DO 1 K=1,MAX	LOC	22
	1 LOCA(K) = 0	LOC	23
	C	LOC	24
	J = 0	LOC	25
25	DO 2 I=1,NL	LOC	26
	LL=LIST(I)	LOC	27
	NAMEI = MNAME(LL)	LOC	28
	IF (NAME.NE.NAMEI) GO TO 2	LOC	29
	J = J + 1	LOC	30
30	LOCA(J) = I	LOC	31
	IF (J.EQ.MAX) GO TO 3	LOC	32
	2 CONTINUE	LOC	33
	C	LOC	34
	3 NLOC = J	LOC	35
35	LOC = LOCA(NUM)	LOC	36
	RETURN	LOC	37
	END	LOC	38

1		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	LOC2	4
5		DATA MAX/100/	LOC2	4
	C		LOC2	5
		DO 1 K=1,MAX	LOC2	6
	1	LOCA(K) = 0	LOC2	7
	C		LOC2	8
10		J = 0	LOC2	9
		DO 2 I=1,NL	LOC2	10
		LL=LIST(I)	LOC2	11
		NAMEI = MNAME(LL)	LOC2	12
		IF (NAME.NE.NAMEI) GO TO 2	LOC2	13
15		J = J + 1	LOC2	14
		LOCA(J) = I	LOC2	15
		IF (J.EQ.MAX) GO TO 3	LOC2	16
	2	CONTINUE	LOC2	17
	C		LOC2	18
20	3	NLOC = J	LOC2	19
		LOCC = LOCA(NUM)	LOC2	20
		RETURN	LOC2	21
		END	LOC2	22

1		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
5			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
10	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
15		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
20		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
25	20	CONTINUE	LOCS	19
		RETURN	LOCS	20
		END	LOCS	21

1		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
5	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
10	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
15	C		LPAR	10
		ITS = 0	LPAR	11
		C = COS(THETA)	LPAR	12
		S = SIN(THETA)	LPAR	13
		E = EXP(THETA)	LPAR	14
20		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
25		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
30		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
35	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
40		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
45		GO TO 100	LPAR	40
	120	ERL = LT - LT0	LPAR	41
		RETURN	LPAR	42
	130	WRITE (3,1000)	LPAR	43
		ERROR = .TRUE.	LPAR	44
50		RETURN	LPAR	45
	1000	FORMAT (*0DLSS COMPUTATIONS DO NOT CONVERGE.*)	LPAR	46
		END	LPAR	47

1	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
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
10	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
15	* 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
20	10 CONTINUE	MAGMATS	17
		MAGMATS	18
20	* STORE MAGNET INPUT	MAGMATS	19
	* COMPUTE AND STORE MAGNET MATRICES	MAGMATS	20
	M1 = M - 1	MAGMATS	21
	DO 25 K=1,KA	MAGMATS	22
25	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
30	* 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
35	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
40	END	MAGMATS	38

1	SUBROUTINE MAGNET(M,PAR)	MAGNET	2
	C SET UP MAGNET MATRICES.	MAGNET	3
		MAGNET	4
	COMMON/SWTC/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	SWTC	2
5	LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS	SWTC	3
		SWTC	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
10	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
15	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
20	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
25	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
30	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
35	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
40	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
45	* 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
50	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
55	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	MAGNET	57
	C FOCUSING MATRIX	MAGNET	58
60	2 S = SIN(PHI)	MAGNET	59
	C = COS(PHI)	MAGNET	60
	F(1,1) = C	MAGNET	61
	F(1,2) = S/K	MAGNET	62
	F(1,3) = (CURV*(1.-C)/(K*K))*BEND	MAGNET	63
65	F(2,1) = -K*S	MAGNET	64
	F(2,2) = C	MAGNET	65
	F(2,3) = (CURV*S/K)*BEND	MAGNET	66
	IF (ID.EQ.1) PL = -(PHI-S)*CURV*CURV/(K*K*K)	MAGNET	67
	GO TO 10	MAGNET	68
70	C DEFOCUSING MATRIX	MAGNET	69
	3 E = EXP(PHI)	MAGNET	70
	EI = 1./E	MAGNET	71
	CH = 0.5*(E+EI)	MAGNET	72
	SH = 0.5*(E-EI)	MAGNET	73
75	D(1,1) = CH	MAGNET	74
	D(1,2) = SH/K	MAGNET	75
	D(2,1) = SH*K	MAGNET	76
	D(1,3) = (CURV*(CH-1.)/(K*K))*BEND	MAGNET	77
	D(2,2) = CH	MAGNET	78
80	D(2,3) = (CURV*SH/K)*BEND	MAGNET	79
	IF (ID.EQ.1) PL = -(SH-PHI)*CURV*CURV/(K*K*K)	MAGNET	80
	GO TO 10	MAGNET	81
	C KP = 0 CASE	MAGNET	82
85	7 A(1,1) = 1.	MAGNET	83
	A(1,2) = L	MAGNET	84
	A(1,3) = (0.5*CURV*L*L)*BEND	MAGNET	85
	A(2,1) = 0.	MAGNET	86
	A(2,2) = 1.	MAGNET	87
	A(2,3) = (CURV*L)*BEND	MAGNET	88
90	IF (ID.EQ.1) PL = -L*L*L*CURV*CURV/6.	MAGNET	89
	C CORRECT FOR EDGE FOCUSING.	MAGNET	90
	10 A(1,1)=A(1,1)+A(1,2)*EFIN	MAGNET	91
	A(2,1) = A(1,1)*EFOUT + A(2,1) + A(2,2)*EFIN	MAGNET	92
	A(2,2) = A(1,2)*EFOUT + A(2,2)	MAGNET	93
95	A(2,3) = A(1,3)*EFOUT + A(2,3)	MAGNET	94
	IF (ID.EQ.2) GO TO 20	MAGNET	95
	C SET UP RADIAL TRANSFER MATRIX.	MAGNET	96
	DO 15 I=1,2	MAGNET	97
	DO 15 J=1,3	MAGNET	98
100	RX(I,J)=A(I,J)	MAGNET	99
	15 CONTINUE	MAGNET	100
	IF (IVERT.EQ.0) GO TO 16	MAGNET	101
	IVERT = 0	MAGNET	102
	GO TO 26	MAGNET	103
105	16 CURV = 0.	MAGNET	104
	ID = 2	MAGNET	105
	EFIN=-EFIN	MAGNET	106
	EFOUT=-EFOUT	MAGNET	107
	FC=-1.	MAGNET	108
110	GO TO 1	MAGNET	109
	C SET UP VERTICAL TRANSFER MATRIX.	MAGNET	110
	20 DO 25 I=1,2	MAGNET	111
	DO 25 J=1,3	MAGNET	112
	RY(I,J)=A(I,J)	MAGNET	113

115	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
120		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
125		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
130		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
135		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
140		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
145		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
150		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.

1	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
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
10	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
15	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

1	SUBROUTINE MAGS(M,N,Z)	MAGS	2
		BLANK	2
	LEVEL 2, STORE,INFF,IWORK	BLANK	3
	COMMON STORE(48000),IWORK(10)	86MARSIZ	1
5	DIMENSION INFF(24,2000)	86MARSIZ	2
	EQUIVALENCE (INFF,STORE)	86MARSIZ	3
		BLANK	5
	COMMON/SWTC/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	SWTC	2
	LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS	SWTC	3
10		SWTC	4
	COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYML	DIM	2
		DIM	3
	COMMON/MI /MLL(3) \$ LEVEL 2,MLL	MAGSLL	1
	LOGICAL BSW	MAGS	6
15	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
20	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
25	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
30	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
35	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
40	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
45	* 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
50	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
55	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
60		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
65		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
70	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
75		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
80		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
85		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
90		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
95	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
100		GO TO 4	MAGS	91
		END	MAGS	92

1	SUBROUTINE MAGSRV(M0,NAME0,KA0,KB0,NAME1)	MAGSRV	2
	COMMON/SWCH/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	SWCH	2
	LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS	SWCH	3
		SWCH	4
5	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
10	RETURN	MAGSRV	9
	END	MAGSRV	10

1	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
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,MSYMP	DIM	2
		DIM	3
	DIMENSION T(49),TR(49),RW(3),PAR(1)	MAGV	9
10	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
15	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

1

SUBROUTINE MAP(V,PAR)

RETURN

END

MAP

2

MAP

3

MAP

4

1 SUBROUTINE MAP1(M,V,PAT)
 RETURN
 END

MAP1 2
MAP1 3
MAP1 4

1			MAP1	5
		SUBROUTINE MAP3(V,PAR)	MAP3	2
			MAP3	3
	C	SIMULATE BEAM-BEAM FORCE WITH ROUND BEAM	MAP3	4
5	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
10		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

1

SUBROUTINE MAP8(V,PAR)
RETURN
END

MAP8 2
MAP8 3
MAP8 4

1

SUBROUTINE MAP9(V,PAR)
RETURN
END

MAP9

2

MAP9

3

MAP9

4

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

MATRSV 2
MATRSV 3
MATRSV 4

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.

1	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
5	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
10	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
15	RETURN	MDAT	11
	END	MDAT	12

1	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
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
10		BLANK	5
	COMMON/SWTCB/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	SWTCB	2
	LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS	SWTCB	3
		SWTCB	4
	COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,	CONTRL	2
15	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
20	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
25	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
30	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
35	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
40	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
45	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
50	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
55	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	MESH	49
	LP(I,JX) = 1	MESH	50
60	20 CONTINUE	MESH	51
	IF (JX.EQ.2) GO TO 30	MESH	52
	MHS=.TRUE.	MESH	53
	GO TO 60	MESH	54
	30 LMX = LL(1,JX)	MESH	55
65	DO 31 I=2,KE	MESH	56
	IF (LL(I,JX).GT.LMX) LMX = LL(I,JX)	MESH	57
	31 CONTINUE	MESH	58
	LMM = 1	MESH	59
	VPR = .TRUE.	MESH	60
70	GO TO 60	MESH	61
	C VPAR CALCULATION. STRAIGHT LINE	MESH	62
	48 CONTINUE	MESH	63
	C TEST TO SEE IF FINISHED	MESH	64
	KE = KEJ(JX)	MESH	65
75	IF(LMM.EQ.LMX) GO TO 63	MESH	66
	LMM = LMM + 1	MESH	67
	DO 36 I=1,KE	MESH	68
	FPAR(I,JX) = FPAR(I,JX) + FDEL(I,JX)	MESH	69
	CALL STDAT(JN(I,JX),1,IND(I,JX),1,FPAR(I,JX))	MESH	70
80	36 CONTINUE	MESH	71
	GO TO 60	MESH	72
	C FULL MESH	MESH	73
	46 CONTINUE	MESH	74
	KE = KEJ(JX)	MESH	75
85	KV = KE	MESH	76
	I = 1	MESH	77
	49 IF (LP(KV,JX).NE.LL(KV,JX))GO TO 55	MESH	78
	C LOOP COMPLETED. RESET	MESH	79
	LP(KV,JX) = 1	MESH	80
90	FPAR(KV,JX) = FMIN(KV,JX)	MESH	81
	CALL STDAT(JN(KV,JX),1,IND(KV,JX),1,FPAR(KV,JX))	MESH	82
	KV = KV - 1	MESH	83
	IF (I.EQ.KE) GO TO 64	MESH	84
	I = I + 1	MESH	85
95	GO TO 49	MESH	86
	55 LP(KV,JX) = LP(KV,JX) + 1	MESH	87
	FPAR(KV,JX) = FPAR(KV,JX) + FDEL(KV,JX)	MESH	88
	CALL STDAT(JN(KV,JX),1,IND(KV,JX),1,FPAR(KV,JX))	MESH	89
	60 KA = 1	MESH	90
100	WRITE (3,100)	MESH	91
	DO 62 I=1,KE	MESH	92
	WRITE (3,101) IND(I,JX),NM(I,JX),FPAR(I,JX)	MESH	93
	62 CONTINUE	MESH	94
	CALL SUBR(3)	MESH	95
105	MSSW = .TRUE.	MESH	96
	MODS=.TRUE.	MESH	97
	RETURN	MESH	98
	C FINISHED. REPLACE ORIGINAL VALUES OF VARIABLES	MESH	99
	63 VPR=.FALSE.	MESH	100
110	MODS=.FALSE.	MESH	101
	GO TO 65	MESH	102
	64 MHS=.FALSE.	MESH	103
	MODS=.FALSE.	MESH	104
	65 DO 70 I=1,KE	MESH	105

115	F(I) = VSAV(I,JX)	MESH	106
	CALL STDAT(JN(I,JX),1,IND(I,JX),1,F(I))	MESH	107
70	CONTINUE	MESH	108
	IF(MHS) RETURN	MESH	109
	IF (VPR) RETURN	MESH	110
120	MSSW = .FALSE.	MESH	111
	MODE = INFF(19,M)	MESH	112
	RETURN	MESH	113
100	FORMAT (/15H CURRENT VALUES/)	MESH	114
101	FORMAT (5X,I5,4H OF ,A5,3H = ,F10.5)	MESH	115
125	END	MESH	116

1	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
5	F=F * FACTOR	MFD	6
	CALL REPFLT(M,J,F)	MFD	7
	RETURN	MFD	8
	END	MFD	9

1	SUBROUTINE MIEXP(M,J,NDAT,NELS,ML)	MIEXP	2
		MIEXP	3
	DIMENSION ML(1),KBEG(10),NREP(10),NR(10)	MIEXP	4
	LEVEL 2, ML	MIXL	1
5		MIXL	2
	INTEGER BK,BDAT	MIEXP	5
	DATA IBLANK/10H /	MIEXP	6
	DATA MASK/00000077000000000000B/	MIEXP	7
	DATA IPL/00000051000000000000B/	MIEXP	8
10	DATA IPR/00000052000000000000B/	MIEXP	9
		MIEXP	10
	C EXPAND PARENTHESES AND STORE IN ML. KI INPUT IS NUMBER OF	MIEXP	11
	C 5-CHARACTER BCD DATA TO PROCESS, ON OUTPUT IS NUMBER OF M'S STORED.	MIEXP	12
		MIEXP	13
15	I=0	MIEXP	14
	LOOP=0	MIEXP	15
	K=1	MIEXP	16
	1 IF (K.GT.NDAT) GO TO 5	MIEXP	17
	BK=BDAT(M,K+J-1)	MIEXP	18
20	IF (BK.EQ.IBLANK) GO TO 6	MIEXP	19
	LBK=BK.AND.MASK	MIEXP	20
	IF (LBK.EQ.IPL) GO TO 2	MIEXP	21
	IF (LBK.EQ.IPR) GO TO 3	MIEXP	22
	I=I+1	MIEXP	23
25	ML(I)=MDAT(M,K+J-1)	MIEXP	24
	6 K=K+1	MIEXP	25
	GO TO 1	MIEXP	26
	2 DECODE(10,1000,BK) NUM	MIEXP	27
	1000 FORMAT (I3,7X)	MIEXP	28
30	IF(NUM.EQ.0) GO TO 3	MIEXP	29
	LOOP=LOOP+1	MIEXP	30
	KBEG(LOOP)=K+1	MIEXP	31
	NREP(LOOP)=NUM	MIEXP	32
	NR(LOOP)=0	MIEXP	33
35	K=K+1	MIEXP	34
	GO TO 1	MIEXP	35
	3 NR(LOOP)=NR(LOOP)+1	MIEXP	36
	IF(NR(LOOP).EQ.NREP(LOOP)) GO TO 4	MIEXP	37
	K=KBEG(LOOP)	MIEXP	38
40	GO TO 1	MIEXP	39
	4 K=K+1	MIEXP	40
	LOOP=LOOP-1	MIEXP	41
	GO TO 1	MIEXP	42
	5 NELS = I	MIEXP	43
45	RETURN	MIEXP	44
	END	MIEXP	45

1	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
5	DIMENSION INFF(24,2000)	86MARSIZ	2
	EQUIVALENCE (INFF,STORE)	86MARSIZ	3
		BLANK	5
	DIMENSION MI(1),ME(16000)	NOV3MFLI	1
	COMMON/ME/ME	MIFILL	2
10	LEVEL 2,MI,ME	MIFILL	3
		MIFILL	4
	INTEGER OPL	MIFILL	5
		MIFILL	6
	CALL MIEXP(M,J,NDAT,NELS,ME)	MIFILL	7
15	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
20		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
25		MIFILL	18
	DO 26 N=1,NELS	MIFILL	19
26	MI(N) = ME(N)	MIFILL	20
		MIFILL	21
	27 KE = NELS	MIFILL	22
30	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
35		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
40	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
45	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
50		MIFILL	43
	12 MI(L)=ME(NL-KBL+1)	MIFILL	44
		MIFILL	45
	13 KBL=KBL+1	MIFILL	46
		MIFILL	47
55	IF (KBL.LE.NL) GO TO 14	MIFILL	48
	KBL=1	MIFILL	49
	14 INFF(5,ML)=KBL	MIFILL	50

		GO TO 1	MIFILL	51
60	11	KEL=KE-L	MIFILL	52
		N0=1	MIFILL	53
		IF(KAL.GE.0) GO TO 4	MIFILL	54
		N0=0	MIFILL	55
65		KE=KE+1	MIFILL	56
		KREF=-KREF	MIFILL	57
	4	DO 5 N=1,KEL	MIFILL	58
	5	MI(KE+NL-N)=MI(KE-N+N0)	MIFILL	59
70		IF(KREF.LT.0) GO TO 7	MIFILL	60
		DO 6 N=1,NL	MIFILL	61
	6	MI(L+N-1)=ME(N)	MIFILL	62
		GO TO 9	MIFILL	63
75		DO 8 N=1,NL	MIFILL	64
	8	MI(L+N-1)=ME(NL-N+1)	MIFILL	65
			MIFILL	66
			MIFILL	67
			MIFILL	68
			MIFILL	69
			MIFILL	70
			MIFILL	71
			MIFILL	72
80	9	KE=KE+NL-1	MIFILL	73
		IF (KAL.LT.0) MI(L+NL)=999999	MIFILL	74
		L=L-1	MIFILL	75
		GO TO 1	MIFILL	76
			MIFILL	77
85	3	KEL=KE-L	MIFILL	78
		DO 10 N=1,KEL	MIFILL	79
	10	MI(L+N-1)=MI(L+N)	MIFILL	80
		KE=KE-1	MIFILL	81
		L=L-1	MIFILL	82
90		KREF=-KREF	MIFILL	83
		GO TO 1	MIFILL	84
			MIFILL	85
	2	NELS=KE	MIFILL	86
			MIFILL	87
95		RETURN	MIFILL	88
		END	MIFILL	89

1			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
5		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
10		DO 1 I=1,K	MLIST	8
	1	MI(I)=ELNUM(ML(I))	MLISTL	4
			MLISTL	5
		RETURN	MLIST	10
		END	MLIST	11

1	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
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
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
15	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,MSYML	DIM	2
20		DIM	3
	DIMENSION MI(1)	MMM	16
	LEVEL 2,MI	MMML	1
		MMML	2
		MMM	17
25	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
30	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
35	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
40	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
45	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
50	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
55	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
60	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,KA)	MMM	51
	10 CONTINUE	MMM	52
65	11 RETURN	MMM	53
	END	MMM	54

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

48 I MI ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

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

1		SUBROUTINE MODQ (M,N,V,W)	MODQ	2
	C		MODQ	3
		COMMON/DIM/LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMP	MODQ	4
		COMMON /MJ/ ML \$ LEVEL 2, ML	MODQLL	1
5		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
10		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
15		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
20		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
25		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
30		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
35		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
40	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
45		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
50	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
55		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.

1	SUBROUTINE MOVE(M,N,V)	MOVE	2
		MOVE	3
	COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYML	DIM	2
		DIM	3
5		MOVE	5
		BLANK	2
	LEVEL 2, STORE,INFF,IWORK	BLANK	3
	COMMON STORE(48000),IWORK(10)	86MARSIZ	1
	DIMENSION INFF(24,2000)	86MARSIZ	2
10	EQUIVALENCE (INFF,STORE)	86MARSIZ	3
		BLANK	5
	C	MOVTOG	2
	COMMON/MOVTOG/TOG	MOVTOG	3
	LOGICAL TOG	MOVTOG	4
15	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
20	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
25	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
30	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
35	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
40	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
45	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
50	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
55	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
60	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
65	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
70	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
75	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
80	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
85	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
90	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
95	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
100	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
105	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
110		MOVE	92
2	ML(1)=M7END-1	MOVE	93
	ML(2)=M7END-2	MOVE	94
	ML(3)=M7END-6	MOVE	95
		MOVE	96

115			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
120		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
125		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
130	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
135		GO TO 8	MOVE	115
	20	IF (.NOT.SXSW) GO TO 21	MOVE	116
		CALL SOTP(N,V,PAT)	MOVE	117
		GO TO 14	MOVE	118
	21	CALL DATA(MQ,1,1,6,PP)	MOV883	5
140		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
145	C		MOVE	121
	8	XIN=V(1)	MOVE	122
		YIN=V(3)	MOVE	123
		XOUT=W(1)	MOVE	124
		YOUT=W(3)	MOVE	125
150		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
155			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
160		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
165		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
170		ML(2)=N	MOVE	145
		ML(3) = M7END - 7	MOVE	146

	CALL RTRV7(M7END-7,T,RW)	MOVE	147
		MOVELL	5
		MOVELL	6
175	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
180	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
185	TOG = .TRUE.	MY3MOVE	19
	RETURN	MY3MOVE	20
	115 CALL RANSET(NRN)	MY3MOVE	21
	TOG = .FALSE.	MY3MOVE	22
	RETURN	MOVE	158
190	END	MOVE	159

1

SUBROUTINE MPCONJ(A,B,C,DC)
RETURN
END

MOVE 160
MPCONJ 2
MPCONJ 3
MPCONJ 4

1

SUBROUTINE MPY2(A,B,C)
RETURN
END

MPY2 2
MPY2 3
MPY2 4

1	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
5		MXD	6
	COMMON/SWTCB/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
10	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
15	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
20	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
25	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
30	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
35	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
40	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
45	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
50	RETURN	MXD	49
	END	MXD	50

1	SUBROUTINE MXDIV(V, IN, IM, DET, IDIM)	MXDIV	2
	C MATRIX DIVISION	MXDIV	3
	C REPLACES B BY (A INVERSE)*B. DESTROYS A.	MXDIV	4
	C A IS N BY N, B IS N BY M. A AND B ARE STORED IN V(N,N+M).	MXDIV	5
5	C N MUST NOT EXCEED 4 .	MXDIV	6
	INTEGER PERM,P	MXDIV	7
	DIMENSION PERM(4)	MXDIV	8
	DIMENSION V(IDIM,1)	MXDIV	9
	N=IN	MXDIV	10
10	M=MAX0(IM,0)	MXDIV	11
	C	MXDIV	12
	NM1=N-1	MXDIV	13
	NP1=N+1	MXDIV	14
	NPM=N+M	MXDIV	15
15	200 DO 201 I=1,N	MXDIV	16
	PERM(I)=I	MXDIV	17
	201 CONTINUE	MXDIV	18
	DETERM=1.0	MXDIV	19
	P=1	MXDIV	20
20	202 CONTINUE	MXDIV	21
	IMIN=P+1	MXDIV	22
	C FIND PIVOT	MXDIV	23
	PIVOT=0.	MXDIV	24
	DO 209 I=P,N	MXDIV	25
25	DO 208 J=P,N	MXDIV	26
	SAVE= ABS(V(I,J))	MXDIV	27
	IF (SAVE .LE. PIVOT) GO TO 208	MXDIV	28
	PIVOT=SAVE	MXDIV	29
	IBIG=I	MXDIV	30
30	JBIG=J	MXDIV	31
	208 CONTINUE	MXDIV	32
	209 CONTINUE	MXDIV	33
	IF (PIVOT .GT. 0.0) GO TO 210	MXDIV	34
	DETERM=0.0	MXDIV	35
35	GO TO 290	MXDIV	36
	C ROW EXCHANGE AND ROW REDUCTION	MXDIV	37
	210 PIVOT=V(IBIG,JBIG)	MXDIV	38
	DETERM=DETERM*PIVOT	MXDIV	39
	DO 219 J=P,NPM	MXDIV	40
40	SAVE=V(IBIG,J)/PIVOT	MXDIV	41
	V(IBIG,J)=V(P,J)	MXDIV	42
	V(P,J)=SAVE	MXDIV	43
	219 CONTINUE	MXDIV	44
	IF (IBIG .NE. P) DETERM=-DETERM	MXDIV	45
45	220 CONTINUE	MXDIV	46
	IF (P .GE. N) GO TO 250	MXDIV	47
	C COLUMN EXCHANGE	MXDIV	48
	IF (JBIG .EQ. P) GO TO 230	MXDIV	49
	ISAVE=PERM(P)	MXDIV	50
50	PERM(P)=PERM(JBIG)	MXDIV	51
	PERM(JBIG)=ISAVE	MXDIV	52
	DO 229 I=1,N	MXDIV	53
	SAVE=V(I,P)	MXDIV	54
	V(I,P)=V(I,JBIG)	MXDIV	55
55	V(I,JBIG)=SAVE	MXDIV	56
	229 CONTINUE	MXDIV	57
	DETERM=-DETERM	MXDIV	58

	230	CONTINUE		MXDIV	59
		V(P,P)=1.0		MXDIV	60
60	C		BLOCK REDUCTION	MXDIV	61
		DO 239 I=IMIN,N		MXDIV	62
		DO 238 J=IMIN,NPM		MXDIV	63
		V(I,J)=V(I,J)-V(I,P)*V(P,J)		MXDIV	64
	238	CONTINUE		MXDIV	65
65		V(I,P)=0.0		MXDIV	66
	239	CONTINUE		MXDIV	67
		P=P+1		MXDIV	68
	249	GO TO 202		MXDIV	69
	C		BACK SUBSTITUTION	MXDIV	70
70	250	CONTINUE		MXDIV	71
		IF (M .LE. 0) GO TO 290		MXDIV	72
		DO 259 K=NP1,NPM		MXDIV	73
		V(N,1)=V(N,K)		MXDIV	74
		DO 253 P=1,NM1		MXDIV	75
75		I=N-P		MXDIV	76
		V(I,1)=V(I,K)		MXDIV	77
		DO 252 J=I,NM1		MXDIV	78
		V(I,1)=V(I,1)-V(J+1,1)*V(I,J+1)		MXDIV	79
	252	CONTINUE		MXDIV	80
80	253	CONTINUE		MXDIV	81
		DO 256 J=1,N		MXDIV	82
		P=PERM(J)		MXDIV	83
		V(P,K)=V(J,1)		MXDIV	84
	256	CONTINUE		MXDIV	85
85	259	CONTINUE		MXDIV	86
	C			MXDIV	87
	290	CONTINUE		MXDIV	88
		DET=DETERM		MXDIV	89
		RETURN		MXDIV	90
90		END		MXDIV	91

1	SUBROUTINE MXM6(A,B,C)	MXM6	2
		MXM6	3
	C MULTIPLES 6 X 6 MATRICES	MXM6	4
	DIMENSION A(6,1),B(6,1),C(6,1),CC(6,6)	MXM6	5
5	DO 100 I=1,6	MXM6	6
	DO 90 J=1,6	MXM6	7
	SUM = 0.	MXM6	8
	DO 80 II=1,6	MXM6	9
	80 SUM = SUM + A(I,II)*B(II,J)	MXM6	10
10	90 CC(I,J) = SUM	MXM6	11
	100 CONTINUE	MXM6	12
	DO 115 I=1,6	MXM6	13
	DO 115 J=1,6	MXM6	14
	C(I,J) = CC(I,J)	MXM6	15
15	115 CONTINUE	MXM6	16
	RETURN	MXM6	17
	END	MXM6	18

1	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
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
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
15	115 CONTINUE	MXM7	16
	RETURN	MXM7	17
	END	MXM7	18

1	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
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

1	SUBROUTINE MXV7(T,V,VN)	MXV7	2
		MXV7	3
	C MULTIPLIES MATRIX BY A VECTOR	MXV7	4
	DIMENSION T(7,7),V(7),VN(7),VV(7)	MXV7	5
5	DO 10 I=1,7	MXV7	6
	VV(I) = 0.	MXV7	7
	DO 10 J=1,7	MXV7	8
	VV(I) = VV(I) + T(I,J)*V(J)	MXV7	9
10	CONTINUE	MXV7	10
10	DO 15 J=1,7	MXV7	11
	VN(J) = VV(J)	MXV7	12
	RETURN	MXV7	13
	END	MXV7	14

1	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
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
10	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
15	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
20	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
25	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
30	RETURN	NAMCHK	24
	120 CONTINUE	NAMCHK	25
	RETURN	NAMCHK	26
	END	NAMCHK	27

1		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
5	C	J =0,1 FOR NORMAL OR SKEW N-POLE LENS	NPOM	6
	C	L = EFFECTIVE 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
10	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
15		COMMON /ML/ ML \$ LEVEL 2, ML	NPOLLL	1
			NPOLLL	2
		LEVEL 2, STORE,INFF,IWORK	BLANK	2
		COMMON STORE(48000),IWORK(10)	BLANK	3
		DIMENSION INFF(24,2000)	86MARSIZ	1
20		EQUIVALENCE (INFF,STORE)	86MARSIZ	2
			86MARSIZ	3
			BLANK	5
		COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMP	DIM	2
			DIM	3
			NPOM	18
25		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
30		DATA II/(0.,1.)/	NPOL682	2
	C		NPOM	23
		NN=M	NPOLJUN83	4
		DO 5 I=1,3	NPOM	24
35	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
40		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
45		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
50	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
55	11	Q=Q*Z/I	NPOL682	9
	12	DP=Q*Z/MP	NPOL682	10
		R(2,1)=-REAL(Q)	NPOL682	11

		R(2,3)=AIMAG(Q)	NPOLJUN83	6
		DPX=-REAL(DP)	NPOL682	13
60		DPY=AIMAG(DP)	NPOL682	14
	C		NPOM	86
	40	R(4,3) = -R(2,1)	NPOM	87
		R(4,1) = R(2,3)	NPOM	88
		V(2) = PX + DPX	NPOM	89
65		V(4) = PY + DPY	NPOM	90
		IF (L.EQ.0) GO TO 41	NPOM	91
		V(1) = X + V(2) *LH	NPOM	92
		V(3)=Y+V(4)*LH	NPOLJUN83	7
	C		NPOM	94
70	41	R(2,6) = -DPX	NPOM	95
		R(2,7) = -DPX	NPOM	96
		R(4,6) = -DPY	NPOM	97
		R(4,7) = -DPY	NPOM	98
		R(5,1) = DPX	NPOM	99
75		R(5,3) = DPY	NPOM	100
		RW(2)=-DPX	NPOLJUN83	8
		IF (L.EQ.0) GO TO 45	NPOM	101
		ML(1) = MEND - 3	NPOM	102
		ML(2) = M	NPOM	103
80		ML(3) = MEND - 3	NPOM	104
		CALL DRIFT(MEND-3,LH)	NPOM	105
		CALL STOR7(M,R,RW)	NPOM	106
		CALL MMM(M,3,ML)	NPOM	107
		RETURN	NPOM	108
85	45	CALL STOR7(M,R,RW)	NPOM	109
		RETURN	NPOM	110
		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.

1	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
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
10	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
15		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
20	* 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
25	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
30	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
35	* 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
40	RETURN	OPNUM	32
	110 OPNUM = 69	OPNUM	33
	RETURN	OPNUM	34
	END	OPNUM	35

1	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
5	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
10	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
15	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
20	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
25		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
30		BLANK	5
	COMMON/BMI1/MI1(16000)	NV3BMI1	1
	COMMON/NELS1/NELS1	BMI1L	2
	LEVEL 2,MI1	BMI1L	3
35	C	BMI1L	4
		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
40	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
45	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
50	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
55	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
60	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
65	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
70	9 NAMONV = MON	ORBIT613 8
	NACORV = NACOR	ORBIT613 9
	NNV = N	ORBIT613 10
	26 CONTINUE	ORBIT613 11
	C	ORBIT 35
75	PRINT 3	ORBIT 36
	3 FORMAT(1H1)	ORBIT 37
	PRINT 106,EP SB,EP S	ORBIT 38
	106 FORMAT(1X*DISPERSION DE MESURE *,F10.5/ 1X*C.A.C. RESIDUEL * *,F10.5/)	ORBIT 39
80	PRINT 100	ORBIT 40
	100 FORMAT(1X*CORRECTEURS*/)	ORBIT 41
	PRINT 2,(NX(I),I=1,N)	ORBIT 42
	2 FORMAT (2I5)	ORBIT 43
	PRINT 4	ORBITC 12
85	PRINT 101	ORBIT 45
	101 FORMAT(1X*STATIONS*/)	ORBIT 46
	PRINT 1,(NB(I),I=1,M)	ORBIT 47
	1 FORMAT (2I5)	ORBIT 48
	PRINT 4	ORBITC 13
90	4 FORMAT(1H0)	ORBIT 50
	IB=1	ORBIT 51
	IC=1	ORBIT 52
	REWIND 5	ORBIT 53
	IF(IFLAG.EQ.0) GO TO 10	ORBITC 14
95	PX=0.	ORBIT 54
	PY=1.	ORBIT 55
	GO TO 11	ORBIT 56
	10 PX=1.	ORBIT 57
	PY=0.	ORBIT 58
100	11 READ(5) IEL	ORBIT 59
	READ (5)	ORBIT 60
	PI=4.*ATAN(1.)	NOV28ORB 4
	ICC=0	ORBIT 61
	DO 12 I=1,IEL	ORBIT 62
105	READ(5) IE ,NO,NAME,DUM,BY,BX,AY,AX,MY,MX	ORBIT 63
	READ (5) EX,EXP,EY,EYP,XP,DXP,Y,DYP	ORBIT 64
	IF(NO.EQ.NB(IB)) GO TO 13	NOV28ORB 5
	IF(NO.EQ.NX(IC)) GO TO 14	ORBIT 66
	GO TO 12	ORBIT 67
110	13 IA=1	ORBIT 68
	IX=IB	ORBIT 69
	IB=IB+1	ORBIT 70
	B (IX)=PR(XP, Y,PX,PY)+(RGEN(0)-.5)*EPSB	ORBIT 71
	AP(IX) = 0.	ORBIT 72
		ORBITC 16

115	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
120	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
125	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
130	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
135	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
140	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
145	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
150	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
155	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
160	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
165	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
170	118 CONTINUE	ORBITD	17
	IF(.NOT.KPP) GO TO 1181	ORBITPRNT	31

	PRINT 4	ORBIT	117
	PRINT 103	ORBIT	118
175	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
180	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
185	JTER=J	ORBIT	129
	CALL MICADO(AA,BB,84,M,N,AP,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
190	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
195	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
200	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
205	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
210	201 FORMAT(1H0,5HAFTER,13,22H 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
215	22 KPP=KPR	ORBITPRNT	44
	6 FORMAT(7(I5,F10.5))	ORBIT	150
	RETURN	ORBIT	151
	END	ORBIT	152

1	SUBROUTINE MICADO(A,B,NDIM,M,N,AP,XA,NA,NB,NC,EPS,ITER,DP,X,NX,R, 1RHO)	MICADO MICADO	2 3
	C*****	MICADO	4
	LEVEL 2,A,B,AP,X,NX,R,RHO	MICADO4	1
5	DIMENSION A(NDIM,1),B(1),AP(1),XA(1),NA(1),NB(1),NC(1),X(1),NX(1) 1,R(1),RHO(1)	MICADO MICADO	5 6
	C	MICADO	7
	C REDUCTION DE AP ET DE A	MICADO	8
	CALL NBR(NA,N1)	MICADO	9
10	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
15	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
20	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
25	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
30	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
35	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
40	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
45	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
50	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
55	9 IF(NX(I).GE.NB(J)) NX(I)=NX(I)+1	MICADO	55
	RETURN	MICADO	56
	END	MICADO	57

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

1	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
5	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
10	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
15	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
20	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
25	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
30	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
35	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
40	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
45	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
50	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
55	C TRANSFORMATION DE A DANS HTAL	HTLS	55
	C	HTLS	56
	CALL HTAL(A,M,N,K,BETA,M1)	HTLS	57

	C		HTLS	58
	C	TRANSFORMATION DE B DANS HTBL	HTLS	59
60	C		HTLS	60
		13 CALL HTBL(A,B,M,N,K,BETA,M1)	HTLS	61
	C		HTLS	62
	C	RECHERCHE DU PIVOT (K+1)	HTLS	63
	C		HTLS	64
65		AUX(K)=SQRT(PIV)	HTLS	65
		IF(K.EQ.N) GO TO 11	HTLS	66
		PIV=0.0	HTLS	67
		KPIV = K + 1	HTLS	68
		J1 = KPIV	HTLS	69
70		K2=N + J1	HTLS	70
		DO 18 J=J1,N	HTLS	71
		H=AUX(J)-DBLE(A(K,J))*DBLE(A(K,J))	HTLS	72
		AUX(J)=H	HTLS	73
		G=AUX(K2)-DBLE(A(K,J))*DBLE(B(K))	HTLS	74
75		AUX(K2) = G	HTLS	75
		PIVT = G*G/H	HTLS	76
		IF(PIVT.LT.PIV)GO TO 18	HTLS	77
		KPIV=J	HTLS	78
		PIV=PIVT	HTLS	79
80		18 K2 = K2 + 1	HTLS	80
	C		HTLS	81
	C	CALCUL DES X	HTLS	82
	C		HTLS	83
		11 X(K)=B(K)/AUX(N+K)	HTLS	84
85		IF(K.EQ.1)GO TO 27	HTLS	85
		DO 15 I=2,K	HTLS	86
		KK=K-I+1	HTLS	87
		X(KK)=B(KK)	HTLS	88
		KI=KK+1	HTLS	89
90		DO 25 J=KI,K	HTLS	90
		25 X(KK)=X(KK)-A(KK,J)*X(J)	HTLS	91
		X(KK)=X(KK)/AUX(N+KK)	HTLS	92
		15 CONTINUE	HTLS	93
	C	27 CALL UCOPY(B,R,M)	HTLS	94
95		27 CONTINUE	HTLS	95
		DO 29 III=1,M	HTLS	96
		R(III) = B(III)	HTLS	97
		29 CONTINUE	HTLS	98
	C		HTLS	99
100	C	CALCUL DU VECTEUR RESIDUEL DANS HTRL	HTLS	100
		CALL HTRL(A,R,M,N,K,AUX,M1)	HTLS	101
	C	IMAX=MAXFZE(R(1),M)	HTLS	102
	C	IMIN=MINFZE(R(1),M)	HTLS	103
		IMAX = MXMIN(R(1),M,1)	HTLS	104
105		IMIN = MXMIN(R(1),M,0)	HTLS	105
		EPS1=R(IMAX)-R(IMIN)	HTLS	106
		GOTO 1000	HTLS	107
	C	PRINT OUT OF INTERMEDIATE CORRECTOR STRENGTHS/ORBITS	HTLS	108
		PRINT 6000,K,R(IMAX),IMAX,R(IMIN),IMIN	HTLS	109
110		PRINT 6002, (IX,IPI(IX),X(IX),IX=1,K,1)	HTLS	110
		PRINT 6004	HTLS	111
		PRINT 6006, (IX,R(IX),IX=1,M,1)	HTLS	112
	1000	CONTINUE	HTLS	113
		IF(EPS1.GT.EPS)GO TO 21	HTLS	114

115	GO TO 202	HTLS	115
	21 CONTINUE	HTLS	116
	RETURN	HTLS	119
	202 EPS=EPS1	HTLS	120
	ITER=K	HTLS	121
120	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
125	. 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
130	. 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.

1	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
5	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
10	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
15	BETA=H + SIG	HTLS	148
	A(K,K)=BETA	HTLS	149
	BETA=1./(SIG*BETA)	HTLS	150
	RETURN	HTLS	151
	END	HTLS	152

1	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
5	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
10	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

1	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
5	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
10	21 B(K1)=B(K1)-A(K1,K)*H	HTLS	174
	RETURN	HTLS	175
	END	HTLS	176

1	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
5	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
10	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
15	CALL HTBL(A,B,M,N,KL,BETA,M1)	HTLS	190
	30 CONTINUE	HTLS	191
	RETURN	HTLS	192
	END	HTLS	193

1	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
5	LEVEL 2,A	DEBUG	12
	DIMENSION A(N)	HTLS	198
	MXMIN = 1	HTLS	199
	IF (N.LT.1) RETURN	HTLS	200
	CURRENT=A(1)	HTLS	201
10	DO 10 I=2,N	HTLS	202
	IF ((M.EQ.0).AND.(A(I).GE.CURRENT)) GO TO 10	HTLS	203
	IF ((M.EQ.1).AND.(A(I).LE.CURRENT)) GO TO 10	HTLS	204
	CURRENT=A(I)	HTLS	205
	MXMIN = I	HTLS	206
15	10 CONTINUE	HTLS	207
	RETURN	HTLS	208
	END	HTLS	209

1	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
5	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
10	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
15	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
20	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
25	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
30	END	PARSR	28

1	SUBROUTINE PBML(M)	PBMLL	2
	C PRINT NAMED BEAM ELEMENTS WITH LENGTHS AND ACCUMULATED LENGTHS.	PBMLL	3
	LEVEL 2, STORE,INFF,IWORK	BLANK	2
5	COMMON STORE(48000),IWORK(10)	BLANK	3
	DIMENSION INFF(24,2000)	86MARSIZ	1
	EQUIVALENCE (INFF,STORE)	86MARSIZ	2
		86MARSIZ	3
		BLANK	5
	COMMON/BMI1/MI1(16000)	NV3BMI1	1
10	COMMON/NELS1/NELS1	BMI1L	2
	LEVEL 2,MI1	BMI1L	3
		BMI1L	4
	C	BMI1	3
	C	PBMLL	6
15	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
20	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
25	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
30	ELT = ELT + EL	PBMLL	22
	WRITE (3,2) MNAM,EL,ELT	PBMLL	23
	110 CONTINUE	PBMLL	24
	100 CONTINUE	PBMLL	25
	C	PBMLL	26
35	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

1 SUBROUTINE PARTYP(IP,ITYP)
RETURN
END

PARTYP 2
PARTYP 3
PARTYP 4

1	SUBROUTINE PLPAR(PM,FL,IT)	PLPAR	2
		PLPAR	3
*	ROUND LIMITS FOR PLOT OF BETATRON FUNCTIONS. SET TICK INTERVALS.	PLPAR	4
		PLPAR	5
5	IF (PM.GT.5.) GO TO 43	PLPAR	6
	FL = 5.	PLPAR	7
	IT = 1	PLPAR	8
	RETURN	PLPAR	9
43	IF (PM.GT.10.) GO TO 44	PLPAR	10
10	FL = 10.	PLPAR	11
	IT = 2	PLPAR	12
	RETURN	PLPAR	13
44	IF (PM.GT.25.) GO TO 45	PLPAR	14
	FL = 25.	PLPAR	15
15	IT = 5	PLPAR	16
	RETURN	PLPAR	17
45	IF (PM.GT.50.) GO TO 46	PLPAR	18
	FL = 50.	PLPAR	19
	IT = 10	PLPAR	20
20	RETURN	PLPAR	21
46	IF (PM.GT.100.) GO TO 471	PLPAR	22
	FL = 100.	PLPAR	23
	IT = 20	PLPAR	24
	RETURN	PLPAR	25
25	471 IF (PM.GT.150.) GO TO 47	PLPAR	26
	FL = 150.	PLPAR	27
	IT = 30	PLPAR	28
	RETURN	PLPAR	29
30	47 IF (PM.GT.250.) GO TO 48	PLPAR	30
	FL = 250.	PLPAR	31
	IT = 50	PLPAR	32
	RETURN	PLPAR	33
35	48 IF (PM.GT.500.) GO TO 49	PLPAR	34
	FL = 500.	PLPAR	35
	IT = 100	PLPAR	36
	RETURN	PLPAR	37
40	49 IF (PM.GT.1000.) GO TO 50	PLPAR	38
	FL = 1000.	PLPAR	39
	IT = 200	PLPAR	40
	RETURN	PLPAR	41
45	50 IF (PM.GT.1500.) GO TO 51	PLPAR	42
	FL = 1500.	PLPAR	43
	IT = 300	PLPAR	44
	RETURN	PLPAR	45
50	51 IF (PM.GT.2500.) GO TO 52	PLPAR	46
	FL = 2500.	PLPAR	47
	IT = 500	PLPAR	48
	RETURN	PLPAR	49
55	52 IF (PM.GT.5000.) GO TO 53	PLPAR	50
	FL = 5000.	PLPAR	51
	IT = 1000	PLPAR	52
	RETURN	PLPAR	53
	53 IP = PM/1000.	PLPAR	54
	FL = (IP+1)*1000	PLPAR	55
	IT = 2000	PLPAR	56
	RETURN	PLPAR	57
	END	PLPAR	58

1	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
5		PLTB682	3
		BLANK	2
	LEVEL 2, STORE,INFF,IWORK	BLANK	3
	COMMON STORE(48000),IWORK(10)	86MARSIZ	1
	DIMENSION INFF(24,2000)	86MARSIZ	2
10	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
15	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
20	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
25	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
30	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
35		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
40	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
45	*** 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
50	****	PLTB682	32
	LIMSW = .FALSE.	PLOTBEV	35
	BXMX = BXX	PLOTBEV	36
	IF(NPLT.EQ.0) CALL SETPLT	PLT	129
	REWIND 98	PLT	130
55	NPLT=NPLT+1	PLT	131
	BYMX = BYX	PLOTBEV	37
		PLOTBEV	38

	*	GET INPUT DATA	PLOTBEV	39
		INS=0	PLDRF	1
60		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
65		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
70	*	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
75		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
80		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
85	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
90		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
95	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
100		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
105		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
110		*****	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

115	* 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
120	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
125	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
130	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
135	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
140	* 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
145	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
150	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
155	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
160	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
165	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
170	GO TO 63	PLTB682	61
	57 IF (ABP.GT.10.) GO TO 62	PLTB682	62

	IP = ABP	PLOTBEV	132
	APP = IP	PLOTBEV	133
175	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
180	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
185	IFLT(I) = FLT(I)	PLOTBEV	145
	66 CONTINUE	PLOTBEV	146
	* DETERMINE LIMITS IN X SCALE	PLOTBEV	147
	IF (IPRS(1).GT.IPRS(2)) GO TO 205	PLOTBEV	148
190	IF (IPRS(2).NE.0.) GO TO 70	PLOTBEV	149
	** DEFAULT — ENTIRE BEAM LINE	PLOTBEV	150
	IPRS(1) = 0	PLOTBEV	151
	IPRS(2) = NZ	PLOTBEV	152
		PLOTBEV	153
195	*****	PLOTBEV	154
	* ORIGINALLY SET UP TO LOOP FROM HERE TO 200 TO PLOT MULTIPLE FRAMES	PLOTBEV	155
	* INPUT CHANGED TO REQUIRE CARD INPUT FOR EACH FRAME.	PLOTBEV	156
	*****	PLOTBEV	157
	70 DO 200 KK=1,KO,2	PLOTBEV	158
200	N1 = IPRS(KK) + 1	PLOTBEV	159
	NBT = IPRS(KK+1) + 1	PLOTBEV	160
	NBE = NBT - 1	PLOTBEV	161
	NP = NBT - N1 + 1	PLOTBEV	162
	* LIMITS OF X FROM S1 TO SND	PLOTBEV	163
205	S1 = XX(N1)	PLOTBEV	164
	SND = XX(NBT)	PLOTBEV	165
	J1 = N1	PLOTBEV	166
		PLOTBEV	167
		PLOTBEV	168
210	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
215	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
220		PLOTBEV	181
	* DRAW BOUNDARIES	PLOTBEV	182
		PLOTBEV	183
	PTOP=850.	PLFORM	4
	PBOT = 60.	PLOTBEV	185
225	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
230	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
235	X(3)=X(2) \$ Y(3)=Y(2)	PLFORM	5
	CALL CCPLLOT(X,Y,3,4HJOIN,1,1,1HI)	PLFORM	6
	Y(1) = CYT	PLOTBEV	199
	X(2) = SND	PLOTBEV	200
	X(1) = SND	PLOTBEV	204
240	Y(2) = CYB	PLOTBEV	205
	X(3)=X(2) \$ Y(3)=Y(2)	PLFORM	7
	CALL CCPLLOT(X,Y,3,4HJOIN,1,1,1HI)	PLFORM	8
	X(2) = S1	PLOTBEV	209
	Y(1) = CYB	PLOTBEV	210
245	X(3)=X(2) \$ Y(3)=Y(2)	PLFORM	9
	CALL CCPLLOT(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
250	*****	PLOTBEV	215
		PLOTBEV	216
	* DRAW BEAM SYSTEM	PLOTBEV	217
	S = S1	PLOTBEV	218
	DO 140 K=N1,NBE	PLOTBEV	219
255	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
260	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
265	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
270	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
275	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
280	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
285		PLOTBEV	247

	* DRAW MAGNET ABOVE OR BELOW THE LINE FOR FOCUSSING OR	PLOTBEV	248
	* DEFOCUSSING GRADIENT.	PLOTBEV	249
	* STRADDLE THE LINE IF ZERO GRADIENT.	PLOTBEV	250
	G = FDAT(NN,2)	PLOTBEV	251
290	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
295	* FOCUSSING	PLOTBEV	257
	120 PCT = PTOP + 50.	PLOTBEV	258
	PCB = PTOP	PLOTBEV	259
	121 TY = PCB + 20.	PLOTBEV	260
	GO TO 125	PLOTBEV	261
300	* 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
305		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
310	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
315	CALL CCPLLOT(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
320	X(3)=X(2) \$ Y(3)=Y(2)	PLFORM	14
	CALL CCPLLOT(X,Y,3,4HJOIN,1,1,1H.)	PLFORM	15
	128 CONTINUE	PLOTBEV	282
	127 X(1) = SE	PLOTBEV	283
	Y(2) = PT	PLOTBEV	284
325	DO 130 I=1,3	PLOTBEV	285
	X(3)=X(2) \$ Y(3)=Y(2)	PLFORM	16
	CALL CCPLLOT(X,Y,3,4HJOIN,1,1,1H.)	PLFORM	17
	130 CONTINUE	PLOTBEV	287
	X(2) = S	PLOTBEV	288
330	Y(1) = PT	PLOTBEV	289
	DO 132 I=1,3	PLOTBEV	290
	X(3)=X(2) \$ Y(3)=Y(2)	PLFORM	18
	CALL CCPLLOT(X,Y,3,4HJOIN,1,1,1H.)	PLFORM	19
	132 CONTINUE	PLOTBEV	292
335	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
340	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
345	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
350	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
355	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
360	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
365	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
370	* 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
375	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
380	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
385	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
390	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
395	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

400	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
405		PLOTBEV	312
		PLOTBEV	313
	*****	PLOTBEV	314
		PLOTBEV	315
	CCYMAX = PTOP	PLOTBEV	316
410	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
415	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
420	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
425	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
430	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
435	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
440	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
445	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
450		PLOTBEV	363
	*****	PLOTBEV	364
		PLOTBEV	365
	* DETERMINE TICK INTERVALS FOR X AXIS AND PLOT THEM.	PLOTBEV	366
	CCYMAX = PTOP	PLOTBEV	367
455	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
460	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
465	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
470	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
475	CALL CCPLLOT(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
480	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
485	J = 1	PLOTBEV	398
	YMIN = 0.	PLOTBEV	399
	166 YMAX = FLT(J)	PLOTBEV	400
	CCX = CCXMIN + T1	PLOTBEV	401
	CXTK1 = XX(J1)	PLOTBEV	402
490	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
495	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
500	Y(2) = TK	PLTB682	113
	CALL CCPLLOT(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
505	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
510	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
515	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 CCPLLOT(X,Y,3,4HJOIN,1,1,1H-)	PLFORM	21
520	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
525	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
530	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
535	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
540	CALL CCPLLOT(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
545	X(3)=X(2) \$ Y(3)=Y(2)	PLFORM	24
	CALL CCPLLOT(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
550	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
555	CCXMIN = SAVXMN	PLOTBEV	437
		PLOTBEV	438
	*****	PLOTBEV	439
	C SET UP PLOT	PLOTBEV	440
		PLOTBEV	441
560	CCYMAX = PTOP	PLOTBEV	442
	CCYMIN = PBOT	PLOTBEV	443
	YMAX = FLT(1)	PLOTBEV	444
	YMIN = 0.	PLOTBEV	445
	NN = NP	PLOTBEV	446
565	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
570	*****	PLOTBEV	452

		PLOTBEV	453
	* PLOT BETAX	PLOTBEV	455
	DO 173 I=1,KW	PLOTBEV	456
	173 YPL(I)=Y1(I)	PLOTBEV	457
575	CALL CCPLLOT(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
580	CALL CCPLLOT(XPL(J1),YPL(J1),NN,4HJOIN,1,1,1HV)	PLB882	2
	* FIND PAPER COORDINATES OF MAXIMUM POINTS	PLOTBEV	467
		PLOTBEV	468
		PLOTBEV	469
	M1 = J1	PLOTBEV	470
	M2 = J1	PLOTBEV	471
585	M3 = J1	PLOTBEV	472
	M4 = J1	PLOTBEV	473
	JJ = J1 + 1	PLOTBEV	474
	Y1MX = Y1(J1)	PLOTBEV	475
	Y2MX = Y2(J1)	PLOTBEV	476
590	Y3MX = Y3(J1)	PLOTBEV	477
	Y4MX = Y4(J1)	PLOTBEV	478
	JN = J1 + NN - 1	PLOTBEV	479
	DO 184 J=J,J,JN	PLOTBEV	480
	IF (Y1(J).LE.Y1MX) GO TO 178	PLOTBEV	481
595	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
600	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
605	M4 = J	PLOTBEV	492
	184 CONTINUE	PLOTBEV	493
		PLOTBEV	494
	PY = Y1(M1)	PLOTBEV	495
	IF (PY.GT.YMAX) GO TO 183	PLOTBEV	496
610	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
615	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
620	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
625	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
630	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
635	187 CONTINUE	PLOTBEV	522
		PLOTBEV	523
	* PLOT XEQ,YEQ	PLOTBEV	524
	YMAX = FLT(2)	PLOTBEV	525
	YMIN = FLT(3)	PLOTBEV	526
640	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 CCPLT(XPL(J1),YPL(J1),NN,4HJOIN,1,1,1HY)	PRCH	8
645	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
650	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 CCPLT(XPL(J1),YPL(J1),NN,4HJOIN,1,1,1HX)	PLTB682	144
655	* 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
660	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
665	Y(2) = 0.	PLTB682	155
	CALL CCPLT(X,Y,2,4HJOIN)	PLTB682	156
	185 CONTINUE	PLTB682	157
		PLTB682	158
	PY = Y2(M2)	PLTB682	159
670	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
675	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
680	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

685		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
690	*	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
695	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
700	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
705	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.

1	SUBROUTINE PRINTV(M,K)	PRINTV	2
		BLANK	2
	LEVEL 2, STORE, INFF, IWORK	BLANK	3
	COMMON STORE(48000), IWORK(10)	86MARSIZ	1
5	DIMENSION INFF(24,2000)	86MARSIZ	2
	EQUIVALENCE (INFF,STORE)	86MARSIZ	3
		BLANK	5
	COMMON/SWTCB/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	SWTCH	2
	LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS	SWTCH	3
10		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
15	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
20	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
25	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
30	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
35	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
40	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
45	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
50	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
55	GO TO 20	PRINTV	49
	16 CALL DATA(M1,5,LOC,NROW,V(1,JJ))	PRINTV	50
	LOC = LOC + NROW	PRINTV	51

	20	CONTINUE	PRINTV	52
		IF (NTYPE.NE.4HSNGL) GO TO 35	PRINTV	53
60		BEGIN = .FALSE.	PRINTV	54
		IF (ISUB.EQ.7.OR.IJ.EQ.K) GO TO 22	PRINTV	55
		NROW1 = NROW	PRINTV	56
		GO TO 100	PRINTV	57
	21	ASSIGN 15 TO IRET	PRINTV	58
65		NR = NROW1	PRINTV	59
		GO TO 25	PRINTV	60
	22	ASSIGN 100 TO IRET	PRINTV	61
		NR = NROW	PRINTV	62
	C	TIME TO PRINT	PRINTV	63
70	25	WRITE (3,1)	PRINTV	64
		CALL DASH	PRINTV	65
		WRITE (3,4)(NAM(J),J=1,JJ)	PRINTV	66
		DO 30 L=1,NR	PRINTV	67
		WRITE (3,5)(V(L,J),J=1,JJ)	PRINTV	68
75	30	CONTINUE	PRINTV	69
		GO TO 40	PRINTV	70
	C	PRINT COMPLEX OR DOUBLE PRECISION VECTORS	PRINTV	71
	35	WRITE (3,1)	PRINTV	72
		CALL DASH	PRINTV	73
80		IF (NTYPE.EQ.4HCPLX) GO TO 37	PRINTV	74
		WRITE (3,9) NAM(1)	PRINTV	75
		GO TO 38	PRINTV	76
	37	WRITE (3,8) NAM(1)	PRINTV	77
	38	WRITE (3,10) (CV(J),J=1,NC)	PRINTV	78
85		ASSIGN 100 TO IRET	PRINTV	79
	40	BEGIN = .TRUE.	PRINTV	80
		JJ = 0	PRINTV	81
		GO TO IRET (15,100)	PRINTV	82
	100	CONTINUE	PRINTV	83
90		CALL DASH	PRINTV	84
		RETURN	PRINTV	85
	1	FORMAT (//)	PRINTV	86
	4	FORMAT (4X,7(9X,A7))	PRINTV	87
	5	FORMAT (4X,7(2X,F14.8))	PRINTV	88
95	6	FORMAT (1X,I1,8X)	PRINTV	89
	7	FORMAT (A4,A1,A2,3X)	PRINTV	90
	8	FORMAT (16H COMPLEX VECTOR ,A5)	PRINTV	91
	9	FORMAT (25H DOUBLE PRECISION VECTOR ,A5)	PRINTV	92
	10	FORMAT (30X,2(E14.8,5X))	PRINTV	93
100		END	PRINTV	94

1	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
5	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
10	COMMON/BMI/MI(5000)	BMIL	1
	COMMON/NELS/NELS	BMIL	2
	LEVEL 2,MI	BMIL	3
		BMIL	4
		BMI	3
15	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
20	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
25	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
30	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
35	NAM(J) = NAME	PRNT	27
	MI(J) = ELNUM(NAME)	PRNT	28
	10 CONTINUE	PRNT	29
	NUM = J	PRNT	30
	C PRINT HEADING	PRNT	31
40	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) KTYPE	PRNT	35
	12 IVAL(I) = BDAT(NK,KNDX)	PRNT	36
45	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
50	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
55	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	PRNT	50
60	21	WRITE (3,106) (VAL(I),I=I1,NM)	PRNT	51
		GO TO 25	PRNT	52
	22	WRITE (3,107) (IVAL(I),I=I1,NM)	PRNT	53
		GO TO 25	PRNT	54
	23	WRITE (3,108) (IVAL(I),I=I1,NM)	PRNT	55
		GO TO 25	PRNT	56
65	24	RETURN	PRNT	57
	25	IF(NM.NE.NUM) GO TO 20	PRNT	58
		WRITE (3,110)	PRNT	59
		RETURN	PRNT	60
	C		PRNT	61
70	100	FORMAT(1X,130(1H-)/2X,*VARIABLE*,I3,1X,*OF TYPE*,I2,1X,*(*, 1 A5,*))*	PRNT	62
	105	FORMAT(/1X,10(7XA5))	PRNT	63
	106	FORMAT(2X,10(F12.6))	PRNT	64
	107	FORMAT(10(8X,A4))	PRNT	65
75	108	FORMAT(10(2X,I10))	PRNT	66
	110	FORMAT(1X,130(1H-))	PRNT	67
		END	PRNT	68
			PRNT	69

1	SUBROUTINE PRNTAB(M)	PRNTAB	2
		BLANK	2
	LEVEL 2, STORE,INFF,IWORK	BLANK	3
	COMMON STORE(48000),IWORK(10)	86MARSIZ	1
5	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
10	INTEGER ELMUM	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
15	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
20	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
25	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 = ELMUM(NAM(I))	PRNTAB	25
30	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
35	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
40	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
45	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
50	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
55	IF (PRES) WRITE (3,100)	PRNTAB	51
	100 FORMAT (1H1)	PRNTAB	52
	RETURN	PRNTAB	53

60

1 FORMAT (1H1,6X,10(7X,A5)//)
2 FORMAT (4X,A1,2X,10F12.8)
END

PRNTAB 54
PRNTAB 55
PRNTAB 56

1	SUBROUTINE PRNTV7(M,K)	PRNTV7	2
		PRNTV7	3
		BLANK	2
	LEVEL 2, STORE,INFF,IWORK	BLANK	3
5	COMMON STORE(48000),IWORK(10)	86MARSIZ	1
	DIMENSION INFF(24,2000)	86MARSIZ	2
	EQUIVALENCE (INFF,STORE)	86MARSIZ	3
		BLANK	5
	COMMON/SWTC/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	SWTC	2
10	LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS	SWTC	3
		SWTC	4
	LOGICAL PH	PRNTV7	6
	DIMENSION V(7)	PRNTV7	7
	INTEGER ELNUM,BDAT	PRNTV7	8
15		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
20	PH = .FALSE.	PRNTV7	14
	CALL DASH	PRNTV7	15
	DO 100 I=1,K	PRNTV7	16
	PV=.FALSE.	PRNTV7	17
	LOC = 1	PRNTV7	18
25	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
30	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
35	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
40	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
45	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
50	30 CONTINUE	PRNTV7	44
	PH = .TRUE.	PRNTV7	45
100	CONTINUE	PRNTV7	46
	CALL DASH	PRNTV7	47
	WRITE (3,14)	PRNTV7	48
55	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)
60 13  FORMAT (5X,A4,I2,2X,7(F10.5,1X))
    14  FORMAT (///1H )
      END
```

PRNTV7	52
PRNTV7	53
PRNTV7	54
PRNTV7	55

1	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
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.

1		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
5	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
10	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

1	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
5	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
10		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
15	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
20	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
25	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
30	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
35	5 DQY = CQY - CQYZ	QUEY	34
	G = DQY	QUEY	35
	RETURN	QUEY	36
	END	QUEY	37

1	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
5	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
10	DO 20 I=1,24	RDINF	6
	20 INF(I) = INFF(I,M)	RDINF	7
	RETURN	RDINF	8
	END	RDINF	9

1	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
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/STORE/LMAX,LINF,LFILE,MAX,MIN,ICARD(11)	STORE	2
10	* 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
15		CONTRL	5
	COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYML	DIM	2
		DIM	3
	INTEGER OPNUM,HPERIOD	RDINST	9
	DIMENSION IC(24)	RDINST	10
20	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
25	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
30	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
35	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
40	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
45	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
50	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
55	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)	RDINST	49
	C	KA,KB BECOME -0 IF BLANK. RESET TO +0.	RDINST	50
60		IF (KA.EQ.0) KA=0	RDINST	51
		IF (KB.EQ.0) KB=0	RDINST	52
		OP = OPNUM(OPNAME)	RDINST	53
		IF (OPNAME.EQ.3HRUN) M = M7END -ISAV7	RDINST	54
		IF (ISIGN.EQ.1H-) OP = -OP	RDINST	55
65		CALL WRTINF(IC,M)	RDINST	56
		RETURN	RDINST	57
			RDINST	58
	100	OPNAME = NULL	RDINST	59
		RETURN	RDINST	60
70		END	RDINST	61

1	FUNCTION REALNUM(LOCI)	REALNUM	2
	C	REALNUM	3
		BLANK	2
	LEVEL 2, STORE, INFF, IWORK	BLANK	3
5	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
10		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
15	LOGICAL ERROR, RSRV, STOR, XEQ, TRASW, NFSW, EMPTY, INDEF, LDFLG, FIN	CONTRL	4
		CONTRL	5
	C	REALNUM	7
	INTEGER ELMUM, HRAND, RTBLNK	REALNUM	8
	LOGICAL NEG, INEG	REALNUM	9
20	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
25	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
30	NEG = .FALSE.	REALNUM	19
	ITR = 0	REALNUM	20
	LOC = LOCI	REALNUM	21
	C	REALNUM	22
	20 FVAR = STORE(LOC)	REALNUM	23
35	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
40	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
45	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
50	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
55	DECODE(6,105,NMVAR) NMVAR	RLNTIM	6
	105 FORMAT(1XA5)	RLNTIM	7
	30 MN = ELMUM(NMVAR)	REALNUM	56

	C	LQ IS DEFAULT	REALNUM	57
		IF (NPT.EQ.0) NPT = -1	REALNUM	58
60		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
65	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
70	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
75		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
80		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
85		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
90		ERROR = .TRUE.	REALNUM	89
		RETURN	REALNUM	90
		END	REALNUM	91

1

SUBROUTINE REAR(NC)
RETURN
END

REAR 2
REAR 3
REAR 4

1	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,MSYMP	DIM	2
5		DIM	3
		BLANK	2
	LEVEL 2, STORE,INFF,IWORK	BLANK	3
	COMMON STORE(48000),IWORK(10)	86MARSIZ	1
	DIMENSION INFF(24,2000)	86MARSIZ	2
10	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
15		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
20	* 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
25	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
30	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
35	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
40	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
45	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

1	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
5	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
10	STORE(LOC)=B	REPBCD	6
	RETURN	REPBCD	7
	END	REPBCD	8

1	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
5	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
10	STORE(LOC)=F	REPFLT	6
	RETURN	REPFLT	7
	END	REPFLT	8

1	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
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
10	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
15	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
20		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
25	* 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
30	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
35	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
40	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
45	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
50	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
55	ISTORE(LOCN) = ISTORE(LOCM)	REPL	48
	RETURN	REPL	49
		REPL	50

	*	HOLLERITH (BCD)	REPL	51
60	110	IP = BDAT(M,3)	REPL	52
		IF (ERROR) GO TO 125	REPL	53
		CALL REPBCD(N,K,IP)	REPL	54
		IF (MODE.EQ.3) WRITE (3,111) K,NM,IP	REPL	55
	111	FORMAT (6H *** ,8X,*REPLACE *,I3,* OF *,A5,* WITH *,A5)	REPL	56
		RETURN	REPL	57
65			REPL	58
	*	INTEGER	REPL	59
	115	IP = IDAT(M,1)	REPL	60
		IF (ERROR) GO TO 125	REPL	61
		CALL REPINT(N,K,IP)	REPL	62
70		IF (MODE.EQ.3) WRITE (3,116) K,NM,IP	REPL	63
	116	FORMAT (6H *** ,8X,*REPLACE *,I3,* OF *,A5,* WITH *,I5)	REPL	64
		RETURN	REPL	65
			REPL	66
	*	FLOATING POINT VARIABLE INSERTED IN LQ STORAGE	REPL	67
75	117	F = FDAT(M,1)	REPL	68
		CALL REPLQ(N,1,K,F)	REPL	69
		IF (MODE.EQ.3) WRITE(3,101) K,NM,F	REPL	70
		RETURN	REPL	71
			REPL	72
80	*	ERROR	REPL	73
	120	WRITE (3,121) KT	REPL	74
	121	FORMAT (//6H *** ,*ERROR IN REPL INSTRUCTION. *,A5,* IS AN UNKNO 1WN TYPE.*)	REPL	75
			REPL	76
	130	ERROR = .TRUE.	REPL	77
85		RETURN	REPL	78
			REPL	79
	125	WRITE(3,126) KT	REPL	79
	126	FORMAT (//6H *** ,*ERROR IN REPL INSTRUCTION. NO VALUES OF TYPE*, 1 A5,*.*)	REPL	80
			REPL	81
		RETURN	REPL	82
90		END	REPL	83

1	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
5	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
10	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
15	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
20	15 STORE(LOC) = F	REPLQ	16
	RETURN	REPLQ	17
	END	REPLQ	18

1		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
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
10		BLANK	5
	LOC=INFF(12,M)+J-1	REPINT	7
	STORE(LOC)=FI	REPINT	8
	RETURN	REPINT	9
	END	REPINT	10

```

1      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
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
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
15     C UNTIL IT FINDS A NEW INSTRUCTION CARD, AND SET NBC=12*CARDS READ. RESRV     16
      C                                         RESRV     17
                                         BLANK      2
                                         BLANK      3
20     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
25     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
30     * 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
35     COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMP L 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
40     13HPRD,4HSCAL,1H /                   RESRV     26
      C                                         RESRV     27
      RESRV     28
      IF(MODE.EQ.3) RETURN                   RESRV     29
      IF (.NOT.RSRV) RETURN                  RESRV     30
45     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
50     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
55     NFL =0                                RESRV     41
      NBC =0                                 RESRV     42
      LQ = LFILE+ 1                          RESRV     43

```

	LQ2 = LQ	RESRV	44
	LQ3 = LQ	RESRV	45
60	C IF IQ.GE.0, NQ=IQ=NO. LOCATIONS RESERVED STARTING AT LQ	RESRV	46
	C IF IQ.EQ.-0, NQ WILL BE CALCULATED ACCORDING TO KIND	RESRV	47
	C IF IQ.LE.-1, ABS(IQ) REFERS TO THE IQ-TH SET OF DATA.	RESRV	48
	C LQ WILL BE SET = LIN=LBC,OR LFL, DEPENDING ON DATA TYPE	RESRV	49
	IF (IQ.EQ.0) GO TO 10	RESRV	50
65	IF (IQ.GE.0) GO TO 15	RESRV	51
	C IQ IS NEGATIVE NUMBER.	RESRV	52
	C NQ WILL BE DETERMINED BY THE IQ-TH SET OF INPUT DATA	RESRV	53
	NQ = 0	RESRV	54
	GO TO 16	RESRV	55
70	C DETERMINE SIGN OF ZERO	RESRV	56
	10 IZ = IQ.AND.1	RESRV	57
	IF (IZ.NE.0) NQSW = .TRUE.	RESRV	58
	IF (NQSW) GO TO 18	RESRV	59
	C IQ IS POSITIVE NUMBER. SET NQ=IQ	RESRV	60
75	15 NQ = IQ	RESRV	61
	C IF KIND NOT SPECIFIED, SET IT TO OPNAME	RESRV	62
	16 IF (KKND.EQ.0) KIND = INFF(1,M0)	RESRV	63
	IF (KIND.EQ.0) KIND = NKIND(KKND)	RESRV	64
	IF (OPNAME.NE.4HFXT) GO TO 30	RESRV	65
80	C FXPT. SET UP 3 LQ LOCATIONS	RESRV	66
	LQ2 = LQ + KADD + 49	RESRV	67
	LQ3 = LQ2 + 35	RESRV	68
	NQ = LQ3 + 15 - LQ	RESRV	69
	GO TO 30	RESRV	70
85	C IQ IS -0. NQ DETERMINED BY KIND	RESRV	71
	18 IF (KKND.NE.4H) GO TO 181	RESRV	72
	C KIND IS UNKNOWN. SET NQ = 0. LOAD WILL SET LQ AND NQ AFTER LOADING.	RESRV	73
	KIND = KKND	RESRV	74
	NQ = 0	RESRV	75
90	GO TO 30	RESRV	76
	C KIND IS KNOWN. FIND OUT WHICH ONE.	RESRV	77
	181 NK = 1	RESRV	78
	DO 19 J=1,10	RESRV	79
	IF (KKND.EQ.NK) GO TO 20	RESRV	80
95	NK=NK+1	RESRV	81
	19 CONTINUE	RESRV	82
	WRITE (3,110)	RESRV	83
	110 FORMAT (/5X,25H ILLEGAL KIND ENCOUNTERED//)	RESRV	84
	ERROR=.TRUE.	RESRV	85
100	RETURN	RESRV	86
	20 KIND=NKIND(NK)	RESRV	87
	GO TO (21,22,23,24,25,26,25,25,29), NK	RESRV	88
	C KIND IS MXP MY. MATRIX PAIRS RX AND RY	RESRV	89
	21 NQ = 12 + KADDR	RESRV	90
105	IF (.NOT.ROWSW) ROWS=2	RESRV	91
	IF (.NOT.COLSW) COLS=7	RESRV	92
	GO TO 30	RESRV	93
	C KIND IS PVECS, PARTICLE VECTOR	RESRV	94
	C IF NOT OTHERWISE INDICATED, 7 ROWS BY KA COLUMNS	RESRV	95
110	22 NQ = 7*KA	RESRV	96
	IF (.NOT.ROWSW) ROWS = 7	RESRV	97
	IF (.NOT.COLSW) COLS= KA	RESRV	98
	GO TO 30	RESRV	99
	C KIND IS SCALAR	RESRV	100

115	29	NQ = 1	RESRV	101
		IF (TYPE.NE.4HSNGL) NQ = 2	RESRV	102
		GO TO 30	RESRV	103
		C KIND IS MAT, MATRIX OF KA ROWS AND KB COLUMNS	RESRV	104
	23	GO TO 28	RESRV	105
120		C KIND IS 6X6 MATRIX	RESRV	106
	24	NQ = 22 + KADD	RESRV	107
	28	IF (.NOT.ROWSW) ROWS = KA	RESRV	108
		IF (.NOT.COLSW) COLS=KB	RESRV	109
		GO TO 30	RESRV	110
125		C KIND IS R7, MAP, OR PRD (7X7 MATRIX REPRESENTATION)	RESRV	111
	25	NQ = 49 + KADD	RESRV	112
		ROWS = 7	RESRV	113
		COLS = 7	RESRV	114
		GO TO 30	RESRV	115
130		C KIND IS ROT,ROTATION	RESRV	116
	26	NQ = KADD	RESRV	117
		KIND = NKIND(NK)	RESRV	118
		C SET LFMX FOR NSETS=0 CASE. IF NSETS NOT 0, LFMX COMPUTED BELOW.	RESRV	119
	30	LFMX=LQ+NQ-1	RESRV	120
135		IF(NSETS.EQ.0) GO TO 60	RESRV	121
		IF (I(NSETS).GE.0) GO TO 31	RESRV	122
		C NUMBER OF DATA IN LAST SET IS INDEFINITE.	RESRV	123
		INDEF=.TRUE.	RESRV	124
		I(NSETS) = -I(NSETS)	RESRV	125
140	31	IF(I(1).NE.0) GO TO 32	RESRV	126
		C STORE VARIABLE-INPUT-LENGTH NUMBERS IN NVAR(I).	RESRV	127
		DECODE (50,100,ICARD(3)) (NVAR(K),K=1,N1)	RESRV	128
	100	FORMAT (10I5)	RESRV	129
		C CALCULATE NUMBER OF DATA OF EACH TYPE.	RESRV	130
145	32	DO 44 J=1,NSETS	RESRV	131
		NAB = IABS(ND(J))	RESRV	132
		IF (ND(J).LT.0) ND(J) = NVAR(NAB)	RESRV	133
		IJ = I(J) + 1	RESRV	134
		GO TO (44,41,42,43,44),IJ	RESRV	135
150	41	NFL=NFL + ND(J)	RESRV	136
		GO TO 44	RESRV	137
	42	NBC=NBC + ND(J)	RESRV	138
		GO TO 44	RESRV	139
	43	NIN=NIN + ND(J)	RESRV	140
155	44	CONTINUE	RESRV	141
		C IF A CYX INSTRUCTION, SET UP LQ2	RESRV	142
		CYSW =.FALSE.	RESRV	143
		IF ((OPNAME.EQ.3HCYB).OR.(OPNAME.EQ.3HCYC)) CYSW = .TRUE.	RESRV	144
		IF ((OPNAME.EQ.3HCYA).OR.(OPNAME.EQ.4HBEST)) CYSW=.TRUE.	RESRV	145
160		IF (.NOT.CYSW) GO TO 45	RESRV	146
		LQ2 = LQ + NQ	RESRV	147
		NQ = NQ + 10	RESRV	148
	45	CONTINUE	RESRV	149
		IF (OPNAME.NE.3HTAB) GO TO 46	RESRV	150
165		C SETS UP EXTRA LQ FOR TABLE USE	RESRV	151
		LQ2 = LQ + NQ	RESRV	152
		LQ3 = LQ2 + 10	RESRV	153
		NQ = NQ + 20	RESRV	154
	46	CONTINUE	RESRV	155
170		C CALCULATE FILE LOCATIONS.	RESRV	156
		C ORDER SETS WITH FLOATING POINT AFTER LQ	RESRV	157

	NIWS = NIN	RESRV	158
	NBWS = NBC	RESRV	159
53	LFL = LQ + NQ	RESRV	160
175	LBC = LFL + NFL	RESRV	161
	LIN = LBC + NBWS	RESRV	162
	LFMX = LIN + NIWS - 1	RESRV	163
60	NTOT = LFMX - LFILE	RESRV	164
	IF (IQ.GE.0) GO TO 65	RESRV	165
180	C RESET NQ. RESET LQ ACCORDING TO TYPE OF DATA IN IQ-TH SET	RESRV	166
	C IQ REFERS TO THE ABS IQ-TH SET OF INPUT DATA	RESRV	167
	IQ=-IQ	RESRV	168
	NQ=ND(IQ)	RESRV	169
	IN=I(IQ)	RESRV	170
185	GO TO (61,62,63),IN	RESRV	171
	61 LQ=LFL	RESRV	172
	NQ = NFL	RESRV	173
	GO TO 65	RESRV	174
	62 LQ=LBC	RESRV	175
190	GO TO 65	RESRV	176
	63 LQ=LIN	RESRV	177
	C PACK LOCATIONS AND LENGTHS INTO INFO(I,N).	RESRV	178
	65 IF(NIN.EQ.0) LIN=0	RESRV	179
	IF(NFL.EQ.0) LFL=0	RESRV	180
195	IF(NBC.EQ.0) LBC=0	RESRV	181
	DO 70 J=1,9	RESRV	182
	70 INF(J) = 1H*	RESRV	183
	INF(10) = LQ	RESRV	184
	INF(11) = NQ	RESRV	185
200	INF(12) = LIN	RESRV	186
	INF(13) = NIN	RESRV	187
	INF(14) = LFL	RESRV	188
	INF(15) = NFL	RESRV	189
	INF(16) = LBC	RESRV	190
205	INF(17) = NBC	RESRV	191
	INF(18) = NTOT	RESRV	192
	INF(19)=LQ2	RESRV	193
	INF(20) = KIND	RESRV	194
	INF(21)=TYPE	RESRV	195
210	INF(22) = ROWS	RESRV	196
	INF(23) = COLS	RESRV	197
	INF(24) = LQ3	RESRV	198
	IF (M0.GT.0) CALL WRTINF(INF,M0)	RESRV	199
	C RESET BOTTOM OF FILE,LFILE,TO BOTTOM OF MTH SECTION OF THE FILE.	RESRV	200
215	LFILE= LFMX	RESRV	201
	C CHECK FOR STORAGE OVERFLOW	RESRV	202
	NSTO = LMAX - LFILE	RESRV	203
	NINFF = MAX - MIN	RESRV	204
	IF (LFILE.LE.LMAX) GO TO 80	RESRV	205
220	WRITE (3,201)	RESRV	206
	WRITE (3,202) LMAX,LFILE,NSTO,MAX,NINFF,M0	RESRV	207
	ERROR = .TRUE.	RESRV	208
	80 IF (M0.GT.0) RETURN	RESRV	209
	WRITE (3,200)	RESRV	210
225	IF (.NOT.ERROR) WRITE (3,202) LMAX,LFILE,NSTO,MAX,NINFF,M0	RESRV	211
	ERROR = .TRUE.	RESRV	212
	200 FORMAT (6H0*****,*ELEMENTS OVERFLOWED. INCREASE INFF ARRAY AND *	RESRV	213
	1 *MAX.*)	RESRV	214

230	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 DEFINITIONS)*,10X,I10,1X,*(MAX)*,5X,I10,5X,	RESRV	219
	3	I10)	RESRV	220
235		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

1		REVMAT	2
	SUBROUTINE REVMAT(N,M,V)	REVMAT	3
		REVMAT	4
	C MAPS PARTICLE VECTOR V BY ELEMENT WITH INDEX M.	REVMAT	5
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
10	C THE OLD ONE IS STORED AT N.	REVMAT	11
		REVMAT	12
	C ELEMENTS DEFINED BY MAG AND SXTIP WILL BE RECALCULATED BY REVMAT,	REVMAT	13
	C INCLUDING BR=BR*(1+DP/P).	REVMAT	14
		REVMAT	15
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
20	DIMENSION INFF(24,2000)	86MARSIZ	2
	EQUIVALENCE (INFF,STORE)	86MARSIZ	3
		BLANK	5
	COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMP	DIM	2
		DIM	3
25	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
30	DATA RADEG/.0174532925/	REVMAT	22
		REVMAT	23
	NOP = INFF(1,M)	REVMAT	24
		REVMAT	25
	C MOVE	REVMAT	26
35	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
40	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
45	CALL DATA(M,1,1,4,PAR)	REVMAT	37
	INFF(4,N)=INFF(4,M)	REVMATLTP	1
	INFF(5,N)=INFF(5,M)	REVMATLTP	2
	CALL SXTIP(N,V,PAR)	REVMAT	38
	RETURN	REVMAT	39
50		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
55	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
60	IF ((GRAD.NE.0.).OR.(IRECT.EQ.0.).OR.(BZ.EQ.0.)) GO TO 11	REVMAT	50
	C RECTANGULAR,ZERO-GRADIENT BENDING MAGNET	REVMAT	51
	CALL BEND(N,V,PAR,2)	REVMAT	52
	GO TO 9	REVMAT	53
65	11 CONTINUE	REVMAT	54
	IF (BZ.NE.0.) GO TO 111	REVMAT	55
	C QUADRUPOLE	REVMAT	56
	CALL MODQ (M,N,V,W)	REVMAT	57
	GO TO 45	REVMAT	58
70	111 BR = BR*(1.+V(6))	REVMAT	59
	CALL MAGNET(N,PAR)	REVMAT	60
	CALL RTRV7(N,T,RW)	REVMAT	61
	CALL MXV7(T,V,W)	REVMAT	62
		REVMAT	63
		REVMAT	64
75	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
80	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
85	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
90	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
95	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
100	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
105		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
110		REVMAT	100
		REVMAT	101
	C DRIFT	REVMAT	102
	7 IF(NOP.NE.3HDRF) GO TO 8	REVMAT	103
	EL = FDAT(M,1)	REVMAT	104

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

1

FUNCTION RGEN(D)
RGEN=RANF(D)
END

RGEN 2
RGEN 3
RGEN 4

1	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
5	COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYMP	DIM	2
		DIM	3
	COMMON /MO/ ML(3) \$ LEVEL 2, ML	ROTL	1
	EQUIVALENCE(ML,MI)	ROTL	2
		ROTL	3
10	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
15	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
20	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
25	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
30	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.

1	SUBROUTINE ROTM(M,MB,THETA)	ROTM	2
		ROTM	3
	* ROTATE MATRIX MB BY THETA RADIANS AND STORE IN MATRIX M.	ROTM	4
	* MB AND M ARE INDECES OF MATRICES	ROTM	5
5		ROTM	6
	DIMENSION T(49),TR(49),RW(3)	ROTM	7
		ROTM	8
	CALL RTRV7(MB,T,RW)	ROTM	9
	CALL ROTATE (THETA,T,TR,RW)	ROTM	10
10	CALL STOR7(M,TR,RW)	ROTM	11
	RETURN	ROTM	12
	END	ROTM	13

1	SUBROUTINE ROTZ(THET,T)	ROTZ	2
	* DEFINE A MATRIX REPRESENTING A ROTATION ABOUT THE Z-AXIS	ROTZ	3
	* THET IS IN RADIANŠ	ROTZ	4
5	DIMENSION T(7,7)	ROTZ	5
	DO 10 I=1,7	ROTZ	6
	DO 10 J=1,7	ROTZ	7
	T(I,J) = 0.	ROTZ	8
	10 CONTINUE	ROTZ	9
10	SN = SIN(THET)	ROTZ	10
	CS = COS(THET)	ROTZ	11
	T(1,1) = CS	ROTZ	12
	T(2,2) = CS	ROTZ	13
	T(3,3) = CS	ROTZ	14
15	T(4,4) = CS	ROTZ	15
	T(1,3) = SN	ROTZ	16
	T(2,4) = SN	ROTZ	17
	T(3,1) = -SN	ROTZ	18
	T(4,2) = - SN	ROTZ	19
20	T(5,5) = 1.	ROTZ	20
	T(6,6) = 1.	ROTZ	21
	T(7,7) = 1.	ROTZ	22
	RETURN	ROTZ	23
	END	ROTZ	24
		ROTZ	25

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

1	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
5	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
10	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
15	COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYML	DIM	2
		DIM	3
	EQUIVALENCE (T,TM)	RTRV7	9
	LOGICAL INV	RTRV7	10
	DATA RADEG/.0174532925/	RTRV7	11
20	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
25	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
30	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
35	KD = 7	RTRV7	27
	CALL CONVMT(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
40	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
45	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
50	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
55	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.

1

SUBROUTINE R7TOU(R7,U)
RETURN
END

R7TOU 2
R7TOU 3
R7TOU 4

1	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
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
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
15	1 CONTINUE	RXRY	16
	CALL STXY(M,RX,RY,RW)	RXRY	17
	RETURN	RXRY	18
	END	RXRY	19

1	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
5	C IF ELEMENT NUMBER IS --, RETURN INVERSE	RXY	6
	LEVEL 2, STORE, INFF, IWORK	BLANK	2
	COMMON STORE(48000), IWORK(10)	BLANK	3
	DIMENSION INFF(24,2000)	86MARSIZ	1
10	EQUIVALENCE (INFF,STORE)	86MARSIZ	2
		86MARSIZ	3
	COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,	BLANK	5
	1 LDFLG,FIN	CONTRL	2
	LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN	CONTRL	3
15		CONTRL	4
	COMMON/SWTC/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	CONTRL	5
	LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS	SWTCH	2
		SWTCH	3
	COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYML	SWTCH	4
20		DIM	2
	LOGICAL INV	DIM	3
	DIMENSION RX(2,3),RY(2,3),AX(2,3),AY(2,3),RW(3)	RXY	11
	DIMENSION TT(49),T(7,7),RPQ(4)	RXY	12
	EQUIVALENCE (T,TT)	RXY	13
25	C	RXY	14
	INV = .FALSE.	RXY	15
	IF (M.GE.0) GO TO 5	RXY	16
	4 M = - M	RXY	17
	INV = .TRUE.	RXY	18
30	5 LQ = INFF(10,M)	RXY	19
	NQ = INFF(11,M)	RXY	20
	DO 7 J=1,3	RXY	21
	DO 3 I=1,2	RXY	22
	RX(I,J) = 0.	RXY	23
35	RY(I,J) = 0.	RXY	24
	3 CONTINUE	RXY	25
	RW(J) = 0.	RXY	26
	7 CONTINUE	RXY	27
	KIND = INFF(20,M)	RXY	28
40	IF (KIND.EQ.3HROT) GO TO 12	RXY	29
	IF(KIND.EQ.3HMXY) GO TO 6	RXY	30
	C KIND IS R7. RETRIEVE AND CONVERT	RXY	31
	NN = M	RXY	32
	IF (INV) NN = -NN	RXY	33
45	CALL RTRV7(NN,T,RW)	RXY	34
	KD = 3	RXY	35
	CALL CONVMT(KD,T,RX,RY,RW,RPQ)	RXY	36
	RETURN	RXY	37
	C KIND IS ROT	RXY	38
50	12 THETA = FDAT(M,1)	RXY	39
	CS = COS(THETA)	RXY	40
	DO 14 I=1,2	RXY	41
	RX(I,I) = CS	RXY	42
	RY(I,I) = CS	RXY	43
55	14 CONTINUE	RXY	44
	GO TO 24	RXY	45
	C CHECK STORAGE	RXY	46
		RXY	47

6	CONTINUE	RXY	48
	NNQ = 12 + KADDR	RXY	49
60	IF (NNQ.GT.NQ) GO TO 20	RXY	50
	NJ = LQ + KADDR	RXY	51
	NI = NJ + 6	RXY	52
	DO 10 IJ=1,3	RXY	53
	DO 10 II=1,2	RXY	54
65	RX(II,IJ)=STORE(NJ)	RXY	55
	RY(II,IJ)=STORE(NI)	RXY	56
	NJ=NJ+1	RXY	57
	NI=NI+1	RXY	58
10	CONTINUE	RXY	59
70	CALL DATA(M,5,1,3,RW)	RXY	60
24	IF (.NOT.INV) RETURN	RXY	61
	INV = .FALSE.	RXY	62
	DO 25 J=1,3	RXY	63
	DO 25 I=1,2	RXY	64
75	AX(I,J) = RX(I,J)	RXY	65
25	AY(I,J) = RY(I,J)	RXY	66
	RX(1,1) = AX(2,2)	RXY	67
	RX(2,2) = AX(1,1)	RXY	68
	RX(1,2) = -AX(1,2)	RXY	69
80	RX(2,1) = -AX(2,1)	RXY	70
	RX(1,3) = AX(1,2)*AX(2,3) - AX(2,2)*AX(1,3)	RXY	71
	RX(2,3) = AX(2,1)*AX(1,3) - AX(1,1)*AX(2,3)	RXY	72
	RY(1,1) = AY(2,2)	RXY	73
	RY(2,2) = AY(1,1)	RXY	74
85	RY(1,2) = -AY(1,2)	RXY	75
	RY(2,1) = -AY(2,1)	RXY	76
	RY(1,3) = AY(1,2)*AY(2,3) - AY(2,2)*AY(1,3)	RXY	77
	RY(2,3) = AY(2,1)*AY(1,3) - AY(1,1)*AY(2,3)	RXY	78
	RW(1) = -RW(1)	RXY	79
90	RW(2) = -RW(2)	RXY	80
	IF (BEND.NE.0.) RW(3)= - RW(3)	RXY	81
	RETURN	RXY	82
20	WRITE (3,1) M	RXY	83
1	FORMAT (66H *** ERROR *** RETRIEVAL REQUEST EXCEEDS STORAGE RESERV	RXY	84
95	1E AT INDEX I5)	RXY	85
	ERROR=.TRUE.	RXY	86
	RETURN	RXY	87
	END	RXY	88

1

SUBROUTINE SCOPCN
RETURN
END

SCOPCN 2
SCOPCN 3
SCOPCN 4

1		SUBROUTINE SETALPH (M)	SETALPH	2
			BLANK	2
		LEVEL 2, STORE, INFF, IWORK	BLANK	3
		COMMON STORE(48000), IWORK(10)	86MARSIZ	1
5		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
10	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
15		RETURN	SETALPH	10
		END	SETALPH	11

1	SUBROUTINE SETCYB(M,KA)	SETCYB	2
		SETCYB	3
		SETCYB	4
		SETCYB	5
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
10		CONTRL	5
	COMMON/BMI/MI(5000)	BMIL	1
	COMMON/NELS/NELS	BMIL	2
	LEVEL 2,MI	BMIL	3
		BMIL	4
15		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
20	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,MSYMP	DIM	2
25		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
30	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
35	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
40	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
45	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
50	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
55		SETCYB	43
	20 CONTINUE	SETCYB	44
	C ** SYNCH INSTR. GET M, DETERMINE TYPE AND FORM MI LIST.	SETCYB	45

	MM = MDAT(MPR,1)	SETCYB	46
	NB = INFF(1,MM)	SETCYB	47
60	ASSIGN 25 TO NGO	SETCYB	48
	GO TO 55	SETCYB	49
25	CONTINUE	SETCYB	50
	IF (ITP.GT.2) GO TO 61	SETCYB	51
	GO TO (26,27) ITP	SETCYB	52
65	26 IF (KA.GT.12) GO TO 65	SETCYB	53
	CALL DATA (MM,3,1,12,MI)	SETCYB	54
	GO TO 30	SETCYB	55
27	CONTINUE	SETCYB	56
	KNB = INFF(4,MM)	SETCYB	57
70	IF (KA.GT.KNB) GO TO 65	SETCYB	58
	CALL MLIST(MM,1,KNB,MI)	SETCYB	59
30	CONTINUE	SETCYB	60
	* IF KN IS 5, IT IS AN R7 MATRIX.	SETCYB	61
	IF (KN.EQ.5) GO TO 50	SETCYB	62
75		SETCYB	63
	* TRANSFER MATRIX FROM INDEX MPR TO WORKING STORAGE WITH INDEX MEND-1.	SETCYB	64
	CALL RXY(MPR,RX,RY,RW)	SETCYB	65
	CALL STXY (NT1,RX,RY,RW)	SETCYB	66
	RETURN	SETCYB	67
80		SETCYB	68
	* 7 X 7 MATRIX	SETCYB	69
50	NT1 = M7END - 1	SETCYB	70
	CALL RTRV7(MPR,T,RW)	SETCYB	71
	CALL STOR7 (NT1,T,RW)	SETCYB	72
85	RETURN	SETCYB	73
		SETCYB	74
	*****	SETCYB	75
		SETCYB	76
	* ANALYZE TYPE OF ELEMENT TO BE USED	SETCYB	77
90	55 IF ((NB.EQ.4HCELL).OR.(NB.EQ.4HALTC)) GO TO 51	SETCYB	78
	IF ((NB.EQ.4HCFD).OR.(NB.EQ.4HDCFD)) GO TO 51	SETCYB	79
	IF (NB.NE.4HCELC) GO TO 52	SETCYB	80
	51 ITP = 1	SETCYB	81
	GO TO NGO,(10,25)	SETCYB	82
95	52 IGO = 2	SETCYB	83
	IF ((NB.EQ.4HCYA).OR.(NB.EQ.4HCYC)) GO TO 53	SETCYB	84
	IF (NB.EQ.4HMMM) GO TO 53	SETCYB	85
	IF (NB.NE.4HBML) GO TO 54	SETCYB	86
	IGO = 1	SETCYB	87
100	53 ITP = 2	SETCYB	88
	GO TO NGO,(10,25)	SETCYB	89
	54 IF (NB.NE.4H**) GO TO 60	SETCYB	90
	ITP = 3	SETCYB	91
	GO TO NGO (10,25)	SETCYB	92
105	*****	SETCYB	93
		SETCYB	94
	*****	SETCYB	95
	* ERROR EXITS	SETCYB	96
60	WRITE (3,2) NB	SETCYB	97
110	2 FORMAT (5X,5H*** ,*CYB CANNOT USE A *,A5,* SYNCH INSTRUCTION.*)	SETCYB	98
	GO TO 70	SETCYB	99
	61 WRITE (3,3)	SETCYB	100
	3 FORMAT (5X,5H*** ,* CANNOT GET ELEMENTS FOR CYB. STAR INSTR.	SETCYB	101
	1 REFERS TO A BML OR ANOTHER STAR INSTR.*)	SETCYB	102

115	GO TO 70	SETCYB 103
65	WRITE (3,1) KA,NB,KNB	SETCYB 104
1	FORMAT (5X,5H*** ,*ERROR*,1H*, *CYB WANTS *,I5,* ELEMENTS FOR 1CYCLING.*,A5,* HAS ONLY *,I5)	SETCYB 105
70	ERROR = .TRUE.	SETCYB 106
120	RETURN	SETCYB 107
	END	SETCYB 108
		SETCYB 109

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

1	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
5		SETOPS	6
	COMMON/OPLIST/LIST(200)	OPLIST	2
		OPLIST	3
	DATA LIST/200(1H)/	SETOPS	8
		SETOPS	9
10	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
15	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
20	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
25	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
30	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
35	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
40	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
45	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
50	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
55	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
60		SETOPS	60
	LIST(51) = 4HSUM	SETOPS	61
	LIST(52) = 4HACT	SETOPS	62
	LIST(53) = 4HDELE	SETOPS	63
	LIST(54) = 4HSCOP	SETOPS	64
65	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
70	LIST(60) = 4HDCCFD	SETOPS	70
	LIST(61) = 4HSTR2	SETOPS	71
	LIST(62) = 4HSTR4	SETOPS	72
	LIST(63) = 4HSTRP	SETOPS	73
	LIST(64) = 4HSTRN	SETOPS	74
75		SETOPS	75
	LIST(65) = 4HFXPT	SETOPS	76
	LIST(66) = 4HTRK	SETOPS	77
	LIST(67) = 4HPRD	SETOPS	78
	LIST(68) = 4HSXTP	SETOPS	79
80	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
85	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
90	LIST(81) = 4HVAR	SETOPS	90
	LIST(82) = 4HNPOL	SETOPS	91
	LIST(83) = 4HORBC	STPS683	1
		SETOPS	92
	LIST(100) = 4HPBML	MY3STOP	1
95	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
100	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
105	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
110	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

115		LIST(121) = 4HATAN	SETOPS	113
		LIST(122) = 4HABS	SETOPS	114
		LIST(123) = 4HCALC	SETOPS	115
			SETOPS	116
		LIST(151) = 4HBEAM	SETOPS	117
120		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
125	C		SETOPS	123
		LIST(161) = 4HTRKB	SETOPS	124
		LIST(162) = 4HSOLV	SETOPS	125
		LIST(164) = 4HTRKE	86SETOPS	1
		LIST(165) = 4HTRKM	86SETOPS	2
130		LIST(166) = 4HIBET	86SETOPS	3
		LIST(163) = 4HSMIN	SETOPS	126
		RETURN	SETOPS	127
		END	SETOPS	128

```

1      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
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
10         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
15                                           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
20         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
25                                           SETFIT 15
      QSW = .FALSE.                                 SETFIT 16
      BSW = .FALSE.                                 SETFIT 17
      VSW = .FALSE.                                 SETFIT 18
      RSW = .FALSE.                                 SETFIT 19
30         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
35         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
40         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
45         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
50         .)                                         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
55         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)	SETFIT	48
		SETFIT	49
60	110 IF (MODE.EQ.2) RETURN	SETFIT	50
	* XSW SUPPRESSES EXECUTION	SETFIT	51
	115 IF (XSW) RETURN	SETFIT	52
		SETFIT	53
	116 CONTINUE	SETFIT	54
65	CALL DATA (M,2,3,2,MV)	SETFIT	55
	CALL DATA(M,3,1,2,IV)	SETFIT	56
	CALL DATA(M,1,1,2,Q)	SETFIT	57
	MSR = MDAT(M,1)	SETFIT	58
	MHZ = MDAT(M,2)	SETFIT	59
70	K1 = KA	SETFIT	60
	K2 = KB	SETFIT	61
	IF (.NOT.RSW) GO TO 117	SETFIT	62
	K1 = BDAT(M,5)	SETFIT	63
	K2 = BDAT(M,6)	SETFIT	64
75	117 IF (FSW) GO TO 120	SETFIT	65
		SETFIT	66
	* CALL FITT EXCEPT ON OPTION IN CASE OF FITQ.	SETFIT	67
	CALL FITT(MSR,MHZ,MV,IV,Q,K1,K2)	SETFIT	68
	RETURN	SETFIT	69
80	120 CALL FITE(MSR,MHZ,MV,IV,Q,K1,K2)	SETFIT	70
	RETURN	SETFIT	71
	END	SETFIT	72

1	SUBROUTINE SETMAT(MT)	SETMAT	2
	C ROUTINE SETS UP AND STORES UNIT AND TRANSPOSE MATRICES	SETMAT	3
	DIMENSION T(7,7),RW(3)	SETMAT	4
	ASSIGN 6 TO NEXT	SETMAT	5
5	DO 3 I=1,3	SETMAT	6
	3 RW(I) = 0.	SETMAT	7
	4 DO 5 J=1,7	SETMAT	8
	DO 5 I=1,7	SETMAT	9
	5 T(I,J) = 0.	SETMAT	10
10	GO TO NEXT,(6,10)	SETMAT	11
	C COMPUTE UNIT MATRIX NAMED (1)	SETMAT	12
	6 DO 7 I=1,7	SETMAT	13
	7 T(I,I) = 1.	SETMAT	14
	RW(3) = T(5,6)	SETMAT	15
15	CALL STOR7(MT,T,RW)	SETMAT	16
	MT = MT - 1	SETMAT	17
	ASSIGN 10 TO NEXT	SETMAT	18
	GO TO 4	SETMAT	19
	C COMPUTE TRANSPOSE NAMED (S)	SETMAT	20
20	10 DO 12 I=1,5,2	SETMAT	21
	T(I,I + 1) = -1.	SETMAT	22
	12 T(I+1,I) = + 1.	SETMAT	23
	T(7,7) = +1.	SETMAT	24
	RW(3) = T(5,6)	SETMAT	25
25	CALL STOR7(MT,T,RW)	SETMAT	26
	RETURN	SETMAT	27
	END	SETMAT	28

1	SUBROUTINE SHF(M,PAR)	SHF	2
		BLANK	2
	LEVEL 2, STORE,INFF,IWORK	BLANK	3
	COMMON STORE(48000),IWORK(10)	86MARSIZ	1
5	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
10	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
15	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
20	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
25	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
30	CALL STXY(M,RX,RY,RW)	SHF	26
	RETURN	SHF	27
	END	SHF	28

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

SKIP1	2
SKIP1	3
SKIP1	4
SKIP1	5
SKIP1	6
SKIP1	7

1	SUBROUTINE SHF7(M,PAR)	SHF7	2
	C	SHF7	3
	C SQ SHF7 R17 R27 R37 R47 R57 R67	SHF7	4
	C	SHF7	5
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
10	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
15	CALL STOR7(M,T,RW)	SHF7	15
	RETURN	SHF7	16
	END	SHF7	17

1	SUBROUTINE SMIN(M)	SMIN	2
		BLANK	2
	LEVEL 2, STORE, INFF, IWORK	BLANK	3
	COMMON STORE(48000), IWORK(10)	86MARSIZ	1
5	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
10	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
15	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
20	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
25	C SIMPLEX 1500. 0.1	SMIN	16
	C END RETURN	SMIN	17
		SMIN	18
	SMINCAL = .TRUE.	SMIN	19
	KA=INFF(4,M)	SMIN	20
30		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
35	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
40	CALL DATA(M,1,LFPI,5,DWORD7(1,I))	SMIN	31
	1 CONTINUE	SMIN	32
		SMIN	33
	RETURN	SMIN	34
	END	SMIN	35

1		SUBROUTINE SOL(M,PAR)	SOL	2
	C	FORMS A MATRIX REPRESENTING A SOLENOID.	SOL	3
	C	SOL	4
	C	S SOL L (BLANK) BRHO B0	SOL	5
5	C	SOL	6
	C		SOL	7
	C	L = LENGTH OF SOLENOID	SOL	8
	C	BRHO = B*RHO (THE RIGIDITY OF THE PARTICLE)	SOL	9
	C	B0 = LONGITUDINAL FIELD IN SOLENOID	SOL	10
10	C		SOL	11
		REAL K,L	SOL	12
		DIMENSION PAR(1),T(7,7),RW(3)	SOL	13
	C		SOL	14
		DO 11 J=1,7	SOL	15
15		DO 10 I=1,7	SOL	16
	10	T(I,J) = 0.	SOL	17
	11	CONTINUE	SOL	18
		DO 12 I=1,3	SOL	19
	12	RW(I) = 0.	SOL	20
20	C		SOL	21
		L = PAR(1)	SOL	22
		BRHO = PAR(3)	SOL	23
		B0 = PAR(4)	SOL	24
		K = B0/(2.*BRHO)	SOL	25
25		C = COS(K*L)	SOL	26
		S = SIN(K*L)	SOL	27
		SC = S*C	SOL	28
		SSQ = S*S	SOL	29
		CSQ = C*C	SOL	30
30		T(1,1) = CSQ	SOL	31
		T(2,1) = -K*SC	SOL	32
		T(3,1) = -SC	SOL	33
		T(4,1) = K*SSQ	SOL	34
		T(1,2) = SC/K	SOL	35
35		T(2,2) = CSQ	SOL	36
		T(3,2) = -SSQ/K	SOL	37
		T(4,2) = -SC	SOL	38
		T(1,3) = SC	SOL	39
		T(2,3) = -K*SSQ	SOL	40
40		T(3,3) = CSQ	SOL	41
		T(4,3) = -K*SC	SOL	42
		T(1,4) = SSQ/K	SOL	43
		T(2,4) = SC	SOL	44
		T(3,4) = SC/K	SOL	45
45		T(4,4) = CSQ	SOL	46
		T(5,5) = 1.	SOL	47
		T(6,6) = 1.	SOL	48
		T(7,7) = 1.	SOL	49
	C		SOL	50
50		RW(1) = L	SOL	51
		CALL STOR7(M,T,RW)	SOL	52
		RETURN	SOL	53
		END	SOL	54

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

22	I	PAR	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
23	I	PAR	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

1	SUBROUTINE SOLV(M)	SOLV	2
		BLANK	2
	LEVEL 2, STORE, INFF, IWORK	BLANK	3
	COMMON STORE(48000), IWORK(10)	86MARSIZ	1
5	DIMENSION INFF(24,2000)	86MARSIZ	2
	EQUIVALENCE (INFF,STORE)	86MARSIZ	3
		BLANK	5
	COMMON/BMI1/MI1(16000)	NV3BMI1	1
	COMMON/NELS1/NELS1	BMI1L	2
10	LEVEL 2,MI1	BMI1L	3
		BMI1L	4
		BMI1	3
	COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,	CONTRL	2
	1 LDFLG,FIN	CONTRL	3
15	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
20	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
25	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
30	9/CONVER/ EPSI ,APSI ,VTEST ,NSTEPQ ,NFCN ,NFCNMX	COMMONT	29
	A/CARD / CWORD ,CWORD2 ,CWORD3 ,WORD7(7)	COMMONT	30
	B/MINIMA/ AMIN ,UP ,NEWMIN ,ITAU ,SIGMA	COMMONT	31
	K/COMMND/ NNAME ,CNAME(25) ,CNAM2(25) ,CNAM3(25)	COMMONT	32
	COMMON /MATCH/ KAT,KBT,MPAR(30),IPAR(30),MSR,MTR,LRTAB,FNORM,	MATCH	2
35	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
40	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
45	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
50	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
55	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/	SOLV	24
	DATA LIM1,LIM2/7HINITIAL,7H FINAL/	SOLV	25
60	DATA MINZER/77777 77777 77777 77777B/	SOLV	26
	DATA MASK1/77000 00000 00000 00000B/	SOLV	27
	DATA IBLNK1/55000 00000 00000 00000B/	SOLV	28
	DATA NPR/6/	SOLV	29
	DATA IBLNK/1H /	SOLV	30
65	C _____	SOLV	31
		SOLV	32
	C INSTRUCTION SOLV — BETATRON FUNCTION FITTING USING MINUIT	SOLV	33
		SOLV	34
	C	SOLV	35
70	C XX SOLV KA KB//SR TR IP1 IP2 ITS TOL PRN SAV PRI LAMDA	SOLV	36
	C (CONSTR CD 1) BNAM TYPE POS1 POS2 POS3 POS4 POS5 POS6 BFIT SIG	SOLV	37
	C (CONSTR CD 2) BNAM TYPE POS1 POS2 POS3 POS4 POS5 POS6 BFIT SIG	SOLV	38
		SOLV	39
	C (CONSTR CD KA) BNAM TYPE POS1 POS2 POS3 POS4 POS5 POS6 BFIT SIG	SOLV	40
75	C (VARIABLE CD 1) PAR I1 I2 I3 I4 I5 LOWER UPPER STP	SOLV	41
	C (VARIABLE CD 2) PAR I1 I2 I3 I4 I5 LOWER UPPER STP	SOLV	42
		SOLV	43
	C (VARIABLE CD KB) PAR I1 I2 I3 I4 I5 LOWER UPPER STP	SOLV	44
	C	SOLV	45
80	C 2ND FORMAT FOR VARIABLE CARDS —	SOLV	46
	C	SOLV	47
	C PAR1 PAR1 PAR3 PAR4 PAR5 I LOWER UPPER STO	SOLV	48
	C	SOLV	49
	C PARK IS NAME OF KTH ELEMENT WHOSE ITH INPUT NUMBER IS TO BE VARIED..	SOLV	50
85	C _____	SOLV	51
		SOLV	52
	C FIRST CARD —	SOLV	53
	C KA = NUMBER OF CONSTRAINT CARDS	SOLV	54
	C KB = NUMBER OF VARIABLE CARDS	SOLV	55
90	C SR = NAME OF SYNCH SUBROUTINE WITH VARIABLES AND A TRKB INSTRUCTION	SOLV	56
	C TR = NAME OF TRKB INSTRUCTION IN SR.	SOLV	57
	C TR MAY ALSO BE A CYC INSTRUCTION, WITH ITS KA SET TO 21,	SOLV	58
	C OR AN MMM OR EQUIVALENT INSTRUCTION THAT DEFINES A SINGLE	SOLV	59
	C TRANSFER MATRIX FOR THE LATTICE PERIOD.	SOLV	60
95	C IP1 = START POSITION OF TRKB RUN	SOLV	61
	C IP2 = END POSITION OF TRKB RUN. IF ZERO, USE VALUES ON TRKB CARD.	SOLV	62
	C ITS = MAX NO. OF CALLS TO FCN BY MINUIT	SOLV	63
	C TOL = DESIRED TOLERANCE ON FCN (FCN .LE. 10**TOL)—AN INTEGER	SOLV	64
	C PRN = PRINTOUT LEVEL FROM MINUIT (0 GIVES LEAST OUTPUT)	SOLV	65
100		SOLV	66
	C SAV——INITIALIZED TO 0 BY SRUN. IF SAV DIFFERENT FROM PREVIOUS	SOLV	67
	C VALUE, CURRENT VARIABLE VALUES ARE SAVED IN ARRAY V.	SOLV	68
	C SOLV ALWAYS TAKES INITIAL VARIABLE VALUES FROM V.	SOLV	69
	C NORMALLY SET SAV=0, THEN VALUES FROM PREVIOUS FIT ARE USED	SOLV	70
105	C FOR STARTING VALUES. SET SAV=1 TO ALWAYS START FROM	SOLV	71
	C SAME INITIAL VALUES.	SOLV	72
	C PRI = 0 (OR BLANK) — PRINT BETAS BEFORE AND AFTER FITTING	SOLV	73
	C 1 — PRINT BETAS AFTER FITTING ONLY	SOLV	74
	C 2 — DO NOT PRINT BETAS	SOLV	75
110	C 10,11,12 — SAME AS 0,1,2 AND FORCE DOIT TO ALWAYS CALL MI	SOLV	76
		SOLV	77
	C LAMDA — SEE BELOW UNDER VARIABLES	SOLV	78
	C _____	SOLV	79
	C CONSTRAINT CARDS —	SOLV	80

```

115      C          BNAM TYPE IP1 IP2 IP3 IP4 IP5 IP6 BFIT SIG SOLV      81
      C  EACH CONSTRAINT CARD MAY INVOLVE UP TO 6 SEPARATE POSITIONS      SOLV      82
      C  BNAM = NAMES OF BETA FUNCTION CONSTRAINTS——      SOLV      83
      C      NUX BX AX GX X DX NUY BY AY GY Y DY      SOLV      84
      C      S THET —— WST MATRX      SOLV      85
120      C      SOLV      86
      C      S = PATH LENGTH, THET = HORIZONTAL BEND ANGLE,      SOLV      87
      C      WST COMBINES THREE CONSTRAINTS AX, AY, AND X      SOLV      88
      C      MTRX —— CONSTRAIN MATRIX ELEMENT I1,I2 OF MATRIX      SOLV      89
      C      CORRESPONDING TO TR.      SOLV      90
125      C      GAMT —— TRANSITION GAMMA ( TR MUST BE A CYC 21 INSTR.)      SOLV      91
      C      SOLV      92
      C      TYPE = STD OR BLANK ——BETAS SHOULD FIT SPECIFIED VALUES      SOLV      93
      C      TYPE = DIF —— DIFFERENCE OF BETAS BETWEEN SUCCESSIVE POSITIONS      SOLV      94
      C      SUM —— SUM (DITTO)      SOLV      95
130      C      SHOULD FIT SPECIFIED VALUE BFIT.      SOLV      96
      C      WHEN BFIT = S, DIF AFFORDS A WAY TO OBTAIN GIVEN SEPARATIONS      SOLV      97
      C      BETWEEN SPECIFIED POSITIONS.      SOLV      98
      C      TYPE = BFIT —— BNAM IS THE NAME OF A PREVIOUSLY DEFINED MATRIX;      SOLV      99
      C      THE BETA FUNCTIONS OF THAT MATRIX ARE CALCULATED;      SOLV     100
135      C      THEIR VALUES ARE THOSE THAT SOLV SEEKS TO FIT.      SOLV     101
      C      TYPE = NO —— PASS OVER THIS CONSTRAINT      SOLV     102
      C      SOLV     103
      C      I1,I2, ...I5 ARE POSITIONS AT WHICH THE CONSTRAINT APPLIES.      SOLV     104
      C      THE POSITIONS CAN BE ENTERED SYMBOLICALLY IN THE FORM AAANN,      SOLV     105
140      C      WHERE AAA IS THE NAME OF A BEAMLINE ELEMENT, AND NN DENOTES      SOLV     106
      C      WHICH OCCURRENCE OF AAA IN THE BEAMLINE IS MEANT.      SOLV     107
      C      IF NN IS BLANK, THE FIRST OCCURENCE OF THE NAME IS UNDERSTOOD.      SOLV     108
      C      SOLV     109
      C      SOLV     110
145      C      SIGMA = GOODNESS OF FIT DESIRED (INVERSE WEIGHT).      SOLV     111
      C      SOLV     112
      C      SOLV     113
      C      IF BNAM IS NOT A RECOGNIZED NAME, CONSTRAINT IGNORED, BUT POSITIONS      SOLV     114
      C      CAN BE USED TO MARK FIT POINTS FOR SUBSEQUENT CONSTRAINT CARDS.      SOLV     115
150      C      IF A CONSTRAINT CARD HAS NO POSITIONS, THE POSITIONS ON THE NEAREST      SOLV     116
      C      PREVIOUS CARD WITH POSITIONS PUNCHED WILL APPLY.      SOLV     117
      C      SOLV     118
      C      FCN = FUNCTION TO BE MINIMIZED BY SUBROUTINE FCN IS      SOLV     119
      C      FCN = SUM[( VALUE-FITVALUE)/TOLERANCE]**2]/(NUMBER OF CONSTRAINTS)      SOLV     120
155      C      SOLV     121
      C      VARIABLE CARDS ——      SOLV     122
      C      KB = NUMBER OF VARIABLE CARDS, EACH MAY HAVE 1 PARAMETER NAME      SOLV     123
      C      AND UP TO 5 PARAMETER INDICES.      SOLV     124
      C      SECOND FORMAT ——UP TO 5 PARAMETER NAMES FOLLOWED BY 1 INDEX.      SOLV     125
160      C      IF A VARIABLE CARD HAS NO POSITIONS, THE POSITIONS ON THE NEAREST      SOLV     126
      C      PREVIOUS CARD WITH POSITIONS PRESENT WILL APPLY.      SOLV     127
      C      SOLV     128
      C      IF STEPSIZ NOT ENTERED, DEFAULT IS (UPRBOUND-LOWBOUND)/2.      SOLV     129
      C      IF STEPSIZ IS NEGATIVE, VARIABLE ENTERS INTO FCN WITH      SOLV     130
165      C      FIT VALUE = (UPPER+LOWER)/2, TOLERANCE = LAMDA*(UPPER-LOWER)/2      SOLV     131
      C      SOLV     132
      C      PASS ITS, TOL, PRN TO MINUIT      SOLV     133
      C      IF (SMINCAL) GO TO 5      SOLV     134
      C      SAVE DEFAULT VALUES      SOLV     135
170      C      ISV1 = DWORD7(1,1)      SOLV     136
      C      ISV2 = DWORD7(1,2)      SOLV     137

```

		ISV3 = DWORD7(2,2)	SOLV	138
		ITS = IDAT(M,1)	SOLV	139
		IF(ITS.EQ.0) GO TO 3	SOLV	140
175		DWORD7(1,2)=ITS	SOLV	141
	3	ITOL = IDAT(M,2)	SOLV	142
		DWORD7(2,2)=10.**ITOL	SOLV	143
	4	IPR = IDAT(M,3)	SOLV	144
		IF(IPR.EQ.0) GO TO 5	SOLV	145
180		DWORD7(1,1)=IPR	SOLV	146
	5	CONTINUE	SOLV	147
		IVSAV0=IVSAV	SOLV	148
		IVSAV = IDAT(M,4)	SOLV	149
		IPP = IDAT(M,5)	SOLV	150
185		MIFLG = .FALSE.	SOLV	151
		MATFLG = .FALSE.	SOLV	152
		GLOBAL = .FALSE.	SOLV	153
		IF (IPP.LT.10) GO TO 52	SOLV	154
		MIFLG = .TRUE.	SOLV	155
190		IPP = IPP - 10	SOLV	156
	52	KAT=INFF(4,M)	SOLV	157
		KAR=KAT	SOLV	158
		KBT=INFF(5,M)	SOLV	159
		KBR=KBT	SOLV	160
195		KBTS=KBT	SOLV	161
		MSR=MDAT(M,1)	SOLV	162
		MTR=MDAT(M,2)	SOLV	163
		KATR = 0	SOLV	164
		OPTR = INFF(1,MTR)	SOLV	165
200		IF ((OPTR.NE.HTRKB).AND.(OPTR.NE.HCYC)) GO TO 55	SOLV	166
		MBML = MDAT(MTR,1)	SOLV	167
		NDAT = INFF(17,MBML)	SOLV	168
		CALL MIFILL(MBML,1,NDAT,NELS1,MI1)	SOLV	169
		NELS2 = NELS1	SOLV	170
205		KBTR = INFF(5,MTR)	SOLV	171
		IF (OPTR.NE.HTRKB) GO TO 53	SOLV	172
		KATR = INFF(4,MTR)	SOLV	173
		CALL DATA (M,2,3,2,NIP)	SOLV	174
		CALL LOCS(NIP,2,NELS1,MI1,IP)	SOLV	175
210		IP1 = IP(1)	SOLV	176
		IP2 = IP(2)	SOLV	177
		IF (IP1.EQ.0.AND.IP2.EQ.0) GO TO 54	SOLV	178
		CALL REPINT(MTR,1,IP1)	SOLV	179
		CALL REPINT(MTR,2,IP2)	SOLV	180
215		GO TO 54	SOLV	181
	53	IF (KBTR.LT.0) NELS2 = NELS1 + 1	SOLV	182
	54	LQTR = INFF(24,MTR)	SOLV	183
		LTRTAB = LQTR - 14*KATR	SOLV	184
		GO TO 51	SOLV	185
220	55	MATFLG = .TRUE.	SOLV	186
	C 51	NASR = INFF(2,MSR)	SOLV	187
	51	CONTINUE	SOLV	188
		XLAMDA=FDAT(M,1)	SOLV	189
			SOLV	190
225			SOLV	191
	C	GET CONSTRAINT DATA AND STORE IN ARRAYS FOR FCN	SOLV	192
			SOLV	193
		DO 61 KK=1,30	SOLV	194

	61	IPOS(7, KK)=0	SOLV	195
230		NCONSTR = 0	SOLV	196
		K=0	SOLV	197
		DO 2 N=1, KAR	SOLV	199
235		K=K+1	SOLV	200
		ICON(K)=0	SOLV	201
		NTYPE = BDAT(M, 8*N-2)	SOLV	202
		ITYPE(K)=0	SOLV	203
		IF(NTYPE.EQ.5HNO) ITYPE(K)=-1	SOLV	204
240		IF(NTYPE.EQ.5HSTD) ITYPE(K)=0	SOLV	205
		IF(NTYPE.EQ.5HDIF) ITYPE(K)=1	SOLV	206
		IF (NTYPE.EQ.5HSUM) ITYPE(K)=2	SOLV	207
		NCONK = BDAT(M, 8*N-3)	SOLV	208
		IF (NTYPE.NE.5HBFIT) GO TO 62	SOLV	209
245		ITYPE(K)=3	SOLV	210
		GO TO 63	SOLV	211
			SOLV	212
			SOLV	213
	62	DO 6 I=1, 42	SOLV	214
250	6	IF (NCONK.EQ.NAMBFC(I)) ICON(K)=I	SOLV	215
			SOLV	216
		ICONK = ICON(K)	SOLV	217
		IF (ICON(K).GE.34.AND.ICON(K).LE.37) MIFLG = .TRUE.	SOLV	218
		IF(ICON(K).EQ.0) ITYPE(K)=-1	SOLV	219
255	63	LPOSK = 7 + 8*(N-1)	SOLV	220
		IF (ICON(K).LE.14.OR.ICON(K).GE.34) GO TO 65	SOLV	221
		IPOS(1, K) = NEL2	SOLV	222
		NPOS(K) = 1	SOLV	223
		GLOBAL = .TRUE.	SOLV	224
		GO TO 8	SOLV	225
260	65	CONTINUE	SOLV	226
		CALL DATA (M, 2, LPOSK, 6, NAIPOS)	SOLV	227
		CALL LOCS(NAIPOS, 6, NEL2, MI1, IPOS(1, K))	SOLV	228
		NPOS(K)=0	SOLV	229
			SOLV	230
265		I=0	SOLV	231
	7	I=I+1	SOLV	232
		IF (NAIPOS(I).NE.IBLNK) GO TO 72	SOLV	233
		NP1=0	SOLV	234
		I1=I	SOLV	235
270			SOLV	236
		DO 71 II=I1, 6	SOLV	237
		IPOS(II, K) = IPOS(II+1, K)	SOLV	238
	71	IF (NAIPOS(I).NE.IBLNK) NP1 = NP1 + 1	SOLV	239
			SOLV	240
275		I=I-1	SOLV	241
		IF(NP1.EQ.0) GO TO 73	SOLV	242
		GO TO 7	SOLV	243
	72	NPOS(K)=NPOS(K)+1	SOLV	244
		IF(I.LT.6) GO TO 7	SOLV	245
280			SOLV	246
	73	CONTINUE	SOLV	247
		IF(NPOS(K).NE.0) GO TO 8	SOLV	248
		IF (ICON(K).NE.0) GO TO 74	SOLV	249
		K=K-1	SOLV	250
285		GO TO 2	SOLV	251

	74	CONTINUE	SOLV	252
		IF (K.GT.1) GO TO 10	SOLV	253
		NUMCON=1	SOLV	254
		GO TO 32	SOLV	255
290	10	NPOS(K)=NPOS(K-1)	SOLV	256
		DO 9 I=1,6	SOLV	257
	9	IPOS(I,K)=IPOS(I,K-1)	SOLV	258
			SOLV	259
	8	CONTINUE	SOLV	260
295		NUMCON=NPOS(K)	SOLV	261
	32	CONTINUE	SOLV	262
		IF (ITYPE(K).EQ.-1) NUMCON=0	SOLV	263
		IF ((ITYPE(K).EQ.1).OR.(ITYPE(K).EQ.2)) NUMCON=NUMCON-1	SOLV	264
			SOLV	265
300		LFPK=2*(N-1)+1	SOLV	266
		BET0(K) = FDAT(M,LFPK+1)	SOLV	267
		SIG(K) = FDAT(M,LFPK+2)	SOLV	268
		IF(SIG(K).EQ.0.) SIG(K)=1.E20	SOLV	269
		WT(K)=1./SIG(K)	SOLV	270
305			SOLV	271
		IF (ICON(K).LT.38) GO TO 13	SOLV	272
		LL = 1	SOLV	273
		IF (ICONK.NE.40.AND.ICONK.NE.39) GO TO 28	SOLV	274
		LL = 2	SOLV	275
310		ICON(K) =3	SOLV	276
		ICON(K+1)=6	SOLV	277
		ICON(K+2)=9	SOLV	278
			SOLV	279
		IF (ICONK.EQ.40) GO TO 27	SOLV	280
315		LL = 3	SOLV	281
		ICON(K+3) = 37	SOLV	282
		BET0F = BET0(K)	SOLV	283
		BET0(K) = 0.	SOLV	284
		GO TO 27	SOLV	285
320			SOLV	286
	28	IF (ICONK.NE.41) GO TO 29	SOLV	287
		ICON(K) = 5	SOLV	288
		ICON(K+1) = 6	SOLV	289
		GO TO 27	SOLV	290
325			SOLV	291
	29	IF (ICONK.NE.42.AND.ICONK.NE.38) GO TO 27	SOLV	292
		ICON(K) = 3	SOLV	293
		ICON(K+1) = 9	SOLV	294
			SOLV	295
330		IF (ICONK.EQ.42) GO TO 27	SOLV	296
		LL = 2	SOLV	297
		ICON(K+2) = 37	SOLV	298
		BET0F = BET0(K)	SOLV	299
		BET0(K) = 0.	SOLV	300
335			SOLV	301
			SOLV	302
	27	CONTINUE	SOLV	303
		KL=K	SOLV	304
		DO 11 L=1,LL	SOLV	305
340		KL=KL+1	SOLV	306
		ITYPE(KL)=ITYPE(K)	SOLV	307
		NPOS(KL)=NPOS(K)	SOLV	308

	BET0(KL)=BET0(K)	SOLV	309
	SIG(KL)=SIG(K)	SOLV	310
345	WT(KL)=WT(K)	SOLV	311
	DO 12 I=1,6	SOLV	312
12	IPOS(I,KL)=IPOS(I,K)	SOLV	313
11	CONTINUE	SOLV	314
		SOLV	315
350	IF (ICONK.EQ.38.OR.ICONK.EQ.39) BET0(KL)=BET0F	SOLV	316
	NUMCON = (LL+1)*NUMCON	SOLV	317
	K = KL	SOLV	318
	KAT = KAT + LL	SOLV	319
	GO TO 131	SOLV	320
355		SOLV	321
	13 CONTINUE	SOLV	322
		SOLV	323
	IF ((ICON(K).NE.35).AND.(ICON(K).NE.36)) GO TO 14	SOLV	324
	NPOS(K) = 4	SOLV	325
360	NUMCON = 1	SOLV	326
	IF (IPOS(3,K).EQ.0) IPOS(3,K)=1	SOLV	327
	IF (IPOS(4,K).EQ.0) IPOS(4,K)=1	SOLV	328
	14 CONTINUE	SOLV	329
	IF (ICON(K).NE.36) GO TO 24	SOLV	330
365	NN = IPOS(1,K)/2	SOLV	331
	ISGN = 1	SOLV	332
	IF (IPOS(2,K).LT.0) ISGN = -1	SOLV	333
	IPOS(2,K) = 0	SOLV	334
	ICON(K) = 35	SOLV	335
370	KK = K	SOLV	336
	DO 25 II=1,NN	SOLV	337
	KK = KK + 1	SOLV	338
	DO 26 JJ=3,4	SOLV	339
	26 IPOS(JJ,KK) = IPOS(JJ,K)	SOLV	340
375	ICON(KK) = 35	SOLV	341
	NPOS(KK) = 4	SOLV	342
	SIG(KK) = SIG(K)	SOLV	343
	BET0(KK) = BET0(K)	SOLV	344
	WT(KK) = WT(K)	SOLV	345
380	IPOS(1,KK) = IPOS(1,KK-1) -2	SOLV	346
	IPOS(2,KK) = IPOS(2,KK-1) + 2*ISGN	SOLV	347
	25 CONTINUE	SOLV	348
	K = K + NN	SOLV	349
	KAT = KAT + NN	SOLV	350
385	NUMCON = NN + 1	SOLV	351
	24 CONTINUE	SOLV	352
		SOLV	353
	IF (ITYPE(K).NE.3) GO TO 131	SOLV	354
	MBET=ELNUM(NCONK)	SOLV	355
390	CALL BET(MBET,BXI,BYI,0)	SOLV	356
	ICON(K) =2	SOLV	357
	ICON(K+1)=3	SOLV	358
	ICON(K+2)=5	SOLV	359
	ICON(K+3)=6	SOLV	360
395	ICON(K+4)=8	SOLV	361
	ICON(K+5)=9	SOLV	362
	BET0(K) =BXI(2)	SOLV	363
	BET0(K+1)=BXI(3)	SOLV	364
	BET0(K+2)=BXI(5)	SOLV	365

400	BET0(K+3)=BXI(6)	SOLV	366
	BET0(K+4)=BYI(2)	SOLV	367
	BET0(K+5)=BYI(3)	SOLV	368
		SOLV	369
	DO 132 L=1,5	SOLV	370
405	ITYPE(K+L)=ITYPE(K)	SOLV	371
	NPOS(K+L)=NPOS(K)	SOLV	372
	SIG(K+L)=SIG(K)	SOLV	373
	WT(K+L)=WT(K)	SOLV	374
		SOLV	375
410	DO 133 I=1,6	SOLV	376
	133 IPOS(I,K+L)=IPOS(I,K)	SOLV	377
		SOLV	378
	132 CONTINUE	SOLV	379
		SOLV	380
415	NUMCON=6*NUMCON	SOLV	381
	K=K+5	SOLV	382
	KAT=KAT+5	SOLV	383
	131 CONTINUE	SOLV	384
		SOLV	385
420	NCONSTR = NCONSTR + NUMCON	SOLV	386
	2 CONTINUE	SOLV	387
	DO 82 K=1,30	SOLV	388
	DO 82 I=1,6	SOLV	389
	82 IPRPOS(I,K) = IBLNK	SOLV	390
425	DO 90 K=1,NCONSTR	SOLV	391
	NP=NPOS(K)	SOLV	392
	IF (ICON(K).GT.14) GO TO 84	SOLV	393
	IF ((OPTR.NE.HTRKB).AND.(OPTR.NE.HCYC)) GO TO 90	SOLV	394
	GO TO 86	SOLV	395
430	84 IF (ICON(K).LE.32) GO TO 90	SOLV	396
	IF (ICON(K).LT.38) GO TO 86	SOLV	397
	IF ((OPTR.NE.HTRKB).AND.(OPTR.NE.HCYC)) GO TO 90	SOLV	398
	86 DO 88 I=1,NP	SOLV	399
	88 ENCODE(5,106,IPRPOS(I,K)) IPOS(I,K)	SOLV	400
435	106 FORMAT(I5)	SOLV	401
	90 CONTINUE	SOLV	402
		SOLV	403
		SOLV	404
	C GET DATA ON VARIABLES AND LOAD COMMON BLOCKS FOR FCN AND MIDATA	SOLV	405
440	C MPAR(J) = M-NO OF JTH VARIABLE	SOLV	406
		SOLV	407
	VARWTS=.FALSE.	SOLV	408
	NVARWT=0	SOLV	409
	IF(XLAMDA.EQ.0.) XLAMDA=1.0	SOLV	410
445	L=0	SOLV	411
		SOLV	412
	DO 1 J=1,KBR	SOLV	413
	L2=1	SOLV	414
	LOCJ=2*KAR+3*J-2	SOLV	415
450	AAJ=FDAT(M,LOCJ+1)	SOLV	416
	BBJ=FDAT(M,LOCJ+2)	SOLV	417
	WRJ=FDAT(M,LOCJ+3)	SOLV	418
	VALJ=(BBJ+AAJ)/2.	SOLV	419
	DEL=ABS(BBJ-AAJ)/2.	SOLV	420
455	IF (DEL.EQ.0.) DEL=1.	SOLV	421
	SIGV=XLAMDA*DEL	SOLV	422

	WTVARJ=1./SIGV	SOLV	423
	JWT=.FALSE.	SOLV	424
	IF(WRJ.EQ.0.) WRJ=DEL	SOLV	425
460	IF(WRJ.GE.0.) GO TO 23	SOLV	426
	VARWTS=.TRUE.	SOLV	427
	JWT=.TRUE.	SOLV	428
	WRJ=-WRJ	SOLV	429
	23 CONTINUE	SOLV	430
465	LVARJ = 8*KAR + 6*J -1	SOLV	431
	MPARJ=MDAT(M, LVARJ)	SOLV	432
	MPAR(L+1)=MPARJ	SOLV	433
	NAMPAR(1)=BDAT(M, LVARJ)	SOLV	434
		SOLV	435
470	C FIND OUT FORMAT TYPE	SOLV	436
	IFORM=1	SOLV	437
	LI=L+1	SOLV	438
		SOLV	439
	DO 17 I=1,5	SOLV	440
475	ERPR=.FALSE.	SOLV	441
	MPARI=MDAT(M, LVARJ+I)	SOLV	442
	IF (MPARI.EQ.MINZER) GO TO 16	SOLV	443
	IFORM=2	SOLV	444
	L2=L2+1	SOLV	445
480	LI=LI+1	SOLV	446
	MPAR(LI)=MPARI	SOLV	447
	NAMPAR(L2)=BDAT(M, LVARJ+I)	SOLV	448
	GO TO 17	SOLV	449
	16 NAIND(I)=BDAT(M, LVARJ+I)	SOLV	450
485	DECODE(5,100,NAIND(I)) IND(I)	SOLV	451
	100 FORMAT(I5)	SOLV	452
	C TO MAKE INPUT COMPATIBLE WITH VAX VERSION	86SOLV	1
	DECODE(5,2000, NAIND(I))LL	86SOLV	2
	2000 FORMAT(4XA1)	86SOLV	3
490	IF(LL.EQ.1H) IND(I)=IND(I)/10	86SOLV	4
	IF(IND(I).EQ.0) GO TO 17	SOLV	453
	NAIND2=NAIND(I)	SOLV	454
	IND2=IND(I)	SOLV	455
	17 CONTINUE	SOLV	456
495		SOLV	457
	L0=L+1	SOLV	458
	IF(IFORM.EQ.2) GO TO 18	SOLV	459
	C FORMAT 1	SOLV	460
		SOLV	461
500	DO 15 I=1,5	SOLV	462
	IF(IND(I).EQ.0) GO TO 15	SOLV	463
	ENCODE(10,101,NAMI(I)) NAMPAR(1),NAIND(I)	SOLV	464
	101 FORMAT(2A5)	SOLV	465
	L=L+1	SOLV	466
505	NAM(L)=NAMI(I)	SOLV	467
	MPAR(L)=MPARJ	SOLV	468
	IPAR(L)=IND(I)	SOLV	469
	15 CONTINUE	SOLV	470
	C IF NO INDEX WAS PRESENT, USE PREVIOUSLY DEFINED ONE.	SOLV	471
510	IF (L.GE.L0) GO TO 20	SOLV	472
	L = L + 1	SOLV	473
	ENCODE (10,101,NAM(L)) NAMPAR(1),NAIND2	SOLV	474
	MPAR(L) = MPARJ	SOLV	475

	IPAR(L) = IND2	SOLV	476
515	GO TO 20	SOLV	477
		SOLV	478
	C FORMAT 2	SOLV	479
	18 LL2=L2	SOLV	480
		SOLV	481
520	DO 19 L2=1,LL2	SOLV	482
	L=L+1	SOLV	483
	ENCODE(10,101,NAM(L)) NAMPAR(L2),NAIND2	SOLV	484
	IPAR(L)=IND2	SOLV	485
	19 CONTINUE	SOLV	486
525		SOLV	487
	20 L1=L	SOLV	488
		SOLV	489
	DO 21 L=L0,L1	SOLV	490
	U(L)=FDAT(MPAR(L),IPAR(L))	SOLV	491
530	IF ((IVSAV.NE.0).AND.(IVSAV.EQ.IVSAV0).AND.(M.EQ.M0)) GO TO 22	SOLV	492
	U0(L)=U(L)	SOLV	493
	22 U(L)=U0(L)	SOLV	494
	AA(L)=AAJ	SOLV	495
	BB(L)=BBJ	SOLV	496
535	WERR(L)=WRJ	SOLV	497
	LVAR(L)=.FALSE.	SOLV	498
	WTVAR(L)=0.	SOLV	499
	IF(.NOT.JWT) GO TO 21	SOLV	500
	LVAR(L)=.TRUE.	SOLV	501
540	WTVAR(L)=WTVARJ	SOLV	502
	VAL(L)=VALJ	SOLV	503
	NVARWT=NVARWT+1	SOLV	504
	21 CONTINUE	SOLV	505
	C	SOLV	506
545	L = L1	SOLV	507
		SOLV	508
	L = L1	SOLV	509
	1 CONTINUE	SOLV	510
		SOLV	511
550	NCONTOT=NCONSTR+NVARWT	SOLV	512
	FNORM=1./FLOAT(NCONTOT)	SOLV	513
	M0=M	SOLV	514
	NPAR=L	SOLV	515
	KBT=L	SOLV	516
555	KBTS=KBT	SOLV	517
	CALL DASH	SOLV	518
	WRITE (3,102)	SOLV	519
	WRITE (3,105) LIM1,(I,NAM(I),U(I),I=1,NPAR)	SOLV	520
	CALL FCN(NPAR,G,AMIN,U,1)	SOLV	521
560	CALL DASH	SOLV	522
		SOLV	523
	CALL MINNEW	SOLV	524
		SOLV	525
	WRITE (3,105) LIM2,(I,NAM(I),U(I),I=1,NPAR)	SOLV	526
565	CALL FCN(NPAR,G,F,U,3)	SOLV	527
	MINFLG=1	SOLV	528
	CALL DASH	SOLV	529
	C RESTORE DEFAULTS IF SAVED	SOLV	530
	MIFLG = .FALSE.	SOLV	531
570	NOPR = .FALSE.	SOLV	532

	IF (SMINCAL) RETURN	SOLV	533
	DWORD7(1,1) = ISV1	SOLV	534
	DWORD7(1,2) = ISV2	SOLV	535
	DWORD7(2,2) = ISV3	SOLV	536
575	RETURN	SOLV	537
102	FORMAT (10X,*— SOLV — BETA-FUNCTION FITTING *)	SOLV	538
105	FORMAT (1H0,5X,A7,* VALUES OF VARIABLES*/	SOLV	539
	1 (7X,I5,3X,A5,5X,E15.8))	SOLV	540
		SOLV	541
580	END	SOLV	542

1	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
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
10	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
15	* 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
20		CONTRL	5
	COMMON/COPY/CPYSW	COPY	2
	LOGICAL CPYSW	COPY	3
		COPY	4
		SPLOAD	14
25	INTEGER HPERIOD,HREM	SPLOAD	15
	DATA IB,HPERIOD,HREM/1H,1H.,3HREM/	SPLOAD	16
	DATA MASK1,MASK2/0077777777777777777B,5500000000000000000B/	SPLOAD	17
		SPLOAD	18
	INS = 1H	SPLOAD	19
30	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
35	* 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
40	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
45	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
50	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
55	3 FORMAT (4X,A4,16A5)	SPLOAD	45
		SPLOAD	46
	END	SPLOAD	47

1	SUBROUTINE SRUN(IENTRY)	SRUN	2
		BLANK	2
	LEVEL 2, STORE, INFF, IWORK	BLANK	3
	COMMON STORE(48000), IWORK(10)	86MARSIZ	1
5	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
10	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
15	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, MSYMP	DIM	2
		DIM	3
	COMMON/FLTN/IFL(15)	FLTN	2
20		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/SWCH/BEND, MSIZE, CYCSWT, VCSW, PV, MSSW, VPR, MHS, MODS, MINZER	SWTCH	2
25	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
30	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
35	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
40	COMMON/SVNAM/NAMRUN	SVNAM	2
	DATA IBLNK/1H /, MXPMPY, R7/1,5/, (INAM(I), I=1,2)/3H(1),3H(S)/	SRUN	16
	DATA IBLNK /1H /	SRUN	17
	DATA MXPMPY, R7/1,5/	SRUN	18
	C	SRUN	19
45	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
50	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
55	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	SRUN	33
25	STORE(I) = 0.	SRUN	34
60	DO 26 I=1,MAX	SRUN	35
	DO 26 J=1,24	SRUN	36
26	INFF(J,I)=0	SRUN	37
	EMPTY=.FALSE.	SRUN	38
	INDEF=.FALSE.	SRUN	39
65	XEQ=.TRUE.	SRUN	40
	STOR=.TRUE.	SRUN	41
	RSRV=.TRUE.	SRUN	42
	ERROR=.FALSE.	SRUN	43
	CYCSWT=.FALSE.	SRUN	44
70	IERR=.FALSE.	SRUN	45
	PV = .FALSE.	SRUN	46
	MSSW = .FALSE.	SRUN	47
	VPR = .FALSE.	SRUN	48
	MHS=.FALSE.	SRUN	49
75	MODS=.FALSE.	SRUN	50
	ERPR=.TRUE.	SRUN	51
	IVSAV=0	SRUN	52
	MODE=1	SRUN	53
	NF=0	SRUN	54
80	NB=0	SRUN	55
	NI=0	SRUN	56
	NAMRUN=NAME	SRUN	57
	BEND=1.	SRUN	58
	LFILE=1	SRUN	59
85	MS=MEND	SRUN	60
	DO 3 I=1,ISAV	SRUN	61
	CALL RESRV(MS,MXPMY,MINZER,	SRUN	62
	0,0,0,0,0,0,0,0	SRUN	63
	.)	SRUN	64
90	INFF(1,MS) = IBLNK	SRUN	65
	INFF(2,MS) = IBLNK	SRUN	66
	MS=MS-1	SRUN	67
3	CONTINUE	SRUN	68
	MSIZE = 37	SRUN	69
95	I7 = ISAV7 - 2	SRUN	70
	IF (ISAV7.EQ.0) GO TO 7	SRUN	71
	DO 10 I=1,I7	SRUN	72
	CALL RESRV(MS,R7,MINZER,	SRUN	73
	0,0,0,0,0,0,0,0	SRUN	74
100	.)	SRUN	75
	INFF (1,MS) = IBLNK	SRUN	76
	INFF(2,MS) = IBLNK	SRUN	77
	MS = MS - 1	SRUN	78
10	CONTINUE	SRUN	79
105	C SET UP INFO AND STORAGE FOR UNIT AND TRANSPOSE MATRICES	SRUN	80
7	CONTINUE	SRUN	81
	MT = MS	SRUN	82
	MUNIT = MS	SRUN	83
	DO 15 I=1,2	SRUN	84
110	INFF(1,MS) = IBLNK	SRUN	85
	INFF(2,MS) = INAM(I)	SRUN	86
	CALL RESRV(MS,R7,MINZER,	SRUN	87
	0,0,0,0,0,0,0,0	SRUN	88
	.)	SRUN	89

115	15	MS = MS - 1	SRUN	90
		MSYMP = MS	SRUN	91
	C	DEFINE AND STORE MATRICES	SRUN	92
		CALL SETMAT(MT)	SRUN	93
	*	RESTORE INFO FOR RUN	SRUN	94
120		CALL WRTINF(OPNAME,MS)	SRUN	95
		RETURN	SRUN	96
	C	END OF RUN	SRUN	97
	5	CONTINUE	SRUN	98
		EMPTY=.FALSE.	SRUN	99
125		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
130	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
135	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.

1

SUBROUTINE SSQMIN(M,N,F,X,E,ESCALE,IPRINT,MAXFUN)
RETURN
END

SSQMIN 2
SSQMIN 3
SSQMIN 4

1	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
5	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
10	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
15	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

1	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
5	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
10	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
15	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
20	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
25	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
30	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
35	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
40	END	MY3TKPL	141

```

1      C *****TRKB***** 86TRKB 1
      C BETATRON FUNCTION TRACKING ROUTINE 9-19-85 86TRKB 2
      C 86TRKB 3
          SUBROUTINE TRKB(M) 86TRKB 4
5      LEVEL 2, STORE, INFF, IWORK 86TRKB 2
          COMMON STORE(48000), IWORK(10) 86MARSIZ 3
          DIMENSION INFF(24,2000) 86MARSIZ 2
          EQUIVALENCE (INFF,STORE) 86MARSIZ 3
10     COMMON/BMI1/MI1(16000) 86TRKB 5
          COMMON/NELS1/NELS1 86TRKB 1
          LEVEL 2,MI1 86TRKB 2
          BMI1L 86TRKB 3
          BMI1L 86TRKB 4
15     C BMI1 86TRKB 3
          DIMENSION MI(1) 86TRKB 5
          EQUIVALENCE (MI,MI1), (NELS,NELS1) 86TRKB 6
          COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF, 86TRKB 2
          1 LDFLG,FIN 86TRKB 3
20     LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN 86TRKB 4
          86TRKB 5
          TRKB 86TRKB 8
          COMMON /MATCH3/NOPR,NONU,FLAG,IPR,IPP,MINFLG,MIFLG,MATFLG,GLOBAL 86TRKB 2
          LOGICAL FLAG,NOPR,NONU,MIFLG,MATFLG,GLOBAL 86TRKB 3
25     COMMON/CDERIV/IFLAG,MR,NR,DP,NV(11),DAT(20),UZ,STOT, 86TRKB 2
          1 DAT2(7),MP2FLG,NF,H0,LOCALPH,NUMALPH 86TRKB 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
30     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
35     LOGICAL RADFLG 86TRKB 13
          DATA DZERO/0.0/ 86TRKB 14
          DATA MP2FLG/0/ 86TRKB 15
      C 86TRKB 16
          DATA IBLANK/4H / 86TRKB 17
40     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
45     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
50     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
55     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.	86TRKB	36
	C	BET0 = /NAME/ —INITIAL BETAS FROM ARRAY/NAME/, FOR AN IBET,	86TRKB	37
60	C	AND FROM BETA-FUNCTIONS OF /NAME/, IF A MATRIX.	86TRKB	38
	C	IRAD = 1 — OUTPUT PHASE ADVANCES IN RADIANS	86TRKB	39
	C	= 0 — IN UNITS OF 2PI	86TRKB	40
	C		86TRKB	41
	C	DEFAULTS —	86TRKB	42
65	C		86TRKB	43
	C	IF KA = KB = 0 (OR BLANK), A TABLE IS RESERVED SUFFICIENT TO STORE	86TRKB	44
	C	THE COMPLETE BEAM LINE.	86TRKB	45
	C	IF KA = -1 NO TABLE IS RESERVED, NO BETAS ARE STORE	86TRKB	46
	C		86TRKB	47
70	C	IF KT1 = IT2 = 0 (OR BLANK), TRACKING IS DONE THROUGH COMPLETE BEAM	86TRKB	48
	C		86TRKB	49
			86TRKB	50
		RADFLG = .FALSE.	86TRKB	51
		IF (MINFLG.NE.4) GO TO 23	86TRKB	52
75		IF (MIFLG) GO TO 152	86TRKB	53
		GO TO 4	86TRKB	54
	23	NOSTOR = .FALSE.	86TRKB	55
		TPI=4.*ACOS(DZERO)	86TRKB	56
		ELNAMEI=IBLANK	86TRKB	57
80		KB=INFF(5,M)	86TRKB	58
		KA=INFF(4,M)	86TRKB	59
		LQ3 = INFF(24,M)	86TRKB	60
		NCOL = IDAT(M,5)	86TRKB	61
		IF (KA.GT.-1) GO TO 151	86TRKB	62
85		NOSTOR = .TRUE.	86TRKB	63
		GO TO 152	86TRKB	64
	151	LTAB0 = LQ3 - NCOL*KA - 1	86TRKB	65
	152	MBML = MDAT(M,1)	86TRKB	66
		NDAT=INFF(17,MBML)	86TRKB	67
90		CALL MIFILL(MBML,1,NDAT,NELS,MI)	86TRKB	68
		CALL MMM(M,NELS,MI)	86TRKB	69
		IF (MINFLG.EQ.4) GO TO 4	86TRKB	70
		IP1=IDAT(M,1)	86TRKB	71
		IP2=IDAT(M,2)	86TRKB	72
95		ICALL=IDAT(M,6)	86TRKB	73
		IRAD = IDAT(M,4)	86TRKB	74
		SI = FDAT(M,1)	86TRKB	75
		TI = FDAT(M,2)	86TRKB	76
	C	SET UP CONSTANTS FOR CALCULATING PSI IN DEGREES OR RADIANS	86TRKB	77
100		IF (IRAD.EQ.1) RADFLG=.TRUE.	86TRKB	78
		DNOM = TPI	86TRKB	79
		CCON = 1.	86TRKB	80
		IF (.NOT.RADFLG) GO TO 153	86TRKB	81
		DNOM = 1.	86TRKB	82
105		CCON = TPI	86TRKB	83
	153	CONTINUE	86TRKB	84
	C	GET VECTOR	86TRKB	85
		NAMVEC = BDAT(M,3)	86TRKB	86
		VECFLAG = .FALSE.	86TRKB	87
110		IF(NAMVEC.EQ.IBLANK) GO TO 160	86TRKB	88
		VECFLAG = .TRUE.	86TRKB	89
		MVEC = MDAT(M,3)	86TRKB	90
		CALL DATA(MVEC,1,1,6,VI)	86TRKB	91
	160	CONTINUE	86TRKB	92

115			86TRKB	93
		NAMBET=BDAT(M,2)	86TRKB	94
		NAME=INFF(2,M)	86TRKB	95
			86TRKB	96
		ITABLE = IDAT(M,3)	86TRKB	97
120		IF(ITABLE.EQ.1.AND..NOT.NOSTOR) GO TO 10	86TRKB	98
		GO TO 12	86TRKB	99
			86TRKB	100
	C 10	IF ((ICALL.EQ.0).OR.(IP1.EQ.0)) GO TO 12	86TRKB	101
	10	CONTINUE	86TRKB	102
125	C	INITIAL BETAS FROM BETA FUNCTION TABLE OF TRKB INSTRUCTION,	86TRKB	103
	C	EXCEPT THAT FOR FIRST CALL, OR IF IP1=0, THEY ARE TAKEN FROM IBET OR	86TRKB	104
		LOCBET=NCOL*(IP1-KA)+1	86TRKB	105
		MBET=M	86TRKB	106
		NTYP=8	86TRKB	107
130		CALL DATA(MBET,NTYP,LOCBET+12,1,SI)	86TRKB	108
		CALL DATA(MBET,NTYP,LOCBET+13,1,TI)	86TRKB	109
		IF(VECFLAG) CALL DATA(MBET,NTYP,LOCBET+14,6,VI)	86TRKB	110
		GO TO 13	86TRKB	111
			86TRKB	112
135	C	INITIAL BETAS FROM ARRAY SPECIFIED BY NAMBET	86TRKB	113
	12	LOCBET=1	86TRKB	114
		NTYP=1	86TRKB	115
		MBET=MDAT(M,2)	86TRKB	116
			86TRKB	117
140		NAMOP=INFF(1,MBET)	86TRKB	118
			86TRKB	119
		IF(NAMOP.EQ.4HIBET) GO TO 13	86TRKB	120
	C	INITIAL BETAS ARE BETAS OF MATRICES MBET.	86TRKB	121
		CALL BET(MBET,BXI,BYI,1)	86TRKB	122
145		BXI(1)=0.	86TRKB	123
		BYI(1)=0.	86TRKB	124
			86TRKB	125
		GO TO 16	86TRKB	126
			86TRKB	127
150	13	CALL DATA(MBET,NTYP,LOCBET ,6,BXI)	86TRKB	128
		CALL DATA(MBET,NTYP,LOCBET+6,6,BYI)	86TRKB	129
	C		86TRKB	130
	16	CONTINUE	86TRKB	131
			86TRKB	132
155			86TRKB	133
	26	IF (IP1.EQ.0) GO TO 9	86TRKB	134
		NELI = MI(IP1)	86TRKB	135
		ELNAMEI=MNAME(NELI)	86TRKB	136
	9	IF(IP1.LT.KA) NOSTOR=.TRUE.	86TRKB	137
160		IF(IP2.GT.KB) NOSTOR=.TRUE.	86TRKB	138
		ICALL=1	86TRKB	139
		CALL STDAT(M,3,3,1,ICALL)	86TRKB	140
			86TRKB	141
			86TRKB	142
165	4	CONTINUE	86TRKB	143
			86TRKB	144
		S = SI	86TRKB	145
		TH = TI	86TRKB	146
			86TRKB	147
170		DO 3 I=1,6	86TRKB	148
		BX(I) = BXI(I)	86TRKB	149

	BY(I) = BYI(I)	86TRKB	150
	V(I) = VI(I)	86TRKB	151
175	3 CONTINUE	86TRKB	152
	IF (NOPR) GO TO 7	86TRKB	153
	CALL DASH	86TRKB	154
	LNUM = 0	86TRKB	155
	IF (I2.GT.45) WRITE (3,1004)	86TRKB	156
180	IF(.NOT.VECFLAG) GO TO 406	86TRKB	157
	WRITE (3,1002)	86TRKB	158
	GO TO 407	86TRKB	159
	406 WRITE(3,1000)	86TRKB	160
	407 CONTINUE	86TRKB	161
185		86TRKB	162
	ELNAME=ELNAMEI	86TRKB	163
	7 CONTINUE	86TRKB	164
	I1 = IP1 + 1	86TRKB	165
	I2 = IP2	86TRKB	166
190		86TRKB	167
	IF (BX(4).EQ.0.) BX(4)=(1.+BX(3)*BX(3))/BX(2)	86TRKB	168
	IF (BY(4).EQ.0.) BY(4)=(1.+BY(3)*BY(3))/BY(2)	86TRKB	169
		86TRKB	170
		86TRKB	171
	C LOOP OVER POSITIONS	86TRKB	172
195	19 DO 1 IBS = I1,I2	86TRKB	173
	IPOS = IBS - 1	86TRKB	174
	IF(NOPR) GO TO 6	86TRKB	175
	IF(.NOT.VECFLAG) GO TO 190	86TRKB	176
	IF (MOD(LNUM,45).NE.0.OR.LNUM.EQ.0) GO TO 191	86TRKB	177
200	WRITE (3,1004)	86TRKB	178
	WRITE (3,1002)	86TRKB	179
	LNUM = 0	86TRKB	180
	191 IF (MOD(LNUM,5).EQ.0.AND.LNUM.NE.0) WRITE (3,1010)	86TRKB	181
	WRITE(3,1003) IPOS,ELNAME,S,(BX(I),I=1,3),BX(5),BX(6),	86TRKB	182
205	1 (BY(I),I=1,3),BY(5),BY(6),(V(I),I=1,4)	86TRKB	183
	LNUM = LNUM + 1	86TRKB	184
	GO TO 6	86TRKB	185
	190 IF (MOD(LNUM,45).NE.0.OR.LNUM.EQ.0) GO TO 192	86TRKB	186
	WRITE (3,1004)	86TRKB	187
210	WRITE (3,1000)	86TRKB	188
	LNUM = 0	86TRKB	189
	192 IF (MOD(LNUM,5).EQ.0.AND.LNUM.NE.0) WRITE (3,1010)	86TRKB	190
	WRITE(3,1001) IPOS,ELNAME,S,(BX(I),I=1,3),BX(5),BX(6),	86TRKB	191
	1 (BY(I),I=1,3),BY(5),BY(6)	86TRKB	192
215	LNUM = LNUM + 1	86TRKB	193
	6 NEL = MI(IBS)	86TRKB	194
	IF(NOSTOR) GO TO 8	86TRKB	195
	J=LTAB0+NCOL*IPOS	86TRKB	196
	DO 11 I=1,14	86TRKB	197
220	11 STORE(J+I)=BETF(I)	86TRKB	198
	IF(.NOT.VECFLAG) GO TO 8	86TRKB	199
	DO 110 I=15,20	86TRKB	200
	110 STORE(J+I) = VF(I)	86TRKB	201
	8 CONTINUE	86TRKB	202
225		86TRKB	203
	ELNAME=MNAME(NEL)	86TRKB	204
	STOT = S	86TRKB	205
	C	86TRKB	206

	18	CALL RXY(NEL,RX,RY,RW)	86TRKB	207
230		IF(.NOT.VECFLAG) GO TO 180	86TRKB	208
		DO 181 I=1,6	86TRKB	209
		BX0(I)=BX(I)	86TRKB	210
		BY0(I)=BY(I)	86TRKB	211
235	181	V0(I) = V(I)	86TRKB	212
		GO TO 182	86TRKB	213
			86TRKB	214
			86TRKB	215
	180	DO 2 I=1,6	86TRKB	216
		BX0(I)=BX(I)	86TRKB	217
240	2	BY0(I)=BY(I)	86TRKB	218
			86TRKB	219
	182	S = S + RW(1)	86TRKB	220
		TH = TH + RW(2)	86TRKB	221
			86TRKB	222
245		BX(2)=RX(1,1)*RX(1,1)*BX0(2)-2.*RX(1,1)*RX(1,2)*BX0(3)	86TRKB	223
	1	+ RX(1,2)*RX(1,2)*BX0(4)	86TRKB	224
			86TRKB	225
		BY(2)=RY(1,1)*RY(1,1)*BY0(2)-2.*RY(1,1)*RY(1,2)*BY0(3)	86TRKB	226
	1	+ RY(1,2)*RY(1,2)*BY0(4)	86TRKB	227
250			86TRKB	228
		BX(3) = -RX(1,1)*RX(2,1)*BX0(2)	86TRKB	229
	1	+ (RX(1,1)*RX(2,2)+RX(1,2)*RX(2,1))*BX0(3)	86TRKB	230
	2	- RX(1,2)*RX(2,2)*BX0(4)	86TRKB	231
			86TRKB	232
255		BY(3) = -RY(1,1)*RY(2,1)*BY0(2)	86TRKB	233
	1	+ (RY(1,1)*RY(2,2)+RY(1,2)*RY(2,1))*BY0(3)	86TRKB	234
	2	- RY(1,2)*RY(2,2)*BY0(4)	86TRKB	235
			86TRKB	236
		BX(4) = RX(2,1)*RX(2,1)*BX0(2) - 2.*RX(2,1)*RX(2,2)*BX0(3)	86TRKB	237
260	1	+ RX(2,2)*RX(2,2)*BX0(4)	86TRKB	238
			86TRKB	239
		BY(4) = RY(2,1)*RY(2,1)*BY0(2) - 2.*RY(2,1)*RY(2,2)*BY0(3)	86TRKB	240
	1	+ RY(2,2)*RY(2,2)*BY0(4)	86TRKB	241
			86TRKB	242
265		BX(5) = RX(1,1)*BX0(5) + RX(1,2)*BX0(6) + RX(1,3)	86TRKB	243
			86TRKB	244
		BY(5) = RY(1,1)*BY0(5) + RY(1,2)*BY0(6) + RY(1,3)	86TRKB	245
			86TRKB	246
		BX(6) = RX(2,1)*BX0(5) + RX(2,2)*BX0(6) + RX(2,3)	86TRKB	247
270			86TRKB	248
		BY(6) = RY(2,1)*BY0(5) + RY(2,2)*BY0(6) + RY(2,3)	86TRKB	249
			86TRKB	250
		IF(NONU) GO TO 183	86TRKB	251
			86TRKB	252
275		BETAV=SQRT(BX(2)*BX0(2))	86TRKB	253
		BERAT=SQRT(BX0(2)/BX(2))	86TRKB	254
		SINE=RX(1,2)/BETAV	86TRKB	255
		COSINE=RX(1,1)*BERAT-BX0(3)*SINE	86TRKB	256
	C	DNOM OR CCON SET TO TPI OR 1. DEPENDING ON DEGREES OR RADIAN.	86TRKB	257
280		DNU = ATAN2(SINE,COSINE)/DNOM	86TRKB	258
		IF (DNU.LT.0) DNU = DNU + CCON	86TRKB	259
		BX(1) = BX0(1) + DNU	86TRKB	260
		BETAV=SQRT(BY(2)*BY0(2))	86TRKB	261
		BERAT=SQRT(BY0(2)/BY(2))	86TRKB	262
285		SINE=RY(1,2)/BETAV	86TRKB	263

		COSINE=RY(1,1)*BERAT-BY0(3)*SINE	86TRKB	264
		DNU = ATAN2(SINE,COSINE)/DNOM	86TRKB	265
		IF (DNU.LT.0) DNU = DNU + CCON	86TRKB	266
		BY(1) = BY0(1) + DNU	86TRKB	267
290			86TRKB	268
	183	CONTINUE	86TRKB	269
			86TRKB	270
		IF(.NOT.VECFLAG) GO TO 1	86TRKB	271
		CALL RTRV7(NEL,T,RW)	86TRKB	272
295		CALL MXV7(T,V0,V)	86TRKB	273
			86TRKB	274
	1	CONTINUE	86TRKB	275
			86TRKB	276
	17	IPOS = I2	86TRKB	277
300		MP2FLG = 0	86TRKB	278
		IF(NOSTOR) GO TO 5	86TRKB	279
			86TRKB	280
		J = J ₂ + NCOL	86TRKB	281
		DO 14 I=1,14	86TRKB	282
305	14	STORE(J+I)=BETF(I)	86TRKB	283
		IF(.NOT.VECFLAG) GO TO 5	86TRKB	284
		DO 140 I=15,20	86TRKB	285
	140	STORE(J+I) = VF(I)	86TRKB	286
			86TRKB	287
310	5	CONTINUE	86TRKB	288
			86TRKB	289
		IF(NOPR) RETURN	86TRKB	290
			86TRKB	291
		IF(.NOT.VECFLAG) GO TO 150	86TRKB	292
315			86TRKB	293
		WRITE(3,1003) IPOS,ELNAME,S,(BX(I),I=1,3),BX(5),BX(6),	86TRKB	294
	1	(BY(I),I=1,3),BY(5),BY(6),(V(I),I=1,4)	86TRKB	295
		GO TO 154	86TRKB	296
	150	WRITE(3,1001) IPOS,ELNAME,S,(BX(I),I=1,3),BX(5),BX(6),	86TRKB	297
320	1	(BY(I),I=1,3),BY(5),BY(6)	86TRKB	298
			86TRKB	299
	154	CALL DASH	86TRKB	300
			86TRKB	301
		RETURN	86TRKB	302
325			86TRKB	303
	1000	FORMAT (" POS",11X,"S",7X,"QX",10X,"BX",8X,"AX",8X,"X",9X,	86TRKB	304
	1	"DX",14X,"QY",10X,"BY",8X,"AY",8X,"Y",9X,"DY")	86TRKB	305
			86TRKB	306
	1001	FORMAT (I4,1X,A4,1X,F10.4,2(F10.6,F11.4,F11.6,F10.6,F10.6,6X))	86TRKB	307
330			86TRKB	308
	1002	FORMAT (" POS",10X,"S",7X,"QX",7X,"BX",7X,"AX",6X,"DX",6X,	86TRKB	309
	1	"DDX",6X,"QY",7X,"BY",7X,"AY",6X,"DY",6X,"DDY",4X,"X(MM)",	86TRKB	310
	2	1X,"DX(MR)",1X,"Y(MM)",1X,"DY(MR)")	86TRKB	311
			86TRKB	312
335	1003	FORMAT (I4,1X,A4,F10.4,2(F8.4,2F9.4,2F8.4,1X),2(F6.2,F7.3))	86TRKB	313
	1004	FORMAT (1H1)	86TRKB	314
	1010	FORMAT (1H)	86TRKB	315
		END	86TRKB	316

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

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

```

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

```

	C	ITBL = 1 — INPUT BETAS FROM TRKB TABLE, AT IT1, EXCEPT	86TRKE	45
	C	ON 1ST CALL, AND/OR IF IT1=0, FROM IBET INPUT.	86TRKE	46
60	C	BET0 = /NAME/ —INITIAL BETAS FROM ARRAY/NAME/,FOR AN IBET,	86TRKE	47
	C	AND FROM BETA-FUNCTIONS OF /NAME/, IF A MATRIX.	86TRKE	48
	C	IADD = 0 — ADD XB AND XP IN QUADRATURE, ETC.	86TRKE	49
	C	IADD = 1 — ADD XB AND XP ALGEBRAICALLY, ETC.	86TRKE	50
	C		86TRKE	51
65	C	DEFAULTS —	86TRKE	52
	C		86TRKE	53
	C	IF KA = KB = 0 (OR BLANK), A TABLE IS RESERVED SUFFICIENT TO STORE	86TRKE	54
	C	THE COMPLETE BEAM LINE.	86TRKE	55
	C	IF KA = -1	86TRKE	56
70	C	NO TABLE IS RESERVED, NO BETAS ARE STORE	86TRKE	57
	C		86TRKE	58
	C	IF KT1 = IT2 = 0 (OR BLANK), TRACKING IS DONE THROUGH COMPLETE BEAM	86TRKE	59
	C		86TRKE	60
		RADFLG = .FALSE.	86TRKE	61
75		IF (MINFLG.NE.4) GO TO 23	86TRKE	62
		IF (MIFLG) GO TO 152	86TRKE	63
		GO TO 4	86TRKE	64
	23	NOSTOR = .FALSE.	86TRKE	65
		TPI=4.*ACOS(DZERO)	86TRKE	66
80		ELNAMEI=IBLANK	86TRKE	67
		KB=INFF(5,M)	86TRKE	68
		KA=INFF(4,M)	86TRKE	69
		LQ3 = INFF(24,M)	86TRKE	70
		NCOL = IDAT(M,5)	86TRKE	71
85		IF (KA.GT.-1) GO TO 151	86TRKE	72
		NOSTOR = .TRUE.	86TRKE	73
		GO TO 152	86TRKE	74
	151	LTAB0 = LQ3 - NCOL*KA - 1	86TRKE	75
	152	MBML = MDAT(M,1)	86TRKE	76
90			86TRKE	77
		NDAT=INFF(17,MBML)	86TRKE	78
		CALL MIFILL(MBML,1,NDAT,NELS,MI)	86TRKE	79
		CALL MMM(M,NELS,MI)	86TRKE	80
		IF (MINFLG.EQ.4) GO TO 4	86TRKE	81
95		IP1=IDAT(M,1)	86TRKE	82
		IP2=IDAT(M,2)	86TRKE	83
		ICALL=IDAT(M,6)	86TRKE	84
		IADD = IDAT(M,4)	86TRKE	85
		SI = 0.0	86TRKE	86
100		TI = 0.0	86TRKE	87
		IG = 1	86TRKE	88
		IF(IADD.EQ.0) IG = 2	86TRKE	89
		DNOM = TPI	86TRKE	90
		CCON = 1.	86TRKE	91
105			86TRKE	92
	C	GET VECTOR	86TRKE	93
		NAMVEC = BDAT(M,3)	86TRKE	94
		VECFLAG = .FALSE.	86TRKE	95
		IF(NAMVEC.EQ.IBLANK) GO TO 160	86TRKE	96
110		VECFLAG = .TRUE.	86TRKE	97
		MVEC = MDAT(M,3)	86TRKE	98
		CALL DATA(MVEC,1,1,6,VI)	86TRKE	99
	160	CONTINUE	86TRKE	100
			86TRKE	101

115	MBVAL = MDAT(M,4)	86TRKE	102
	CALL DATA(MBVAL,5,1,4,P)	86TRKE	103
	NAMBET=BDAT(M,2)	86TRKE	104
	NAME=INFF(2,M)	86TRKE	105
		86TRKE	106
120	ITABLE = IDAT(M,3)	86TRKE	107
	IF(ITABLE.EQ.1.AND..NOT.NOSTOR) GO TO 10	86TRKE	108
	GO TO 12	86TRKE	109
		86TRKE	110
C 10	IF ((ICALL.EQ.0).OR.(IP1.EQ.0)) GO TO 12	86TRKE	111
125	10 CONTINUE	86TRKE	112
	C INITIAL BETAS FROM BETA FUNCTION TABLE OF TRKB INSTRUCTION,	86TRKE	113
	C EXCEPT THAT FOR FIRST CALL, OR IF IP1=0, THEY ARE TAKEN FROM IBET OR	86TRKE	114
	LOCBET=NCOL*(IP1-KA)+1	86TRKE	115
	MBET=M	86TRKE	116
130	NTYP=8	86TRKE	117
	CALL DATA(MBET,NTYP,LOCBET+12,1,SI)	86TRKE	118
	CALL DATA(MBET,NTYP,LOCBET+13,1,TI)	86TRKE	119
	IF(VECFLAG) CALL DATA(MBET,NTYP,LOCBET+14,6,VI)	86TRKE	120
	GO TO 13	86TRKE	121
		86TRKE	122
135	C INITIAL BETAS FROM ARRAY SPECIFIED BY NAMBET	86TRKE	123
	12 LOCBET=1	86TRKE	124
	NTYP=1	86TRKE	125
	MBET=MDAT(M,2)	86TRKE	126
140		86TRKE	127
	NAMOP=INFF(1,MBET)	86TRKE	128
		86TRKE	129
	IF(NAMOP.EQ.4HIBET) GO TO 13	86TRKE	130
	C INITIAL BETAS ARE BETAS OF MATRICES MBET.	86TRKE	131
145	CALL BET(MBET,BXI,BYI,1)	86TRKE	132
	BXI(1)=0.	86TRKE	133
	BYI(1)=0.	86TRKE	134
		86TRKE	135
	GO TO 16	86TRKE	136
		86TRKE	137
150	13 CALL DATA(MBET,NTYP,LOCBET ,6,BXI)	86TRKE	138
	CALL DATA(MBET,NTYP,LOCBET+6,6,BYI)	86TRKE	139
	C	86TRKE	140
	16 CONTINUE	86TRKE	141
155		86TRKE	142
		86TRKE	143
	C GET EMITTANCES	86TRKE	144
	EPXCO = FDAT(M,1)	86TRKE	145
	EPYCO = FDAT(M,2)	86TRKE	146
160	RATX = SQRT(EPXCO/EPX)	86TRKE	147
	RATY = SQRT(EPYCO/EPY)	86TRKE	148
		86TRKE	149
	26 IF (IP1.EQ.0) GO TO 9	86TRKE	150
	NELI = MI(IP1)	86TRKE	151
165	ELNAMEI=MNAME(NELI)	86TRKE	152
	9 IF(IP1.LT.KA) NOSTOR=.TRUE.	86TRKE	153
	IF(IP2.GT.KB) NOSTOR=.TRUE.	86TRKE	154
	ICALL=1	86TRKE	155
	CALL STDAT(M,3,3,1,ICALL)	86TRKE	156
170		86TRKE	157
		86TRKE	158

	4	CONTINUE	86TRKE	159
		S = SI	86TRKE	160
175		TH = TI	86TRKE	161
			86TRKE	162
			86TRKE	163
		DO 3 I=1,6	86TRKE	164
		BX(I) = BXI(I)	86TRKE	165
		BY(I) = BYI(I)	86TRKE	166
180		V(I) = VI(I)	86TRKE	167
	3	CONTINUE	86TRKE	168
			86TRKE	169
		I1 = IP1 + 1	86TRKE	170
		I2 = IP2	86TRKE	171
185		IF (NOPR) GO TO 7	86TRKE	172
		LNUM = 0	86TRKE	173
		IF (I2.GT.40) WRITE (3,4004)	86TRKE	174
		CALL DASH	86TRKE	175
		WRITE(3,4001) P(1),EPXCO,P(3),P(2),EPYCO,P(4)	86TRKE	176
190		GO TO (404,405) IG	86TRKE	177
	404	WRITE (3,4006)	86TRKE	178
		GO TO 406	86TRKE	179
	405	WRITE (3,4008)	86TRKE	180
		IF(.NOT.VECFLAG) GO TO 406	86TRKE	181
195		WRITE (3,4011)	86TRKE	182
		GO TO 407	86TRKE	183
	406	WRITE(3,4010)	86TRKE	184
	407	CONTINUE	86TRKE	185
			86TRKE	186
200		ELNAME=ELNAMEI	86TRKE	187
	7	CONTINUE	86TRKE	188
			86TRKE	189
		IF (BX(4).EQ.0.) BX(4)=(1.+BX(3)*BX(3))/BX(2)	86TRKE	190
		IF (BY(4).EQ.0.) BY(4)=(1.+BY(3)*BY(3))/BY(2)	86TRKE	191
205			86TRKE	192
	C	COMPUTE AND PRINT BEAM ENVELOPES	86TRKE	193
		X=SQRT(EPX*BX(2))	86TRKE	194
		Y=SQRT(EPY*BY(2))	86TRKE	195
		XP=BX(5)*DPP	86TRKE	196
210		YP=BY(5)*DPP	86TRKE	197
		XCO=RATX*X	86TRKE	198
		YCO=RATY*Y	86TRKE	199
		XPR=SQRT(EPX*BX(4))	86TRKE	200
		YPR=SQRT(EPY*BY(4))	86TRKE	201
215		XPPR=BX(6)*DPP	86TRKE	202
		YPPR=BY(6)*DPP	86TRKE	203
		XCOPR=RATX*XPR	86TRKE	204
		YCOPR=RATY*YPR	86TRKE	205
		GO TO (4116,4117) IG	86TRKE	206
220	4116	XTOT = X + ABS(XP)	86TRKE	207
		XPRTOT = XPR + ABS(XPPR)	86TRKE	208
		YTOT = Y + ABS(YP)	86TRKE	209
		YPRTOT = YPR + ABS(YPPR)	86TRKE	210
		GO TO 4118	86TRKE	211
225	4117	CONTINUE	86TRKE	212
		XTOT = SQRT(X * X + XP * XP)	86TRKE	213
		XPRTOT = SQRT(XPR*XPR + XPPR*XPPR)	86TRKE	214
		YTOT = SQRT(Y*Y + YP*YP)	86TRKE	215

		YPRTOT = SQRT(YPR*YPR + YPPR*YPPR)	86TRKE	216
230	4118	CONTINUE	86TRKE	217
		XTOT=XTOT+XCO	86TRKE	218
		YTOT=YTOT+YCO	86TRKE	219
		XPRTOT=XPRTOT+XCOPR	86TRKE	220
		YPRTOT=YPRTOT+YCOPR	86TRKE	221
235			86TRKE	222
	C	LOOP OVER POSITIONS	86TRKE	223
	19	DO 1 IBS = I1,I2	86TRKE	224
		IPOS = IBS - 1	86TRKE	225
		IF(NOPR) GO TO 6	86TRKE	226
240		IF(.NOT.VECFLAG) GO TO 190	86TRKE	227
		IF (MOD(LNUM,40).NE.0.OR.LNUM.EQ.0) GO TO 191	86TRKE	228
		WRITE (3,4004)	86TRKE	229
		WRITE (3,4011)	86TRKE	230
		LNUM = 0	86TRKE	231
245	191	IF (MOD(LNUM,40).NE.0.OR.LNUM.NE.0) WRITE (3,4040)	86TRKE	232
		WRITE(3,4012) IPOS,ELNAME,S,X,XP,XCO,XTOT,Y,YP,YCO,YTOT,	86TRKE	233
	1	XPRTOT,YPRTOT,(V(I),I=1,4)	86TRKE	234
		LNUM = LNUM + 1	86TRKE	235
		GO TO 6	86TRKE	236
250	190	IF (MOD(LNUM,40).NE.0.OR.LNUM.EQ.0) GO TO 192	86TRKE	237
		WRITE (3,4004)	86TRKE	238
		WRITE (3,4010)	86TRKE	239
		LNUM = 0	86TRKE	240
255	192	IF (MOD(LNUM,5).EQ.0.AND.LNUM.NE.0) WRITE (3,4040)	86TRKE	241
		WRITE(3,4012) IPOS,ELNAME,S,X,XP,XCO,XTOT,Y,YP,YCO,YTOT,	86TRKE	242
	1	XPR,XPPR,XPRTOT,YPR,YPPR,YPRTOT	86TRKE	243
		LNUM = LNUM + 1	86TRKE	244
	6	NEL = MI(IBS)	86TRKE	245
		IF(NOSTOR) GO TO 8	86TRKE	246
260		J=LTAB0+NCOL*IPOS	86TRKE	247
		DO 11 I=1,14	86TRKE	248
	11	STORE(J+I)=BETF(I)	86TRKE	249
		IF(.NOT.VECFLAG) GO TO 8	86TRKE	250
		DO 110 I=15,20	86TRKE	251
265	110	STORE(J+I) = VF(I)	86TRKE	252
	8	CONTINUE	86TRKE	253
			86TRKE	254
		ELNAME=MNAME(NEL)	86TRKE	255
		STOT = S	86TRKE	256
270	C		86TRKE	257
	18	CALL RXY(NEL,RX,RY,RW)	86TRKE	258
			86TRKE	259
		IF(.NOT.VECFLAG) GO TO 180	86TRKE	260
		DO 181 I=1,6	86TRKE	261
275		BX0(I)=BX(I)	86TRKE	262
		BY0(I)=BY(I)	86TRKE	263
	181	V0(I) = V(I)	86TRKE	264
		GO TO 182	86TRKE	265
			86TRKE	266
280	180	DO 2 I=1,6	86TRKE	267
		BX0(I)=BX(I)	86TRKE	268
	2	BY0(I)=BY(I)	86TRKE	269
			86TRKE	270
	182	S = S + RW(1)	86TRKE	271
285		TH = TH + RW(2)	86TRKE	272

			86TRKE	273
		BX(2)=RX(1,1)*RX(1,1)*BX0(2)-2.*RX(1,1)*RX(1,2)*BX0(3)	86TRKE	274
	1	+ RX(1,2)*RX(1,2)*BX0(4)	86TRKE	275
290			86TRKE	276
		BY(2)=RY(1,1)*RY(1,1)*BY0(2)-2.*RY(1,1)*RY(1,2)*BY0(3)	86TRKE	277
	1	+ RY(1,2)*RY(1,2)*BY0(4)	86TRKE	278
			86TRKE	279
		BX(3) = -RX(1,1)*RX(2,1)*BX0(2)	86TRKE	280
	1	+ (RX(1,1)*RX(2,2)+RX(1,2)*RX(2,1))*BX0(3)	86TRKE	281
295	2	- RX(1,2)*RX(2,2)*BX0(4)	86TRKE	282
			86TRKE	283
		BY(3) = -RY(1,1)*RY(2,1)*BY0(2)	86TRKE	284
	1	+ (RY(1,1)*RY(2,2)+RY(1,2)*RY(2,1))*BY0(3)	86TRKE	285
	2	- RY(1,2)*RY(2,2)*BY0(4)	86TRKE	286
300			86TRKE	287
		BX(4) = RX(2,1)*RX(2,1)*BX0(2) - 2.*RX(2,1)*RX(2,2)*BX0(3)	86TRKE	288
	1	+ RX(2,2)*RX(2,2)*BX0(4)	86TRKE	289
			86TRKE	290
		BY(4) = RY(2,1)*RY(2,1)*BY0(2) - 2.*RY(2,1)*RY(2,2)*BY0(3)	86TRKE	291
305	1	+ RY(2,2)*RY(2,2)*BY0(4)	86TRKE	292
			86TRKE	293
		BX(5) = RX(1,1)*BX0(5) + RX(1,2)*BX0(6) + RX(1,3)	86TRKE	294
			86TRKE	295
		BY(5) = RY(1,1)*BY0(5) + RY(1,2)*BY0(6) + RY(1,3)	86TRKE	296
310			86TRKE	297
		BX(6) = RX(2,1)*BX0(5) + RX(2,2)*BX0(6) + RX(2,3)	86TRKE	298
			86TRKE	299
		BY(6) = RY(2,1)*BY0(5) + RY(2,2)*BY0(6) + RY(2,3)	86TRKE	300
			86TRKE	301
315		IF(NONU) GO TO 183	86TRKE	302
			86TRKE	303
		BETAV=SQRT(BX(2)*BX0(2))	86TRKE	304
		BERAT=SQRT(BX0(2)/BX(2))	86TRKE	305
		SINE=RX(1,2)/BETAV	86TRKE	306
320		COSINE=RX(1,1)*BERAT-BX0(3)*SINE	86TRKE	307
	C	DNOM OR CCON SET TO TPI OR 1. DEPENDING ON DEGREES OR RADIANS.	86TRKE	308
		DNU = ATAN2(SINE,COSINE)/DNOM	86TRKE	309
		IF (DNU.LT.0) DNU = DNU + CCON	86TRKE	310
		BX(1) = BX0(1) + DNU	86TRKE	311
325			86TRKE	312
		BETAV=SQRT(BY(2)*BY0(2))	86TRKE	313
		BERAT=SQRT(BY0(2)/BY(2))	86TRKE	314
		SINE=RY(1,2)/BETAV	86TRKE	315
		COSINE=RY(1,1)*BERAT-BY0(3)*SINE	86TRKE	316
		DNU = ATAN2(SINE,COSINE)/DNOM	86TRKE	317
330		IF (DNU.LT.0) DNU = DNU + CCON	86TRKE	318
		BY(1) = BY0(1) + DNU	86TRKE	319
			86TRKE	320
	183	CONTINUE	86TRKE	321
			86TRKE	322
335		IF(.NOT.VECFLAG) GO TO 184	86TRKE	323
		CALL RTRV7(NEL,T,RW)	86TRKE	324
		CALL MXV7(T,V0,V)	86TRKE	325
			86TRKE	326
	184	CONTINUE	86TRKE	327
340			86TRKE	328
	C	COMPUTE AND PRINT BEAM ENVELOPES	86TRKE	329
		X=SQRT(EPX*BX(2))	86TRKE	329

		Y=SQRT(EPY*BY(2))	86TRKE	330
		XP=BX(5)*DPP	86TRKE	331
345		YP=BY(5)*DPP	86TRKE	332
		XCO=RATX*X	86TRKE	333
		YCO=RATY*Y	86TRKE	334
		XPR=SQRT(EPX*BX(4))	86TRKE	335
		YPR=SQRT(EPY*BY(4))	86TRKE	336
350		XPPR=BX(6)*DPP	86TRKE	337
		YPPR=BY(6)*DPP	86TRKE	338
		XCOPR=RATX*XPR	86TRKE	339
		YCOPR=RATY*YPR	86TRKE	340
		GO TO (5116,5117) IG	86TRKE	341
355	5116	XTOT = X + ABS(XP)	86TRKE	342
		XPRTOT = XPR + ABS(XPPR)	86TRKE	343
		YTOT = Y + ABS(YP)	86TRKE	344
		YPRTOT = YPR + ABS(YPPR)	86TRKE	345
		GO TO 5118	86TRKE	346
360	5117	CONTINUE	86TRKE	347
		XTOT = SQRT(X * X + XP * XP)	86TRKE	348
		XPRTOT = SQRT(XPR*XPR + XPPR*XPPR)	86TRKE	349
		YTOT = SQRT(Y*Y + YP*YP)	86TRKE	350
		YPRTOT = SQRT(YPR*YPR + YPPR*YPPR)	86TRKE	351
365	5118	CONTINUE	86TRKE	352
		XTOT=XTOT+XCO	86TRKE	353
		YTOT=YTOT+YCO	86TRKE	354
		XPRTOT=XPRTOT+XCOPR	86TRKE	355
		YPRTOT=YPRTOT+YCOPR	86TRKE	356
370			86TRKE	357
			86TRKE	358
	1	CONTINUE	86TRKE	359
			86TRKE	360
	17	IPOS = I2	86TRKE	361
375		MP2FLG = 0	86TRKE	362
		IF(NOSTOR) GO TO 5	86TRKE	363
			86TRKE	364
		J = J + NCOL	86TRKE	365
		DO 14 I=1,14	86TRKE	366
380	14	STORE(J+I)=BETF(I)	86TRKE	367
		IF(.NOT.VECFLAG) GO TO 5	86TRKE	368
		DO 140 I=15,20	86TRKE	369
	140	STORE(J+I) = VF(I)	86TRKE	370
			86TRKE	371
385	5	CONTINUE	86TRKE	372
			86TRKE	373
		IF(NOPR) RETURN	86TRKE	374
			86TRKE	375
		IF(.NOT.VECFLAG) GO TO 150	86TRKE	376
390			86TRKE	377
		WRITE(3,4012) IPOS,ELNAME,S,X,XP,XCO,XTOT,Y,YP,YCO,YTOT,	86TRKE	378
	1	XPRTOT,YPRTOT,(V(I),I=1,4)	86TRKE	379
		GO TO 153	86TRKE	380
	150	WRITE(3,4012) IPOS,ELNAME,S,X,XP,XCO,XTOT,Y,YP,YCO,YTOT,	86TRKE	381
395	1	XPR,XPPR,XPRTOT,YPR,YPPR,YPRTOT	86TRKE	382
	153	CALL DASH	86TRKE	383
			86TRKE	384
		RETURN	86TRKE	385
			86TRKE	386

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

```

1      C *****TRKM***** 86TRKM 2
      C MAP-TYPE ELEMENT TRACKING ROUTINE 9-19-85 86TRKM 3
      C 86TRKM 4
          SUBROUTINE TRKM(M) 86TRKM 5
5      LEVEL 2, STORE,INFF,IWORK 86TRKM 2
          COMMON STORE(48000),IWORK(10) 86MARSIZ 1
          DIMENSION INFF(24,2000) 86MARSIZ 2
          EQUIVALENCE (INFF,STORE) 86MARSIZ 3
10     COMMON/BMI1/MI1(16000) 86TRKM 5
          COMMON/NELS1/NELS1 86TRKM 2
          LEVEL 2,MI1 86TRKM 3
15     C 86TRKM 3
          DIMENSION MI(1) 86TRKM 8
          EQUIVALENCE (MI,MI1),(NELS,NELS1) 86TRKM 9
          COMMON/CONTRL/ERROR,MODE,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF, 86TRKM 2
1      LDFLG,FIN 86TRKM 3
20     LOGICAL ERROR,RSRV,STOR,XEQ,TRASW,NFSW,EMPTY,INDEF,LDFLG,FIN 86TRKM 4
          86TRKM 5
          COMMON /MATCH3/NOPR,NONU,FLAG,IPR,IPP,MINFLG,MIFLG,MATFLG,GLOBAL 86TRKM 2
          LOGICAL FLAG,NOPR,NONU,MIFLG,MATFLG,GLOBAL 86TRKM 3
          COMMON/CDERIV/IFLAG,MR,NR,DP,NV(11),DAT(20),UZ,STOT, 86TRKM 2
25     1 DAT2(7),MP2FLG,NF,H0,LOCALPH,NUMALPH 86TRKM 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
30     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
35     DATA DZERO/0.0/ 86TRKM 22
          DATA MP2FLG/0/ 86TRKM 23
      C 86TRKM 24
          DATA IBLANK/" 86TRKM 25
          86TRKM 26
40     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
45     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
50     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
55     C 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,	86TRKM	45
	C	AND FROM BETA-FUNCTIONS OF /NAME/, IF A MATRIX.	86TRKM	46
60	C	IRAD = 1 — OUTPUT PHASE ADVANCES IN RADIANS	86TRKM	47
	C	= 0 — IN UNITS OF 2PI	86TRKM	48
	C		86TRKM	49
	C	DEFAULTS —	86TRKM	50
	C		86TRKM	51
65	C	IF KA = KB = 0 (OR BLANK), A TABLE IS RESERVED SUFFICIENT TO STORE	86TRKM	52
	C	THE COMPLETE BEAM LINE.	86TRKM	53
	C	IF KA = -1 NO TABLE IS RESERVED, NO BETAS ARE STORE	86TRKM	54
	C		86TRKM	55
70	C	IF KT1 = IT2 = 0 (OR BLANK), TRACKING IS DONE THROUGH COMPLETE BEAM	86TRKM	56
	C		86TRKM	57
			86TRKM	58
		RADFLG = .FALSE.	86TRKM	59
		IF (MINFLG.NE.4) GO TO 23	86TRKM	60
		IF (MIFLG) GO TO 152	86TRKM	61
75		GO TO 4	86TRKM	62
	23	NOSTOR = .FALSE.	86TRKM	63
		TPI=4.*ACOS(DZERO)	86TRKM	64
		ELNAMEI=IBLANK	86TRKM	65
		KB=INFF(5,M)	86TRKM	66
80		KA=INFF(4,M)	86TRKM	67
		LQ3 = INFF(24,M)	86TRKM	68
		NCOL = IDAT(M,5)	86TRKM	69
		IF (KA.GT.-1) GO TO 151	86TRKM	70
		NOSTOR = .TRUE.	86TRKM	71
85		GO TO 152	86TRKM	72
	151	LTAB0 = LQ3 - NCOL*KA - 1	86TRKM	73
	152	MBML = MDAT(M,1)	86TRKM	74
			86TRKM	75
		NDAT=INFF(17,MBML)	86TRKM	76
90		CALL MIFILL(MBML,1,NDAT,NELS,MI)	86TRKM	77
		CALL MMM(M,NELS,MI)	86TRKM	78
		IF (MINFLG.EQ.4) GO TO 4	86TRKM	79
		IP1=IDAT(M,1)	86TRKM	80
		IP2=IDAT(M,2)	86TRKM	81
95		ICALL=IDAT(M,6)	86TRKM	82
		SI = FDAT(M,2)	86TRKM	83
		TI = 0.	86TRKM	84
		H0 = FDAT(M,1)	86TRKM	85
		DAT2(7) = H0	86TRKM	86
100		MAPK = BDAT(M,3)	86TRKM	87
		IRAD = IDAT(M,4)	86TRKM	88
	C	SET UP CONSTANTS FOR CALCULATING PSI IN DEGREES OR RADIANS	86TRKM	89
		IF (IRAD.EQ.1) RADFLG=.TRUE.	86TRKM	90
		DNOM = TPI	86TRKM	91
105		CCON = 1.	86TRKM	92
		IF (.NOT.RADFLG) GO TO 153	86TRKM	93
		DNOM = 1.	86TRKM	94
		CCON = TPI	86TRKM	95
	153	CONTINUE	86TRKM	96
110			86TRKM	97
		NAMBET=BDAT(M,2)	86TRKM	98
		NAMVEC = BDAT(M,3)	86TRKM	99
		NAMPAR = BDAT(M,4)	86TRKM	100
		NAME=INFF(2,M)	86TRKM	101

115		ITABLE = IDAT(M,3)	86TRKM	102
		IF(ITABLE.EQ.1.AND..NOT.NOSTOR) GO TO 10	86TRKM	103
		GO TO 12	86TRKM	104
			86TRKM	105
			86TRKM	106
120	C 10	IF ((ICALL.EQ.0).OR.(IP1.EQ.0)) GO TO 12	86TRKM	107
	10	CONTINUE	86TRKM	108
	C	INITIAL BETAS FROM BETA FUNCTION TABLE OF TRKB INSTRUCTION,	86TRKM	109
	C	EXCEPT THAT FOR FIRST CALL, OR IF IP1=0, THEY ARE TAKEN FROM IBET OR	86TRKM	110
		LOCBET=NCOL*(IP1-KA)+1	86TRKM	111
125		MBET=M	86TRKM	112
		NTYP=8	86TRKM	113
		CALL DATA(MBET,NTYP,LOCBET+12,1,SI)	86TRKM	114
		CALL DATA(MBET,NTYP,LOCBET+13,1,TI)	86TRKM	115
		GO TO 13	86TRKM	116
			86TRKM	117
130	C	INITIAL BETAS FROM ARRAY SPECIFIED BY NAMBET	86TRKM	118
	12.	LOCBET=1	86TRKM	119
		NTYP=1	86TRKM	120
		MBET=MDAT(M,2)	86TRKM	121
135			86TRKM	122
		NAMOP=INFF(1,MBET)	86TRKM	123
			86TRKM	124
		IF(NAMOP.EQ.4HIBET) GO TO 13	86TRKM	125
	C	INITIAL BETAS ARE BETAS OF MATRICES MBET.	86TRKM	126
140		CALL BET(MBET,BXI,BYI,1)	86TRKM	127
		BXI(1)=0.	86TRKM	128
		BYI(1)=0.	86TRKM	129
			86TRKM	130
		GO TO 16	86TRKM	131
			86TRKM	132
145	13	CALL DATA(MBET,NTYP,LOCBET ,6,BXI)	86TRKM	133
		CALL DATA(MBET,NTYP,LOCBET+6,6,BYI)	86TRKM	134
			86TRKM	135
	16	CONTINUE	86TRKM	136
			86TRKM	137
150		IF (MAPK.EQ.4HMAP2) MP2FLG=1	86TRKM	138
		IF (MAPK.NE.3HELQ) GO TO 34	86TRKM	139
		NUM = 7	86TRKM	140
		GO TO 35	86TRKM	141
155	34	DECODE(10,1002,MAPK) NUM	86TRKM	142
	1002	FORMAT(3X,I1)	86TRKM	143
	35	IF (NAMPAR.EQ.IBLANK) GO TO 26	86TRKM	144
		MPAR = MDAT(M,4)	86TRKM	145
		CALL DATA(MPAR,1,1,6,DAT2)	86TRKM	146
			86TRKM	147
160	26	IF (IP1.EQ.0) GO TO 9	86TRKM	148
		NELI = MI(IP1)	86TRKM	149
		ELNAMEI=MNAME(NELI)	86TRKM	150
	9	IF(IP1.LT.KA) NOSTOR=.TRUE.	86TRKM	151
165		IF(IP2.GT.KB) NOSTOR=.TRUE.	86TRKM	152
		ICALL=1	86TRKM	153
		CALL STDAT(M,3,3,1,ICALL)	86TRKM	154
			86TRKM	155
			86TRKM	156
170	4	CONTINUE	86TRKM	157
		S = SI	86TRKM	158

	TH = TI	86TRKM	159
		86TRKM	160
	DO 3 I=1,6	86TRKM	161
175	BX(I) = BXI(I)	86TRKM	162
	BY(I) = BYI(I)	86TRKM	163
	3 CONTINUE	86TRKM	164
	I1 = IP1 + 1	86TRKM	165
	I2 = IP2	86TRKM	166
180		86TRKM	167
	IF (NOPR) GO TO 7	86TRKM	168
	IF (I2.GT.45) WRITE (3,1004)	86TRKM	169
	LNUM = 0	86TRKM	170
	CALL DASH	86TRKM	171
185	406 WRITE(3,1000)	86TRKM	172
		86TRKM	173
	ELNAME=ELNAMEI	86TRKM	174
	7 CONTINUE	86TRKM	175
		86TRKM	176
190	C ALL ELEMENTS ARE DEFINED BY MAPS	86TRKM	177
	21 NR = 1	86TRKM	178
	IFLAG = 1	86TRKM	179
	NV(1) = 1	86TRKM	180
	S = 0.	86TRKM	181
195	TH = 0.	86TRKM	182
	IF (NUM-3) 27,28,29	86TRKM	183
		86TRKM	184
	C MAP2	86TRKM	185
	27 MR = 8	86TRKM	186
200	GO TO 19	86TRKM	187
		86TRKM	188
	C MAP3 — TRANSVERSE AND LONGITUDINAL ENVELOPES	86TRKM	189
	28 MR = 6	86TRKM	190
	GO TO 19	86TRKM	191
205		86TRKM	192
	C MAP4 — TRANSVERSE ENVELOPES	86TRKM	193
	29 MR = 4	86TRKM	194
		86TRKM	195
	C LOOP OVER POSITIONS	86TRKM	196
210	19 DO 1 IBS = I1,I2	86TRKM	197
	IPOS = IBS - 1	86TRKM	198
	IF(NOPR) GO TO 6	86TRKM	199
		86TRKM	200
	190 IF (MOD(LNUM,45).NE.0.OR.LNUM.EQ.0) GO TO 191	86TRKM	201
215	WRITE (3,1004)	86TRKM	202
	WRITE (3,1000)	86TRKM	203
	LNUM = 0	86TRKM	204
	191 IF (MOD(LNUM,5).EQ.0.AND.LNUM.NE.0) WRITE (3,1003)	86TRKM	205
	WRITE(3,1001) IPOS,ELNAME,S,(BX(I),I=1,3),BX(5),BX(6),	86TRKM	206
220	1 (BY(I),I=1,3),BY(5),BY(6)	86TRKM	207
	LNUM = LNUM + 1	86TRKM	208
	6 NEL = MI(IBS)	86TRKM	209
	IF(NOSTOR) GO TO 8	86TRKM	210
	J=LTAB0+NCOL*IPOS	86TRKM	211
225	DO 11 I=1,14	86TRKM	212
	11 STORE(J+I)=BETF(I)	86TRKM	213
	8 CONTINUE	86TRKM	214
		86TRKM	215

		ELNAME=MNAME(NEL)	86TRKM	216
230		STOT = S	86TRKM	217
	C		86TRKM	218
		MAPK2=INFF(1,NEL)	86TRKM	219
		IF (MAPK2.EQ.MAPK) GO TO 20	86TRKM	220
		MAPK=MAPK2	86TRKM	221
235		IF (MAPK.EQ.3HELQ) GO TO 22	86TRKM	222
		NUM=4	86TRKM	223
		GO TO 32	86TRKM	224
	22	NUM=7	86TRKM	225
		GO TO 36	86TRKM	226
240	20	IF (NUM.EQ.7) GO TO 36	86TRKM	227
		IF (NUM-3) 30,31,32	86TRKM	228
			86TRKM	229
	C	MAP2	86TRKM	230
	30	P(1) = BX(2)	86TRKM	231
245		P(2) = BX(3)	86TRKM	232
		P(3) = BX(5)	86TRKM	233
		P(4) = BX(6)	86TRKM	234
		P(5) = BY(2)	86TRKM	235
		P(6) = BY(3)	86TRKM	236
250		P(7) = BY(5)	86TRKM	237
		P(8) = BY(6)	86TRKM	238
		CALL DIFEQ(NEL,P,DER2)	86TRKM	239
		BX(2) = P(1)	86TRKM	240
		BX(3) = P(2)	86TRKM	241
255		BX(5) = P(3)	86TRKM	242
		BX(6) = P(4)	86TRKM	243
		BY(2) = P(5)	86TRKM	244
		BY(3) = P(6)	86TRKM	245
		BY(5) = P(7)	86TRKM	246
260		BY(6) = P(8)	86TRKM	247
		GO TO 33	86TRKM	248
			86TRKM	249
	C	MAP3	86TRKM	250
	31	P(1) = BX(2)	86TRKM	251
265		P(2) = BX(3)	86TRKM	252
		P(3) = BY(2)	86TRKM	253
		P(4) = BY(3)	86TRKM	254
		P(5) = BX(5)	86TRKM	255
		P(6) = BX(6)	86TRKM	256
270		P(7) = 0.	86TRKM	257
		CALL DIFEQ(NEL,P,DER3)	86TRKM	258
		BX(2) = P(1)	86TRKM	259
		BX(3) = P(2)	86TRKM	260
		BY(2) = P(3)	86TRKM	261
275		BY(3) = P(4)	86TRKM	262
		BX(5) = P(5)	86TRKM	263
		BX(6) = P(6)	86TRKM	264
		GO TO 33	86TRKM	265
			86TRKM	266
280	C	ELQ	86TRKM	267
	36	P(1) = BX(2)	86TRKM	268
		P(2) = BX(3)	86TRKM	269
		P(3) = BY(2)	86TRKM	270
		P(4) = BY(3)	86TRKM	271
285		CALL DIFEQ(NEL,P,DELQ)	86TRKM	272

	GO TO 37	86TRKM	273
		86TRKM	274
	C MAP4	86TRKM	275
290	32 P(1) = BX(2)	86TRKM	276
	P(2) = BX(3)	86TRKM	277
	P(3) = BY(2)	86TRKM	278
	P(4) = BY(3)	86TRKM	279
	P(5) = 0.	86TRKM	280
	P(6) = 0.	86TRKM	281
295	CALL DIFEQ(NEL,P,DER4)	86TRKM	282
	37 BX(2) = P(1)	86TRKM	283
	BX(3) = P(2)	86TRKM	284
	BY(2) = P(3)	86TRKM	285
	BY(3) = P(4)	86TRKM	286
300	33 S = S + DAT(1)	86TRKM	287
		86TRKM	288
	1 CONTINUE	86TRKM	289
		86TRKM	290
	17 IPOS = I2	86TRKM	291
305	MP2FLG = 0	86TRKM	292
	IF(NOSTOR) GO TO 5	86TRKM	293
		86TRKM	294
	J = J + NCOL	86TRKM	295
	DO 14 I=1,14	86TRKM	296
310	14 STORE(J+I)=BETF(I)	86TRKM	297
		86TRKM	298
	5 CONTINUE	86TRKM	299
		86TRKM	300
	IF(NOPR) RETURN	86TRKM	301
315		86TRKM	302
	150 WRITE(3,1001) IPOS,ELNAME,S,(BX(I),I=1,3),BX(5),BX(6),	86TRKM	303
	1 (BY(I),I=1,3),BY(5),BY(6)	86TRKM	304
		86TRKM	305
	154 CALL DASH	86TRKM	306
320		86TRKM	307
	RETURN	86TRKM	308
		86TRKM	309
	1000 FORMAT (" POS",11X,"S",7X,"QX",10X,"BX",8X,"AX",8X,"X",9X,	86TRKM	310
	1 "DX",14X,"QY",10X,"BY",8X,"AY",8X,"Y",9X,"DY")	86TRKM	311
325		86TRKM	312
	1001 FORMAT (I4,1X,A4,1X,F10.4,2(F10.6,F11.4,F11.6,F10.6,F10.6,6X))	86TRKM	313
	1004 FORMAT (1H1)	86TRKM	314
	1003 FORMAT (1H)	86TRKM	315
	END	86TRKM	316

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

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

1		TRKCN	2
	SUBROUTINE TRKCN (MFXPT, NEL, P, NP, IOP)	TRKCN	3
		TRKCN	4
5	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
10	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, MSYMP	DIM	2
		DIM	3
15	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
20	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
25	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
30	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
35		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
40	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
45	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
50	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
55		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
60	C LOOP OVER PRD FACTORS	TRKCN	50
	21 I = I + 1	TRKCN	51
	IF (I.GT.K) GO TO 45	TRKCN	52
	22 NN = MDAT(MM,I)	TRKCN	53
65	KIND = INFF(20,NN)	TRKCN	54
	IF (KIND.NE.3HPRD) GO TO 222	TRKCN	55
	N2 = NN	TRKCN	56
	K2 = 0	TRKCN	57
	ASSIGN 321 TO RET	TRKCN	58
70	KN = INFF(4,N2)	TRKCN	59
	321 K2 = K2 + 1	TRKCN	60
	NN = MDAT(N2,K2)	TRKCN	61
	KIND = INFF(20,NN)	TRKCN	62
	322 IF (K2.EQ.KN) ASSIGN 21 TO RET	TRKCN	63
75	IF (KIND.NE.3HPRD) GO TO 222	TRKCN	64
	N3 = NN	TRKCN	65
	K3 = 0	TRKCN	66
	ASSIGN 421 TO RET	TRKCN	67
	KN3 = INFF(4,N3)	TRKCN	68
80	421 K3 = K3 + 1	TRKCN	69
	NN = MDAT(N3,K3)	TRKCN	70
	KIND = INFF(20,NN)	TRKCN	71
	IF (K3.EQ.KN3) ASSIGN 321 TO RET	TRKCN	72
	IF (K2.EQ.KN.AND.K3.EQ.KN3) ASSIGN 21 TO RET	TRKCN	73
85	IF (KIND.NE.3HPRD) GO TO 222	TRKCN	74
	WRITE (3,1000)	TRKCN	75
	1000 FORMAT (* PRD INSTRUCTIONS NESTED MORE THAN THREE DEEP - ERROR*)	TRKCN	76
	ERROR = .TRUE.	TRKCN	77
	RETURN	TRKCN	78
90	222 IF (KIND.NE.3HMAP) GO TO 40	TRKCN	79
		TRKCN	80
	* PICK UP SUBROUTINE NAME	TRKCN	81
	221 OPNAME = INFF(1,NN)	TRKCN	82
	IF (OPNAME.NE.HSXTP) GO TO 223	TRKCN	83
95	NUM = 11	TRKCN	84
	GO TO 9	TRKCN	85
	223 IF (OPNAME.EQ.HMOVE) GO TO 27	TRKCN	86
	IF (OPNAME.NE.HNPOL) GO TO 25	TRKCN	87
	NUM = 12	TRKCN	88
100	GO TO 9	TRKCN	89
	* STRIP OFF MAP SUBROUTINE NUMBER	TRKCN	90
	25 DECODE (10,100,OPNAME) NUM	TRKCN	91
	100 FORMAT (3X,I1,6X)	TRKCN	92
	IF (NUM.EQ.0) NUM = + 0	TRKCN	93
105	NUM = NUM + 1	TRKCN	94
		TRKCN	95
	* GET DATA OF MAP INSTRUCTION. TRANSMIT IN ARGUMENT.	TRKCN	96
	9 NFL = INFF(15,NN)	TRKCN	97
	CALL DATA(NN,1,1,NFL,PAR)	TRKCN	98
110	GO TO (10,11,12,13,14,15,16,17,18,19,23,28) NUM	TRKCN	99
	10 CALL MAP (P,PAR)	TRKCN	100
	GO TO RET,(21,43,45,321,421)	TRKCN	101
	C MAP1 — WIGGLER MAGNET	TRKCN	102
	11 CALL DIFEQ(NN,P,DER1)	TRKCN	103
		TRKCN	104

115	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
120	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
125	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
130	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
135	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
140	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
145	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
150		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
155	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
160	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
165	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
170	37 CALL SOL(NEL,P,NP)	TRKCN	158
		TRKCN	159

45 RETURN
END

TRKCN 160
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.

1			TRKCN	162	
		SUBROUTINE WBE(K,L)	WBE	2	
	C	WRITES BETATRON FUNCTIONS OF ELEMENTS MI(I),I=1,K	WBE	3	
5		LEVEL 2, STORE,INFF,IWORK	BLANK	2	
		COMMON STORE(48000),IWORK(10)	BLANK	3	
		DIMENSION INFF(24,2000)	86MARSIZ	1	
		EQUIVALENCE (INFF,STORE)	86MARSIZ	2	
			86MARSIZ	3	
			BLANK	5	
10		COMMON/BMI/MI(5000)	BMIL	1	
		COMMON/NELS/NELS	BMIL	2	
		LEVEL 2,MI	BMIL	3	
			BMIL	4	
			BMI	3	
15		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	
20	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	
25		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	
30		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	
35		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
40		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	
45	53	FORMAT(1X,A5,F12.8,F10.5,4F10.6,2X,F12.8,F10.5,4F10.6)	WBE	36	
		END	WBE	37	

1	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
5	C IF KB NOT 0 AND KA=1, WORKING STORAGE IS ALSO PRINTED.	WFLSR	6
	LEVEL 2, STORE,INFF,IWORK	BLANK	2
	COMMON STORE(48000),IWORK(10)	BLANK	3
	DIMENSION INFF(24,2000)	86MARSIZ	1
10	EQUIVALENCE (INFF,STORE)	86MARSIZ	2
		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
15	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
20		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,MSYMP	DIM	2
25		DIM	3
	LOGICAL ENDL	WFLSR	12
C		WFLSR	13
	IF (MODE.EQ.3) GO TO 16	WFLSR	14
	ISGN = 1H	WFLSR	15
30	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
35	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
40	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
45	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
50	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)=05,8H),I=1,4),4(2X,020))	WFLSR	38
	WRITE (3,23) MX,(INFF(J,MX),J=5,8)	WFLSR	39
23	FORMAT (16X,8H(M(DEC)=15,8H),I=5,8),4(2X,020))	WFLSR	40
55	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)	WFLSR	44
	J1 = J2 + 1	WFLSR	45
60	30 FORMAT (21X,5H(I = ,I2,1H,I2,1H),5X,4(2X,020))	WFLSR	46
	31 CONTINUE	WFLSR	47
	MX=MX-1	WFLSR	48
	10 CONTINUE	WFLSR	49
	C PRINT STOR	WFLSR	50
65	IF (ERROR) RETURN	WFLSR	51
	IF (KB.EQ.0) RETURN	WFLSR	52
	LX=LFILE	WFLSR	53
	L = ISAV*(14+KADD)	WFLSR	54
	IF(KA.EQ.1) L=1	WFLSR	55
70	C ENDL IS SWITCH TO TERMINATE PRINTING	WFLSR	56
	ENDL=.FALSE.	WFLSR	57
	WRITE (3,28)	WFLSR	58
	28 FORMAT (31H1 CONTENTS OF STORE IN OCTAL//	WFLSR	59
	141H OCTAL LOCATION PLUS 0 1 2 3/	WFLSR	60
75	241H 4 5 6 7//)	WFLSR	61
	17 J=L+3	WFLSR	62
	IF (J.LT.LX) GO TO 18	WFLSR	63
	J=LX	WFLSR	64
	ENDL=.TRUE.	WFLSR	65
80	18 WRITE (3,25) L,(STORE(I),I=L,J)	WFLSR	66
	25 FORMAT (5X,06,5X,4(020,2X))	WFLSR	67
	IF (ENDL) GO TO 20	WFLSR	68
	L=J+1	WFLSR	69
	J=L+3	WFLSR	70
85	IF (J.LT.LX) GO TO 19	WFLSR	71
	J=LX	WFLSR	72
	ENDL=.TRUE.	WFLSR	73
	19 WRITE (3,26) (STORE(I),I=L,J)	WFLSR	74
	26 FORMAT (16X,4(020,2X)//)	WFLSR	75
90	IF (ENDL) GO TO 20	WFLSR	76
	L=J+1	WFLSR	77
	GO TO 17	WFLSR	78
	20 WRITE (3,27)	WFLSR	79
	27 FORMAT (1H1)	WFLSR	80
95	RETURN	WFLSR	81
	END	WFLSR	82

1	SUBROUTINE WMA(KA)	WMA	2
	C WRITES MATRICES OF ELEMENTS MI(I), I=1,KA.	WMA	3
	LEVEL 2, STORE, INFF, IWORK	BLANK	2
5	COMMON STORE(48000), IWORK(10)	BLANK	3
	DIMENSION INFF(24,2000)	86MARSIZ	1
	EQUIVALENCE (INFF,STORE)	86MARSIZ	2
		86MARSIZ	3
		BLANK	5
	COMMON/SWTC/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	SWTCH	2
10	LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS	SWTCH	3
		SWTCH	4
	COMMON/BMI/MI(5000)	BMIL	1
	COMMON/NELS/NELS	BMIL	2
15	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
20		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
25	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
30	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
35	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
40	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
45	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
50	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
55	DO 350 I=1,KA	WMA	45
	M1 = MI(I)	WMA	46
	N = MNAME(M1)	WMA	47

	CALL RTRV7(M1,T,RW)	WMA	48
	DO 305 II=1,3	WMA	49
60	WRITE (3,34) (T(II,J),J=1,7)	WMA	50
	305 CONTINUE	WMA	51
	WRITE (3,35) N,(T(4,J),J=1,7)	WMA	52
	DO 310 II=5,7	WMA	53
	WRITE (3,34) (T(II,J),J=1,7)	WMA	54
65	310 CONTINUE	WMA	55
	WRITE (3,15) RW(1),RW(2)	WMA	56
	350 CONTINUE	WMA	57
	355 CALL DASH	WMA	58
	RETURN	WMA	59
70	11 FORMAT (20H TRANSFER MATRICES,25X,7HRX(I,J),40X,7HRY(I,J)//	WMA	60
	134X,1HX,11X,2HDX,11X,1H1,20X,1HY,12X,2HDY,11X,1H1/9X,7HELEMENT)	WMA	61
	12 FORMAT (27X,3F13.8,8X,3F13.8)	WMA	62
	13 FORMAT (10X,A5,12X,3F13.8,8X,3F13.8/30X,2H0.,11X,2H0.,11X,2H1.,19X	WMA	63
	1,2H0.,11X,2H0.,11X,2H1./)	WMA	64
75	15 FORMAT (/30X,9HLENGTH = ,F13.8,5X,9H THETA = ,F13.8///)	WMA	65
	21 FORMAT (20H TRANSFER MATRICES,29X,7HRX(I,J),35X,7HRY(I,J)//	WMA	66
	134X,1HX,9X,5HDX/DS,9X,3H-DS,8X,4HDP/P,15X,1HY,10X,5HDY/DS/	WMA	67
	29X,7HELEMENT)	WMA	68
	22 FORMAT (27X,2F13.8,3X,2H0.,8X,F13.8,3X,2F13.8)	WMA	69
80	23 FORMAT (10X,A5,12X,2F13.8,3X,2H0.,8X,F13.8,3X,2F13.8)	WMA	70
	24 FORMAT (27X,2F13.8,3X,10H1.00000000,F13.8/30X,2H0.,11X,2H0.,11X,	WMA	71
	12H0.,11X,10H1.00000000)	WMA	72
	34 FORMAT (30X,7F13.8)	WMA	73
	35 FORMAT (10X,A5,15X,7F13.8)	WMA	74
85	36 FORMAT (20H TRANSFER MATRICES,53X,6HR(I,J)//36X,1HX,10X,5HDX/DS,	WMA	75
	110X,1HY,10X,5HDY/DS,9X,3H-DS,9X,4HDP/P,10X,1H1/9X,7HELEMENT)	WMA	76
	END	WMA	77

1	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
5	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
10	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
15	END	WRTINF	11

1	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
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
10		BLANK	5
	LOGICAL SEQ,ACT	XEQCON	7
	INTEGER ELNUM	XEQCON	8
	DIMENSION NDAT(50)	XEQCON	9
	DATA IBLNK/1H /	XEQCON	10
15		XEQCON	11
	SEQ = .FALSE.	XEQCON	12
	KA = INFF(4,M)	XEQCON	13
	KB = INFF(5,M)	XEQCON	14
	NAM = INFF(1,M)	XEQCON	15
20		XEQCON	16
	ACT = .TRUE.	XEQCON	17
	IF (NAM.EQ.4HDELE) ACT = .FALSE.	XEQCON	18
	IF (KB.EQ.1) SEQ = .TRUE.	XEQCON	19
	KC = KA	XEQCON	20
	IF (SEQ) KC = 1	XEQCON	21
25		XEQCON	22
	CALL DATA(M,2,1,KC,NDAT)	XEQCON	23
	IF(SEQ) GO TO 15	XEQCON	24
	C CHANGE EXECUTION MODE OF ALL ELEMENTS IN NDAT ARRAY	XEQCON	25
	DO 10 I=1,KA	XEQCON	26
	IF (NDAT(I).EQ.IBLNK) GO TO 10	XEQCON	27
30		XEQCON	28
	NN = ELNUM(NDAT(I))	XEQCON	29
	IOP = IABS(INFF(3,NN))	XEQCON	30
	IF (.NOT.ACT) IOP = -IOP	XEQCON	31
	INFF(3,NN) = IOP	XEQCON	32
35	10 CONTINUE	XEQCON	33
	RETURN	XEQCON	34
40		XEQCON	35
	15 CONTINUE	XEQCON	36
	NN = ELNUM(NDAT(1))	XEQCON	37
	DO 20 I=1,KA	XEQCON	38
	IOP = IABS(INFF(3,NN))	XEQCON	39
	IF (.NOT.ACT) IOP = -IOP	XEQCON	40
	INFF(3,NN) = IOP	XEQCON	41
45	NN = INFF(8,NN)	XEQCON	42
	20 CONTINUE	XEQCON	43
	RETURN	XEQCON	44
	END	XEQCON	45

1		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
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
10	C		ZAM0	11
		HPR=HPRO	ZAM0	12
		XX=DBLE(X)	ZAM0	13
		N=NO	ZAM0	14
		EUB=EMAX	ZAM0	15
15		ELB=EMAX/SSSR	ZAM0	16
		IP=1	ZAM0	17
		IT=0	ZAM0	18
		NRKS=0	ZAM0	19
		SWIN=SWEX	ZAM0	20
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
25		END	ZAM0	26

1		SUBROUTINE ZAM(X,DERI ,Y,F,T,SWPR)	ZAM	2
		COMMON /INTC/ IPMX,AREF,EMAX,SSSR,HFAC,SWAM,SWEX	ZAM	3
		COMMON /INTP/ HPR,XX,N,EUB,ELB,IP,IT,NRKS,SWIN	ZAM	4
	C		ZAM	5
5		DIMENSION Y(1),F(1),T(8,1)	ZAM	6
		LOGICAL SWAM,SWEX,SWIN	ZAM	7
		LOGICAL SWPR	ZAM	8
		INTEGER HFAC	ZAM	9
		DOUBLE PRECISION T,HPR,XX	ZAM	10
10		DOUBLE PRECISION D,H	ZAM	11
	6000	FORMAT (36H0 CANNOT DECREASE H BECAUSE OF HMIN. ,1PE16.8,I20)	ZAM	12
	C		ZAM	13
	1	CONTINUE	ZAM	14
		SWPR=.FALSE.	ZAM	15
15		TEST=0.0	ZAM	16
		H=HPR/DBLE(FLOAT(IP*24))	ZAM	17
		IF ((NRKS .LT. 3) .OR. (.NOT. SWAM)) GO TO 200	ZAM	18
	C		ZAM	19
	C	ADAMS-MOULTON STEP.	ZAM	20
20	100	CONTINUE	ZAM	21
		DO 109 I=1,N	ZAM	22
		D=DBLE(F(I))	ZAM	23
		T(4,I)=D	ZAM	24
		Y(I)=SNGL(T(5,I)+H*(ZAM	25
25	X	55.0D0*D-59.0D0*T(3,I)+37.0D0*T(2,I)- 9.0D0*T(1,I))	ZAM	26
	109	CONTINUE	ZAM	27
		X=SNGL(XX+24.0D0*H)	ZAM	28
		CALL DERI (X,Y,F)	ZAM	29
		DO 119 I=1,N	ZAM	30
30		D=DBLE(F(I))	ZAM	31
		D=(T(5,I)+H*(ZAM	32
	X	9.0D0*D+19.0D0*T(4,I)- 5.0D0*T(3,I)+ T(2,I))	ZAM	33
		T(6,I)=D	ZAM	34
		E=ABS(SNGL(D)-Y(I))/14.0	ZAM	35
35		TEST=AMAX1(E/AMAX1(AREF,ABS(SNGL(D))),TEST)	ZAM	36
	119	CONTINUE	ZAM	37
	C		ZAM	38
		GO TO 300	ZAM	39
	C		ZAM	40
40	C	ZONNEVELD STEP.	ZAM	41
	200	CONTINUE	ZAM	42
		DO 209 I=1,N	ZAM	43
		D=DBLE(F(I))	ZAM	44
		T(4,I)=D	ZAM	45
45	C	1	ZAM	46
		Y(I)=SNGL(T(5,I)+H*(ZAM	47
	X	12.0D0*D)	ZAM	48
	209	CONTINUE	ZAM	49
		X=SNGL(XX+12.0D0*H)	ZAM	50
50		CALL DERI (X,Y,F)	ZAM	51
		DO 219 I=1,N	ZAM	52
		D=DBLE(F(I))	ZAM	53
		T(6,I)=D	ZAM	54
	C	2	ZAM	55
55		Y(I)=SNGL(T(5,I)+H*(ZAM	56
	X	12.0D0*D)	ZAM	57
	219	CONTINUE	ZAM	58

	CALL DERI (X,Y,F)	ZAM	59
	DO 229 I=1,N	ZAM	60
60	D=DBLE(F(I))	ZAM	61
	T(7,I)=D	ZAM	62
	C 3	ZAM	63
	Y(I)=SNGL(T(5,I)+H*(ZAM	64
	X 24.0D0*D))	ZAM	65
65	229 CONTINUE	ZAM	66
	X=SNGL(XX+24.0D0*H)	ZAM	67
	CALL DERI (X,Y,F)	ZAM	68
	DO 239 I=1,N	ZAM	69
	D=DBLE(F(I))	ZAM	70
70	T(8,I)=D	ZAM	71
	C 4	ZAM	72
	Y(I)=SNGL(T(5,I)+H*(ZAM	73
	X 3.75D0*T(4,I)+5.25D0*T(6,I)+9.75D0*T(7,I)-0.75D0*D))	ZAM	74
	239 CONTINUE	ZAM	75
75	X=SNGL(XX+18.0D0*H)	ZAM	76
	CALL DERI (X,Y,F)	ZAM	77
	DO 249 I=1,N	ZAM	78
	D=DBLE(F(I))	ZAM	79
	E=ABS(SNGL(H*(ZAM	80
80	X -16.0D0*T(4,I)+48.0D0*T(6,I)+48.0D0*T(7,I)+48.0D0*T(8,I)	ZAM	81
	X -128.0D0*D)))	ZAM	82
	C 5	ZAM	83
	D=(T(5,I)+H*(ZAM	84
	X 4.0D0*T(4,I)+ 8.0D0*T(6,I)+ 8.0D0*T(7,I)+ 4.0D0*T(8,I)	ZAM	85
85	X))	ZAM	86
	T(6,I)=D	ZAM	87
	TEST=AMAX1(E/AMAX1(AREF,ABS(SNGL(D))),TEST)	ZAM	88
	249 CONTINUE	ZAM	89
	C	ZAM	90
90	C BOTH ADAMS-MOULTON AND ZONNEVELD METHODS CONTINUE FROM HERE.	ZAM	91
	300 CONTINUE	ZAM	92
	X=SNGL(XX+24.0D0*H)	ZAM	93
	IF (TEST .LE. EUB) GO TO 310	ZAM	94
	IF (IP*HFAC .GT. IPMX) GO TO 309	ZAM	95
95	C	ZAM	96
	C REPEAT STEP WITH SMALLER H.	ZAM	97
	NRKS=0	ZAM	98
	IP=IP*HFAC	ZAM	99
	IT=IT*HFAC	ZAM	100
100	DO 305 I=1,N	ZAM	101
	Y(I)=SNGL(T(5,I))	ZAM	102
	F(I)=SNGL(T(4,I))	ZAM	103
	305 CONTINUE	ZAM	104
	GO TO 1	ZAM	105
105	C	ZAM	106
	C CANNOT DECREASE H BECAUSE OF HMIN.	ZAM	107
	309 CONTINUE	ZAM	108
	IF (.NOT. SWIN) GO TO 310	ZAM	109
	PRINT 6000, X,IPMX	ZAM	110
110	SWIN=.FALSE.	ZAM	111
	C	ZAM	112
	310 CONTINUE	ZAM	113
	C	ZAM	114
	C	ZAM	115

115	C	ACCEPT CURRENT STEP.	ZAM	116
	C		ZAM	117
	C	XX STILL HAS NOT BEEN CHANGED SINCE ENTRY.	ZAM	118
	C	YY(XX) IS STILL IN T(5,).	ZAM	119
	C	F(YY) IS IN T(4,).	ZAM	120
120	C		ZAM	121
		IT=IT+1	ZAM	122
		XX=XX+HPR/DBLE(FLOAT(IP))	ZAM	123
		NRKS=MIN0(NRKS+1,4)	ZAM	124
		DO 319 I=1,N	ZAM	125
125		D=T(6,I)	ZAM	126
		T(5,I)=D	ZAM	127
		Y(I)=SNGL(D)	ZAM	128
	319	CONTINUE	ZAM	129
		X=SNGL(XX)	ZAM	130
130		CALL DERI (X,Y,F)	ZAM	131
		IF (IT .LT. IP) GO TO 320	ZAM	132
	C		ZAM	133
	C	X IS A MULTIPLE OF HPRINT.	ZAM	134
		SWPR=.TRUE.	ZAM	135
135		IT=IT-IP	ZAM	136
	C		ZAM	137
	320	CONTINUE	ZAM	138
		IF (TEST .GE. ELB) GO TO 330	ZAM	139
		IF (MOD(IP,HFAC)+MOD(IT,HFAC) .NE. 0) GO TO 330	ZAM	140
140	C		ZAM	141
	C	PROCEED TO NEXT STEP WITH LARGER H, USING ZONNEVELD METHOD.	ZAM	142
		NRKS=0	ZAM	143
		IP=IP/HFAC	ZAM	144
		IT=IT/HFAC	ZAM	145
145		RETURN	ZAM	146
	C		ZAM	147
	C		ZAM	148
	C	PROCEED TO NEXT STEP WITH SAME H.	ZAM	149
	330	CONTINUE	ZAM	150
150		DO 339 I=1,N	ZAM	151
		T(1,I)=T(2,I)	ZAM	152
		T(2,I)=T(3,I)	ZAM	153
		T(3,I)=T(4,I)	ZAM	154
	339	CONTINUE	ZAM	155
155		RETURN	ZAM	156
		END	ZAM	157

1

SUBROUTINE MINUITS
CC AUG. 17, 1975
CALL MINNEW
5 RETURN
END

ZAM 158
MINUITS 3
MINUITS 4
MINUITS 9
MINUITS 10
MINUITS 11

1	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
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
10	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
15	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
20	A/CARD / CWORD ,CWORD2 ,CWORD3 ,WORD7(7)	COMMONU	30
	B/MINIMA/ AMIN ,UP ,NEWMIN ,ITAU ,SIGMA	COMMONU	31
	K/COMMND/ NNAME ,CNAME(25) ,CNAM2(25) ,CNAM3(25)	COMMONU	32
	DIMENSION PVEC(15)	CALFCN	8
	CALL INTOEX(PVEC)	CALFCN	9
25	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
30	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
35	IF (DENOM .LE. 0.) DENOM = 1.0	CALFCN	20
	CALFCN = (F-APSI) / DENOM	CALFCN	21
	RETURN	CALFCN	22
	END	CALFCN	23

1	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
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
10	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
15	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
20	A/CARD / CWORD ,CWORD2 ,CWORD3 ,WORD7(7)	COMMONU	30
	B/MINIMA/ AMIN ,UP ,NEWMIN ,ITAU ,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
25	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
30	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
35	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
40	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
45	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
	ITAU = 0	COMAND	42
50	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
55	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
60		ITAU = 0		COMAND	53
		EPSI = 0.1*UP		COMAND	54
	C	 MINIMIZE	COMAND	55
		100 NF = NFCN		COMAND	56
		CALL SIMPLX		COMAND	57
65		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
70		200 CALL SEEK		COMAND	63
		GO TO 50		COMAND	64
	C	 SIMPLEX	COMAND	65
		300 CALL SIMPLX		COMAND	66
		GO TO 50		COMAND	67
75	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
80		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
85		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
90	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
95		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
100	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
105		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
110		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

115	GO TO 50	COMAND	108
	C	RESTORE	COMAND 109
	1000 IT = WORD7(1) + 0.5	COMAND	110
	CALL RESTOR(IT)	COMAND	111
	GO TO 50	COMAND	112
120	C	END, EXIT	COMAND 113
	1100 IT = WORD7(1) + 0.5	COMAND	114
	IF (ISW(4) .EQ. 1 .OR. IT .GT. 0) GO TO 1150	COMAND	115
	CSY IFLAG = 3	COMAND	116
	CSY WRITE (ISYSWR,5009)	COMAND	117
125	CSY CALL FCN(NPAR,G,F,U,IFLAG)	COMAND	118
	NFCN = NFCN + 1	COMAND	119
	1150 IF (I .EQ. 11) STOP	COMAND	120
	RETURN	COMAND	121
	C	GRADIENT	COMAND 122
130	1200 ISW(3) = 1	COMAND	123
	IF (WORD7(1) .GT. 0.0) GO TO 50	COMAND	124
	DO 1230 I= 1, NU	COMAND	125
	1230 GF(I) = 0.0	COMAND	126
	CALL INTOEX(X)	COMAND	127
135	CALL FCN(NPAR,GF,AMIN,U,2)	COMAND	128
	NFCN = NFCN + 1	COMAND	129
	CALL DERIVE(GF,G2)	COMAND	130
	ISW(3) = 0	COMAND	131
	CALL DERIVE(G, G2)	COMAND	132
140	WRITE (ISYSWR,5013)	COMAND	133
	ISW(3) = 1	COMAND	134
	DO 1250 I= 1, NU	COMAND	135
	LC = LCORSP(I)	COMAND	136
	IF (LC .EQ. 0) GO TO 1250	COMAND	137
145	AWD = AGOOD	COMAND	138
	IF (ABS(GF(LC)-G(LC)) .GT. ABS(G2(LC))) AWD = ABAD	COMAND	139
	IF (GF(LC) .EQ. 0.) AWD = ANONE	COMAND	140
	IF (AWD .NE. AGOOD) ISW(3) = 0	COMAND	141
	WRITE(ISYSWR,5014) I,NAM(I),GF(LC),G(LC),G2(LC),AWD	COMAND	142
150	1250 CONTINUE	COMAND	143
	IF (ISW(3) .EQ. 0) WRITE (ISYSWR,5015)	COMAND	144
	GO TO 50	COMAND	145
	C	CALL FCN	COMAND 146
	1300 IFLAG = WORD7(1)	COMAND	147
155	IF (IFLAG .EQ. 3) ISW(4) = 1	COMAND	148
	CALL FCN(NPAR,G,F,U,IFLAG)	COMAND	149
	NFCN = NFCN + 1	COMAND	150
	IF(IFLAG.LE.5) GO TO 50	COMAND	151
	CALL EXTTOIN(X)	COMAND	152
160	CALL FCN(NPAR,G,AMIN,U,4)	COMAND	153
	NFCN=NFCN+1	COMAND	154
	IF (ISW(2) .LE. 1) GO TO 1350	COMAND	155
	ISW(2) = 1	COMAND	156
	WRITE (ISYSWR, 5010)	COMAND	157
165	1350 CALL MPRINT(1,AMIN)	COMAND	158
	GO TO 50	COMAND	159
	C	MATOUT	COMAND 160
	1400 IF(ISW(2).GE.2) GO TO 1405	COMAND	161
	C	HESSE	COMAND 162
170	1402 CALL HESSE	COMAND	163
	CALL MPRINT(1, AMIN)	COMAND	164

	1405 CALL MATOUT(0.0 , 1)	COMAND	165	
	GO TO 50	COMAND	166	
	C COVARIANCE	COMAND	167
175	1500 NRAPE = WORD7(1) + 0.5	COMAND	168	
	IF (NRAPE .NE. NPAR) GO TO 1550	COMAND	169	
	READ (ISYSRD,5002) ((V(I,J),I=1,NRAPE),J=1,NRAPE)	COMAND	170	
	ISW(2) = 3	COMAND	171	
	CALL MATOUT(0.0 , 1)	COMAND	172	
180	CALL MPRINT(1,AMIN)	COMAND	173	
	GO TO 50	COMAND	174	
	1550 WRITE (ISYSWR,5006)	COMAND	175	
	WRITE (ISYSWR,5008)	COMAND	176	
	NRAP2 = NRAPE**2	COMAND	177	
185	READ (ISYSRD,5002) (G(1),I=1,NRAP2)	COMAND	178	
	GO TO 50	COMAND	179	
	C IMPROVE	COMAND	180
	1600 CONTINUE	COMAND	181	
	IF (ISW(2) .LT. 2) GO TO 550	COMAND	182	
190	CALL IMPROV	COMAND	183	
	IF (NEWMIN .EQ. 1) GO TO 150	COMAND	184	
	GO TO 50	COMAND	185	
	C ERROR DEF	COMAND	186
	1700 CONTINUE	COMAND	187	
195	UP = WORD7(1)	COMAND	188	
	IF (UP .LE. 0.) UP = 1.0	COMAND	189	
	IF (ISW(2) .GE. 1) CALL MPRINT(1,AMIN)	COMAND	190	
	GO TO 50	COMAND	191	
	C PAGE	COMAND	192
200	1800 WRITE (ISYSWR,5005)	COMAND	193	
	GO TO 50	COMAND	194	
	C CONTOUR	COMAND	195
	2000 CONTINUE	COMAND	196	
	IF (ISW(2) .LT. 1) GO TO 550	COMAND	197	
205	KE1 = WORD7(1)	COMAND	198	
	KE2 = WORD7(2)	COMAND	199	
	IF (KE1 .LE. 0 .OR. KE2 .LE. 0) GO TO 905	COMAND	200	
	IF (KE1 .GT. NU .OR. KE2 .GT. NU) GO TO 905	COMAND	201	
	IF (LCORSP(KE1) .LE. 0 .OR. LCORSP(KE2) .LE. 0) GO TO 905	COMAND	202	
210	NFCNMX = 1000	COMAND	203	
	CALL CONTOU	COMAND	204	
	CALL MPRINT(1,AMIN)	COMAND	205	
	IF (NEWMIN .LE. 0) GO TO 50	COMAND	206	
	GO TO 95	COMAND	207	
215	C STANDARD	COMAND	208
	2100 CALL STAND	COMAND	209	
	GO TO 50	COMAND	210	
	C RELEASE	COMAND	211
	2200 CONTINUE	COMAND	212	
220	DO 2220 IRL=1,7	COMAND	213	
	KRL = WORD7(IRL)	COMAND	214	
	IF (KRL .EQ. 0) GO TO 50	COMAND	215	
	KRL = -IABS(KRL)	COMAND	216	
	CALL RESTOR(KRL)	COMAND	217	
225	2220 CONTINUE	COMAND	218	
	GO TO 50	COMAND	219	
	C BLANK COMMAND CARD	COMAND	220
	2300 CONTINUE	COMAND	221	

	GO TO 50	COMAND	222
230	5000 FORMAT (2A4, A2, 7F10.0)	COMAND	223
	5001 FORMAT (1H 10(1H*)/1H 3H*** I3,4H****,2A4,A2,7F15.5)	COMAND	224
	5002 FORMAT (7F10.0,10X)	COMAND	225
	5003 FORMAT (13E10.2)	COMAND	226
	5004 FORMAT (1H 10(1H*))	COMAND	227
235	5005 FORMAT (1H1)	COMAND	228
	5006 FORMAT (1H+10X 17H(COMMAND IGNORED))	COMAND	229
	5007 FORMAT (118H0THE ABOVE COMMAND CANNOT BE EXECUTED BECAUSE COVARIA	COMAND	230
	1NCE MATRIX WAS NEITHER CALCULATED NOR SUPPLIED, OR WAS DESTROYED./	COMAND	231
	2/)	COMAND	232
240	5008 FORMAT(102H SIZE OF COVARIANCE MATRIX TO BE READ DOES NOT CORRESP	COMAND	233
	1OND TO NUMBER OF CURRENTLY VARIABLE PARAMETERS./)	COMAND	234
	5009 FORMAT (28H0CALL TO FCN WITH IFLAG = 3 /)	COMAND	235
	5010 FORMAT (51H NEW START POINT ASSUMED COVARIANCE MATRIX LOST)	COMAND	236
	5011 FORMAT (50H0NEW MINIMUM FOUND. GO BACK TO MINIMIZATION STEP./1H	COMAND	237
245	1 , 60(1H=)/60X1HV/60X1HV/60X1HV/57X7HVVVVVVV/58X5HVVVVV/59	COMAND	238
	2X 3HVVV/60X1HV///)	COMAND	239
	5012 FORMAT (91H0THE ABOVE COMMAND CANNOT BE EXECUTED BECAUSE IT REQU	COMAND	240
	1ESTS A PARAMETER THAT IS NOT VARIABLE./)	COMAND	241
	5013 FORMAT (37H0CHECK OF GRADIENT CALCULATION IN FCN 12X 9HPARAME	COMAND	242
250	1TER 6X 9HG(IN FCN) 3X 9HG(MINUIT) 4X 5HERROR 5X	COMAND	243
	2 9HAGREEMENT)	COMAND	244
	5014 FORMAT (45X I5,2X A10,3E12.4,4X A4)	COMAND	245
	5015 FORMAT (54H0MINUIT DOES NOT ACCEPT DERIVATIVE CALCULATIONS BY FC	COMAND	246
	1N)	COMAND	247
255	END	COMAND	248

1	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
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
10	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
15	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
20	A/CARD / CWORD ,CWORD2 ,CWORD3 ,WORD7(7)	COMMONU	30
	B/MINIMA/ AMIN ,UP ,NEWMIN ,ITAU ,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
25	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
30	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
35	SPT(3,1) = 1H	CONTOU	20
	NSPT = 1	CONTOU	21
	NEWMIN = 0	CONTOU	22
	KI1 = LCORSP(KE1)	CONTOU	23
	KI2 = LCORSP(KE2)	CONTOU	24
40	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
45	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
50	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
55	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	CONTOU	43
	A22 = VT(KI1,KI1) * DETR	CONTOU	44
60	A12 = -VT(KI1,KI2) * DETR	CONTOU	45
	75 CONTINUE	CONTOU	46
	RHO = A12/SQRT(A11*A22)	CONTOU	47
	XMIN = U(KE1)	CONTOU	48
	YMIN = U(KE2)	CONTOU	49
65	IF (LCODE(KE1) .GT. 1) GO TO 110	CONTOU	50
	ALIM(KE1) = XMIN - TEN*WERR(KE1)	CONTOU	51
	BLIM(KE1) = XMIN + TEN*WERR(KE1)	CONTOU	52
	EXIN1 = 1.0	CONTOU	53
	GO TO 115	CONTOU	54
70	110 EXIN1 = 0.5*(BLIM(KE1)-ALIM(KE1))*COS(X(KI1))	CONTOU	55
	115 CONTINUE	CONTOU	56
	IF (LCODE(KE2) .GT. 1) GO TO 120	CONTOU	57
	ALIM(KE2) = YMIN - TEN*WERR(KE2)	CONTOU	58
	BLIM(KE2) = YMIN + TEN*WERR(KE2)	CONTOU	59
75	EXIN2 = 1.0	CONTOU	60
	GO TO 125	CONTOU	61
	120 CONTINUE	CONTOU	62
	EXIN2 = 0.5*(BLIM(KE2)-ALIM(KE2)) * COS(X(KI2))	CONTOU	63
	125 CONTINUE	CONTOU	64
80	A11 = A11 * EXIN1**2	CONTOU	65
	A22 = A22 * EXIN2**2	CONTOU	66
	A12 = A12 * ABS(EXIN1*EXIN2)	CONTOU	67
	CIRCUM = 4.0*SQRT(A11 + A22)	CONTOU	68
	IFLAG = 4	CONTOU	69
85	C . CALCULATE THE CONTOURS	CONTOU	70
	DO 800 ICON= 1, NCON	CONTOU	71
	SYM = DIG(ICON)	CONTOU	72
	ANP = 50 - 5*(NCON-ICON)	CONTOU	73
	DCA = 6.2832/ANP	CONTOU	74
90	RAVE = FLOAT(ICON) * CIRCUM / ANP	CONTOU	75
	RMAXAB = 3.0*RAVE	CONTOU	76
	RMIN = 0.1*RAVE	CONTOU	77
	DIFF = UP* FLOAT(ICON**2)	CONTOU	78
	AIM = AMIN + DIFF	CONTOU	79
95	TOLERN = 0.05*DIFF	CONTOU	80
	NPTMX = 200	CONTOU	81
	IF (NPTDIM-NSPT .LT. NPTMX) NPTMX = NPTDIM-NSPT	CONTOU	82
	NPT = 0	CONTOU	83
	IF (NPTMX .LT. 1) GO TO 810	CONTOU	84
100	WRITE (ISYSWR, 1000) KE1, KE2, SYM,DIFF	CONTOU	85
	C . SET UP FIRST POINT LONG ARM	CONTOU	86
	STEP2 = FLOAT(ICON) * SQRT(A11*(1.-RHO**2))	CONTOU	87
	U(KE1) = XMIN + STEP2	CONTOU	88
	IF (ALLOW(KE1) .LT. 0.) GO TO 731	CONTOU	89
105	CALL FCN(NPAR, G, F2, U, IFLAG)	CONTOU	90
	NFCN = NFCN + 1	CONTOU	91
	IF (F2 .LE. AMIN) GO TO 741	CONTOU	92
	IF (ISW(5) .GE. 2) WRITE (ISYSWR,900)	CONTOU	93
	THETB = 1.5708 - A12/SQRT(A11**2 + A12**2)	CONTOU	94
110	THET1 = THETB	CONTOU	95
	R = 10.0 * RMAXAB	CONTOU	96
	XPT = U(KE1) - R*COS(THET1)	CONTOU	97
	YPT = U(KE2) - R*SIN(THET1)	CONTOU	98
	SLOPE = 2.0 * (F2-AMIN) / (STEP2* SIN(THET1))	CONTOU	99

115	TOLER = TOLERN	CONTOU	100
	NSTEP = 0	CONTOU	101
	RMAX = RAVE	CONTOU	102
	GO TO 407	CONTOU	103
	C CUT STEP LENGTH R	CONTOU	104
120	350 THET1 = 0.25 * (3.0*THETB + THET1)	CONTOU	105
	R = R * 0.5	CONTOU	106
	RMAX = R	CONTOU	107
	TOLER = TOLERN*R/RMAXAB	CONTOU	108
	SLOP = SLOPE	CONTOU	109
125	NSTEP = 0	CONTOU	110
	C FIND NEXT POINT	CONTOU	111
	400 U(KE1) = XPT + R*COS(THET1)	CONTOU	112
	U(KE2) = YPT + R*SIN(THET1)	CONTOU	113
	IF (ALLOW(KE1) .LT. 0. .OR. ALLOW(KE2) .LT. 0.) GO TO 731	CONTOU	114
130	CALL FCN(NPAR,G,F2,U,IFLAG)	CONTOU	115
	NFCN = NFCN + 1	CONTOU	116
	IF (F2 .LE. AMIN) GO TO 741	CONTOU	117
	C . CHECK IF TWO TRIES ARE NEEDED	CONTOU	118
	F1 = F2	CONTOU	119
135	THET2 = THET1	CONTOU	120
	AFA = ABS(F2-AIM)	CONTOU	121
	IF (AFA .GT. 10.*TOLER .AND. R .GT. RMIN) GO TO 350	CONTOU	122
	IF (AFA .LT. 0.2*TOLER) GO TO 420	CONTOU	123
	407 DTHET = (F2-AIM) / (R*SLOPE)	CONTOU	124
140	THMIN = 0.001*RAVE/R	CONTOU	125
	IF (ABS(DTHET) .GT. THMAX) DTHET = SIGN(THMAX,DTHET)	CONTOU	126
	IF (ABS(DTHET) .LT. THMIN) DTHET = SIGN(THMIN,DTHET)	CONTOU	127
	C . SUBSEQUENT TRIALS FOR NEXT PT	CONTOU	128
	410 CONTINUE	CONTOU	129
145	F1 = F2	CONTOU	130
	THET2 = THET1 + DTHET	CONTOU	131
	U(KE1) = XPT + R*COS(THET2)	CONTOU	132
	U(KE2) = YPT + R*SIN(THET2)	CONTOU	133
	IF (ALLOW(KE1) .LT. 0. .OR. ALLOW(KE2) .LT. 0.) GO TO 731	CONTOU	134
150	CALL FCN(NPAR,G,F2,U,IFLAG)	CONTOU	135
	NFCN = NFCN + 1	CONTOU	136
	IF (F2 .LE. AMIN) GO TO 741	CONTOU	137
	SLOPE = (F1-F2)/(DTHET*R)	CONTOU	138
	IF (SLOPE .GT. 0.) GO TO 420	CONTOU	139
155	IF (NPT .GT. 0) GO TO 415	CONTOU	140
	SLOPE = 2.0 * (AMIN-F2) / (SIN(THET2) *(XMIN-U(KE1)))	CONTOU	141
	GO TO 420	CONTOU	142
	415 IF (R .GT. RMIN) GO TO 350	CONTOU	143
	SLOPE = SLOP	CONTOU	144
160	420 CHANG = (F2-AIM)/(SLOPE*R)	CONTOU	145
	IF (ABS(CHANG) .GT. THMAX) CHANG = SIGN(THMAX, CHANG)	CONTOU	146
	THETA = THET2 + CHANG	CONTOU	147
	C CHECK IF BEST TRIAL POINT IS ACCURATE ENOUGH	CONTOU	148
	DELTA = AMIN1(ABS(F2-AIM), ABS(F1-AIM))	CONTOU	149
165	IF (DELTA .LT. TOLER) GO TO 430	CONTOU	150
	IF (NPT .EQ. 0) GO TO 425	CONTOU	151
	IF (NSTEP .EQ. 1 .AND. R .GT. RMIN) GO TO 350	CONTOU	152
	425 NSTEP = NSTEP + 1	CONTOU	153
	IF (NSTEP .GT. NSTMX) GO TO 751	CONTOU	154
170	THET1 = THET2	CONTOU	155
	DTHET = THETA - THET1	CONTOU	156

	GO TO 410	CONTOU	157
C ACCEPT NEW POINT	CONTOU	158
175	430 XPT = XPT + R*COS(THETA)	CONTOU	159
	YPT = YPT + R*SIN(THETA)	CONTOU	160
	U(KE1) = XPT	CONTOU	161
	U(KE2) = YPT	CONTOU	162
	IF (ALLOW(KE1) .LT. 0. .OR. ALLOW(KE2) .LT. 0.) GO TO 731	CONTOU	163
	THETAB = ABS(THETA-THETB)	CONTOU	164
180	NSTEP = 0	CONTOU	165
	440 IF (NPT .GT. 1) GO TO 445	CONTOU	166
	THBEG = THETA	CONTOU	167
	XBEG = XPT	CONTOU	168
	YBEG = YPT	CONTOU	169
185	445 CONTINUE	CONTOU	170
	NPT = NPT + 1	CONTOU	171
	NSPT = NSPT + 1	CONTOU	172
	SPT(1,NSPT) = XPT	CONTOU	173
	SPT(2,NSPT) = YPT	CONTOU	174
190	SPT(3,NSPT) = SYM	CONTOU	175
	XPM = XPT - XMIN	CONTOU	176
	YPM = YPT - YMIN	CONTOU	177
	IF (ISW(5) .LT. 2) GO TO 450	CONTOU	178
	THDEG = THETA*180. / 3.14159265	CONTOU	179
195	THDEG = AMOD(THDEG,360.0)	CONTOU	180
	WRITE (ISYSWR,901) NPT,XPT,YPT,XPM,YPM,DELTA,R,THDEG,NFCN	CONTOU	181
	450 CONTINUE	CONTOU	182
C TEST IF CONTOUR COMPLETED	CONTOU	183
	IF (NPT .LT. MINPT) GO TO 500	CONTOU	184
200	IF ((YPT-YBEG)**2 +(XPT-XBEG)**2 .GT. R**2) GO TO 500	CONTOU	185
	IF (COS(THETA-THBEG) .GT. 0.5) GO TO 600	CONTOU	186
	500 IF (NPT .GE. NPTMX) GO TO 761	CONTOU	187
C ESTIMATE NEW POINT	CONTOU	188
	RINV = THETAB**2 / (R*DCA**2)	CONTOU	189
205	IF (RINV .GT. 1.0/RMAX) GO TO 522	CONTOU	190
	R2 = RMAX	CONTOU	191
	GO TO 540	CONTOU	192
	522 R2 = 1.0/RINV	CONTOU	193
	IF (R2 .LT. RMIN) R2 = RMIN	CONTOU	194
210	540 CHANG = R2*RINV*DCA	CONTOU	195
	IF (CHANG .GT. THMAX) CHANG = THMAX	CONTOU	196
	IF (THETB .GT. THETA) CHANG = -CHANG	CONTOU	197
	THET1 = THETA + CHANG	CONTOU	198
	THETB = THETA	CONTOU	199
215	R = R2	CONTOU	200
	RMAX = AMIN1(2.0*R , RMAXAB)	CONTOU	201
	TOLER = TOLERN * R/RMAXAB	CONTOU	202
	SLOP = SLOPE	CONTOU	203
	GO TO 400	CONTOU	204
220	C CONTOUR COMPLETED NORMALLY	CONTOU	205
	600 WRITE (ISYSWR,605) SYM,NPT	CONTOU	206
	605 FORMAT (9H0CONTOUR A3, 18H IS COMPLETED WITHI5, 7H POINTS//)	CONTOU	207
	GO TO 790	CONTOU	208
C ERROR RETURNS	CONTOU	209
225	731 WRITE (ISYSWR, 735) U(KE1), U(KE2)	CONTOU	210
	735 FORMAT (34H0CONTOUR LEAVING ALLOWED REGION AT2E16.7)	CONTOU	211
	GO TO 780	CONTOU	212
	741 WRITE (ISYSWR, 745)	CONTOU	213

	745	FORMAT (38H0NEW ABSOLUTE MINIMUM FOUND BY CONTOUR/)	CONTOU	214
230		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
235		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
240	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
245	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
250		WRITE (ISYSWR,1002) KE1,KE2,UP,KE2	CONTOU	235
		CALL PLTCON(NSPT,SPT)	CONTOU	236
		WRITE (ISYSWR,1003) KE1	CONTOU	237
		RETURN	CONTOU	238
255	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
260	1001	FORMAT (55H CONTOUR NOT COMPLETED BECAUSE OF ABOVE ERROR CONDITI	CONTOU	245
		10N/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
265	1003	FORMAT (25H+ PARAMETER I3,1H,/)	CONTOU	250
		END	CONTOU	251

60

150 CONTINUE
200 RETURN
END

DERIVE 48
DERIVE 49
DERIVE 50

1	SUBROUTINE EXTGIN(PINT)	EXTGIN	2
CC	TRANSFORMS THE EXTERNAL PARAMETER VALUES X TO INTERNAL	EXTGIN	3
CC	VALUES IN THE DENSE ARRAY PINT. FUNCTION PINTF IS USED.	EXTGIN	4
CC		EXTGIN	5
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
10	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
15	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
20	B/MINIMA/ AMIN ,UP ,NEWMIN ,ITAU ,SIGMA	COMMONU	31
	K/COMMND/ NNAME ,CNAME(25) ,CNAM2(25) ,CNAM3(25)	COMMONU	32
	DIMENSION PINT(2)	EXTGIN	7
	LIMSET=0	EXTGIN	8
	DO 100 I= 1, NU	EXTGIN	9
25	J = LCORSP(I)	EXTGIN	10
	IF (J) 100,100,50	EXTGIN	11
	50 PINT(J) = PINTF(U(I),I)	EXTGIN	12
100	CONTINUE	EXTGIN	13
	RETURN	EXTGIN	14
30	END	EXTGIN	15

1	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
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
10	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
15	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
20	9/CONVER/ EPSI ,APSI ,VTEST ,NSTEPQ ,NFCN ,NFCNMX	COMMONU	29
	A/CARD / CWORD ,CWORD2 ,CWORD3 ,WORD7(7)	COMMONU	30
	B/MINIMA/ AMIN ,UP ,NEWMIN ,ITAU ,SIGMA	COMMONU	31
	K/COMMND/ NNAME ,CNAME(25) ,CNAM2(25) ,CNAM3(25)	COMMONU	32
	DIMENSION V1(2),YY(15)	FIXPAR	10
25	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
30	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
35	GO TO 300	FIXPAR	25
	80 LC = LCORSP(I)	FIXPAR	26
	IT = LC	FIXPAR	27
	LCORSP(I) = 0	FIXPAR	28
	ILAX = I	FIXPAR	29
40	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
45	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
50	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
55	DIRIN(LC) = DIRIN(LC+1)	FIXPAR	45
100	CONTINUE	FIXPAR	46
	IF (ISW(2) .GT. 1) GO TO 250	FIXPAR	47

	ISW(2) = 0	FIXPAR	48
	GO TO 300	FIXPAR	49
60	C . INT PARAM NO. SPECIFIED	FIXPAR	50
	150 CONTINUE	FIXPAR	51
	DO 200 IQ= 1, NU	FIXPAR	52
	IF (LCORSP(IQ) .NE. I2) GO TO 200	FIXPAR	53
	I = IQ	FIXPAR	54
65	GO TO 60	FIXPAR	55
	200 CONTINUE	FIXPAR	56
	GO TO 70	FIXPAR	57
	C REMOVE ONE ROW AND ONE COLUMN FROM VARIANCE MATRIX	FIXPAR	58
	250 KON = 0	FIXPAR	59
70	IF (NPAR .LE. 0) GO TO 300	FIXPAR	60
	KON2 = 0	FIXPAR	61
	MPAR = NPAR + 1	FIXPAR	62
	DO 260 I= 1, MPAR	FIXPAR	63
	260 YY(I)=V(I,IT)	FIXPAR	64
75	DO 294 I= 1, MPAR	FIXPAR	65
	IF (I.EQ.IT) GO TO 294	FIXPAR	66
	KON2 = KON2 + 1	FIXPAR	67
	DO 292 J= 1, MPAR	FIXPAR	68
	IF (J .EQ. IT) GO TO 292	FIXPAR	69
80	KON = KON + 1	FIXPAR	70
	V1(KON)=V(J,I) - YY(J)*YY(I)/YY(IT)	FIXPAR	71
	292 CONTINUE	FIXPAR	72
	KON = MAXINT*KON2	FIXPAR	73
	294 CONTINUE	FIXPAR	74
85	C CHECK FOR WELL-BEHAVED FINAL MATRIX	FIXPAR	75
	DO 295 I= 1, NPAR	FIXPAR	76
	IF (V(I,I) .LE. 0.) GO TO 296	FIXPAR	77
	DO 295 J= 1, NPAR	FIXPAR	78
	IF (I .EQ. J) GO TO 295	FIXPAR	79
90	IF (V(I,J)**2 .GE. V(I,I)*V(J,J)) V(I,J) = 0.	FIXPAR	80
	295 CONTINUE	FIXPAR	81
	GO TO 300	FIXPAR	82
	296 ISW(2) = 0	FIXPAR	83
	WRITE (ISYSWR, 501)	FIXPAR	84
95	300 RETURN	FIXPAR	85
	500 FORMAT (28H0ERROR IN FIXPAR. PARAMETERI3, 18H WAS NOT VARIABLE	FIXPAR	86
	1./)	FIXPAR	87
	501 FORMAT (72H0COVARIANCE MATRIX WAS ILL-CONDITIONED AND HAS BEEN D	FIXPAR	88
	1ESTROYED BY FIXPAR./)	FIXPAR	89
100	END	FIXPAR	90

1	FUNCTION FTIME(BIDON)	FIXPAR	91
	CC GIVES THE ELAPSED JOB TIME IN FLOATING-POINT MINUTES	FIXPAR	92
	CC BY CALLING SOME INSTALLATION-DEPENDENT SUBROUTINE	FIXPAR	93
	CC	FIXPAR	94
5	CALL SECOND(SEC)	FIXPAR	95
	FTIME=SEC	FIXPAR	96
	RETURN	FIXPAR	97
	END	FIXPAR	98

```

1      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
5      CC      OFF-DIAGONAL ELEMENTS TO ZERO IN ATTEMPT TO FORCE POSITIVENESS      HESSE      6
      CC      HESSE      7
      COMMON
      1/MINERR/ ERP(30)      ,ERN(30)      COMMONU      2
      2/PARINT/ X(15)      ,XT(15)      ,DIRIN(15)      ,MAXINT      ,NPAR      COMMONU      4
10     3/PAREXT/ U(30)      ,NAM(30)      ,WERR(30)      ,MAXEXT      ,NU      COMMONU      5
      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
15     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
20     9/CONVER/ EPSI ,APSI      ,VTEST      ,NSTEPQ      ,NFCN      ,NFCNMX      COMMONU      29
      A/CARD / CWORD      ,CWORD2      ,CWORD3      ,WORD7(7)      COMMONU      30
      B/MINIMA/ AMIN      ,UP      ,NEWMIN      ,ITAU      ,SIGMA      COMMONU      31
      K/COMMND/ NNAME      ,CNAME(25)      ,CNAM2(25)      ,CNAM3(25)      COMMONU      32
      DIMENSION YY(15),GY(30)      HESSE      10
25     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
30     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
35     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     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
45     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
50     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
55     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
60		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
65	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
70	C	FIX A PARAMETER IF G = G2 = 0.0	HESSE	60
		IF (ITAU .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
75		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
80		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
85	C	OFF-DIAGONAL ELEMENTS	HESSE	75
		IF (NPAR .EQ. 1) GO TO 214	HESSE	76
		NPARAM1 = NPAR - 1	HESSE	77
		DO 200 I= 1, NPARAM1	HESSE	78
		IP1 = I + 1	HESSE	79
90		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
95		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
100		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
105		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
110		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

115	WRITE (ISYSWR,520)	HESSE	105
	C DIAGONAL MATRIX ONLY . . .	HESSE	106
	216 WRITE (ISYSWR,540)	HESSE	107
	ISW(2) = 1	HESSE	108
	DO 220 I= 1, NPAR	HESSE	109
120	DO 218 J= 1, NPAR	HESSE	110
	218 V(I,J) = 0.	HESSE	111
	220 V(I,I) = 1.0/G2(I)	HESSE	112
	MDIAG = 0	HESSE	113
	GO TO 223	HESSE	114
125	222 WRITE (ISYSWR, 480)	HESSE	115
	ISW(2) = 2	HESSE	116
	C CALCULATE E D M	HESSE	117
	223 DO 225 I= 1, NPAR	HESSE	118
	DO 225 J= 1, NPAR	HESSE	119
130	225 V(I,J) = 2.0 * V(I,J)	HESSE	120
	SIGMA = 0.	HESSE	121
	DO 250 I= 1, NPAR	HESSE	122
	IF (V(I,I) .GT. 0.) GO TO 228	HESSE	123
	WRITE (ISYSWR,510) I	HESSE	124
135	MDIAG = 1	HESSE	125
	228 R = 0.	HESSE	126
	DO 240 J= 1, NPAR	HESSE	127
	IF (I .EQ. J) GO TO 230	HESSE	128
	IF (V(I,J)**2 .LT. ABS(V(I,I)*V(J,J))) GO TO 230	HESSE	129
140	WRITE (ISYSWR, 470) I,J	HESSE	130
	V(I,J) = 0.	HESSE	131
	V(J,I) = 0.	HESSE	132
	230 CONTINUE	HESSE	133
	240 R = R + V(I,J) * G(J)	HESSE	134
145	250 SIGMA = SIGMA + 0.5 *R *G(I)	HESSE	135
	IF (MDIAG .EQ. 1) GO TO 390	HESSE	136
	IF (SIGMA .GT. 0.) GO TO 400	HESSE	137
	WRITE (ISYSWR,530)	HESSE	138
	GO TO 216	HESSE	139
150	390 ISW(2) = 0	HESSE	140
	400 RETURN	HESSE	141
	460 FORMAT(10H0PARAMETER I3,2H, A10, 16H, HAS BEEN FIXED 9X 19HFIRS	HESSE	142
	1T DERIVATIVE IS E11.3/1X40(1H*),8X 20HSECOND DERIVATIVE IS E11.3	HESSE	143
	2/)	HESSE	144
155	470 FORMAT (71H COVARIANCE MATRIX NOT POSITIVE-DEFINITE. FAULTY E	HESSE	145
	1LEMENT IN POSITION2I3)	HESSE	146
	480 FORMAT (35H SECOND DERIVATIVE MATRIX INVERTED)	HESSE	147
	490 FORMAT (76H0CALL LIMIT IN HESSE. OFF-DIAGONAL ELEMENTS CALCULAT	HESSE	148
	1ED ONLY UP TO POSITION2I3/)	HESSE	149
160	500 FORMAT (37H START SECOND DERIVATIVE CALCULATION)	HESSE	150
	510 FORMAT (17H DIAGONAL ELEMENT I5, 20H IS ZERO OR NEGATIVE)	HESSE	151
	520 FORMAT (23H MATRIX INVERSION FAILS)	HESSE	152
	530 FORMAT (29H MATRIX NOT POSITIVE-DEFINITE)	HESSE	153
	540 FORMAT (30H ONLY DIAGONAL MATRIX PRODUCED)	HESSE	154
165	END	HESSE	155

	XI = X(I)	IMPROV	43
	X(I)= XI-DIRIN(I)*(RGEN(I)-0.5)	IMPROV	44
60	Y(I) = CALFCN(X)	IMPROV	45
	IF (Y(I) .GE. AMIN) GO TO 7	IMPROV	46
	AMIN = Y(I)	IMPROV	47
	JL = I	IMPROV	48
	7 IF (Y(I) .LE. AMAX) GO TO 8	IMPROV	49
65	AMAX = Y(I)	IMPROV	50
	JH = I	IMPROV	51
	8 CONTINUE	IMPROV	52
	DO 10 J= 1, NPAR	IMPROV	53
	10 P(J,I) = X(J)	IMPROV	54
70	P(I,NPAR1) = XI	IMPROV	55
	15 X(I) = XI	IMPROV	56
	SIGMA = AMIN	IMPROV	57
	SIG2 = SIGMA	IMPROV	58
	C START MAIN LOOP	IMPROV	59
75	50 CONTINUE	IMPROV	60
	IF (AMIN .LT. 0.) GO TO 95	IMPROV	61
	IF (ISW(2) .LT. 2) GO TO 280	IMPROV	62
	EP = 0.1*AMIN	IMPROV	63
	IF (SIG2 .LT. EP .AND. SIGMA.LT.EP) GO TO 100	IMPROV	64
80	SIG2 = SIGMA	IMPROV	65
	IF ((NFCN-NPFN) .GT. NFCNMX) GO TO 300	IMPROV	66
	C CALCULATE NEW POINT * BY REFLECTION	IMPROV	67
	DO 60 I= 1, NPAR	IMPROV	68
	PB = 0.	IMPROV	69
85	DO 59 J= 1, NPAR1	IMPROV	70
	59 PB = PB + WG * P(I,J)	IMPROV	71
	PBAR(I) = PB - WG * P(I,JH)	IMPROV	72
	60 PSTAR(I)=(1.+ALPHA)*PBAR(I)-ALPHA*P(I,JH)	IMPROV	73
	YSTAR = CALFCN(PSTAR)	IMPROV	74
90	IF(YSTAR.GE.AMIN) GO TO 70	IMPROV	75
	C POINT * BETTER THAN JL, CALCULATE NEW POINT **	IMPROV	76
	DO 61 I=1,NPAR	IMPROV	77
	61 PSTST(I)=GAMMA*PSTAR(I)+(1.-GAMMA)*PBAR(I)	IMPROV	78
	YSTST = CALFCN(PSTST)	IMPROV	79
95	66 IF (YSTST .LT. Y(JL)) GO TO 67	IMPROV	80
	CALL RAZZIA(YSTAR,PSTAR)	IMPROV	81
	GO TO 50	IMPROV	82
	67 CALL RAZZIA(YSTST,PSTST)	IMPROV	83
	GO TO 50	IMPROV	84
100	C POINT * IS NOT AS GOOD AS JL	IMPROV	85
	70 IF (YSTAR .GE. Y(JH)) GO TO 73	IMPROV	86
	JHOLD = JH	IMPROV	87
	CALL RAZZIA(YSTAR,PSTAR)	IMPROV	88
	IF (JHOLD .NE. JH) GO TO 50	IMPROV	89
105	C CALCULATE NEW POINT **	IMPROV	90
	73 DO 74 I=1,NPAR	IMPROV	91
	74 PSTST(I)=BETA*P(I,JH)+(1.-BETA)*PBAR(I)	IMPROV	92
	YSTST = CALFCN(PSTST)	IMPROV	93
	IF(YSTST.GT.Y(JH)) GO TO 5	IMPROV	94
110	C POINT ** IS BETTER THAN JH	IMPROV	95
	IF (YSTST .LT. AMIN) GO TO 67	IMPROV	96
	CALL RAZZIA(YSTST,PSTST)	IMPROV	97
	GO TO 50	IMPROV	98
	C END MAIN LOOP	IMPROV	99

115	95 WRITE (ISYSWR,1000)	IMPROV	100
	REG = 0.1	IMPROV	101
	C ASK IF POINT IS NEW	IMPROV	102
	100 CALL INTOEX(X)	IMPROV	103
	CALL FCN(NPAR,G,AMIN,U,4)	IMPROV	104
120	NFCN = NFCN + 1	IMPROV	105
	DO 120 I= 1, NPAR	IMPROV	106
	DIRIN(I) = REG*G2(I)	IMPROV	107
	IF (ABS(X(I)-XT(I)) .GT. DIRIN(I)) GO TO 150	IMPROV	108
	120 CONTINUE	IMPROV	109
125	GO TO 230	IMPROV	110
	150 NFCNMX = NFCNMX + NPFN - NFCN	IMPROV	111
	NPFN = NFCN	IMPROV	112
	CALL SIMPLX	IMPROV	113
	IF (AMIN .GE. APSI) GO TO 325	IMPROV	114
130	DO 220 I= 1, NPAR	IMPROV	115
	DIRIN(I) = 0.1 *G2(I)	IMPROV	116
	IF (ABS(X(I)-XT(I)) .GT. DIRIN(I)) GO TO 250	IMPROV	117
	220 CONTINUE	IMPROV	118
	230 IF (AMIN .LT. APSI) GO TO 350	IMPROV	119
135	GO TO 325	IMPROV	120
	C TRULY NEW MINIMUM	IMPROV	121
	250 NEWMIN = 1	IMPROV	122
	ISW(2) = 0	IMPROV	123
	ITAU = 0	IMPROV	124
140	NFCNMX = NFCNMX + NPFN - NFCN	IMPROV	125
	WRITE (ISYSWR,1030)	IMPROV	126
	RETURN	IMPROV	127
	C . . . RETURN TO PREVIOUS REGION	IMPROV	128
	280 WRITE (ISYSWR,1020)	IMPROV	129
145	ISW(2) = 0	IMPROV	130
	GO TO 325	IMPROV	131
	300 ISW(1) = 1	IMPROV	132
	325 DO 330 I= 1, NPAR	IMPROV	133
	DIRIN(I) = 0.01*G2(I)	IMPROV	134
150	330 X(I) = XT(I)	IMPROV	135
	AMIN = APSI	IMPROV	136
	SIGMA = SIGSAV	IMPROV	137
	350 CALL INTOEX(X)	IMPROV	138
	WRITE (ISYSWR,1010)	IMPROV	139
155	IF (ISW(2) .LT. 2) GO TO 380	IMPROV	140
	IF (LOOP .LT. NLOOP .AND. ISW(1) .LT. 1) GO TO 3	IMPROV	141
	ISW(2) = 3	IMPROV	142
	380 CALL MPRINT (1,AMIN)	IMPROV	143
	RETURN	IMPROV	144
160	1000 FORMAT (54H AN IMPROVEMENT ON THE PREVIOUS MINIMUM HAS BEEN FOUN	IMPROV	145
	1D)	IMPROV	146
	1010 FORMAT (51H IMPROVE HAS RETURNED TO REGION OF ORIGINAL MINIMUM)	IMPROV	147
	1020 FORMAT (44H0COVARIANCE MATRIX WAS NOT POSITIVE-DEFINITE)	IMPROV	148
	1030 FORMAT (38H0IMPROVE HAS FOUND A TRULY NEW MINIMUM/1H 37(1H*)//)	IMPROV	149
165	1040 FORMAT (18H0START ATTEMPT NO.I2, 20H TO FIND NEW MINIMUM)	IMPROV	150
	END	IMPROV	151

1	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
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
10	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
15	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
20	B/MINIMA/ AMIN ,UP ,NEWMIN ,ITAU ,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
25	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
30	GO TO 100	INTOX	15
	80 U(I) = PINT(J)	INTOX	16
100	CONTINUE	INTOX	17
	RETURN	INTOX	18
	END	INTOX	19

```

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

```

	500 RETURN		MATOUT	48
	600 FORMAT(/11X 26HINTERNAL COVARIANCE MATRIX)		MATOUT	49
60	610 FORMAT (1H+43X 28H LAST FRACTIONAL CHANGE WAS F10.6)		MATOUT	50
	620 FORMAT (1H 9X11E10.3)		MATOUT	51
	630 FORMAT (1H)		MATOUT	52
	650 FORMAT (11X 24HCORRELATION COEFFICIENTS)		MATOUT	53
	660 FORMAT (8X,I3,18F6.3/(20F6.3))		MATOUT	54
65	670 FORMAT (64X 18HGLOBAL CORRELATION/ 54X 25HPARAMETER	COEFFI	MATOUT	55
	1CIENT)		MATOUT	56
	680 FORMAT (47X,I5,2X,A10,F13.5)		MATOUT	57
	690 FORMAT (8X,4HINT.,I4,17I6)		MATOUT	58
	END		MATOUT	59

```

1          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
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
10         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
15         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
20         A/CARD / CWORD      ,CWORD2      ,CWORD3      ,WORD7(7)   COMMONU  30
          B/MINIMA/ AMIN       ,UP          ,NEWMIN     ,ITAU      ,SIGMA     COMMONU  31
          K/COMMND/ NNAME      ,CNAME(25)   ,CNAM2(25)   ,CNAM3(25)   COMMONU  32
          COMMON /MATCH2/ AA(30),BB(30),KBTS,SMINCAL,
          1          DWORD(10),DWORD2(10),DWORD3(10),DWORD7(7,10),MINCOM
25         LOGICAL SMINCAL                                             MATCH4   2
          C          . INITIALIZE NEW DATA BLOCK . . . . . MIDATA    9
          NBLOCK = NBLOCK + 1                                           MIDATA  10
          VERSN = 1.75                                                  MIDATA  11
          WRITE (ISYSWR,1004) VERSN,NBLOCK                             MIDATA  12
30         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
35         NPFIX = 0                                                  MIDATA  18
          NINT = 0                                                    MIDATA  19
          NU = 0                                                       MIDATA  20
          NPAR = 0                                                     MIDATA  21
          IFATAL = 0                                                  MIDATA  22
40         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
45         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
50         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
55         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	MIDATA	41
	WRITE (ISYSWR,1002) K,NAMK,UK,WK,A,B	MIDATA	42
60	GO TO 160	MIDATA	43
	115 CONTINUE	MIDATA	44
	IF (WK .GT. 0.0) GO TO 122	MIDATA	45
	C . . . FIXED PARAMETER	MIDATA	46
	WRITE (ISYSWR, 1002) K,NAMK,UK	MIDATA	47
65	LCODE(K) = 0	MIDATA	48
	GO TO 160	MIDATA	49
	C . . . VARIABLE PARAMETER	MIDATA	50
	122 WRITE (ISYSWR, 1002) K,NAMK,UK,WK,A,B	MIDATA	51
	NINT = NINT + 1	MIDATA	52
70	IF (A) 140,130,140	MIDATA	53
	130 IF (B) 140,135,140	MIDATA	54
	135 LCODE(K) = 1	MIDATA	55
	GO TO 160	MIDATA	56
	140 IF (B-A) 145,142,150	MIDATA	57
75	142 IFATAL = IFATAL + 1	MIDATA	58
	WRITE (ISYSWR,1010)	MIDATA	59
	GO TO 150	MIDATA	60
	145 SAV = B	MIDATA	61
	B = A	MIDATA	62
80	A = SAV	MIDATA	63
	150 ALIM(K) = A	MIDATA	64
	BLIM(K) = B	MIDATA	65
	LCODE(K) = 4	MIDATA	66
	IF ((B-U(K))*(U(K)-A)) 153,155,160	MIDATA	67
85	153 IFATAL = IFATAL + 1	MIDATA	68
	WRITE (ISYSWR,1011)	MIDATA	69
	GO TO 160	MIDATA	70
	155 WRITE (ISYSWR,1006)	MIDATA	71
	160 CONTINUE	MIDATA	72
90	200 CONTINUE	MIDATA	73
	C . . . END PARAMETER CARDS	MIDATA	74
	C . . . STOP IF FATAL ERROR	MIDATA	75
	250 WRITE (ISYSWR,1005)	MIDATA	76
	IF (NINT .LE. MAXINT) GO TO 253	MIDATA	77
95	WRITE (ISYSWR,1008) NINT,MAXINT	MIDATA	78
	IFATAL = IFATAL + 1	MIDATA	79
	253 IF (IFATAL .LE. 0) GO TO 280	MIDATA	80
	WRITE (ISYSWR,1013) IFATAL	MIDATA	81
	STOP	MIDATA	82
100	C CALCULATE STEP SIZES DIRIN	MIDATA	83
	280 NPAR = 0	MIDATA	84
	DO 300 K= 1, NU	MIDATA	85
	IF (LCODE(K) .LE. 0) GO TO 300	MIDATA	86
	NPAR = NPAR + 1	MIDATA	87
105	LCORSP(K) = NPAR	MIDATA	88
	SAV = U(K)	MIDATA	89
	X(NPAR) = PINTF(SAV,K)	MIDATA	90
	XT(NPAR) = X(NPAR)	MIDATA	91
	SAV2 = SAV + WERR(K)	MIDATA	92
110	VPLU = PINTF(SAV2,K) - X(NPAR)	MIDATA	93
	SAV2 = SAV - WERR(K)	MIDATA	94
	VMINU = PINTF(SAV2,K) - X(NPAR)	MIDATA	95
	DIRIN(NPAR) = 0.5 * (ABS(VPLU) +ABS(VMINU))	MIDATA	96
	300 CONTINUE	MIDATA	97

115	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
120	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
125	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 FORMÁT (46H0 TOO MANY VARIABLE PARAMETERS. YOU REQUESTI5/ 49	MIDATA	117
130	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
135	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

```

1      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
5      CC                                               MIGRAD  6
          COMMON
          1/MINERR/ ERP(30)      ,ERN(30)                COMMONU  2
          2/PARINT/ X(15)       ,XT(15)      ,DIRIN(15)  ,MAXINT      ,NPAR  COMMONU  4
          3/PAREXT/ U(30)       ,NAM(30)      ,WERR(30)  ,MAXEXT      ,NU   COMMONU  5
10     4/LIMITS/ ALIM(30)     ,BLIM(30)     ,LCODE(30)  ,LCORSP(30) ,LIMSET  COMMONU  6
          5/VARIAN/ V(15,15)
          7/FIX / IPFIX(15)     ,XS(15)      ,XTS(15)   ,WTS(15)   ,NPFIX  COMMONU  8
          C/CASC/ JH, JL, Y(16)
          F/DERIVA/ G(30)      ,G2(30)                COMMONU  9
15     G/SIMVEC/ P(15,16)     ,PSTAR(15)  ,PSTST(15)  ,PBAR(15)  ,PRHO(15) COMMONU 10
          J/VARIAT/ VT(15,15)
          6/UNIT / ISYSRD      ,ISYSWR      ,ISYSPU                COMMONU 11
          8/TITLE / TITLE(13)  ,DATE(2)   ,ISW(7)    ,NBLOCK                COMMONU 12
          9/CONVER/ EPSI ,APSI ,VTEST      ,NSTEPQ   ,NFCN      ,NFCNMX  COMMONU 13
20     A/CARD / CWORD        ,CWORD2      ,CWORD3    ,WORD7(7)  COMMONU 14
          B/MINIMA/ AMIN      ,UP           ,NEWMIN   ,ITAUT      ,SIGMA  COMMONU 15
          K/COMMND/ NNAME     ,CNAME(25)   ,CNAM2(25) ,CNAM3(25)  COMMONU 16
          DIMENSION GS(30) , R(15),XXS(15), FLNU(15), VG(15), VII(15)
          DATA SLAMIN,SLAMAX,TLAMIN,TLAMAX/0.2, 3.0, 0.05, 6.0/
25     IF (NPAR .LE. 0) RETURN                                MIGRAD 17
          ISWTR = ISW(5) - ITAUT                                MIGRAD 18
          NPFN = NFCN                                           MIGRAD 19
          PARN=NPAR                                              MIGRAD 20
          RHO2 = 10.*APSI                                         MIGRAD 21
30     ROSTOP = 1.0E-5 * APSI                                     MIGRAD 22
          TRACE=1.                                               MIGRAD 23
          IFLAG=4                                                MIGRAD 24
          IF (ISW(3) .EQ. 1) IFLAG = 2                            MIGRAD 25
          FS = AMIN                                               MIGRAD 26
35     IF (ITAUT .LT. 1) WRITE (ISYSWR,470) ROSTOP, APSI, VTEST
          GO TO 2                                                MIGRAD 27
          1 WRITE (ISYSWR,520)                                    MIGRAD 28
C          . . . . . STEP SIZES DIRIN . . . . .                MIGRAD 29
          2 NPARD = NPAR                                          MIGRAD 30
40     DO 3 I= 1, NPARD                                          MIGRAD 31
          D = 0.02* ABS(DIRIN(I))                                MIGRAD 32
          IF (ISW(2) .GE. 1) D = 0.02* SQRT(ABS(V(I,I))*UP)      MIGRAD 33
          IF (D .LT. 1.0E-8 *ABS(X(I))) D = 1.0E-8 * X(I)      MIGRAD 34
          3 DIRIN(I) = D                                          MIGRAD 35
45     C          . . . . . STARTING GRADIENT                    MIGRAD 36
          NTRY = 0                                               MIGRAD 37
          4 NEGG2 = 0                                             MIGRAD 38
          DO 10 ID= 1, NPARD                                      MIGRAD 39
          I = ID + NPARD - NPARD                                  MIGRAD 40
50     D = DIRIN(I)                                              MIGRAD 41
          XTF = X(I)                                              MIGRAD 42
          X(I) = XTF + D                                          MIGRAD 43
          CALL INTOEX(X)                                          MIGRAD 44
          CALL FCN(NPAR,G,FS1,U,4)                                MIGRAD 45
55     NFCN = NFCN + 1                                           MIGRAD 46
          X(I) = XTF - D                                          MIGRAD 47
          CALL INTOEX(X)                                          MIGRAD 48

```

	CALL FCN(NPAR,G,FS2,U,4)	MIGRAD	48
	NFCN = NFCN + 1	MIGRAD	49
60	X(I) = XTF	MIGRAD	50
	GS(I) = (FS1-FS2)/(2.0 * D)	MIGRAD	51
	G2(I) = (FS1 + FS2 - 2.0*AMIN) / D**2	MIGRAD	52
	IF (G2(I) .GT. 1.0E-30) GO TO 10	MIGRAD	53
	C . . . SEARCH IF G2 .LE. 0. . .	MIGRAD	54
65	WRITE (ISYSWR,520)	MIGRAD	55
	NEGG2 = NEGG2 + 1	MIGRAD	56
	NTRY = NTRY + 1	MIGRAD	57
	IF (NTRY .GT. 4) GO TO 230	MIGRAD	58
	D = 50.*ABS(DIRIN(I))	MIGRAD	59
70	XBEG = XTF	MIGRAD	60
	IF (GS(I) .LT. 0.) DIRIN(I) = -DIRIN(I)	MIGRAD	61
	KG = 0	MIGRAD	62
	NF = 0	MIGRAD	63
	NS = 0	MIGRAD	64
75	5 X(I) = XTF + D	MIGRAD	65
	CALL INTOEX(X)	MIGRAD	66
	CALL FCN(NPAR,G,F,U,4)	MIGRAD	67
	NFCN = NFCN + 1	MIGRAD	68
	IF (F .LE. AMIN) GO TO 6	MIGRAD	69
80	C FAILURE	MIGRAD	70
	IF (KG .EQ. 1) GO TO 8	MIGRAD	71
	KG = -1	MIGRAD	72
	NF = NF + 1	MIGRAD	73
	D = -0.4*D	MIGRAD	74
85	IF (NF .LT. 10) GO TO 5	MIGRAD	75
	D = 1000.*D	MIGRAD	76
	GO TO 7	MIGRAD	77
	C SUCCESS	MIGRAD	78
90	6 XTF = X(I)	MIGRAD	79
	D = 3.0*D	MIGRAD	80
	AMIN = F	MIGRAD	81
	KG = 1	MIGRAD	82
	NS = NS + 1	MIGRAD	83
	IF (NS .LT. 10) GO TO 5	MIGRAD	84
95	IF (AMIN .LT. FS) GO TO 8	MIGRAD	85
	D = 0.001*D	MIGRAD	86
	7 XTF = XBEG	MIGRAD	87
	G2(I) = 1.0	MIGRAD	88
	NEGG2 = NEGG2 - 1	MIGRAD	89
100	8 X(I) = XTF	MIGRAD	90
	DIRIN(I) = 0.1*D	MIGRAD	91
	FS = AMIN	MIGRAD	92
	10 CONTINUE	MIGRAD	93
	IF (NEGG2 .GE. 1) GO TO 4	MIGRAD	94
105	NTRY = 0	MIGRAD	95
	MATGD = 1	MIGRAD	96
	C DIAGONAL MATRIX	MIGRAD	97
	IF (ISW(2) .GT. 1) GO TO 15	MIGRAD	98
110	11 NTRY = 1	MIGRAD	99
	MATGD = 0	MIGRAD	100
	DO 13 I= 1, NPAR	MIGRAD	101
	DO 12 J= 1, NPAR	MIGRAD	102
	12 V(I,J) = 0.	MIGRAD	103
	13 V(I,I) = 2.0/G2(I)	MIGRAD	104

115	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
120		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
125		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
130		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
135		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
140		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
145		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
150		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
155		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
160		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
165		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
170		IF (TLAM .LT. TLAMIN) TLAM=TLAMIN	MIGRAD	160
		GO TO 50	MIGRAD	161

	48 TLAM = TLAMAX	MIGRAD	162
	50 CONTINUE	MIGRAD	163
	DO 51 I= 1, NPAR	MIGRAD	164
175	51 X(I) = XXS(I)+TLAM*DIRIN(I)	MIGRAD	165
	CALL INTOEX(X)	MIGRAD	166
	CALL FCN(NPAR,G,F3,U,4)	MIGRAD	167
	NFCN = NFCN + 1	MIGRAD	168
	IF (F.GE.AMIN .AND. F2.GE.AMIN .AND. F3.GE.AMIN) GO TO 200	MIGRAD	169
180	IF (F .LT. F2 .AND. F .LT. F3) GO TO 61	MIGRAD	170
	IF (F2 .LT. F3) GO TO 58	MIGRAD	171
	55 F = F3	MIGRAD	172
	SLAM = TLAM	MIGRAD	173
	GO TO 65	MIGRAD	174
185	58 F = F2	MIGRAD	175
	GO TO 65	MIGRAD	176
	61 SLAM = 1.0	MIGRAD	177
	65 DO 67 I= 1, NPAR	MIGRAD	178
	DIRIN(I) = DIRIN(I)*SLAM	MIGRAD	179
190	67 X(I) = XXS(I) + DIRIN(I)	MIGRAD	180
	70 AMIN = F	MIGRAD	181
	ISW(2) = 2	MIGRAD	182
	IF (SIGMA+FS-AMIN .LT. ROSTOP) GO TO 170	MIGRAD	183
	IF (SIGMA+RHO2+FS-AMIN .GT. APSI) GO TO 75	MIGRAD	184
195	IF (TRACE .LT. VTEST) GO TO 170	MIGRAD	185
	75 CONTINUE	MIGRAD	186
	IF (NFCN-NPFN .GE. NFCNMX) GO TO 190	MIGRAD	187
	ITER = ITER + 1	MIGRAD	188
	IF (ISWTR.GE. 3 .OR.(ISWTR.EQ. 2 .AND. MOD(ITER,10) .EQ.1))	MIGRAD	189
200	* CALL MPRINT(0,AMIN)	MIGRAD	190
	C . . . GET GRADIENT AND SIGMA .	MIGRAD	191
	IF (ISW(3) .NE. 1) GO TO 80	MIGRAD	192
	CALL INTOEX(X)	MIGRAD	193
	CALL FCN(NPAR,G,AMIN,U,IFLAG)	MIGRAD	194
205	NFCN = NFCN + 1	MIGRAD	195
	80 CALL DERIVE(G,G2)	MIGRAD	196
	RHO2 = SIGMA	MIGRAD	197
	SIGMA = 0.	MIGRAD	198
	GVG = 0.	MIGRAD	199
210	DELGAM = 0.	MIGRAD	200
	DO 100 I= 1, NPAR	MIGRAD	201
	RI = 0.	MIGRAD	202
	VGI = 0.	MIGRAD	203
	DO 90 J= 1, NPAR	MIGRAD	204
215	VGI = VGI + V(I,J)*(G(J)-GS(J))	MIGRAD	205
	90 RI = RI + V(I,J) *G (J)	MIGRAD	206
	R(I) = RI * 0.5	MIGRAD	207
	VG(I) = VGI*0.5	MIGRAD	208
	GAMI = G(I) - GS(I)	MIGRAD	209
220	GVG = GVG + GAMI*VG(I)	MIGRAD	210
	DELGAM = DELGAM + DIRIN(I)*GAMI	MIGRAD	211
	100 SIGMA = SIGMA + G(I)*R(I)	MIGRAD	212
	IF (SIGMA .LT. 0.) GO TO 1	MIGRAD	213
	IF (GVG .LE. 0.) GO TO 105	MIGRAD	214
225	IF (DELGAM .LE. 0.) GO TO 105	MIGRAD	215
	GO TO 107	MIGRAD	216
	105 IF (SIGMA .LT. 0.1*ROSTOP) GO TO 170	MIGRAD	217
	GO TO 1	MIGRAD	218

	107 CONTINUE	MIGRAD	219
230	C UPDATE COVARIANCE MATRIX	MIGRAD	220
	TRACE=0.	MIGRAD	221
	DO 120 I= 1, NPAR	MIGRAD	222
	VII(I) = V(I,I)	MIGRAD	223
	DO 120 J=1,NPAR	MIGRAD	224
235	D = DIRIN(I)*DIRIN(J)/DELGAM - VG(I)*VG(J)/GVG	MIGRAD	225
	120 V(I,J) = V(I,J) + 2.0*D	MIGRAD	226
	IF (DELGAM .LE. GVG) GO TO 135	MIGRAD	227
	DO 125 I= 1, NPAR	MIGRAD	228
240	125 FLNU(I) = DIRIN(I)/DELGAM - VG(I)/GVG	MIGRAD	229
	DO 130 I= 1, NPAR	MIGRAD	230
	DO 130 J= 1, NPAR	MIGRAD	231
	130 V(I,J) = V(I,J) + 2.0*GVG*FLNU(I)*FLNU(J)	MIGRAD	232
	135 CONTINUE	MIGRAD	233
	DO 140 I= 1, NPAR	MIGRAD	234
245	140 TRACE = TRACE + ((V(I,I)-VII(I))/(V(I,I)+VII(I)))**2	MIGRAD	235
	TRACE = SQRT(TRACE/PARN)	MIGRAD	236
	IF (ISWTR .GE. 4) CALL MATOUT(TRACE, 0)	MIGRAD	237
	CALL UCOPY(X,XXS,NPAR)	MIGRAD	238
	CALL UCOPY(G,GS,NPAR)	MIGRAD	239
250	FS = F	MIGRAD	240
	GO TO 24	MIGRAD	241
	C END MAIN LOOP	MIGRAD	242
	170 WRITE(ISYSWR,500)	MIGRAD	243
	ISW(2) = 3	MIGRAD	244
255	IF(ISWTR .GE. 0) CALL MPRINT(1-ITAU,AMIN)	MIGRAD	245
	ISWTR = ISWTR - 3*ITAU	MIGRAD	246
	IF (ISWTR .GT. 0) CALL MATOUT(TRACE,1)	MIGRAD	247
	IF (ITAU .GT. 0) GO TO 435	MIGRAD	248
	IF (MATGD .GT. 0) GO TO 435	MIGRAD	249
260	NPARGD = NPAR*(NPAR+5)/2	MIGRAD	250
	IF (NFCN-NPFN .GE. NPARGD) GO TO 435	MIGRAD	251
	WRITE (ISYSWR,180)	MIGRAD	252
	180 FORMAT (55H COVARIANCE MATRIX INACCURATE. MINUIT WILL RECALCULA	MIGRAD	253
	1TE)	MIGRAD	254
265	CALL HESSE	MIGRAD	255
	CALL MPRINT(1,AMIN)	MIGRAD	256
	CALL MATOUT(0.0, 1)	MIGRAD	257
	IF (ISW(2) .GE. 2) ISW(2) = 3	MIGRAD	258
	GO TO 435	MIGRAD	259
270	190 ISW(1) = 1	MIGRAD	260
	GO TO 230	MIGRAD	261
	200 WRITE (ISYSWR,650)	MIGRAD	262
	CALL UCOPY(XXS,X,NPAR)	MIGRAD	263
	ISW(2) = 1	MIGRAD	264
275	IF (SIGMA .LT. ROSTOP) GO TO 170	MIGRAD	265
	IF (MATGD .GT. 0) GO TO 2	MIGRAD	266
	230 WRITE (ISYSWR,510)	MIGRAD	267
	CALL INTOEX(X)	MIGRAD	268
	CALL MPRINT(1-ITAU, AMIN)	MIGRAD	269
280	ISWTR = ISW(5) - ITAU*3	MIGRAD	270
	IF (ISWTR .LT. 1) GO TO 435	MIGRAD	271
	IF (ISW(2) .LE. 1) GO TO 435	MIGRAD	272
	CALL MATOUT(TRACE,1)	MIGRAD	273
	435 RETURN	MIGRAD	274
285	470 FORMAT (37H START MIGRAD MINIMIZATION. BX 66HCONVERG	MIGRAD	275

1ENCE CRITERIA — ESTIMATED DISTANCE TO MINIMUM (EDM) .LT.E9.2/45 MIGRAD 276
2X 11HOR EDM .LT.E9.2, 46H AND FRACTIONAL CHANGE IN VARIANCE MATR MIGRAD 277
3IX .LT.E9.2) MIGRAD 278
500 FORMAT (34H MIGRAD MINIMIZATION HAS CONVERGED) MIGRAD 279
510 FORMAT (39H MIGRAD TERMINATED WITHOUT CONVERGENCE) MIGRAD 280
520 FORMAT (11X 43HCOVARIANCE MATRIX IS NOT POSITIVE-DEFINITE.) MIGRAD 281
650 FORMAT (34H MIGRAD FAILS TO FIND IMPROVEMENT) MIGRAD 282
END MIGRAD 283

290

	WRITE (ISYSWR,880) AMIN, F	MINNEW	44
	STOP	MINNEW	45
60	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

1	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
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
10	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
15	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
20	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 ,ITAU ,SIGMA	COMMONU	31
	K/COMMND/ NNAME ,CNAME(25) ,CNAM2(25) ,CNAM3(25)	COMMONU	32
25	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
30	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
35	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
40	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
45	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
50	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
55	KNT = KNT + 1	MINOS	45
	LMI(KNT) = I	MINOS	46
	IF (KNT .EQ. 30) GO TO 40	MINOS	47

	30	CONTINUE	MINOS	48
	40	CONTINUE	MINOS	49
60		IF (MARC .EQ. 1) WRITE (ISYSWR,811)	MINOS	50
		WRITE (ISYSWR, 810) (LMI(IZ3),IZ3=1,KNT)	MINOS	51
	C	. . . SAVE AND PREPARE START VALS	MINOS	52
		SIGSAV = SIGMA	MINOS	53
		TOLER = EPSI	MINOS	54
65		APSI = EPSI*0.5	MINOS	55
		ITAUT = 1	MINOS	56
		ABEST=AMIN	MINOS	57
		AIM = AMIN + UP	MINOS	58
		NSAVE = NFCNMX	MINOS	59
70		MPAR=MPAR	MINOS	60
		DO 130 I= 1, MPAR	MINOS	61
		XT(I) = X(I)	MINOS	62
		DO 125 J= 1, MPAR	MINOS	63
75	125	VT(I,J) = V(I,J)	MINOS	64
	130	CONTINUE	MINOS	65
		DO 135 I= 1, NU	MINOS	66
		ERP(I) = 0.	MINOS	67
		ERN(I) = 0.	MINOS	68
80	135	W(I) = WERR(I)	MINOS	69
		KNT = 0	MINOS	70
	C START MAIN LOOP . . .	MINOS	71
	150	KNT = KNT + 1	MINOS	72
		ISW(1) = 0	MINOS	73
		NLIMIT = NFCN + NSAVE	MINOS	74
85		IF (KNT .GT. 30) GO TO 590	MINOS	75
		IF (LMI(KNT) .LT. 1) GO TO 590	MINOS	76
		ILAX = LMI(KNT)	MINOS	77
		ERP(ILAX) = 0.	MINOS	78
		IT = LCORSP(ILAX)	MINOS	79
90		XTIT = XT(IT)	MINOS	80
		CALL INTOEX(XT)	MINOS	81
		UT = U(ILAX)	MINOS	82
		SPT(1,1) = UT	MINOS	83
		SPT(2,1) = AIM	MINOS	84
95		SPT(3,1) = 1H	MINOS	85
		SPT(1,2) = UT	MINOS	86
		SPT(2,2) = ABEST	MINOS	87
		SPT(3,2) = ASTAR	MINOS	88
		SPT(1,3) = UT	MINOS	89
100		SPT(2,3) = AIM + 0.2*(AIM-ABEST)	MINOS	90
		SPT(3,3) = ADOT	MINOS	91
		NSPT = 3	MINOS	92
		IF (LCODE(ILAX) .GT. 1) GO TO 160	MINOS	93
		ALIM(ILAX) = UT -100.*W(ILAX)	MINOS	94
105		BLIM(ILAX) = UT +100.*W(ILAX)	MINOS	95
	160	CONTINUE	MINOS	96
		XUNIT = SQRT(UP/VT(IT,IT))	MINOS	97
		MARC = 0	MINOS	98
		DO 162 I= 1, MPAR	MINOS	99
110		IF (I .EQ. IT) GO TO 162	MINOS	100
		MARC = MARC + 1	MINOS	101
		XDEV(MARC) = XUNIT*VT(IT,I)	MINOS	102
	162	CONTINUE	MINOS	103
		CALL FIXPAR (IT,1,ILAX)	MINOS	104

115	SIG = 1.0	MINOS	105
	ASIG = APOS	MINOS	106
	DULIM = BLIM(ILAX) - UT	MINOS	107
	IF(ISW(2).LT.1) GO TO 460	MINOS	108
	C . SIG=SIGN OF ERROR BEING CALCD	MINOS	109
120	165 WRITE (ISYSWR,806) ASIG,ILAX,NAM(ILAX)	MINOS	110
	ITER = 0	MINOS	111
	LIMSET = 0	MINOS	112
	DU1 = SIG*W(ILAX)	MINOS	113
	IF (ABS(DU1) .LE. DULIM) GO TO 180	MINOS	114
125	LIMSET = 1	MINOS	115
	DU1 = SIG * DULIM	MINOS	116
	IF (DULIM .LT. 1.0E-3*W(ILAX)) GO TO 440	MINOS	117
	180 U(ILAX) = UT + DU1	MINOS	118
	IF (NPAR .EQ. 0) GO TO 205	MINOS	119
130	FAC = DU1/W(ILAX)	MINOS	120
	DO 185 I= 1, NPAR	MINOS	121
	185 X(I) = XT(I) + FAC*XDEV(I)	MINOS	122
	200 CALL INTOEX (X)	MINOS	123
	205 WRITE (ISYSWR,801) ILAX,UT,DU1,U(ILAX)	MINOS	124
135	CALL FCN(NPAR,G,AMIN,U,4)	MINOS	125
	NFCN = NFCN + 1	MINOS	126
	NFCNMX = NLIMIT - NFCN	MINOS	127
	CALL MIGRAD	MINOS	128
	IF (AMIN .LT. ABEST) GO TO 650	MINOS	129
140	IF (ISW(1) .GE. 1) GO TO 450	MINOS	130
	IF (ISW(2) .GE. 2) GO TO 240	MINOS	131
	NFCNMX = NLIMIT - NFCN	MINOS	132
	CALL SIMPLX	MINOS	133
	IF (AMIN .LT. ABEST) GO TO 650	MINOS	134
145	IF (ISW(1) .GE. 1) GO TO 450	MINOS	135
	NFCNMX = NLIMIT - NFCN	MINOS	136
	CALL MIGRAD	MINOS	137
	IF (AMIN .LT. ABEST) GO TO 650	MINOS	138
	IF (ISW(1) .GE. 1) GO TO 450	MINOS	139
150	IF (ISW(2) .LT. 2) GO TO 460	MINOS	140
	240 CREM = AMIN - ABEST	MINOS	141
	NSPT = NSPT + 1	MINOS	142
	SPT(1,NSPT) = U(ILAX)	MINOS	143
	SPT(2,NSPT) = AMIN	MINOS	144
155	SPT(3,NSPT) = ASIG	MINOS	145
	IF (CREM .LE. 0.0) GO TO 650	MINOS	146
	SQUC = SQRT(UP/CREM)	MINOS	147
	IF (ABS(AMIN-AIM) .LT. TOLER) GO TO 400	MINOS	148
	C . . ANOTHER ITERATION NECESSARY	MINOS	149
160	ITER = ITER + 1	MINOS	150
	IF (ITER .GT. 6) GO TO 430	MINOS	151
	IF (ITER .EQ. 1) GO TO 270	MINOS	152
	C CHECK PREVIOUS ITERATION TO AVOID OSCILLATING	MINOS	153
	SQUC2 = SQUC	MINOS	154
165	IF ((SQUC2-1.0) * (SQUC1-1.0) .GT. 0.0) GO TO 270	MINOS	155
	SQUC = 0.65*SQUC2 + 0.35	MINOS	156
	SQUC11 = 1.0/SQUC1	MINOS	157
	IF ((SQUC11-SQUC)*(SQUC-1.0) .GT. 0.0) GO TO 270	MINOS	158
	WRITE (ISYSWR,260)	MINOS	159
170	260 FORMAT (44H MINOS IS HAVING TROUBLE WITH THIS PARAMETER)	MINOS	160
	SQUC = 0.5*SQUC11 + 0.5	MINOS	161

	270 CONTINUE	MINOS	162	
	SQUC1 = SQUC	MINOS	163	
	DU1 = DU1 * SQUC	MINOS	164	
175	IF (ABS(DU1) .LE. DULIM) GO TO 280	MINOS	165	
	IF (LIMSET .EQ. 1) GO TO 440	MINOS	166	
	LIMSET = 1	MINOS	167	
	DU1 = SIG*DULIM	MINOS	168	
	280 U(ILAX) = UT + DU1	MINOS	169	
180	DO 290 I= 1, NPAR	MINOS	170	
	290 X(I) = XT(I) + SQUC*(X(I)-XT(I))	MINOS	171	
	GO TO 200	MINOS	172	
	C	. ERROR SUCCESSFULLY CALCULATED	MINOS	173
	400 EROS = DU1 * SQUC	MINOS	174	
185	WRITE (ISYSWR,808) ASIG,ILAX,NAM(ILAX),EROS	MINOS	175	
	410 WRITE (ISYSWR,812)	MINOS	176	
	IF (SIG .GT. 0.) GO TO 420	MINOS	177	
	ERN(ILAX) = EROS	MINOS	178	
	GO TO 500	MINOS	179	
190	420 ERP(ILAX) = EROS	MINOS	180	
	SIG = -1.0	MINOS	181	
	ASIG = ANEG	MINOS	182	
	DULIM = UT - ALIM(ILAX)	MINOS	183	
	GO TO 165	MINOS	184	
195	C FAILURE RETURNS	MINOS	185
	430 WRITE (ISYSWR, 809)	MINOS	186	
	EROS = 0.	MINOS	187	
	GO TO 410	MINOS	188	
	440 WRITE (ISYSWR, 807) ASIG,ILAX,NAM(ILAX),DULIM	MINOS	189	
200	EROS = 0.	MINOS	190	
	GO TO 410	MINOS	191	
	450 WRITE (ISYSWR, 802) NSAVE	MINOS	192	
	GO TO 500	MINOS	193	
	460 WRITE (ISYSWR, 805)	MINOS	194	
205	C	. . . PARAMETER FINISHED. RESET V	MINOS	195
	500 CONTINUE	MINOS	196	
	CALL RESTOR(1)	MINOS	197	
	ISW(2) = 3	MINOS	198	
	DO 560 I= 1, MPAR	MINOS	199	
210	DO 550 J= 1, MPAR	MINOS	200	
	550 V(I,J) = VT(I,J)	MINOS	201	
	560 CONTINUE	MINOS	202	
	IF (NSPT .LT. 9) GO TO 150	MINOS	203	
	NSPT3 = NSPT - 3	MINOS	204	
215	WRITE (ISYSWR,813) ILAX,NAM(ILAX),NSPT3	MINOS	205	
	CALL PLTCON(NSPT,SPT)	MINOS	206	
	WRITE(ISYSWR,814) ILAX	MINOS	207	
	GO TO 150	MINOS	208	
	C PRINTOUT FINAL VALUES .	MINOS	209
220	590 CALL UCOPY(XT,X,MPAR)	MINOS	210	
	CALL INTOEX (XT)	MINOS	211	
	SIGMA = SIGSAV	MINOS	212	
	AMIN = ABEST	MINOS	213	
	CALL MPRINT(2,AMIN)	MINOS	214	
225	CALL MATOUT(0.0, 1)	MINOS	215	
	GO TO 700	MINOS	216	
	C	. . . NEW MINIMUM FOUND	MINOS	217
	650 NEWMIN = 1	MINOS	218	

	ISW(2) = 0	MINOS	219
230	CALL RESTOR(1)	MINOS	220
	CALL EXTOIN(X)	MINOS	221
	DO 670 I= 1, NPAR	MINOS	222
	670 DIRIN(I) = SQRT(VT(I,I)*UP)	MINOS	223
	700 RETURN	MINOS	224
235	801 FORMAT (10H0PARAMETERI4, 7H SET TOE11.3,3H + E10.3,3H = E12.3)	MINOS	225
	802 FORMAT (87H0CALL LIMIT EXCEEDED. CALCULATION OF MINOS ERROR FOR	MINOS	226
	1 THIS PARAMETER REQUIRES MORE THAN I5, 11H FCN CALLS./)	MINOS	227
	805 FORMAT (/45X 46HMINOS ERROR NOT CALCULATED FOR THIS PARAMETER.)	MINOS	228
	806 FORMAT (18H0DETERMINATION OF A4, 30HTIVE MINOS ERROR FOR PARAME	MINOS	229
240	1TERI3, 2X A10)	MINOS	230
	807 FORMAT (/45X4HTHE A4, 29HTIVE MINOS ERROR OF PARAMETERI3, 2H, A	MINOS	231
	110, 9H, EXCEEDS E12.4)	MINOS	232
	808 FORMAT (/45X4HTHE A4, 29HTIVE MINOS ERROR OF PARAMETERI3, 2H, A	MINOS	233
	110, 4H, IS E12.4)	MINOS	234
245	809 FORMAT (/45X 48HTOO MANY ITERATIONS. SEE PLOT BELOW FOR DETAILS.)	MINOS	235
	810 FORMAT (38H MINOS ERRORS REQUESTED FOR PARAMETERS30I3//)	MINOS	236
	811 FORMAT (55H THERE ARE MISTAKES IN THE MINOS COMMAND CARD JUST RE	MINOS	237
	1AD)	MINOS	238
	812 FORMAT (45X, 75(1H*))	MINOS	239
250	813 FORMAT (50H1MINOS FINDS NON-PARABOLIC BEHAVIOR FOR PARAMETER ,I3,	MINOS	240
	12H, A10/28H ERROR CALCULATION REQUIRED ,I3,63H MINIMIZATIONS.	MINOS	241
	2A SUMMARY OF POINTS FOUND IS PLOTTED BELOW. /13H0 FCN VALUE)	MINOS	242
	814 FORMAT (20H+ PARAMETER I3,1H,/)	MINOS	243
	END	MINOS	244

```

1      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
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
10     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
15     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
20     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   ,ITAU     ,SIGMA   COMMONU 31
      K/COMMND/ NNAME    ,CNAME(25) ,CNAM2(25) ,CNAM3(25) COMMONU 32
25     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
30     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
35     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
40     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
45     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
50     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
55     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
60	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
65	IF (KOUNT) 60,60,70	MPRINT	50
	60 KOUNT = 1	MPRINT	51
	WRITE (ISYSWR,1001) FVAL,NFCN,FI,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
70	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
75	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 I4,4XA10,4E14.5) E	MPRINT	63
	1004 FORMAT (1H 52X 42HWARNING - - ABOVE PARAMETER IS AT LIMIT.)	MPRINT	64
	1005 FORMAT (/45X 40HE:RORS CORRESPOND TO FUNCTION CHANGE OF F10.4)	MPRINT	65
80	1006 FORMAT (1H140X36HRES:ULTS OF FULL MINOS ERROR ANALYSIS/41:(36(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
85	END	MPRINT	71

1	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
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
10	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
15	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
20	A/CARD / CWORD ,CWORD2 ,CWORD3 ,WORD7(7)	COMMONU	30
	B/MINIMA/ AMIN ,UP ,NEWMIN ,ITAU ,SIGMA	COMMONU	31
	K/COMMND/ NNAME ,CNAME(25) ,CNAM2(25) ,CNAM3(25)	COMMONU	32
	DIMENSION VC(7)	MPUNCH	8
	TI = FTIME(DUMMY)	MPUNCH	9
25	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
30	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
35	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
40	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
45	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
50	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
55	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	MPUNCH	43
	300 RETURN	MPUNCH	44
60	1000 FORMAT (15A4,10X,F6.3)	MPUNCH	45
	1001 FORMAT (I10,A10,4E10.4)	MPUNCH	46
	1002 FORMAT (1H)	MPUNCH	47
	1003 FORMAT (10HCOVARIANCE I10)	MPUNCH	48
	1004 FORMAT (7E10.3,F6.3,2I2)	MPUNCH	49
65	END	MPUNCH	50

1	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
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
10	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
15	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
20	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 ,ITAU ,SIGMA	COMMONU	31
	K/COMMND/ NNAME ,CNAME(25) ,CNAM2(25) ,CNAM3(25)	COMMONU	32
25	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
30	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
35	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
40	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
45	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
50	150 CONTINUE	PLTCON	35
	160 CONTINUE	PLTCON	36
C	FIND EXTREME VALUES	PLTCON	37
	XMAX = SPT(1,1)	PLTCON	38
	XMIN = XMAX	PLTCON	39
55	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	PLTCON	43
	CALL BINSIZ(XMIN,XMAX,MAXNX,XMIN,XMAX,NX,BWIDX)	PLTCON	44
60	ANX = NX	PLTCON	45
	YMAX = SPT(2,1)	PLTCON	46
	YMIN = SPT(2,NSPT)	PLTCON	47
	IF (YMAX .EQ. YBEST) YMAX=YBEST + 1.0	PLTCON	48
	CALL BINSIZ(YMIN,YMAX,MAXNY,YMIN,YMAX,NY,BWIDY)	PLTCON	49
65	ANY = NY	PLTCON	50
	IF (CHBEST .EQ. ABLANK) GO TO 50	PLTCON	51
	XBEST = 0.5 * (XMAX+XMIN)	PLTCON	52
	YBEST = 0.5 * (YMAX+YMIN)	PLTCON	53
	50 CONTINUE	PLTCON	54
70	C FIND SCALE CONSTANTS	PLTCON	55
	AX = 1.0/BWIDX	PLTCON	56
	AY = 1.0/BWIDY	PLTCON	57
	BX = -AX*XMIN + 2.0	PLTCON	58
	BY = -AY*YMIN - 2.0	PLTCON	59
75	C CONVERT POINTS TO GRID POSITIONS	PLTCON	60
	DO 300 I= 1, NSPT	PLTCON	61
	SPT(1,I) = AX*SPT(1,I) + BX	PLTCON	62
	300 SPT(2,I) = ANY-AY*SPT(2,I) - BY	PLTCON	63
	NXBEST = AX*XBEST + BX	PLTCON	64
80	NYBEST = ANY - AY*YBEST - BY	PLTCON	65
	C PRINT THE CONTOURS	PLTCON	66
	NY = ANY + 1.	PLTCON	67
	NX = ANX + 2.	PLTCON	68
	ISP1 = 1	PLTCON	69
85	LINODD = 1	PLTCON	70
	DO 400 I= 1, NY	PLTCON	71
	DO 310 IBK= 1, NX	PLTCON	72
	310 ALINE (IBK) = ABLANK	PLTCON	73
	ALINE(1) = DOT	PLTCON	74
90	ALINE(NX) = DOT	PLTCON	75
	ALINE(NXBEST) = DOT	PLTCON	76
	IF (I.NE.1 .AND. I.NE.NYBEST .AND. I.NE.NY) GO TO 320	PLTCON	77
	DO 315 J= 1, NX	PLTCON	78
	315 ALINE(J) = DOT	PLTCON	79
95	320 CONTINUE	PLTCON	80
	YPRT = YMAX - FLOAT(I-1)*BWIDY	PLTCON	81
	IF (ISP1 .GT. NSPT) GO TO 350	PLTCON	82
	DO 340 K= ISP1,NSPT	PLTCON	83
	KS = SPT(2,K)	PLTCON	84
100	IF (KS .GT. I) GO TO 345	PLTCON	85
	IX = SPT(1,K)	PLTCON	86
	340 ALINE(IX) = SPT(3,K)	PLTCON	87
	ISP1 = NSPT + 1	PLTCON	88
	GO TO 350	PLTCON	89
105	345 ISP1 = K	PLTCON	90
	350 CONTINUE	PLTCON	91
	IF (LINODD .EQ. 1 .OR. I .EQ. NY) GO TO 380	PLTCON	92
	LINODD = 1	PLTCON	93
	WRITE (ISYSWR, 1007) (ALINE(IW),IW=1,NX)	PLTCON	94
110	GO TO 400	PLTCON	95
	380 WRITE (ISYSWR, 1001) YPRT, (ALINE(IW),IW=1,NX)	PLTCON	96
	LINODD = 0	PLTCON	97
	400 CONTINUE	PLTCON	98
	DO 410 IBK= 1, NX	PLTCON	99

115	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
120		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
125		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
130		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,13H ONE COLUMN = F13.7)	PLTCON	119
135	1003	FORMAT (5XA10)	PLTCON	120
	1007	FORMAT (18X , 100A1)	PLTCON	121
		END	PLTCON	122

```

1      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
5      CC
      COMMON
      1/MINERR/ ERP(30)      ,ERN(30)                                COMMONU 2
      2/PARINT/ X(15)       ,XT(15)      ,DIRIN(15) ,MAXINT      ,NPAR  COMMONU 4
      3/PAREXT/ U(30)       ,NAM(30)     ,WERR(30)  ,MAXEXT      ,NU   COMMONU 5
10     4/LIMITS/ ALIM(30)   ,BLIM(30)    ,LCODE(30) ,LCORSP(30) ,LIMSET  COMMONU 6
      5/VARIAN/ V(15,15)                                          COMMONU 7
      7/FIX / IPFIX(15)    ,XS(15)      ,XTS(15)   ,WTS(15)   ,NPFIX  COMMONU 8
      C/CASC/ JH, JL, Y(16)                                       COMMONU 9
      F/DERIVA/ G(30)      ,G2(30)                                           COMMONU 10
15     G/SIMVEC/ P(15,16)  ,PSTAR(15)   ,PSTST(15) ,PBAR(15)   ,PRHO(15) COMMONU 11
      J/VARIAT/ VT(15,15)                                       COMMONU 12
      6/UNIT / ISYSRD     ,ISYSWR      ,ISYSPU                                          COMMONU 13
      8/TITLE / TITLE(13) ,DATE(2)     ,ISW(7)    ,NBLOCK                                          COMMONU 27
      9/CONVER/ EPSI ,APSI ,VTEST      ,NSTEPQ   ,NFCN      ,NFCNMX  COMMONU 28
20     A/CARD / CWORD     ,CWORD2     ,CWORD3    ,WORD7(7)   COMMONU 29
      B/MINIMA/ AMIN      ,UP          ,NEWMIN   ,ITAU      ,SIGMA  COMMONU 30
      K/COMMND/ NNAME     ,CNAME(25)   ,CNAM2(25) ,CNAM3(25)  COMMONU 31
      DIMENSION PNEW(15)                                         RAZZIA 32
      DO 10 I=1,NPAR                                             RAZZIA 33
25     10 P(I,JH)=PNEW(I)                                         RAZZIA 34
      Y(JH)=YNEW                                                 RAZZIA 35
      IF(YNEW.GE.AMIN) GO TO 18                                   RAZZIA 36
      DO 15 I=1,NPAR                                             RAZZIA 37
30     15 X(I)=PNEW(I)                                           RAZZIA 38
      CALL INTOEX(X)                                             RAZZIA 39
      AMIN=YNEW                                                 RAZZIA 40
      JL=JH                                                       RAZZIA 41
35     18 CONTINUE                                               RAZZIA 42
      JH=1                                                         RAZZIA 43
      NPARP1=NPAR+1                                             RAZZIA 44
40     20 DO 25 J=2,NPARP1                                        RAZZIA 45
      IF (Y(J) .GT. Y(JH)) JH = J                                RAZZIA 46
45     25 CONTINUE                                               RAZZIA 47
      SIGMA = Y(JH) - Y(JL)                                       RAZZIA 48
      IF (SIGMA .LE. 0.) GO TO 45                                  RAZZIA 49
      US = 1.0/SIGMA                                             RAZZIA 50
      DO 35 I= 1, NPAR                                           RAZZIA 51
      PBIG = P(I,1)                                              RAZZIA 52
      PLIT = PBIG                                                RAZZIA 53
50     45 DO 30 J= 2, NPARP1                                       RAZZIA 54
      IF (P(I,J) .GT. PBIG) PBIG = P(I,J)                        RAZZIA 55
      IF (P(I,J) .LT. PLIT) PLIT = P(I,J)                        RAZZIA 56
55     30 CONTINUE                                               RAZZIA 57
      DIRIN(I) = PBIG - PLIT                                       RAZZIA 58
      IF (ITAU .LT. 1 ) V(I,I) = 0.5*(V(I,I) +US*DIRIN(I)**2)  RAZZIA 59
      35 CONTINUE                                               RAZZIA 60
      40 RETURN                                                  RAZZIA 61
      45 WRITE (ISYSWR, 1000) NPAR                                RAZZIA 62
      GO TO 40                                                  RAZZIA 63
1000  FORMAT ( 59H0***** FUNCTION VALUE DOES NOT SEEM TO DEPEND ON ANY RAZZIA 64
      10F THE I3, 20H VARIABLE PARAMETERS/15X 58HVERIFY THAT STEP SIZ RAZZIA 65
      2ES ARE BIG ENOUGH AND CHECK FCN LOGIC./1X 81(1H*)/1X 81(1H*)///) RAZZIA 66

```

END

RAZZIA 43

```

1      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
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
10     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
15     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
20     A/CARD / CWORD      ,CWORD2      ,CWORD3     ,WORD7(7)   COMMONU 30
      B/MINIMA/ AMIN       ,UP          ,NEWMIN     ,ITAU      ,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
25     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
30     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
35     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
40     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
45     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
50     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
55     XTS(I-1) = XTS(I)                                    RESTOR 40
30     WTS(I-1) = WTS(I)                                    RESTOR 41
      IPFIX(NPFIX) = IPSAV                                  RESTOR 42

```

	XS(NPFIK) = XSSAV	RESTOR	43
	XTS(NPFIK) = XTSSAV	RESTOR	44
60	WTS(NPFIK) = WTSSAV	RESTOR	45
	C RESTORE LAST PARAMETER IN LIST — IPFIK(NPFIK)	RESTOR	46
	50 CONTINUE	RESTOR	47
	IF (NPFIK .LT. 1) GO TO 300	RESTOR	48
	IR = IPFIK(NPFIK)	RESTOR	49
65	IS = 0	RESTOR	50
	DO 100 IJ= IR, NU	RESTOR	51
	IK = NU + IR - IJ	RESTOR	52
	IF (LCORSP(IK)) 100,100,85	RESTOR	53
70	85 LC = LCORSP(IK) + 1	RESTOR	54
	IS = LC - 1	RESTOR	55
	LCORSP(IK) = LC	RESTOR	56
	X(LC) = X(LC-1)	RESTOR	57
	XT(LC) = XT(LC-1)	RESTOR	58
	DIRIN(LC) = DIRIN(LC-1)	RESTOR	59
75	100 CONTINUE	RESTOR	60
	NPAR = NPAR + 1	RESTOR	61
	IF (IS .EQ. 0) IS = NPAR	RESTOR	62
	LCORSP(IR) = IS	RESTOR	63
	IQ = NPFIK	RESTOR	64
80	X(IS) = XS(IQ)	RESTOR	65
	XT(IS) = XTS(IQ)	RESTOR	66
	DIRIN(IS) = WTS(IQ)	RESTOR	67
	NPFIK = NPFIK - 1	RESTOR	68
	ISW(2) = 0	RESTOR	69
85	IF (ITAU .LT. 1) WRITE(ISYSWR,520) IR,NAM(IR)	RESTOR	70
	IF (K.EQ.0) GO TO 50	RESTOR	71
	300 RETURN	RESTOR	72
	500 FORMAT (61H0ERROR IN CALL TO RESTOR. THERE ARE NO MORE FIXED PA	RESTOR	73
	RAMETERS/)	RESTOR	74
90	510 FORMAT (52H0ERROR IN CALL TO RESTOR. ARGUMENT GREATER THAN ONE/	RESTOR	75
	1)	RESTOR	76
	520 FORMAT (20X, 9HPARAMETER,I4,2H, ,A10, 22H RESTORED TO VARIABLE.	RESTOR	77
	1)	RESTOR	78
	END	RESTOR	79

1	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
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
10	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
15	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
20	9/CONVER/ EPSI ,APSI ,VTEST ,NSTEPQ ,NFCN ,NFCNMX	COMMONU	29
	A/CARD / CWORD ,CWORD2 ,CWORD3 ,WORD7(7)	COMMONU	30
	B/MINIMA/ AMIN ,UP ,NEWMIN ,ITAU ,SIGMA	COMMONU	31
	K/COMMND/ NNAME ,CNAME(25) ,CNAM2(25) ,CNAM3(25)	COMMONU	32
	DIMENSION AMID(15) , N(15)	SEEK	10
25	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
30	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
35	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
40	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
45	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
50	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
55	IF (ISW(5) .LT. 2) GO TO 650	SEEK	45
	CALL EXTGIN(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
60	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
65	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

1	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
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
10	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
15	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
20	B/MINIMA/ AMIN ,UP ,NEWMIN ,ITAU ,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
25	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
30	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
35	IF (ABS(DIRIN(I)) .LT. 1.0E-10*ABS(X(I))) DIRIN(I)=1.0E-8*X(I)	SIMPLX	20
	IF(ITAU.LT. 1) V(I,I) = DIRIN(I)**2/UP	SIMPLX	21
	2 CONTINUE	SIMPLX	22
	IF (ITAU .LT. 1) ISW(2) = 1	SIMPLX	23
	C** CHOOSE THE INITIAL SIMPLEX USING SINGLE-PARAMETER SEARCHES	SIMPLX	24
40	1 CONTINUE	SIMPLX	25
	CONVER=.TRUE.	SIMPLX	26
	YNPP1 = AMIN	SIMPLX	27
	JL = NPARP1	SIMPLX	28
	Y(NPARP1) = AMIN	SIMPLX	29
45	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
50	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
55	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	SIMPLX	43
		IF (KG .EQ. 1) GO TO 8	SIMPLX	44
60		KG = -1	SIMPLX	45
		NF = NF + 1	SIMPLX	46
		DIRIN(I) = DIRIN(I) * (-0.4)	SIMPLX	47
		IF (NF .LT. 3) GO TO 4	SIMPLX	48
		NS = 6	SIMPLX	49
65	C	SUCCESS	SIMPLX	50
	6	BESTX = X(I)	SIMPLX	51
		DIRIN(I) = DIRIN(I) * 3.0	SIMPLX	52
		AMING = F	SIMPLX	53
		KG = 1	SIMPLX	54
70		NS = NS + 1	SIMPLX	55
		IF (NS .LT. 6) GO TO 4	SIMPLX	56
	C	LOCAL MINIMUM FOUND IN ITH DIRECTION	SIMPLX	57
	8	Y(I) = AMING	SIMPLX	58
		IF (AMING .LT. ABSMIN) JL = I	SIMPLX	59
75		IF (AMING .LT. ABSMIN) ABSMIN = AMING	SIMPLX	60
		X(I) = BESTX	SIMPLX	61
		XBEST(I)=BESTX	SIMPLX	62
		DO 9 K= 1, NPAR	SIMPLX	63
	9	P(K,I) = X(K)	SIMPLX	64
80	10	CONTINUE	SIMPLX	65
		JH = NPARP1	SIMPLX	66
		AMIN=Y(JL)	SIMPLX	67
		CALL RAZZIA(YNPP1,PBAR)	SIMPLX	68
		DO 20 I= 1, NPAR	SIMPLX	69
85	20	X(I) = P(I,JL)	SIMPLX	70
		CALL INTOEX(X)	SIMPLX	71
		IF (ISW(5) .GE. 1) CALL MPRINT(0,AMIN)	SIMPLX	72
		SIGMA = SIGMA * 10.	SIMPLX	73
		SIG2 = SIGMA	SIMPLX	74
90		IGNAL = 0	SIMPLX	75
		NCYCL=0	SIMPLX	76
	C START MAIN LOOP	SIMPLX	77
	50	CONTINUE	SIMPLX	78
		IF (IGNAL .GE. 10) GO TO 1	SIMPLX	79
95		IF (SIG2 .LT. EPSI .AND. SIGMA.LT.EPSI) GO TO 76	SIMPLX	80
		SIG2 = SIGMA	SIMPLX	81
		IF ((NFCN-NPFN) .GT. NFCNMX) GO TO 78	SIMPLX	82
	C	CALCULATE NEW POINT * BY REFLECTION	SIMPLX	83
		DO 60 I= 1, NPAR	SIMPLX	84
100		PB = 0.	SIMPLX	85
		DO 59 J= 1, NPARP1	SIMPLX	86
	59	PB = PB + WG * P(I,J)	SIMPLX	87
		PBAR(I) = PB - WG * P(I,JH)	SIMPLX	88
	60	PSTAR(I)=(1.+ALPHA)*PBAR(I)-ALPHA*P(I,JH)	SIMPLX	89
105		CALL INTOEX(PSTAR)	SIMPLX	90
		CALL FCN(NPAR,G,YSTAR,U,4)	SIMPLX	91
		NFCN=NFCN+1	SIMPLX	92
		IF(YSTAR.GE.AMIN) GO TO 70	SIMPLX	93
	C	POINT * BETTER THAN JL, CALCULATE NEW POINT **	SIMPLX	94
110		DO 61 I=1,NPAR	SIMPLX	95
	61	PSTST(I)=GAMMA*PSTAR(I)+(1.-GAMMA)*PBAR(I)	SIMPLX	96
		CALL INTOEX(PSTST)	SIMPLX	97
		CALL FCN(NPAR,G,YSTST,U,4)	SIMPLX	98
		NFCN=NFCN+1	SIMPLX	99

115	C	TRY A PARABOLA THROUGH PH, PSTAR, PSTST. MIN = PRHO	SIMPLX	100
		Y1 = (YSTAR-Y(JH)) * RHO2	SIMPLX	101
		Y2 = (YSTST-Y(JH)) * RHO1	SIMPLX	102
		RHO = 0.5 * (RHO2*Y1 -RHO1*Y2) / (Y1 -Y2)	SIMPLX	103
		IF (RHO .LT. RHOMIN) GO TO 66	SIMPLX	104
120		IF (RHO .GT. RHOMAX) RHO = RHOMAX	SIMPLX	105
		DO 64 I= 1, NPAR	SIMPLX	106
	64	PRHO(I) = RHO*PSTAR(I) + (1. -RHO)*P(I, JH)	SIMPLX	107
		CALL INTOEX(PRHO)	SIMPLX	108
		CALL FCN(NPAR, G, YRHO, U, 4)	SIMPLX	109
125		NFCN = NFCN + 1	SIMPLX	110
		IF (YRHO .LT. Y(JL) .AND. YRHO .LT. YSTST) GO TO 65	SIMPLX	111
		IF (YSTST .LT. Y(JL)) GO TO 67	SIMPLX	112
		IF (YRHO .GT. Y(JL)) GO TO 66	SIMPLX	113
	C	ACCEPT MINIMUM POINT OF PARABOLA, PRHO	SIMPLX	114
130	65	CALL RAZZIA (YRHO, PRHO)	SIMPLX	115
		IGNAL = MAX0(IGNAL-2, 0)	SIMPLX	116
		GO TO 68	SIMPLX	117
	66	IF (YSTST .LT. Y(JL)) GO TO 67	SIMPLX	118
		IGNAL = MAX0(IGNAL-1, 0)	SIMPLX	119
135		CALL RAZZIA(YSTAR, PSTAR)	SIMPLX	120
		GO TO 68	SIMPLX	121
	67	IGNAL = MAX0(IGNAL-2, 0)	SIMPLX	122
	675	CALL RAZZIA(YSTST, PSTST)	SIMPLX	123
	68	NCYCL=NCYCL+1	SIMPLX	124
140		IF (ISW(5) .LT. 2) GO TO 50	SIMPLX	125
		IF (ISW(5) .GE. 3 .OR. MOD(NCYCL, 10) .EQ. 0) CALL MPRINT(0, AMIN)	SIMPLX	126
		GO TO 50	SIMPLX	127
	C	POINT * IS NOT AS GOOD AS JL	SIMPLX	128
	70	IF (YSTAR .GE. Y(JH)) GO TO 73	SIMPLX	129
145		JHOLD = JH	SIMPLX	130
		CALL RAZZIA(YSTAR, PSTAR)	SIMPLX	131
		IF (JHOLD .NE. JH) GO TO 50	SIMPLX	132
	C	CALCULATE NEW POINT **	SIMPLX	133
	73	DO 74 I=1, NPAR	SIMPLX	134
150	74	PSTST(I)=BETA*P(I, JH)+(1.-BETA)*PBAR(I)	SIMPLX	135
		CALL INTOEX (PSTST)	SIMPLX	136
		CALL FCN(NPAR, G, YSTST, U, 4)	SIMPLX	137
		NFCN=NFCN+1	SIMPLX	138
		IF(YSTST.GT.Y(JH)) GO TO 1	SIMPLX	139
155	C	POINT ** IS BETTER THAN JH	SIMPLX	140
		IF (YSTST .LT. AMIN) GO TO 675	SIMPLX	141
		IGNAL = IGNAL + 1	SIMPLX	142
		CALL RAZZIA(YSTST, PSTST)	SIMPLX	143
		GO TO 50	SIMPLX	144
160	C END MAIN LOOP	SIMPLX	145
	76	WRITE(ISYSWR, 120)	SIMPLX	146
		GO TO 80	SIMPLX	147
	78	WRITE(ISYSWR, 130)	SIMPLX	148
		ISW(1) = 1	SIMPLX	149
165		CONVER=.FALSE.	SIMPLX	150
	80	DO 82 I=1, NPAR	SIMPLX	151
		PB = 0.	SIMPLX	152
		DO 81 J=1, NPARP1	SIMPLX	153
	81	PB = PB + WG * P(I, J)	SIMPLX	154
170	82	PBAR(I) = PB - WG * P(I, JH)	SIMPLX	155
		CALL INTOEX(PBAR)	SIMPLX	156

	CALL FCN(NPAR,G,YPBAR,U,IFLAG)	SIMPLX	157
	NFCN=NFCN+1	SIMPLX	158
	IF (YPBAR .LT. AMIN) CALL RAZZIA(YPBAR,PBAR)	SIMPLX	159
175	CALL INTOEX(X)	SIMPLX	160
	IF (NFCNMX+NPFN-NFCN .LT. 3*NPAR) GO TO 90	SIMPLX	161
	IF (SIGMA .GT. 2.0*EPSI) GO TO 1	SIMPLX	162
90	IF(CONVER) GO TO 902	SIMPLX	163
	CALL INTOEX(XBEST)	SIMPLX	164
180	902 CALL MPRINT(1-ITAU, AMIN)	SIMPLX	165
	RETURN	SIMPLX	166
	100 FORMAT(37H START SIMPLEX MINIMIZATION 8X 65HCON	SIMPLX	167
	1VERGENCE CRITERION — ESTIMATED DISTANCE TO MINIMUM (EDM) .LT.E10.	SIMPLX	168
	22)	SIMPLX	169
185	120 FORMAT(1H , 34HSIMPLEX MINIMIZATION HAS CONVERGED)	SIMPLX	170
	130 FORMAT(1H , 38HSIMPLEX TERMINATES WITHOUT CONVERGENCE)	SIMPLX	171
	END	SIMPLX	172

1	SUBROUTINE STAND	STAND	2
	CC OPTIONAL USER-SUPPLIED SUBROUTINE IS CALLED WHENEVER THE	STAND	3
	CC COMMAND "STANDARD" APPEARS.	STAND	4
	CC	STAND	5
5	RETURN	STAND	6
	END	STAND	7

```

1      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
5      CC
      COMMON
      1/MINERR/ ERP(30)      ,ERN(30)                                COMMONU   2
      2/PARINT/ X(15)      ,XT(15)      ,DIRIN(15) ,MAXINT      ,NPAR    COMMONU   5
      3/PAREXT/ U(30)      ,NAM(30)      ,WERR(30) ,MAXEXT      ,NU      COMMONU   6
10     4/LIMITS/ ALIM(30)  ,BLIM(30)   ,LCODE(30) ,LCORSP(30) ,LIMSET   COMMONU   7
      5/VARIAN/ V(15,15)
      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)
15     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
      8/TITLE / TITLE(13) ,DATE(2)    ,ISW(7)     ,NBLOCK    COMMONU  28
      9/CONVER/ EPSI ,APSI ,VTEST     ,NSTEPQ    ,NFCN      ,NFCNMX    COMMONU  29
20     A/CARD / CWORD    ,CWORD2     ,CWORD3     ,WORD7(7)   COMMONU  30
      B/MINIMA/ AMIN     ,UP          ,NEWMIN    ,ITAU      ,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
25     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
30     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
35     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
40     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
45     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
50     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
55     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	VERMIN	48
		DO 70 J= 1, N	VERMIN	49
60		DO 70 K= 1, J	VERMIN	50
		A(K,J) = A(K,J) *S(K)*S(J)	VERMIN	51
	70	A(J,K) = A(K,J)	VERMIN	52
		RETURN	VERMIN	53
	C	FAILURE RETURN	VERMIN	54
65	100	IFAIL=1	VERMIN	55
		RETURN	VERMIN	56
		END	VERMIN	57

1	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
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
10	RETURN	UCOPY	11
	END	UCOPY	12

1		SUBROUTINE BINSIZ(A1,A2,NAA,BL,BH,NB,BWID)	BINSIZ	2
	C	SUBROUTINE TO DETERMINE REASONABLE HISTOGRAM INTERVALS	BINSIZ	3
	C	GIVEN ABSOLUTE UPPER AND LOWER BOUNDS A1 AND A2	BINSIZ	4
	C	AND DESIRED MAXIMUM NUMBER OF BINS NAA	BINSIZ	5
5	C	PROGRAM MAKES REASONABLE BINNING FROM BL TO BH OF WIDTH BWID	BINSIZ	6
	C	F. JAMES, AUGUST, 1974	BINSIZ	7
	C	TYPED BY E. CLOSE 9/75 FROM LISTING FROM CERN THAT CAME	BINSIZ	8
	C	WITH THE CODE.	BINSIZ	9
		AL= AMIN1(A1,A2)	BINSIZ	10
10		AH= AMAX1(A1,A2)	BINSIZ	11
		IF(AL.EQ.AH) AH= AL+1.	BINSIZ	12
	C	IF(NAA.EQ.-1), PROGRAM USES BWID INPUT FROM CALLING ROUTINE	BINSIZ	13
		IF(NAA.EQ.-1) GOTO 150	BINSIZ	14
	10	NA=NAA-1	BINSIZ	15
15		IF(NA.LT.1) NA= 1	BINSIZ	16
	C	GET NOMINAL BIN WIDTH IN EXPON FORM	BINSIZ	17
	20	AWID=(AH-AL)/FLOAT(NA)	BINSIZ	18
		LOG= ALOG10(AWID)	BINSIZ	19
		IF(AWID.LE.1.0) LOG= LOG-1	BINSIZ	20
20		SIGFIG= AWID*(10.00**(-LOG))	BINSIZ	21
	C	ROUND MANTISSA UP TO 2,2.5,5,OR 10	BINSIZ	22
		IF(SIGFIG.GT. 2.0) GOTO 40	BINSIZ	23
		SIGRND= 2.0	BINSIZ	24
		GOTO 100	BINSIZ	25
25	40	IF(SIGFIG.GT.2.5) GOTO 50	BINSIZ	26
		SIGRND= 2.5	BINSIZ	27
		GOTO 100	BINSIZ	28
	50	IF(SIGFIG.GT.5.0) GOTO 60	BINSIZ	29
		SIGRND= 5.0	BINSIZ	30
30		GOTO 100	BINSIZ	31
	60	SIGRND=1.0	BINSIZ	32
		LOG= LOG+1	BINSIZ	33
	100	CONTINUE	BINSIZ	34
		BWID= SIGRND*10.0**LOG	BINSIZ	35
35		GOTO 200	BINSIZ	36
	C	GET NEW BOUNDS FROM NEW WIDTH BWID	BINSIZ	37
	150	IF(BWID.LE.0.) GOTO 10	BINSIZ	38
	200	CONTINUE	BINSIZ	39
		ALB= AL/BWID	BINSIZ	40
40		LWID= ALB	BINSIZ	41
		IF(ALB.LT.0.0) LWID= LWID-1	BINSIZ	42
		BL= BWID* FLOAT(LWID)	BINSIZ	43
		ALB= AH/BWID+1.0	BINSIZ	44
		KWID= ALB	BINSIZ	45
45		IF(ALB.LT.0.0) KWID= KWID-1	BINSIZ	46
		BH= BWID*FLOAT(KWID)	BINSIZ	47
		NB= KWID-LWID	BINSIZ	48
		IF(NAA.GT.5) GOTO 240	BINSIZ	49
		IF(NAA.EQ.-1) RETURN	BINSIZ	50
50	C	REQUEST FOR ONE BIN IS DIFFICULT CASE	BINSIZ	51
		IF(NAA.GT.1 .OR. NB .EQ. 1) RETURN	BINSIZ	52
		BWID= BWID*2.0	BINSIZ	53
		NB= 1	BINSIZ	54
		RETURN	BINSIZ	55
55	240	IF(2*NB.NE.NAA) RETURN	BINSIZ	56
		NA= NA+1	BINSIZ	57
		GOTO 20	BINSIZ	58

END

BINSIZ 59

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.      PFL0D - BEGIN FILE DUMTAPE RECOVERY
      16.54.35 00000.156 USR.      PFL0D - ---LOAD BEGINS AGAIN -----AT BN =      0B---
      16.54.35 00000.157 USR.      PFL0D - LOADING BEGINS WITH FILE 112
      16.54.43 00000.312 USR.      PFL0D -      EDC CYCLE 1 BSY86
      16.54.43 00000.312 USR.      PFL0D -      1 PERMANENT FILES LOADED
      16.54.43 00000.313 USR.      PFL0D - LOADPF ERROR COUNT      0
      16.54.43 00000.313 USR.      PFL0D - 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		

1
1
1
1
1
1

1	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 = FOCUSING MAGNET AT M-1.	SSSR	5
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
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
15	C IF KF=2, Q = PRODUCT OF ...QS QF QT QD QS QS QF QT QD QS. (DFD)	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
20	COMMON/BMI/MI(5000)	BMIL	1
	COMMON/NELS/NELS	BMIL	2
	LEVEL 2,MI	BMIL	3
		BMIL	4
		BMI	3
25	EQUIVALENCE (E,T)	SSSR	22
	LOGICAL COLL	SSSR	23
	DIMENSION P(6)	SSSR	24
	C	SSSR	25
	COLL = .FALSE.	SSSR	26
30	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
35	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
40	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
45	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
50	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
55	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	SSSR	55
	MI(2) = M - 1	SSSR	56
60	MI(4) = M - 2	SSSR	57
	2 IF (COLL) GO TO 4	SSSR	58
	C COMPLETE MI LIST	SSSR	59
	DO 3 I=1,5	SSSR	60
	3 MI(I+5) = MI(I)	SSSR	61
65	IM = 10	SSSR	62
	GO TO 5	SSSR	63
	C MAKE CHANGES FOR COLLINS STRAIGHT SECTION	SSSR	64
	4 ISAV = MI(1)	SSSR	65
	MI(1) = MI(3)	SSSR	66
70	MI(3) = ISAV	SSSR	67
	ISAV = MI(4)	SSSR	68
	MI(4) = MI(5)	SSSR	69
	MI(5) = ISAV	SSSR	70
	MI(6) = MI(1)	SSSR	71
75	IM = 6	SSSR	72
	5 L = LM	SSSR	73
	CALL MMM(L,IM,MI)	SSSR	74
	C SAVE THE MILIST AS DATA FOR Q	SSSR	75
	C PI SECTION HAS ONE PIECE OF INTEGER DATA, SO MILIST BEGINS AT 2.	SSSR	76
80	DO 6 I=1,IM	SSSR	77
	6 CALL REPINT(M,I,MI(I))	SSSR	78
	RETURN	SSSR	79
	END	SSSR	80

1	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
5	COMMON/NELS/NELS	BMIL	2
	LEVEL 2,MI	BMIL	3
		BMIL	4
		BMI	3
	N=MDAT(M,1)	STAR	6
10	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
15	END	STAR	12

1	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
5	C IF I=7,STORE AT LQ2. IF I=8, STORE AT LQ3	STDAT	6
	LEVEL 2, STORE,INFF,IWORK	BLANK	2
	COMMON STORE(48000),IWORK(10)	BLANK	3
	DIMENSION INFF(24,2000)	86MARSIZ	1
10	EQUIVALENCE (INFF,STORE)	86MARSIZ	2
		86MARSIZ	3
	DIMENSION A(1),LOC(10)	BLANK	5
	IF (I.EQ.4) RETURN	STDAT	8
	LOC(1) = INFF(14,M)	STDAT	9
15	LOC(2)=INFF(16,M)	STDAT	10
	LOC(3)=INFF(12,M)	STDAT	11
	LOC(5) = INFF(10,M)	STDAT	12
	LOC(7) = INFF(19,M)	STDAT	13
	LOC(8)=INFF(24,M)	STDAT	14
20	LOC I = LOC(I)+J-2	STDAT	15
	DO 10 L=1,K	STDAT	16
	LOC I = LOC I+1	STDAT	17
	10 STORE(LOC I) = A(L)	STDAT	18
	RETURN	STDAT	19
25	END	STDAT	20
		STDAT	21

1	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
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
10	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
15	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
20	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
25	INFF(5,M0) = KB0	STINFO	21
	RETURN	STINFO	22
	END	STINFO	23

1	SUBROUTINE STOR7(M,T,RW)	STOR7	2
	C STORES 7X7 MATRICES	STOR7	3
		BLANK	2
	LEVEL 2, STORE,INFF,IWORK	BLANK	3
5	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
10	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,MSYML	DIM	2
		DIM	3
15	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
20	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
25	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.3HMY) GO TO 11	STOR7	21
30	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
35	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
40	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
45	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

1	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
5	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
10	EQUIVALENCE (BKS, BZ)	STRFCN	5
	PHI = X	STRFCN	6
	C = COS(PHI)	STRFCN	7
	S = SIN(PHI)	STRFCN	8
	E = EXP(PHI)	STRFCN	9
15	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
20	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
25	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
30	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
35	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
40	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
45	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
50	END	STRFCN	45

1	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	
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	
10	C	STRN	11	
	LEVEL 2, STORE,INFF,IWORK	BLANK	2	
	COMMON STORE(48000),IWORK(10)	BLANK	3	
	DIMENSION INFF(24,2000)	86MARSIZ	1	
15	EQUIVALENCE (INFF,STORE)	86MARSIZ	2	
		86MARSIZ	3	
		BLANK	5	
	COMMON/BMI/MI(5000)	BMIL	1	
	COMMON/NELS/NELS	BMIL	2	
20	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	
25	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	
30	DATA IFO,IDFO,IBL/9HF0CUSSING,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	
35	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	
40	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	
45	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	
50	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	
55	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)	STRN	50
	C COMPUT PX AND PY (P AND Q). PY IS Q OF WRITE UP	STRN	51
60	RBX = SQRT(BX(2))	STRN	52
	RBY = SQRT(BY(2))	STRN	53
	PX(1,1) = (RBY*(CSI+BX(3)*SSI))/RBX	STRN	54
	PX(1,2) = RBX*RBY*SSI	STRN	55
	PX(2,1) = ((BX(3)*BY(3)-1.)*SSI+(BX(3)+BY(3))*CSI)/(RBX*RBY)	STRN	56
65	PX(2,2) = (RBX*(CSI+BY(3)*SSI))/RBY	STRN	57
	PY(1,1) = PX(2,2)	STRN	58
	PY(2,2) = PX(1,1)	STRN	59
	PY(1,2) = PX(1,2)	STRN	60
	PY(2,1) = PX(2,1)	STRN	61
70	C GET KIA (UX,UY) (U AND V OF WRITE UP). STORE KIA IN M-1	STRN	62
	CALL RXY(BLBX,RX,RY,RW)	STRN	63
	L = M - 1	STRN	64
	C STORE OMEGA = 0 IN STORAGE OF L	STRN	65
	OMEG = 0.	STRN	66
75	CALL REPFLT(L,4,OMEG)	STRN	67
	CALL STXY(L,RX,RY,RW)	STRN	68
	LZ = RW(1)	STRN	69
	DO 160 J=1,2	STRN	70
	DO 158 I=1,2	STRN	71
80	UX(I,J) = RX(I,J)	STRN	72
	UY(I,J) = RY(I,J)	STRN	73
	158 CONTINUE	STRN	74
	160 CONTINUE	STRN	75
	C COMPUT KIB = WI = WX,WY(OLD,WIX,WIY)	STRN	76
85	L = M - 2	STRN	77
	CALL REF(L,BLBX)	STRN	78
	CALL RXY(L,RX,RY,RW)	STRN	79
	DO 164 I=1,2	STRN	80
	DO 162 J=1,2	STRN	81
90	RXS = RX(I,J)	STRN	82
	RX(I,J) = RY(I,J)	STRN	83
	162 RY(I,J) = RXS	STRN	84
	164 CONTINUE	STRN	85
	CALL REPFLT(L,4,OMEG)	STRN	86
95	CALL STXY(L,RX,RY,RW)	STRN	87
	C SET UP ITERATION FOR LENGTH	STRN	88
	NIT = 1	STRN	89
	OM = SVM	STRN	90
	DCS = 0.	STRN	91
100	DELM = SVM/5.	STRN	92
	OMM = 0.	STRN	93
	SOK = 1.	STRN	94
	185 PHIO = OK=OM	STRN	95
	CS = COS(PHIO)	STRN	96
105	SN = SIN(PHIO)	STRN	97
	EP = EXP(PHIO)	STRN	98
	EM = 1./EP	STRN	99
	CB = (EP+EM)/2.	STRN	100
	SB = CB-EM	STRN	101
110	C COMPUTE F AND G	STRN	102
	C TRY D MAGNET FOR KA	STRN	103
	SOK = 1.	STRN	104
	196 F1 = SB*UX(1,1)+PX(2,2)*SN*UY(1,1)	STRN	105
	F2 = -PX(2,1)*SN*UY(1,1)	STRN	106

115	F3 = CB*UX(2,1)-PX(2,2)*CS*UY(2,1)	STRN	107
	F4 = PX(2,1)*CS*UY(2,1)	STRN	108
	G1 = SB*UX(1,1)-PX(1,1)*SN*UY(1,1)+PX(2,1)*SN*UY(1,2)	STRN	109
	G2 = SB*UX(1,2)+PX(2,2)*SN*UY(1,2)-PX(1,2)*SN*UY(1,1)	STRN	110
	G3 = CB*UX(2,1)+PX(1,1)*CS*UY(2,1)-PX(2,1)*CS*UY(2,2)	STRN	111
120	G4 = CB*UX(2,2)-PX(2,2)*CS*UY(2,2)+PX(1,2)*CS*UY(2,1)	STRN	112
	C COMPUT A,B,C DISCRIMINANT,DEN	STRN	113
207	AFG = F1*G1+F2*G2	STRN	114
	BFG = (F1*G3+F3*G1+F2*G4+F4*G2)*SOK	STRN	115
	CFG = F3*G3+F4*G4	STRN	116
125	DCS = BFG*BFG-4.*AFG*CFG	STRN	117
	IF (SOK.LT.0..OR.DCS.GE.0.) GO TO 2085	STRN	118
	C OTHERWISE TRY F MAGNET FOR KA	STRN	119
2077	SOK = -1.	STRN	120
	SS = SN	STRN	121
130	SN = SB	STRN	122
	SB = SS	STRN	123
	SS = CS	STRN	124
	CS = CB	STRN	125
	CB = SS	STRN	126
135	GO TO 196	STRN	127
2085	DEN = 1.*AFG	STRN	128
	IF (DCS.LT.0.) GO TO 2086	STRN	129
	DIS = SQRT(DCS)	STRN	130
	C COMPUTE K(MO) CALLED TK	STRN	131
140	TK = 0.	STRN	132
	TKP = (-BFG+DIS)/DEN	STRN	133
	IF (TKP.GE.0.) GO TO 2185	STRN	134
	IF (SOK.LT.0.) GO TO 220	STRN	135
	GO TO 2077	STRN	136
145	2086 OMM = OMM + DELM	STRN	137
	OM = OMM	STRN	138
	NIT = NIT + 1	STRN	139
	GO TO 185	STRN	140
	2185 TK = TKP	STRN	141
150	TKM = (-BFG-DIS)/DEN	STRN	142
	IF (TKM.LE.0.) GO TO 220	STRN	143
	DKP = ABS(OK-TKP)	STRN	144
	DKM = ABS(OK-TKM)	STRN	145
	IF (DKM.LT.DKP) TK=TKM	STRN	146
155	C COMPARE TK AND OK,NIT AND ITMX	STRN	147
220	DK = TK-OK	STRN	148
	AD = ABS(DK)	STRN	149
	IF (AD.LE.EPSK.OR.NIT.GE.ITMX) GO TO 313	STRN	150
	C COMPUTE AND ADJUST NEW OM	STRN	151
160	CM = TK*OM/OK	STRN	152
	TM = .5*OM	STRN	153
	IF (CM.LT.TM) CM=TM	STRN	154
	TM = 2.*OM	STRN	155
	IF (CM.GT.TM) CM=TM	STRN	156
165	IF (NIT.EQ.1) GO TO 311	STRN	157
	IF (AD.LT.SD) GO TO 311	STRN	158
	CM = (CM+OM)/2.	STRN	159
	311 OM = CM	STRN	160
	SD = AD	STRN	161
170	NIT = NIT + 1	STRN	162
	GO TO 185	STRN	163

	C COMPUTE MAGNET KB,KA. STORE IN M-3,M-4	STRN	164
	313 BX(1) = OM	STRN	165
	3131 BX(2) = SOK*OKK	STRN	166
175	BX(3) = RHO	STRN	167
	BX(4) = 0.	STRN	168
	BX(5) = 0.	STRN	169
	BX(6) = 0.	STRN	170
	L = M - 3	STRN	171
180	CALL STDAT(L,1,1,6,BX)	STRN	172
	CALL MAGNET(L,BX)	STRN	173
	BX(2) = -SOK*OKK	STRN	174
	L = M - 4	STRN	175
	CALL STDAT(L,1,1,6,BX)	STRN	176
185	CALL MAGNET(L,BX)	STRN	177
	MI(1) = M - 1	STRN	178
	MI(2) = M - 4	STRN	179
	L = M - 5	STRN	180
	CALL REPFLT(L,4,OMEG)	STRN	181
190	CALL MMM(L,2,MI)	STRN	182
	C UX AND UY ARE NOW TO BE X,Y PARTITIONS OF WD(OLD UIJ,VIJ)	STRN	183
	CALL RXY(L,UX,UY,RW)	STRN	184
	L = M - 6	STRN	185
	MI(1) = M - 2	STRN	186
195	MI(2) = M - 3	STRN	187
	CALL REPFLT(L,4,OMEG)	STRN	188
	CALL MMM(L,2,MI)	STRN	189
	ELE = (UX(2,1)+PX(1,1)*UY(2,1)-PX(2,1)*UY(2,2))/(PX(2,1)*UY(2,1))	STRN	190
	DLD = -UX(2,1)+PX(1,1)*UY(2,1)-PX(2,1)*(UY(2,2)+ELE*UY(2,1))	STRN	191
200	ELD = UX(1,1)-PX(1,1)*UY(1,1)+PX(2,1)*(UY(1,2)+ELE*UY(1,1))	STRN	192
	ELD = ELD/DLD	STRN	193
	DLD = 2. * ELD	STRN	194
	IPR = IBL	STRN	195
	IF (SOK.GT.0.) IPR = IDFO	STRN	196
205	WRITE(3,504)IPR,IFO	STRN	197
	3317 L = M - 7	STRN	198
	CALL DRIFT(L,ELE)	STRN	199
	L = L - 8	STRN	200
	CALL DRIFT(L,ELD)	STRN	201
210	MI(1) = M - 7	STRN	202
	MI(2) = M - 1	STRN	203
	MI(3) = M - 4	STRN	204
	MI(4) = M - 8	STRN	205
	MI(5) = M - 8	STRN	206
215	MI(6) = M - 3	STRN	207
	MI(7) = M - 2	STRN	208
	MI(8) = M - 7	STRN	209
	CALL MMM(M,8,MI)	STRN	210
	DLH = DLD/2.	STRN	211
220	CALL RXY(M,RX,RY,RW)	STRN	212
	LTOT = RW(1)	STRN	213
	WRITE (3,505) ELE,LZ,OM,DLH,DLD,LTOT	STRN	214
	WRITE (3,503) NIT	STRN	215
	CALL HED	STRN	216
225	RETURN	STRN	217
	501 FORMAT (/ * MATCHING STRAIGHT SECTION (INSERTION OR HALF CELL * 1 * REPLACEMENT) AT POSITION *,A5)	STRN	218
	502 FORMAT(2X,*MU*,6X,*= *,F13.8,5X,*GRAD*,4X,*= *,F13.8,5X,*RHO*,	STRN	219
		STRN	220

	1	5X,*=*,F13.8/2X,*BETAX =*,F13.8,5X,*BETAY =*,F13.8,	STRN	221
230	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
235	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.

1	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
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 DDFD,K=2 FOR FDFD)	STRPI	9
	C J IS THE POINT OF INSERTION INTO THE CELL.	STRPI	10
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
15	LEVEL 2, STORE,INFF,IWORK	BLANK	2
	COMMON STORE(48000),IWORK(10)	BLANK	3
	DIMENSION INFF(24,2000)	86MARSIZ	1
	EQUIVALENCE (INFF,STORE)	86MARSIZ	2
20		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
25	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
30	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
35	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
40	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
45	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
50	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
55	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)	STRPI	49
	IF (IOP.EQ.HCFD) GO TO 51	STRPI	50
60	C CELL, ALTC,DCFD	STRPI	51
	I = 3	STRPI	52
	GO TO 52	STRPI	53
	C CFD	STRPI	54
51	I = 4	STRPI	55
65	52 RHO = FDAT(CI,I)	STRPI	56
	BL = LTO/2.	STRPI	57
	F = SQRT(LSEP*BL)	STRPI	58
	S = LSEP + BL	STRPI	59
	A = AVAC	STRPI	60
70	AS = A*SQRT((BETA/BETMAX)*((1.+ALPHA*S/BETA)**2+(S/BETA)**2))	STRPI	61
	LK = BQ/(AS*B0)	STRPI	62
	BK = SQRT(LK/RHO)	STRPI	63
	LM = 1./(F*BK**2)	STRPI	64
	THETA = BK*LM	STRPI	65
75	GAM = P	STRPI	66
	CALL GRT(1,THETA,1,LPAR)	STRPI	67
	IF (IERR) RETURN	STRPI	68
	LDRF = 2.*BL	STRPI	69
	GRAD = BQ/AS	STRPI	70
80	CALL HED	STRPI	71
	WRITE(3,3) C,ALPHA,BETA,BETMAX,RHO,B0,BQ,AVAC,AS,LM,LK,LDRF,LSEP,	STRPI	72
	1 LTO,BL,GRAD	STRPI	73
	CALL HED	STRPI	74
	PAR(1) = LM	STRPI	75
85	PAR(2) = LK	STRPI	76
	PAR(3) = RHO	STRPI	77
	PAR(4) = LDRF	STRPI	78
	PAR(5) = LSEP	STRPI	79
	CALL SSSR(M,PAR,KI,0)	STRPI	80
90	RETURN	STRPI	81
	3 FORMAT(* LONG STRAIGHT SECTION IN CELL *,A1,* ./1H0,5X,*ALPHA=*,	STRPI	82
	1 F12.8,*,*,7X,*BETA=*,F12.8,*,*,4X,*BETMAX=*,F12.8,*,*,7X,	STRPI	83
	2 *RHO=*,F12.8/9X,*B0=*,F12.8,*,*,9X,*BQ=*,F12.8,*,*,6X,	STRPI	84
	3 *AVAC=*,F12.8,*,*,8X,*AS=*,F12.8/7X,*LMAG=*,F12.8,*,*,10X,	STRPI	85
95	4 *K=*,F12.8,*,*,6X,*LDRF=*,F12.8,*,*,6X,*LSEP=*,F12.8,/	STRPI	86
	5 7X,*LTOT*,F12.8,*,*,7X,*LEND=*,F12.8,*,*,6X,*GRAD=*,F12.8,	STRPI	87
	6 2X,*KG/M*)	STRPI	88
	END	STRPI	89

1	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
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
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
15	EQUIVALENCE (INFF,STORE)	86MARSIZ	3
		BLANK	5
	COMMON/BMI/MI(5000)	BMIL	1
	COMMON/NELS/NELS	BMIL	2
20	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
25	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
30	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
35	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
40	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
45	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
50	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
55	CALL BET(L,BX,BY,0)	STR2	42
	PALPH = BX(3)	STR2	43
	ALPHA = ABS(PALPH)	STR2	44

	BETA = BX(2)	STR2	45
	GAM = BX(4)	STR2	46
60	C COMPUTE BMAX FOR THE CELL	STR2	47
	L = CI - 2	STR2	48
	CALL BET(L,BX,BY,0)	STR2	49
	BMAX = BX(2)	STR2	50
	C GET VALUE OF RHO FOR THE CELL	STR2	51
65	IOP = INFF(1,CI)	STR2	52
	IF (IOP.EQ.HCFD) GO TO 51	STR2	53
	C CELL,ALTC OR DCFD	STR2	54
	I = 3	STR2	55
	GO TO 52	STR2	56
70	C CFD	STR2	57
	51 I = 4	STR2	58
	52 RHO = FDAT(CI,I)	STR2	59
	SMUD = SIN(MUD)	STR2	60
	CMUD = COS(MUD)	STR2	61
75	A = AVAC	STR2	62
	C COMPUTE FIRST GUESS FOR THETA	STR2	63
	LK = BQ/(A*BZ)	STR2	64
	BK = SQRT(LK/RHO)	STR2	65
	F = ALPHA/GAM	STR2	66
80	LM = 1./(F*(BK*BK))	STR2	67
	THET = BK * LM	STR2	68
	C CALL ROOT FINDING ROUTINE	STR2	69
	CALL GRT (1,THET,1,PARSR)	STR2	70
	IF (IERR) RETURN	STR2	71
85	LTOT = BL+2*LL+2.*LM	STR2	72
	G = BQ/A	STR2	73
	CALL HED	STR2	74
	WRITE(3,1) C,PALPH,BETA,RHO,BZ,BQ,AVAC,LM,LK,BL,LL,LTOT,A,G	STR2	75
	CALL HED	STR2	76
90	PAR(1) = LM	STR2	77
	PAR(2) = LK	STR2	78
	PAR(3) = RHO	STR2	79
	PAR(4) = BL	STR2	80
	PAR(5) = LL	STR2	81
95	CALL SSSR(M,PAR,KI,1)	STR2	82
	C ON RETURNING, THE MI LIST CONTAINS THE LIST FOR THE CELL WITH THE	STR2	83
	C CSS INSETED. I.E., QK, WHETE Q IS THE CELL NAME PREFIX.	STR2	84
	C QK IS COMPUTED AND SAVED.	STR2	85
	J = 2 - KI	STR2	86
100	L = CI - 13	STR2	87
	MI(2) = L	STR2	88
	MI(12) = L	STR2	89
	L = CI - 14	STR2	90
	MI(J+5) = L	STR2	91
105	MI(J+8) = L	STR2	92
	L = CI -15	STR2	93
	MI(1) = L	STR2	94
	MI(J+6) = L	STR2	95
	MI(J+7) = L	STR2	96
110	MI(13) = L	STR2	97
	L = CI - 16	STR2	98
	MI(3) = L	STR2	99
	MI(J+4) = L	STR2	100
	MI(J+9) = L	STR2	101

115	MI(11) = L	STR2	102
	IF (KI.EQ.2) GO TO 53	STR2	103
	MI(4) = M	STR2	104
	GO TO 54	STR2	105
53	MI(10) = M	STR2	106
120	54 L = M - 5	STR2	107
	CALL MMM(L,13,MI)	STR2	108
	C STORE THE MI LIST	STR2	109
	E = 0.	STR2	110
	J = 0	STR2	111
125	DO 55 I =1,13	STR2	112
	J = MI(I)	STR2	113
	CALL REPINT(L,I,J)	STR2	114
	55 CONTINUE	STR2	115
	RETURN	STR2	116
130	1 FORMAT (* COLLINS STRAIGHT SECTION IN CELL*,A1,* .*/1H0,5X,	STR2	117
	1 *ALPHA=*F12.8,*,*,7X,*BETA=*F12.8,*,*,7X,*RHO=*F12.8/	STR2	118
	2 9X,*B0=*F12.8,*,*,9X,*BQ=*F12.8,*,*,6X,*AVAC=*F12.8/	STR2	119
	3 6X,*LMAG=*F12.8,*,*,10X,*K=*F12.8,*,*,6X,*LDRF=*F12.8,	STR2	120
	4 *,*,6X,*LEND=*F12.8/6X,*LTOT=*F12.8,*,*,9X,*AQ=*F12.8,	STR2	121
135	5 *,*,6X,*GRAD=*F12.8,* KG/M*)	STR2	122
	END	STR2	123

1	SUBROUTINE STR4(M,NCELL,PAR)	STR4	2
		BLANK	2
	LEVEL 2, STORE,INFF,IWORK	BLANK	3
	COMMON STORE(48000),IWORK(10)	86MARSIZ	1
5	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
10		GRR	4
	COMMON/BMI/MI(5000)	BMIL	1
	COMMON/NELS/NELS	BMIL	2
	LEVEL 2,MI	BMIL	3
15		BMIL	4
	C	BMI	3
	COMMON/WRK/BZ,BQ,AZ,DPSI,SM,SL,PSI,WW,WT,BETW,P,Q,	WRK	2
	1 T,U,BKZ,BKZP,CZ,SZ,CHZB,SHZB,PHI,BK,SA,BL,BM,	WRK	3
	2 AQ,RHO,BETS,PSIWS,BKZI,BKZPI,G(6),AB,C,S,CHB,	WRK	4
20	3 SHB,E,EI,SZCB,SZBC,CZBC,CZCB,SZSB,SZBS,CZBS,CZSB,	WRK	5
	4 ZW,ZT,BKI,Y,Z,Y1,Y2,Y3,Y4	WRK	6
	C	WRK	7
	DIMENSION BX(9),BY(9),PAR(4),PARQ(7),PARR(7),NCELL(3)	WRK	8
	EQUIVALENCE (BX,G),(BZ,BKS)	STR4	7
25	LOGICAL QMAG,RMAG	STR4	8
	INTEGER PI,QI,RI,ELNUM,HMAG	STR4	9
	EXTERNAL STRFCN	STR4	10
	DATA HMAG/3HMAG/	STR4	11
	C	STR4	12
30	C INPUT...K STR4 P Q R MUS/2PI L M	STR4	13
	C	STR4	14
	PSI = PAR(1)*6.283185307	STR4	15
	SL = PAR(2)	STR4	16
	SM = PAR(3)	STR4	17
35	C FIND INDEX OF PREVIOUSLY DEFINED ELEMENTS	STR4	18
	PI = ELNUM(NCELL(1))	STR4	19
	QI = ELNUM(NCELL(2))	STR4	20
	RI = ELNUM(NCELL(3))	STR4	21
	INAME = MNAME(M)	STR4	22
40	QMAG = .FALSE.	STR4	23
	RMAG = .FALSE.	STR4	24
	C IF INPUT ELEMENTS Q AND R ARE MAGNETS, TURN ON MAG SWITCH	STR4	25
	C IF NOT MAGNETS, IT IS ASSUMED THEY ARE CELLS	STR4	26
	IQP = INFF(1,QI)	STR4	27
45	IRP = INFF(1,RI)	STR4	28
	IF (IQP.EQ.HMAG) QMAG = .TRUE.	STR4	29
	IF (IRP.EQ.HMAG) RMAG = .TRUE.	STR4	30
	C GET PARAMETERS FOR OUTER AND INNER ELEMENTS—IN MAG SEQUENCE.	STR4	31
	IF (QMAG) GO TO 9	STR4	32
50	C Q IS CELL. PUT INTO MAG SEQUENCE	STR4	33
	CALL DATA(QI,1,1,7,PARQ)	STR4	34
	PARQ(4) = PARQ(7)	STR4	35
	PARQ(5) = PARQ(6)	STR4	36
	PARQ(6) = PARQ(5)	STR4	37
55	GO TO 10	STR4	38
	C Q IS MAGNET	STR4	39
	9 CALL DATA(QI,1,1,6,PARQ)	STR4	40
		STR4	41

	10	IF (RMAG) GO TO 15	STR4	42
	C	R IS CELL. PUT INTO MAG SEQUENCE	STR4	43
60		CALL DATA(RI,1,1,7,PARR)	STR4	44
		PARR(4) = PARR(7)	STR4	45
		PARR(5) = PARR(6)	STR4	46
		PARR(6) = PARR(5)	STR4	47
		GO TO 20	STR4	48
65	C	R IS MAGNET	STR4	49
	15	CALL DATA(RI,1,1,6,PARR)	STR4	50
	C	COMPUTE BETATRON FUNCTIONS	STR4	51
	20	IF (PAR(4).NE.0.) PARR(1) = PAR(4)	STR4	52
		CALL BET(PI,BX,BY,0)	STR4	53
70		ALPHX = BX(3)	STR4	54
		ALPHY = BY(3)	STR4	55
		BETW = BX(2)	STR4	56
		BETS = BY(2)	STR4	57
	C		STR4	58
75		SZ = SIN(PHI)	STR4	59
		CZ = COS(PHI)	STR4	60
	C	REQUIRED MATRIX ELEMENTS OF COMPLETE REPLACEMENT ARRAY	STR4	61
	C	(HORIZONTAL PLANE) M11=P, M12=0, M21=T, M22=U	STR4	62
		P = (CZ + BX(3)*SZ) * BY(7)/BX(7)	STR4	63
80		Q = BX(7) * BY(7) * SZ	STR4	64
		T = -((1. - BX(3)*BY(3)) * SZ - (BX(3) + BY(3))*CZ)/(BX(7)*BY(7))	STR4	65
		U = (CZ + BY(3)*SZ) * BX(7)/BY(7)	STR4	66
	C		STR4	67
		BX(1) = ABS(PARQ(2))	STR4	68
85		RHO = PARQ(3)	STR4	69
		BX(4) = 1.0/RHO	STR4	70
		BKZ = SQRT((BX(1) + BX(4)) / RHO)	STR4	71
		BKZP = SQRT(ABS(-BX(1) + BX(4)) / RHO)	STR4	72
		BKZI = 1.0/BKZ	STR4	73
90		BKZPI = 1.0/BKZP	STR4	74
	C		STR4	75
		PHI = BKZ*SM	STR4	76
		CZ = COS(PHI)	STR4	77
		SZ = SIN(PHI)	STR4	78
95		Z = BKZP*SM	STR4	79
		E = EXP(Z)	STR4	80
		EI = 1.0/E	STR4	81
		CHZB = 0.5*(E + EI)	STR4	82
		SHZB = 0.5*(E - EI)	STR4	83
100	C	SET UP KOF AND KOD	STR4	84
		BX(1) = SM	STR4	85
		DO 25 I = 2,6	STR4	86
	25	BX(I) = PARQ(I)	STR4	87
		BX(2) = ABS(BX(2))	STR4	88
105		L = M - 3	STR4	89
		CALL STDAT(L,1,1,6,BX)	STR4	90
		CALL MAGNET (L,BX)	STR4	91
		MI(2) = L	STR4	92
		L = M - 4	STR4	93
110		BX(2) = -BX(2)	STR4	94
		CALL STDAT(L,1,1,6,BX)	STR4	95
		CALL MAGNET(L,BX)	STR4	96
		MI(9) = L	STR4	97
	C	SET UP KE	STR4	98

115	L = M - 5	STR4	99
	CALL DRIFT(L,SL)	STR4	100
	MI(1) = L	STR4	101
	MI(10) = L	STR4	102
	C SET UP MATRIX LIST FOR REST OF MODIFIED HALF-CELL.	STR4	103
120	L = M - 7	STR4	104
	MI(3) = L	STR4	105
	MI(8) = L	STR4	106
	MI(4) = M - 2	STR4	107
	L = M - 6	STR4	108
125	MI(5) = L	STR4	109
	MI(6) = L	STR4	110
	MI(7) = M - 1	STR4	111
	DO 30 I = 1,6	STR4	112
30	G(I) = PARR(I)	STR4	113
130	RHO = G(3)	STR4	114
	BKS = SQRT(ABS((G(2) + G(4)/G(3)) / G(3)))	STR4	115
	PHI = BKS*G(1)	STR4	116
	CALL GRT(1,PHI,7,STRFCN)	STR4	117
	IF (IERR) RETURN	STR4	118
135	CALL DRIFT(M-7,SA)	STR4	119
	CALL DRIFT(M-6,BL)	STR4	120
	G(1) = BM	STR4	121
	G(2) = ABS(G(2))	STR4	122
	CALL STDAT(M-1,1,1,6,G)	STR4	123
140	CALL MAGNET (M-1,G)	STR4	124
	G(2) = -G(2)	STR4	125
	CALL STDAT(M-2,1,1,6,G)	STR4	126
	CALL MAGNET(M-2,G)	STR4	127
	CALL MMM(M,10,MI)	STR4	128
145	C OUTPUT	STR4	129
	CALL HED	STR4	130
	SPRO = -G(2)	STR4	131
	DRF = 2.0*BL	STR4	132
	BX(1) = 2.0*(SL + SM + SA + BM + BL)	STR4	133
150	PHAD = PAR(1)*360.	STR4	134
	WRITE(3,2) INAME,NCELL(1)	STR4	135
	WRITE(3,3) PAR(1),PHAD, INAME, INAME,NCELL(2),SM,PSI, INAME, INAME,	STR4	136
	1 NCELL(3),BM	STR4	137
	WRITE(3,4) BETW,ALPHX,SL,BL,BX(1)	STR4	138
155	WRITE(3,5) BETS,ALPHY,SA,DRF	STR4	139
	CALL HED	STR4	140
	C	STR4	141
	2 FORMAT(1X,*4-ELEMENT ANTISYMMETRIC LONG STRAIGHT SECTION ARRAY *,	STR4	142
	1 A1,* , REPLACES LATTICE FROM *A3,* TO NEXT ANTI HOMOLOGOUS *	STR4	143
160	2 *POINT.*)	STR4	144
	3 FORMAT(3X,* PHASE ADVANCE = *,F9.5,* 2PI*/	STR4	145
	1 18X,* = *,F9.5,* DEG.*,10X,*OUTER ELEMENTS *,A1,*0F,*,A1,	STR4	146
	2 *OD ARE LIKE MAGNETS OF *A3,* , BUT WITH LENGTH MI = *,F14.8,*,*/	STR4	147
	3 18X,* = *,F9.5,* RAD.*,10X,*INNER ELEMENTS *,A1,*IF,*,A1,	STR4	148
165	4 *ID ARE LIKE MAGNETS OF *,A3,* , BUT WITH LENGTH M2 = *,F14.8,	STR4	149
	5 *.*)	STR4	150
	4 FORMAT(4X,*BETAX = *,F14.8,2X,*ALPHAX = *,F14.8,* LEND = *,F14.8,	STR4	151
	1 * LDRF/2 = *,F14.8,11X,*LTOT = *,F14.8)	STR4	152
	5 FORMAT(4X,*BETAY = *,F14.8,* ALPHAY = *,F14.8,* LSEP = *,	STR4	153
170	1 F14.8,* LDRF = *,F14.8)	STR4	154
	RETURN	STR4	155

END

STR4

156

1	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
5	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
10	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/SWTC/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	SWTCH	2
15	LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS	SWTCH	3
		SWTCH	4
	COMMON /DIM/ LEND,MEND,ISAV,ISAV7,M7END,KADD,KADDR,MUNIT,MSYML	DIM	2
		DIM	3
	DIMENSION RX(2,3),RY(2,3),AX(2,3),AY(2,3),RW(3)	STXY	9
20	DIMENSION RPQ(4),I(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
25	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
30	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
35	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
40	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
45	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
50	LQ = INFF(10,M)	STXY	40
	NQ = INFF(11,M)	STXY	41
	KIND = INFF(20,M)	STXY	42
	IF (KIND.NE.3HMYX) GO TO 11	STXY	43
		STXY	44
55	C CHECK STORAGE REQUIREMENT	STXY	45
	NNQ = 12 + KADDR	STXY	46
	IF (NNQ.GT.NQ) GO TO 20	STXY	47

	NJ = LQ + KADDR	STXY	48
	NI = NJ + 6	STXY	49
60	DO 10 IJ=1,3	STXY	50
	DO 10 II=1,2	STXY	51
	STORE(NJ)=RX(II,IJ)	STXY	52
	STORE(NI)=RY(II,IJ)	STXY	53
	NJ=NJ+1	STXY	54
65	NI=NI+1	STXY	55
	10 CONTINUE	STXY	56
	CALL STDAT(M,5,1,3,RW)	STXY	57
	CALL STDAT(M,5,4,4,RPQ)	STXY	58
	RETURN	STXY	59
70		STXY	60
	20 WRITE (3,1) M	STXY	61
	1 FORMAT (55H *** ERROR *** INSUFFICIENT STORAGE RESERVED AT INDEX	STXY	62
	1I5)	STXY	63
	ERROR=.TRUE.	STXY	64
75	RETURN	STXY	65
		STXY	66
	11 CALL CONVMAT(7,T,RX,RY,RW,RPQ)	STXY	67
	CALL STOR7(M,T,RW)	STXY	68
	RETURN	STXY	69
80	END	STXY	70

1		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
5	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
10	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
15	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
20		CONTRL	5
	COMMON/SWCH/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
25	* LMAX AND MAX ARE DIMENSIONS OF STORE AND INFF	STORE	3
		STORE	4
	COMMON/COPY/CPYSW	COPY	2
	LOGICAL CPYSW	COPY	3
		COPY	4
30	LOGICAL MOSS	SUBR	12
	INTEGER BDAT, ELNUM	SUBR	13
	DIMENSION INSUB(10), ITNO(10)	SUBR	14
	LOGICAL XSW	SUBR	15
		SUBR	16
35	XSW = .FALSE.	SUBR	17
	IF (OP.LT.0) XSW = .TRUE.	SUBR	18
	MOSS = .FALSE.	SUBR	19
	C	SUBR	20
	EMPTY=.FALSE.	SUBR	21
40	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
45	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
50	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
55	XEQ=.FALSE.	SUBR	37
	MSUBR=M	SUBR	38
	CALL RESRV(M,0,0,	SUBR	39

	. 4,3,0,0,0,0,0,0	SUBR	40
	.)	SUBR	41
60	RETURN	SUBR	42
	C END ENTRY	SUBR	43
	70 CONTINUE	SUBR	44
	GO TO (4,2,3),MODE	SUBR	45
	4 CONTINUE	SUBR	46
65	IF (MSSW) GO TO 25	SUBR	47
	ERROR = .TRUE.	SUBR	48
	WRITE (3,53)	SUBR	49
	RETURN	SUBR	50
	53 FORMAT (70H ERROR****INSTRUCTION NAMED END MUST BE PRECEDED BY A S	SUBR	51
70	1UB INSTRUCTION.)	SUBR	52
	C SET TRA TO MSUBR	SUBR	53
	2 CALL INFW(MSUBR,9,M)	SUBR	54
	IF (.NOT.CPYSW) GO TO 30	SUBR	55
	WRITE (3,54) NAME,OPNAME,KA,KB,(ICARD(IW),IW=3,8)	SUBR	56
75	CALL HED	SUBR	57
	CALL SKIP1	SUBR	58
	30 MODE = 1	SUBR	59
	XEQ=.TRUE.	SUBR	60
	RETURN	SUBR	61
80	C MOSS CONTROLS MODE TO ALLOW PROPER FLOW OF PROGRAM WHEN MESH OR	SUBR	62
	C VPAR ARE OPERATIVE.	SUBR	63
	25 MOSS = .TRUE.	SUBR	64
	3 MSUBR = INFF(9,M)	SUBR	65
	KT = IDAT(MSUBR,3)	SUBR	66
85	ICOUNT = IDAT(MSUBR,4)	SUBR	67
	ICOUNT = ICOUNT -1	SUBR	68
	KT = KT + 1	SUBR	69
	CALL REPINT(MSUBR,3,KT)	SUBR	70
	CALL REPINT(MSUBR,4,ICOUNT)	SUBR	71
90	CALL REPINT(MSUBR,1,0)	SUBR	72
	CALL REPINT(MSUBR,2,0)	SUBR	73
	NAMESR=MNAME(MSUBR)	SUBR	74
	IF(ICOUNT-1)5,6,7	SUBR	75
	5 MCALL = INFF(9,MSUBR)	SUBR	76
95	ISNB=ISNB-1	SUBR	77
	M=MCALL	SUBR	78
	IF (MOSS) GO TO 8	SUBR	79
	IF(MODS) MODE=1	SUBR	80
	IF(.NOT.MODS) MODE=INFF(19,MCALL)	SUBR	81
100	GO TO (8,9,10),MODE	SUBR	82
	8 XEQ=.TRUE.	SUBR	83
	STOR=.TRUE.	SUBR	84
	RSRV=.TRUE.	SUBR	85
	RETURN	SUBR	86
105	9 STOR=.TRUE.	SUBR	87
	RSRV=.TRUE.	SUBR	88
	XEQ=.FALSE.	SUBR	89
	RETURN	SUBR	90
	10 STOR=.FALSE.	SUBR	91
110	RSRV=.FALSE.	SUBR	92
	XEQ=.TRUE.	SUBR	93
	RETURN	SUBR	94
	6 CALL REPINT(MSUBR,2,1)	SUBR	95
	7 TRA=MSUBR-1	SUBR	96

115	TRASW=.TRUE.	SUBR	97
	DO 20 IS=1,10	SUBR	98
	IST = IS	SUBR	99
	IF (INSUB(IS).EQ.NAMESR) GO TO 21	SUBR	100
	20 CONTINUE	SUBR	101
120	C ERROR HERE AS NAME SHOULD MATCH. PRINT MESSAGE AND CONTINUE	SUBR	102
	WRITE (3,105) NAMESR	SUBR	103
	105 FORMAT (/5X,A5,29H SUBROUTINE NOT IN PRINT LIST//)	SUBR	104
	CALL HED	SUBR	105
	RETURN	SUBR	106
125	21 ITNO(IST)=KT	SUBR	107
	IGO = 2	SUBR	108
	GO TO 200	SUBR	109
	22 RETURN	SUBR	110
	C PRINT ITERATION NUMBERS OF SUBROUTINES	SUBR	111
130	C	SUBR	112
	200 CONTINUE	SUBR	113
	IPMAX = ISNB-1	SUBR	114
	IF (IPMAX.EQ.0) WRITE (3,102) INSUB(1),ITNO(1)	SUBR	115
	102 FORMAT (/9X,4HSUB.,2X,A5,7H, ITER.,2X,I3)	SUBR	116
135	IF (IPMAX.EQ.1) WRITE (3,100) INSUB(1),ITNO(1),INSUB(2),ITNO(2)	SUBR	117
	100 FORMAT (/9X,4HSUB.,2X,A5,7H, ITER.,2X,I3,10X,4HSUB.,2X,A5,	SUBR	118
	1 7H, ITER.,2X,I3)	SUBR	119
	IF (IPMAX.GE.2) GO TO 104	SUBR	120
	GO TO 110	SUBR	121
140	104 CONTINUE	SUBR	122
	DO 201 IS=2,IPMAX	SUBR	123
	WRITE (3,101) INSUB(IS),ITNO(IS),INSUB(IS+1),ITNO(IS+1)	SUBR	124
	101 FORMAT (15X,A5,9X,I3,16X,A5,9X,I3)	SUBR	125
	201 CONTINUE	SUBR	126
145	110 CONTINUE	SUBR	127
	GO TO (55,22),IGO	SUBR	128
	C	SUBR	129
	C CALL ENTRY	SUBR	130
	80 CONTINUE	SUBR	131
150	IF(MODE.EQ.3) GO TO 11	SUBR	132
	IF (MHS) GO TO 11	SUBR	133
	IF (VPR) GO TO 11	SUBR	134
	IF (CPYSW) CALL SKIP1	SUBR	135
	CALL INP(M,0,0,	SUBR	136
155	. 1,2,0,0,0,0,0,0,	SUBR	137
	.)	SUBR	138
	IF (CPYSW) CALL SKIP1	SUBR	139
	IF(MODE.EQ.2) RETURN	SUBR	140
	IF (XSW) RETURN	SUBR	141
160	C MODE IS 1. BEGINNING OF SUBROUTINE LOOP OR LOOPS	SUBR	142
	C ZERO ARRAYS FOR STORAGE OF NAMES AND ITERATION NUMBERS.	SUBR	143
	81 CONTINUE	SUBR	144
	DO 12 KS=1,10	SUBR	145
	INSUB(KS)=0	SUBR	146
165	ITNO(KS)=0	SUBR	147
	12 CONTINUE	SUBR	148
	ISNB=0	SUBR	149
	11 NAMESR = BDAT(M,1)	SUBR	150
	MSUBR = ELMUM(NAMESR)	SUBR	151
170	IF (MHS) GO TO 55	SUBR	152
	IF (VPR) GO TO 55	SUBR	153

		ISNB = ISNB+1	SUBR	154
		INSUB(ISNB)=NAMESR	SUBR	155
		ITNO(ISNB)=1	SUBR	156
175	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
180	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
185		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
190		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
195		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.

1	SUBROUTINE SUM(K,S,SM)	SUM	2
	DIMENSION S(1)	SUM	3
	SM=0.	SUM	4
	DO 10 I=1,K	SUM	5
5	SM = SM + S(I)	SUM	6
10	CONTINUE	SUM	7
	RETURN	SUM	8
	END	SUM	9

1	SUBROUTINE SWITCH	SWITCH	2
		SWITCH	3
	* CONTROLS CALLS TO DIFFERENT SWITCHES OR OVERLAYS, ACCORDING TO	SWITCH	4
	* OP OF SYNCH INSTRUCTION.	SWITCH	5
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
10	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
15	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
20	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
25	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
30	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
35	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
40	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
45	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
50	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
55	50 FORMAT (5X,3H***,I5,* IS AN INVALID OP NUMBER. SYNCH INSTRUCTIO	SWITCH	48
	1N NAMED *A5)	SWITCH	49
	ERROR = .TRUE.	SWITCH	50

RETURN
END

SWITCH 51
SWITCH 52

1			SXTX	2
		SUBROUTINE SXTX(M,V,PAR)	SXTX	3
	C	SX SXTX KB L D2B BRHO EXACT	SXTX	4
	C	-----	SXTX	5
5	C	SX IS A SEXTUPOLE THAT ACTS AT A POINT. IT REPRESENTS THE	SXTX	6
	C	EFFECT OF A SEXTUPOLE OF LENGTH L, WITH D2B = D2B/DX2.	SXTX	7
	C	IF L=0 ON INPUT, L IS SET =1.	SXTX	8
	C	EXACT IS AN OPTION TO CALL FOR AN EXACT CALCULATION, USING	SXTX	9
	C	ELLIPTIC FUNCTIONS ACCORDING TO A METHOD OF B. AUTIN. IT IS	SXTX	10
10	C	EXERCISED BY SETTING EXACT TO A NON-ZERO VALUE, BUT IS ONLY	SXTX	11
	C	OBEYED IF THE RAY IS IN THE HORIZONTAL PLANE AND L IS NOT ZERO.	SXTX	12
	C	-----	SXTX	13
	C		SXTX	14
	C	IF KB NOT 0, SETS R(2,6) = R(4,6) = 0.	SXTX	15
15	C		SXTX	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,MSYML	DIM	2
			DIM	3
20			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
25			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)	SXTX	19
		EQUIVALENCE (X,W(1)),(PX,W(2)),(Y,W(3)),(PY,W(4)),	SXTX	20
30	1	(L,P(1)),(D2B,P(2)),(BRHO,P(3))	SXTX	21
	2	,(LSX,MAG(1)),(K,MAG(2)),(ONE,MAG(3)),(CURV,MAG(4)),	SXTX	22
	3	(EPS0,MAG(5)),(EPS1,MAG(6))	SXTX	23
		REAL L,LSX,LH,K,KP,MAG	SXTX	24
		DATA ONE,S3,RADEG/1.0,1.732050808,0.0174532925/	SXTX	25
35		DATA S32,SS3,S13,S23/.433012702,1.44224957,.33333333,3.464101616/	SXTX	26
	C		SXTX	27
		IF(INFF(4,M).EQ.0) GO TO 15	SXTMPOL	3
		CALL NPOL(M,V,PAR)	SXTMPOL	4
		RETURN	SXTMPOL	5
40	15	CONTINUE	SXTMPOL	6
		DO 1 I=1,4	SXTX	28
		W(I)=V(I)	SXTX	29
	1	P(I)=PAR(I)	SXTX	30
		DPP = V(6)	SXTX	31
45		FAC = 1. + DPP	SXTX	32
		BR = FAC * BRHO	SXTX	33
		F = D2B / BR	SXTX	34
		KP=F	SXTX	35
	C	IF KB=0,TURN OFF COLUMN 6	SXTX	36
50		KB = INFF(5,M)	SXTX	37
			SXTX	38
		IF(X.EQ.0..AND.PX.EQ.0..OR.KP.EQ.0.) GO TO 11	SXTX	39
		IF(P(4).NE.0..AND.L.NE.0..AND.Y.EQ.0..AND.PY.EQ.0.) GO TO 5	SXTX	40
	11	IF (L.EQ.0.) GO TO 2	SXTX	41
55		LH = L/2.	SXTX	42
		X = X + PX * LH	SXTX	43
		Y = Y + PY * LH	SXTX	44

		F = L * F	SXTP	45
			SXTP	46
60	2	CONTINUE	SXTP	47
		DPX=(F/2.)*(X*X-Y*Y)	SXTP	48
		DPY=F*X*Y	SXTP	49
		V(2) = V(2) + DPX	SXTP	50
		V(4) = V(4) + DPY	SXTP	51
65		IF (L.EQ.0) GO TO 3	SXTP	52
		V(1) = X + V(2)*LH	SXTP	53
		V(3) = Y + V(4)*LH	SXTP	54
			SXTP	55
	3	IF (M.EQ.0) RETURN	SXTP	56
70		CALL RTRV7(MUNIT,R,RW)	SXTP	57
		T21 = - F * X	SXTP	58
		T23 = F * Y	SXTP	59
		R(2,1) = T21	SXTP	60
		R(2,3) = T23	SXTP	61
75		R(4,1) = T23	SXTP	62
		R(4,3) = - T21	SXTP	63
		R(2,6)=-DPX	SXTP	64
		R(2,7)=-DPX	SXTP	65
		R(4,6)=-DPY	SXTP	66
80		R(4,7)=-DPY	SXTP	67
		R(5,1)=DPX	SXTP	68
		R(5,3)=DPY	SXTP	69
		IF (KB.EQ.0) GO TO 7	SXTP	70
		R(2,6) = 0.	SXTP	71
85		R(4,6) = 0.	SXTP	72
	7	RW(2) = -DPX	SXTP	73
			SXTP	74
		IF (L.EQ.0.) GO TO 4	SXTP	75
		ML(1)=MEND-3	SXTP	76
90		ML(2)=M	SXTP	77
		ML(3)=MEND-3	SXTP	78
		CALL DRIFT(MEND-3,LH)	SXTP	79
		CALL STOR7(M,R,RW)	SXTP	80
		CALL MMM(M,3,ML)	SXTP	81
95		RETURN	SXTP	82
			SXTP	83
	4	CONTINUE	SXTP	84
		CALL STOR7(M,R,RW)	SXTP	85
		RETURN	SXTP	86
100			SXTP	87
	C	EXACT CALCULATION FOR THICK SEXTUPOLE, H-PLANE ONLY. — B.AUTIN	SXTP	88
	5	CONTINUE	SXTP	89
	C		SXTP	90
	C	CONDITIONS INITIALES C	SXTP	91
105	C		SXTP	92
		SK=SIGN(1.,KP)	SXTP	93
		AK=(ABS(KP))**S13	SXTP	94
		U=KP*PX	SXTP	95
		SU=-SIGN(1.,U)	SXTP	96
110		C=X*X*X*KP/3.+PX*PX	SXTP	97
		IF(C.EQ.0.) GO TO 99	SXTP	98
		AC=ABS(C)**S13	SXTP	99
		SC=SIGN(1.,C)	SXTP	100
		AXM=AC*SS3/AK	SXTP	101

115		XM=SC*SK*AXM	SXTP	102
		TM=SQRT(S3*AC)*AK/SS3	SXTP	103
	C		SXTP	104
	C	MODULE K2	SXTP	105
	C		SXTP	106
120		XK2=.5+SC*S32	SXTP	107
		XK=SQRT(XK2)	SXTP	108
	C		SXTP	109
	C	LIGNES TRIGONOMETRIQUES DE PHI0	SXTP	110
	C		SXTP	111
125		T=ABS(1.-X/XM)/S3	SXTP	112
		D=1.+T	SXTP	113
		CT=(1.-T)/D	SXTP	114
		ST=SU*2.*SQRT(T)/D	SXTP	115
		ST2=ST*ST	SXTP	116
130		DT=SQRT(1.-XK2*ST2)	SXTP	117
	C		SXTP	118
	C	FONCTIONS DE JACOBI	SXTP	119
	C		SXTP	120
		CALL QFUNCT(XK,XKPRIM,Q)	SXTP	121
135		XK=SQRT(XK)	SXTP	122
		XKP=SQRT(XKPRIM)	SXTP	123
		Z=TM*L	SXTP	124
		TH3=THETA3(0.,Q)	SXTP	125
		U=Z/(TH3*TH3)	SXTP	126
140		TH4=THETA4(U,Q)	SXTP	127
		XT=XK*TH4	SXTP	128
		SN=THETA1(U,Q)/XT	SXTP	129
		SN2=SN*SN	SXTP	130
		CN=XKP*THETA2(U,Q)/XT	SXTP	131
145		DN=XKP*THETA3(U,Q)/TH4	SXTP	132
	C		SXTP	133
	C	X1	SXTP	134
	C		SXTP	135
		C1=(CT*CN-ST*SN*DT*DN)/(1.-XK2*ST2*SN2)	SXTP	136
150		XX=SC*S3*(1.-C1)/(1.+C1)	SXTP	137
		X1M=1.-XX	SXTP	138
		X1=X1M*XM	SXTP	139
	C		SXTP	140
	C	XPRIME1	SXTP	141
155			SXTP	142
		S1=ST*CN*DN+CT*SN*DT	SXTP	143
		U=KP*S1	SXTP	144
		SU=-SIGN(1.,U)	SXTP	145
		CX=ABS(C*XX)	SXTP	146
160		PX1=SU*SQRT(CX*(1.+X1M+X1M*X1M))	SXTP	147
		GO TO 100	SXTP	148
	C		SXTP	149
	C	CAS PARTICULIER C=0	SXTP	150
	C		SXTP	151
165	99	XM=-SK/AK	SXTP	152
		TM=AK/S23	SXTP	153
		T=SU*SQRT(X/XM)	SXTP	154
		T1=T/(1.-TM*T*L)	SXTP	155
		T12=T1*T1	SXTP	156
170		X1=XM*T12	SXTP	157
		PX1=-SK*T12*T1/S3	SXTP	158

	100	V(1)=X1	SXTP	159
		V(2)=PX1	SXTP	160
175		IF(M.EQ.0) RETURN	SXTP	161
	C	MAG(I) = LSX,K,1.0,CURV,EPS0,EPS1 — INPUT TO S/R MAGNET	SXTP	162
		LSX=L	SXTP	163
		THETA=PX-PX1	SXTP	164
		XAV=(X+X1)/2. + THETA*L/12.	SXTP	165
180		K=KP*XAV	SXTP	166
		CURV=THETA/L	SXTP	167
		EPS0=PX/RADEG	SXTP	168
		EPS1=-PX1/RADEG	SXTP	169
		CALL MAGNET(M,MAG)	SXTP	170
185		RETURN	SXTP	171
		END	SXTP	172
			SXTP	173
			SXTP	174

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.

1		SXTP	175
	SUBROUTINE TABULAT(M,TABSW)	TABUL	2
		BLANK	2
	LEVEL 2, STORE,INFF,IWORK	BLANK	3
5	COMMON STORE(48000),IWORK(10)	86MARSIZ	1
	DIMENSION INFF(24,2000)	86MARSIZ	2
	EQUIVALENCE (INFF,STORE)	86MARSIZ	3
		BLANK	5
	COMMON/SWTC/BEND,MSIZE,CYCSWT,VCSW,PV,MSSW,VPR,MHS,MODS,MINZER	SWTC	2
10	LOGICAL CYCSWT,MSSW,VCSW,PV,VPR,MHS,MODS	SWTC	3
		SWTC	4
	EQUIVALENCE(IVAR,FVAR)	TABUL	5
	LOGICAL TABSW	TABUL	6
	DIMENSION INDX(10),NAM(10),JND(10),F(10)	TABUL	7
15	INTEGER ELMUM,BDAT	TABUL	8
		TABUL	9
	KA = INFF(4,M)	TABUL	10
	KB = INFF(5,M)	TABUL	11
	KS = KA*60	TABUL	12
20	C IF KB.GT.0, SET IPR=KB, WHICH GOES TO INN(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
25	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
30	INDX(I) = ELMUM(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
35	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.007777777777777777	TABUL	30
	LNUM= SHIFT(LNUM,-48)	TABUL	31
	JND(I) = LNUM	TABUL	32
40	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
45	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
50	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
55	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
60	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
65	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
70	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
75	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
80	GO TO 120	TABUL	73
	END	TABUL	74

1

SUBROUTINE SYCONJ(T,TSC,XTSC)
RETURN
END

SYCONJ 2
SYCONJ 3
SYCONJ 4

1

SUBROUTINE TEST(M)
RETURN
END

TEST 2
TEST 3
TEST 4

1	FUNCTION THETA1(U,Q)	THETA1	2
	A=SIN(U)	THETA1	3
	GO TO 1	THETA1	4
	ENTRY THETA2	THETA1	5
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
10	B=A	THETA1	11
	QM=1.	THETA1	12
	GO TO 3	THETA1	13
	ENTRY THETA3	THETA1	14
	A=COS(U+U)	THETA1	15
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
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
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
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
35	END	THETA1	36

1		THETA1	37
	SUBROUTINE TINV7(TINV,T)	TINV7	2
	C INVERSE OF 7X7 SYMPLECTIC MATRIX	TINV7	3
	C 6X6 MATRIX INVERSE OF S = SC = - U X STR X U K=(T(I,7),I=1,6)	TINV7	4
5	C (SC -SC X K)	TINV7	5
	C 7X7 MATRIX INVERSE OF T =	TINV7	6
	C (0 1)	TINV7	7
	DIMENSION TINV(7,7),T(7,7), STR(6,6),U(6,6),SC(6,6),VK(6)	TINV7	8
	C	TINV7	9
10	C STR IS TRANSPOSE OF 6X6 OF T	TINV7	10
	DO 10 I = 1,6	TINV7	11
	DO 10 J = 1,6	TINV7	12
	STR(I,J) = T(J,I)	TINV7	13
	10 CONTINUE	TINV7	14
15	C SET UP U ARRAY	TINV7	15
	DO 15 J = 1,6	TINV7	16
	DO 15 I = 1,6	TINV7	17
	U(I,J)= 0.0	TINV7	18
	15 CONTINUE	TINV7	19
20	DO 20 I =1,5,2	TINV7	20
	U(I,I+1) = -1.	TINV7	21
	U(I+1,I) = +1.	TINV7	22
	20 CONTINUE	TINV7	23
	CALL MXM6 (STR,U,SC)	TINV7	24
25	CALL MXM6 (U,SC,SC)	TINV7	25
	DO 25 J = 1,6	TINV7	26
	DO 25 I = 1,6	TINV7	27
	TINV(I,J) = - SC(I,J)	TINV7	28
	25 CONTINUE	TINV7	29
30	C SET UP VECTOR K	TINV7	30
	DO 30 I = 1,6	TINV7	31
	VK(I) = T(I,7)	TINV7	32
	30 CONTINUE	TINV7	33
	CALL MXV6(SC,VK,VK)	TINV7	34
35	DO 35 I = 1,6	TINV7	35
	TINV(I,7) = VK(I)	TINV7	36
	TINV(7,I) = 0.0	TINV7	37
	35 CONTINUE	TINV7	38
	TINV(7,7) = 1.0	TINV7	39
40	RETURN	TINV7	40
	END	TINV7	41

1	SUBROUTINE TRNSPOS	TRNSPOS	2
		BLANK	2
	LEVEL 2, STORE, INFF, IWORK	BLANK	3
	COMMON STORE(48000), IWORK(10)	86MARSIZ	1
5	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
10	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
15	C	TRNSPOS	7
	IL = LQ	TRNSPOS	8
	IRR = LFL - 1	TRNSPOS	9
	C	TRNSPOS	10
	DO 1 J=1,KB	TRNSPOS	11
20	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
25	1 IR = IR + KB	TRNSPOS	17
	RETURN	TRNSPOS	18
	END	TRNSPOS	19

1	SUBROUTINE TRK(M)	TRK	2
	LEVEL 2, STORE, INFF, IWORK	BLANK	2
	COMMON STORE(48000), IWORK(10)	BLANK	3
5	DIMENSION INFF(24,2000)	86MARSIZ	1
	EQUIVALENCE (INFF,STORE)	86MARSIZ	2
		86MARSIZ	3
		BLANK	5
	COMMON/BMI/MI(5000)	BMIL	1
	COMMON/NELS/NELS	BMIL	2
10	LEVEL 2,MI	BMIL	3
		BMIL	4
		BMI	3
	COMMON /DIM/ LEND, MEND, ISAV, ISAV7, M7END, KADD, KADDR, MUNIT, MSYMP	DIM	2
		DIM	3
15	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
20	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
25	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
30	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
35	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
40	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
45	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
50	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
55	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
60	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
65	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
70	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
75	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
80	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
85		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
90		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
95		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
100	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
105		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
110		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

115	IMODE = IDAT(M,4)	TRK	94
	IPR = IDAT(M,5)	TRK	95
	IPL = IDAT(M,6)	TRK	96
	IF (IMODE.EQ.0) IMODE = 1	TRK	97
	IF (IMODE.NE.2) GO TO 30	TRK	98
120	NPARTS = 1	TRK	99
	CALL RTRV7(MUNIT,R7,RW)	TRK	100
	CALL STOR7(M,R7,RW)	TRK	101
	30 IF (NBML.NE.IBLNK) GO TO 402	TRK	102
	NELS = IABS(NELS)	TRK	103
125	GO TO 401	TRK	104
	402 CONTINUE	TRK	105
	MBML=MDAT(M,2)	TRK	106
	C PUT BEAM LIST IN MI ARRAY	TRK	107
	NDAT=INFF(17,MBML)	TRK	108
130	CALL MIFILL(MBML,1,NDAT,NELS,MI)	TRK	109
	IF(ERROR) RETURN	TRK	110
	C FIND BETAS FOR REFERENCE MATRIX MAT	TRK	111
	401 NMAT=BDAT(M,3)	TRK	112
	IF (NMAT.EQ.5H) GO TO 403	TRK	113
135	MEL=MDAT(M,3)	TRK	114
	CALL BET(MEL,BX,BY,0)	TRK	115
	403 CONTINUE	TRK	116
	PLT=.FALSE.	TRK	117
	DPR=.FALSE.	TRK	118
140	IF (IPR.LT.0) DPR = .TRUE.	TRK	119
	IF (IPL.GT.0) PLT = .TRUE.	TRK	120
	KREV = IOF/1000	TRK	121
	KPOS = MOD(IOF,1000)	TRK	122
	IF (PLT) WRITE(5) NPARTS,NRR,NVECS,NBML,NMAT,KPOS,NVEC0,	TRK	123
145	1 (U(J),J=1,7)	TRK	124
		TRK	125
	DO 250 IP=1,NPARTS	TRK	126
	BEG=.TRUE.	TRK	127
	IPS=IP*7-6	TRK	128
150	CALL DATA(MVECS,1,IPS,7,V)	TRK	129
	IGO = 0	TRK	130
	IF (IMODE.EQ.2) IGO = 1	TRK	131
	10 IF (IMP.EQ.0) GO TO 8	TRK	132
	EX=V(1)*1.E-6	TRK	133
155	EY=V(3)*1.E-6	TRK	134
	PHIX=V(2)*RADEG	TRK	135
	PHIY=V(4)*RADEG	TRK	136
	AX=SQRT(EX*BX(2))	TRK	137
	AY=SQRT(EY*BY(2))	TRK	138
160	C=COS(PHIX)	TRK	139
	S=SIN(PHIX)	TRK	140
	V(1)=AX*C	TRK	141
	V(2)=-AX*(S+BX(3)*C)/BX(2)	TRK	142
	C=COS(PHIY)	TRK	143
165	S=SIN(PHIY)	TRK	144
	V(3)=AY*C	TRK	145
	V(4)=-AY*(S+BY(3)*C)/BY(2)	TRK	146
	8 IF (NREF.EQ.1) GO TO 81	TRK	147
	DO 41 I=5,7	TRK	148
170	41 V(I) = 0.	TRK	149
	81 DO 82 J=1,7	TRK	150

	82	VV(J) = V(J)	TRK	151
	83	DO 5 J=1,7	TRK	152
		U(J) = V0(J) + V(J)	TRK	153
175	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
180	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
185	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
190		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
195		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
200		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
205		CALL RANGET(NRN)	TRK	184
		DO 212 IBS=1,NELF	TRK	185
		NEL=MI(IBS)	TRK	186
		IPOS=IBS-1	TRK	186
	C	IF INTEGRATION STEPS ARE BEING PRINTED, OMIT TRK PRINT.	TRK	187
210		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
215	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
220		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
225		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
230	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
235	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
240	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
245	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
250	215 CONTINUE	TRK	228
	IF(.NOT.PLT) GO TO 211	TRK	229
	IF(IFS.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
255	C ADVANCE THROUGH BEAM SYSTEM	TRK	231
	211 CONTINUE	TRK	232
	IF (IFS.GT.NELS) GO TO 212	TRK	233
	NEL = MI(IFS)	TRK	234
	CALL TRKCN(M,NEL,U,0,IMODE)	TRK	235
260	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
265	220 CALL RTRV7(M,R7,RW)	TRK	241
	DPRSAV=DPR	TRK	242
	DPR=.TRUE.	TRK	243
	DLP = -DEL P	TRK	244
	DO 222 I=1,7	TRK	245
270	222 UF(I) = U(I)	TRK	246
	GO TO 230	TRK	247
	225 DLP = + DEL P	TRK	248
	DO 227 I=1,7	TRK	249
	227 UM(I) = U(I)	TRK	250
275	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
280	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
285	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
290	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
295	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
300		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
305	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
310	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	
315	1012	FORMAT (1H0,3X,A5,I2)		TRK	291
	1014	FORMAT (10X,2I5,F11.3,8F12.5)		TRK	292
		END		TRK	293

1	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
5	REWIND 6	MY3TKPL	5
	XR1 = SCALE(1)	MY3TKPL	6
	YU1 = SCALE(2)	MY3TKPL	7
	XR2 = SCALE(3)	MY3TKPL	8
	YU2 = SCALE(4)	MY3TKPL	9
10	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
15	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
20	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
25	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
30	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
35	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
40	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
45	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
50	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
55	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
60		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
65		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.

1	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
5	XR=XTOP-XL	MY3TKPL	71
	YR=YTOP-YL	MY3TKPL	72
	XLOW=XL	MY3TKPL	73
	XRANGE=XR	MY3TKPL	74
	YLOW=YL	MY3TKPL	75
10	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
15	RETURN	MY3TKPL	81
	END	MY3TKPL	82