132
1-30-78

BNL 50741

UC-28

(Particle Accelerators and
High-Voltage Machines -TID 4500)

Dr. 1792

MASTER

HOPI - ON-LINE INJECTION OPTIMIZATION PROGRAM

J-L LeMaire*

October 26, 1977

* On leave from CEN, Saclay, France

ACCELERATOR DEPARTMENT

BROOKHAVEN NATIONAL LABORATORY
ASSOCIATED UNIVERSITIES, INC.
UPTON, NEW YORK 11973

under contract No. EY-76-C-02-0016 with the
UNITED STATES DEPARTMENT OF ENERGY

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

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, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness 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. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

N O T I C E

This report was prepared as an account of work sponsored by the United States Government. Neither the U.S. nor the U.S. Department of Energy, 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 usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

PRINTED IN THE UNITED STATES OF AMERICA
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

Price: Domestic \$ 4.50
Microfiche \$ 3.00

ABSTRACT

A method of matching the beam from the 200 MeV linac to the AGS is presented. It has the advantage of allowing one to forget about emittance measurements. An on-line computer program written on the PDP10 computer performs the matching by modifying independently the horizontal and vertical emittance. Experimental results show success with this method, which can be applied to any matching section.

Introduction

It is convenient, for the purpose of injection into the AGS, to change the emittance orientation in the horizontal and vertical planes independently. The matching problem is then simplified. The present manner of tuning the HEBT IV line consists of changing at random each of the 6 quadrupoles of the matching section to maximize the beam circulating in the AGS. This paper describes a more rational and rapid method to perform the matching. No understanding of beam transport is actually required. The magnetic strength of any of the quadrupoles is varied and an interactive program named HOPI (Help Operators Performing Injection) calculates the settings for the remaining quadrupoles in order to keep constant either the orientation of one plane or both orientation and size.

Matching Section - Method of Optimization

1. HEBT IV - Matching Section

The matching transport line is made of 6 quadrupoles^(*) with alternating defocusing and focusing^(†) and several steering dipoles (Fig. 7). The calculation assumes that the beam is well steered through the center of the quadrupoles so that the beam center does not undergo any motion. Most of the time, this can be done easily with the dipoles ND321, ND348 and the pitching magnets NP319 and NP343 using as instrumentation the beam position monitors BPM8 and BPM9. (NT352 - NT377)

For given running conditions, the transport matrices between an origin chosen arbitrarily upstream of Q18 and the injection point (exit of the inflector) are $H = (h_{ij})$ and $V = (v_{ij})$, $i, j = 1, 2$. We define the parameters that we want to keep constant in terms of those transport matrix elements. The Courant and Snyder parameters α , β , γ and the emittance concept are used to represent the beam. The emittance keeps its elliptical shape along the transport line. At this stage, space charge is not considered, but can be added without difficulties, at the expense of computing time. The transport matrix for the CS parameters are:

* The quadrupoles are Q18, Q19, Q20, Q21, Q22, Q23 or NQ381, NQ399, NQ411, NQ423, NQ434, NQ446

† In the horizontal plane

$$\begin{pmatrix} t_{11}^2 & -2t_{11}t_{12} & t_{12}^2 \\ -t_{21}t_{11} & 1+2t_{12}t_{21} & -t_{12}t_{22} \\ t_{21}^2 & -2t_{22}t_{21} & t_{22}^2 \end{pmatrix} \quad \text{where } t_{ij} \text{ are either } h_{ij} \text{ or } v_{ij}$$

Two modes per plane are considered (which makes a total of 4 modes). They are presented in the following.

Mode 1: This mode allows one to modify the orientation of the horizontal emittance, without affecting either the horizontal beam size or the vertical beam characteristics. In order to do so, we have to keep constant the elements h_{11} , h_{12} , v_{11} , v_{12} , v_{21} which makes 5 constraints for this problem.

Mode 2: This mode gives the possibility to vary the parameters α and β (orientation and size) of the vertical plane without changing the horizontal plane. We have to maintain constant the elements v_{11} , v_{12} , v_{21} which makes 3 constraints.

Mode 3: This mode is the complement of Mode 1 so that just the orientation of the vertical emittance is affected.

Mode 4: This mode is the complement of Mode 2. The horizontal plane is varied without changing the vertical plane.

2. Method of Optimization

We have seen that the requirements lead to a maximum of 5 constraints which we call $T_k(p_j)$ $k = 1, 5$. The T_k are a subset of the matrix elements h_{11} , h_{12} , h_{21} , v_{11} , v_{12} , v_{21} . The p_j parameters to adjust are the 6 quadrupole currents of the matching transport line. To perform the calculation, we use the well-known least squares method. It consists of determining the values of the parameters p_j which yield a minimum for the expression

$$\chi^2 = \sum_k \frac{1}{\sigma_k^2} (T_k - T_k^*)^2$$

The weight σ_k is taken to be equal to one in our case. T_k^* is the value of T_k for the $(n-1)^{th}$ iteration.

The method that we use is described in Ref. 1. It combines the gradient search method with the method of linearizing the fitting function. The gradient search of parabolic expansion of χ^2 is used for approaching the minimum and then the linearization method to reach the minimum when close to it.

The program converges very rapidly, in 5 to 10 iterations, depending upon how far the initial conditions are compared to the search solution. A maximum of 26 iterations is allowed in the program for the difficult cases.

3. Program HOPI

In Appendix 1 we give an example of a typical run. The quadrupoles are arranged from NV = 1 to NV = 6 as follows: Q18, Q19, Q20, Q21, Q22 and Q23. The different modes have been explained in the previous section. The first entry consists of selecting the number of parameters and the mode in which one wants to run the program. NPARM = 3 is the number of parameters (total number of quadrupoles one wants to vary). We selected the Mode 2 which means that we are going to modify the vertical plane and keep constant the horizontal.

In the second entry, one gives the number NV = 1,2,3 of those 3 quadrupoles (we select, therefore, Q18, Q19 and Q20).

The third entry is the number NV of one of the remaining quadrupole (5 for Q22) and the amount in DATAON command one wants it to be varied. The result of the calculation is displayed on the following line. It prints out first the present setting and gives the new one for all of the quadrupoles from Q18 to Q23 in DATAON commands (DAC). Then the program loops on the third entry waiting for another calculation.

The listing of the HOPI program is given in Appendix 2.

Remarks

In this program, the iteration can be performed with more parameters than constraints.

The stability of the A/D - D/A converter of the DATAON system is very important because the program uses a calibration which has been measured once, but may vary with time. This applies for any computer program that reads values from devices such as power supplies.

Conclusions

A method of matching the beam from the linac to the AGS has been developed and is presented in this report. It has the advantage of allowing one to forget about emittance measurements. Because of the continuous variation of the partial transverse acceptance during the betatron injection process (collapsing of the local closed orbit distortion and transverse space charge effects) it is impossible to predict what the shape of the emittance at the exit of the inflector should be. An on-line computer program HOPI allows one to modify independently the horizontal emittance (minimization of the losses against the inflector) and the vertical emittance (matching of the injected emittance to the vertical acceptance). In Figs. 1 to 6 we show the corresponding results for the runs 1 to 6 of the example. The beam envelopes are traced using the α , β parameters and an expected emittance of 10π mm mrad. A systematic study was undertaken to determine what quadrupoles are the most likely to perform the matching depending upon the mode one wants to run, but is not completed. This method of tuning has shown experimental success and can be applied to any matching section.

Acknowledgment

I want to thank Al Maschke for suggesting this work, E. Raka and E. Gill who made possible its application to the AGS. I am also indebted to J. Dabrowski who wrote the subroutine for the acquisition of the input parameters. I am grateful for the support of L. Blumberg, L.W. Smith and M.Q. Barton in making this work possible.

References

1. P.R. Bevington, "Data Reduction and Error Analysis for the Physical Sciences," McGraw-Hill, 1969.

Dist. B2

RUN HOPI

ENTER NB OF PARAMETERS, MODE OF CONSTRAINTS 3,2

ENTER CODE OF PARAMETERS 1,2,3

ENTER VARIABLE QUAD AND VARIATION 5,50. (Run 1)

303 DAC	-1325.	3080.	-3920.	3225.	-3380.	990.
303 DAC	-1361.	3072.	-3831.	3225.	-3330.	990.

ENTER VARIABLE QUAD AND VARIATION 5,100. (Run 2)

303 DAC	-1325.	3080.	-3920.	3225.	-3380.	990.
303 DAC	-1397.	3065.	-3722.	3225.	-3280.	990.

ENTER VARIABLE QUAD AND VARIATION 5,150. (Run 3)

303 DAC	-1325.	3080.	-3920.	3225.	-3380.	990.
303 DAC	-1433.	3057.	-3589.	3225.	-3230.	990.

ENTER VARIABLE QUAD AND VARIATION 5,200. (Run 4)

303 DAC	-1325.	3080.	-3920.	3225.	-3380.	990.
303 DAC	-1469.	3050.	-3429.	3225.	-3180.	990.

ENTER VARIABLE QUAD AND VARIATION 5,250. (Run 5)

303 DAC	-1325.	3080.	-3920.	3225.	-3380.	990.
303 DAC	-1505.	3043.	-3237.	3225.	-3130.	990.

ENTER VARIABLE QUAD AND VARIATION 5,300. (Run 6)

303 DAC	-1325.	3080.	-3920.	3225.	-3380.	990.
303 DAC	-1540.	3036.	-3016.	3225.	-3080.	990.

ENTER VARIABLE QUAD AND VARIATION ** END OF EXAMPLE **

IQUADS	-10.34	113.66	-141.75	119.32	-123.21	7.52	(Amperes)
	-1361.0	3072.0	-3831.0	3225.0	-3330.0	990.0	(Command)
ALF, BETF	0.36	8.13	-2.73	12.82			

($\alpha_x, \beta_x, \alpha_z, \beta_z$ at the
exit of the matching section)

X(CM)

FIG. 1

Run 1

IQUADS	-10.62	113.40	-137.71	119.32	-121.36	7.52
	-1397.0	3065.0	-3722.0	3225.0	-3280.0	990.0
ALF, BETF	0.36	8.10	-1.21	5.64		

X(CM)

-8-

Z(CM)

ACS 77-9

FIG. 2

Run 2

	IQUADS	-10.89	113.11	-132.79	119.32	-119.51	7.52
X(CM)		-1433.0	3057.0	-3589.0	3225.0	-3230.0	990.0
	ALF, BETF	0.37	8.12	-0.10	4.79

-9-

Z(CM)

ACS 77-9

FIG. 3
Run 3

IQUADS	-11.17	112.85	-126.87	119.32	-117.66	7.52
	-1470.0	3050.0	-3429.0	3225.0	-3180.0	990.0
ALF, BETF	0.38	8.11	-0.15	15.94		

X(CM)

Z(CM)

FIG. 4

Run 4

IQUADS	-11.44	112.59	-119.77	119.32	-115.81	7.52
	-1505.0	3043.0	-3237.0	3225.0	-3130.0	990.0
ALF, BETF	0.39	8.09	-2.54	46.46		

X(CM)

-11-

Z(CM)

ACS 77-9

FIG. 5

Run 5

IQUADS	-11.70	112.33	-111.52	119.32	-113.96	7.52
X(CM)	-1540.0	3036.0	-3014.0	3225.0	-3080.0	990.0
ALF, BETF	0.40	8.08	-8.67	102.97

Z(CM)

-12-

ACS 77-9

FIG. 6

Run 6

FIG. 7

APPENDIX 2

[illegible]

```

C PROGRAM HELP OPERATORS PERFORMING INJECTION H.O.P.I
C M,K.S.A SYSTEM USED -
  DIMENSION CAL(4),RAD(2),BC(6),CHSQR(30),QPOL(6)
  DIMENSION LCN(9),LCR(11),LCM(9)
  INTEGER RIN,TRC,ERR
  COMMON/BLOK1/AL(10),ALE(10),THETA(4),RCURV(4)
  COMMON/BLOK2/C(6),DELTAC(6),MZ
  COMMON/BLOK3/TM(4,2,2)
  COMMON/BLOK4/A(2,2,2),B(2,2,2),D(2,2,2)
  COMMON/BLOK6/X(8,2),VD(6),VC(6),E,VC(6)
  COMMON/BLOK7/NV(6),NCR(5),NPARM,NCON
  COMMON/BLOK8/CUR(6),DCUR(6)
  DATA AL/7.029,5.193,3.20,3.20,3.20,3.233,1.430,2.327,0.1,0.1/
  DATA ALE/29.3,36.07,36.07,36.07,36.07,29.3,60.6,64.9,64.9,85.3/
  DATA LCN/'IFC','IFS','TRC','NQ381','NQ399','NQ411','NQ423'
1  , 'NQ434','NQ446'/
  DATA QPOL/-1.,1.,-1.,1.,-1.,1./
  LCA='RIN'
  KLEC=0
  DO 500 I=1,6
  CUR(I)=0.
  DCUR(I)=0.
500  CONTINUE
  DO 4 I=1,10
  ALE(I)=ALE(I)*1.E-2
4    CONTINUE
C    TYPE 9999
9999 FORMAT(' DO YOU WANT TO USE COMMAND VALUES(CMD)'
1  /' OR ACTUAL READBACKS(ACT)? CMD OR ACT?',2X,$)
C    ACCEPT 9998,ANSWER
9998 FORMAT(A3)
  ANSWER='CMD'
  IF(ANSWER.EQ.'ACT') GO TO 9995
1418 DO 9996 I=1,9
  CALL GETCMD(LCA,LCN(I),LCM(I),ERR)
  IF(.NOT.ERR) GO TO 9996
C    TYPE 9997,LCA,LCN(I),ERR
9997 FORMAT(' ERROR: COMMAND VALUE FOR',2A5,013)
9996 CONTINUE
  GO TO 9994
C READS FROM DATACON
9995 CALL RD(LCA,LCN,3,10,1,LCR,ERR)
  IF(ERR) TYPE 1
1  FORMAT(1X,' ERROR RD ROUTINE',012)
  LCOUNT=0
  CALL TRANS(LCA,LCN,3,LCR,LCM,LCOUNT,ERR)
  IF(ERR) TYPE 331
331  FORMAT(1X,' ERROR TRANS,TIMED FILED',013)
  CALL RD(LCA,LCN(4),6,0,4,LCR(5),ERR)
  IF(ERR) TYPE 231
231  FORMAT(1X,'ERROR RD,UNTIMED READ',012)
  LCOUNT=0
  CALL TRANS(LCA,LCN(4),6,LCR(5),LCM(4),LCOUNT,ERR)

```

```

IF(ERR) TYPE 431
431  FORMAT(1X,' ERROR TRANS, UNTIMED FILED',012)
9994  CONTINUE
C      TYPE 10,(LCN(I),LCM(I),I=1,9)
10    FORMAT(1X,A5,5X,I6)
      DO 299 I=1,6
      CUR(I)=QPOL(I)*FLOAT(LCM(I+3))
299   CONTINUE
      CINF=FLOAT(LCM(1))
      CIFS=FLOAT(LCM(2))
      CTRC=FLOAT(LCM(3))
      THETA(1)=227,
      THETA(2)=((7625.*CINF/3190.)-(49*CIFS/800.))*0,018984
      THETA(3)=THETA(2)/2,
      THETA(4)=1028.*CTRC*0,012476/3720,
      PI=3,1415927
      DO 7 I=1,4
      IN=I+6
      THETA(I)=THETA(I)*1,E-3
      RCURV(I)=ALE(IN)/(2,*SIN(THETA(I)/2,))
7     CONTINUE
C     PRINT 310,(RCURV(I),I=1,4)
310   FORMAT(' RCURV',4F)
      EO=938,2796
      CAL(1)=74,667E-4
      CAL(2)=11,5E-4
      RAD(2)=5,12E-2
      RAD(1)=4,13E-2
      VCC=3,E+8
      T=200,
      G=1, +T/EO
      G=SQRT(G*G-1,)
C     TYPE,105
105   FORMAT(' ENTER LEC (LEC=2-NEW IQUADS)',2X,5)
C     ACCEPT 106,LEC
106   FORMAT(I2)
      LEC=1
      GO TO (108,109),LEC
108   CALL RCURRENT(1)
      GO TO 118
109   TYPE,101
101   FORMAT(' ENTER QUAD CMD CURRENTS FROM Q18 TO Q23',/)
      ACCEPT 107,(CUR(I),I=1,6)
107   FORMAT(6F)
      CALL RCURRENT(1)
118   C2=CAL(2)*VCC/(RAD(2)*EO*G*1,E+6)
      C1=CAL(1)*VCC/(RAD(1)*EO*G*1,E+6)
      DO 2 I=2,5
      C(I)=CUR(I)*C2
2     CONTINUE
      C(1)=CUR(1)*C1
      C(6)=CUR(6)*C1
C     PRINT 302,(CUR(I),I=1,6)

```



```

C      PRINT 112,(C(I),I=1,6)
112   FORMAT(' C(I)=' ,6(F,1X))
C      STOCKAGE DES PARAMETRES INITIAUX DANS BC(I)
      DO 6 I=1,6
      BC(I)=C(I)
6      CONTINUE
      MZ=1
      CALL TTRANS(1)
111   FORMAT(' VCI(I)',6F)
C      LECTURE DES PARAMETRES DESIRES VD(I)
      CALL SPLEC(KLEC)
      VC(1)=B(1,1,1)
      VC(2)=B(1,1,2)
      VC(3)=B(1,2,1)
      VC(4)=B(2,1,1)
      VC(5)=B(2,1,2)
      VC(6)=B(2,2,1)
C      PRINT 111,(VC(I),I=1,6)
      DO 555 I=1,NCON
      NR=NCR(I)
      VD(I)=VC(NR)
555   CONTINUE
C      PRINT 115,(VD(I),I=1,NCON)
115   FORMAT(' VD(I)',6F)
      CALL RCURRENT(2)
      DO 501 I=1,6
      CUR(I)=CUR(I)+DCUR(I)
501   CONTINUE
      CALL RCURRENT(1)
      DO 502 I=2,5
      C(I)=CUR(I)*C2
502   CONTINUE
      C(1)=CUR(1)*C1
      C(6)=CUR(6)*C1
C      PRINT 112,(C(I),I=1,6)
      CALL TTRANS(1)
      VC(1)=B(1,1,1)
      VC(2)=B(1,1,2)
      VC(3)=B(1,2,1)
      VC(4)=B(2,1,1)
      VC(5)=B(2,1,2)
      VC(6)=B(2,2,1)
C      PRINT 211,(VC(I),I=1,6)
211   FORMAT(' VCOUR',6F)
      DO 65 I=1,NCON
      NR=NCR(I)
      VCI(I)=VC(NR)
65   CONTINUE
C      PRINT 212,(VCI(I),I=1,6)
212   FORMAT(' VCI(I)',6F)
      DO 3 I=1,8
      DELTAC(I)=0,001
3      CONTINUE

```

```
FLAMDA=0.001
240 CALL CURFIT(CHESQ,CHESQ0,FLAMDA)
    KS=KS+1
    IF(KS,GT,1) GO TO 250
    CHSQR(1)=CHESQ0
    GO TO 240
250 CHSQR(KS)=CHESQ
    C PRINT 301,KS,CHSQR(KS)
301 FORMAT(I3,2X,F10.5)
    IF(CHESQ,LT,0.001) GO TO 290
    IF(KS,GT,25) GO TO 290
    GO TO 240
290 DO 295 I=2,5
    CUR(I)=C(I)/C2
295 CONTINUE
    CUR(1)=C(1)/C1
    CUR(6)=C(6)/C1
    IF(KS,GT,20) GO TO 296
    C PRINT 302,(CUR(I),I=1,6)
    C TYPE 303,(CUR(I),I=1,6)
    CALL RCURRENT(2)
302 FORMAT(1X,' CUR(I)=',6(F7.2,2X))
303 FORMAT(' CUR(I)',6(F7.2,2X))
    GO TO 297
296 PRINT 300,KS
    TYPE 300,KS
    C PRINT 302,(CUR(I),I=1,6)
    C TYPE 303,(CUR(I),I=1,6)
    CALL RCURRENT(2)
297 CONTINUE
    GO TO 1418
    STOP
300 FORMAT(1X,I3,' ITERATIONS')
    END
```

SUBROUTINE RCURRENT(K)

-19-

AGS 77-9

COMMON/BLOKE/CUR(6)

DIMENSION CAL(6)

CAL(1)=7.6E-3

CAL(2)=3.7E-2

CAL(3)=3.7E-2

CAL(4)=3.7E-2

CAL(5)=3.7E-2

CAL(6)=7.6E-3

GO TO (2,3),K

2 CONTINUE

C CONVERT COMMAND TO CURRENT

DO 1 I=1,6

CUR(I)=CUR(I)*CAL(I)

1 CONTINUE

C PRINT 402, (CUR(I), I=1,6)

402 FORMAT(1X, ' 402 CUR', 6(F7.1, 2X))

C TYPE 403, (CUR(I), I=1,6)

403 FORMAT(' 403 CUR', 6(F7.1, 2X))

GO TO 5

3 CONTINUE

C CONVERT CURRENT TO COMMAND

DO 6 I=1,6

CUR(I)=CUR(I)/CAL(I)

6 CONTINUE

C PRINT 302, (CUR(I), I=1,6)

302 FORMAT(1X, ' 302 CUR DAC ', 6(F6.0, 2X))

TYPE 303, (CUR(I), I=1,6)

303 FORMAT(' 303 DAC ', 6(F6.0, 2X))

5 CONTINUE

RETURN

END

SUBROUTINE SPLEC(KLEC)

-20-

ACS 77-9

COMMON/BLOK7/NV(6),NCR(5),NPARM,NCON

COMMON/BLOKE/CUR(6),DCUR(6)

IF(KLEC,EQ.1) GO TO 20

TYPE 11

11 FORMAT(' ENTER NB OF PARAMETERS, CODE OF CONSTRAINTS',1X,9)

ACCEPT 12,NPARM,NCOD

12 FORMAT(2I)

TYPE 21

21 FORMAT(' ENTER PARAMETERS, CODE',1X,\$)

ACCEPT 22,(NV(I),I=1,NPARM)

22 FORMAT(6I)

GO TO (1,2,3,4),NCOD

1 NCON=5

NCR(1)=1

NCR(2)=2

NCR(3)=4

NCR(4)=5

NCR(5)=6

GO TO 10

2 NCON=3

NCR(1)=1

NCR(2)=2

NCR(3)=3

GO TO 10

3 NCON=5

NCR(1)=1

NCR(2)=2

NCR(3)=3

NCR(4)=4

NCR(5)=5

GO TO 10

4 NCON=3

NCR(1)=4

NCR(2)=5

NCR(3)=6

10 CONTINUE

20 TYPE 41

41 FORMAT(' ENTER VARIABLE QUADS,VARIATIONS',1X,\$)

DO 5 I=1,6

DCUR(I)=0.

5 CONTINUE

ACCEPT 42,(I,DCUR(I))

42 FORMAT(I,F)

KLEC=1

RETURN

END

```

SUBROUTINE TTRANS(III)
COMMON/BLOK1/AL(10),ALE(10),THETA(4),RCURV(4)
COMMON/BLOK2/C(6),DELTAC(6),MZ
COMMON/BLOK3/TM(4,2,2)
COMMON/BLOK4/A(2,2,2),B(2,2,2),D(2,2,2)
DIMENSION TM1(2,2,2)
DO 4 I=1,2
DO 4 J=1,2
DO 4 K=1,2
A(I,J,K)=0.
B(I,J,K)=0.
D(I,J,K)=0.
4 CONTINUE
DO 1 I=1,2
B(I,1,1)=1.0
B(I,1,2)=AL(1)
B(I,2,1)=0.
B(I,2,2)=1.
1 CONTINUE
II=1
2 KODE=IFIX(C(II)/ABS(C(II)))
C IF(III.NE.1) GO TO 115
C PRINT 112,KODE
112 FORMAT(' KODE',I3)
115 CALL QUAD(ALE(II),C(II),KODE)
CALL MM
II=II+1
CALL DRIFT(AL(II))
CALL MM
IF(II.GE.7) GO TO 3
GO TO 2
3 CONTINUE
GO TO (5,7),MZ
C INFLECTOR MZ=1
5 IK=II-6
CALL BM(THETA(IK),RCURV(IK),ALE(II),2)
CALL MM
II=II+1
IF(II.GE.11) GO TO 6
CALL DRIFT(AL(II))
CALL MM
GO TO 5
C ZERO DEGREE
AL(8)=2.4
7 CALL DRIFT(AL(II))
CALL MM
6 CONTINUE
C PRINT 111,III
111 FORMAT(' III=',I2)
IF(III.GT.1) GO TO 11
DO 10 I=1,2
DO 10 J=1,2
DO 10 K=1,2

```

```

10  TM1(I,J,K)=B(I,J,K)
C    CONTINUE
C    PRINT 110,(((TM1(I,J,K),K=1,2),J=1,2),I=1,2)
11  DO 20 I=1,2
    DO 20 J=1,2
    DO 20 K=1,2
    I2=I+2
    TM(I2,J,K)=B(I,J,K)
20  CONTINUE
C    PRINT 110,(((TM1(I,J,K),K=1,2),J=1,2),I=1,2)
C    PRINT 110,(((TM(I,J,K),K=1,2),J=1,2),I=3,4)
110  FORMAT(' TM1',/,4F)
    RETURN
    END

```

```

SUBROUTINE QUAD(ALE,C,KODE)
COMMON/BLCK4/A(2,2,2),B(2,2,2),D(2,2,2)
IF(KODE)1,2,3
2  WRITE(6,101)
101 FORMAT(' KODE QUAD NUL !')
GO TO 6
1  I=1
  J=2
  GO TO 4
3  I=2
  J=1
4  RS=SQRT(ABS(C))
  EX=EXP(ALE*RS)
  A(I,1,1)=0.5*(EX+1./EX)
  A(I,1,2)=0.5*(EX-1./EX)/RS
  A(I,2,1)=A(I,1,2)*ABS(C)
  A(I,2,2)=A(I,1,1)
C    TYPE 53,RS,EX
  EX=ALE*RS
  A(J,1,1)=COS(EX)
  A(J,1,2)=SIN(EX)/RS
  A(J,2,1)=-A(J,1,2)*ABS(C)
  A(J,2,2)=A(J,1,1)
C    PRINT 53,RS,EX
C    TYPE 53,RS,EX
C    PRINT 53,(((A(I,J,K),K=1,2),J=1,2),I=1,2)
C    TYPE 53,(((A(I,J,K),K=1,2),J=1,2),I=1,2)
53  FORMAT(4F)
6  CONTINUE
    RETURN
    END

```



```

SUBROUTINE BM(THETA,RCURV,ALE,MTYPE)
COMMON/BLOK4/A(2,2,2),B(2,2,2),D(2,2,2)
GO TO (1,2),MTYPE

```

```

1 CONTINUE
C SECTOR MAGNET

```

```

I=1
A(I,1,1)=COS(THETA)
A(I,1,2)=RCURV*SIN(THETA)
A(I,2,1)=-SIN(THETA)/RCURV
A(I,2,2)=A(I,1,1)
J=2
A(J,1,1)=1.
A(J,1,2)=ALE
A(J,2,1)=0.
A(J,2,2)=1.
GO TO 3

```

```

2 CONTINUE
C RECTANGULAR MAGNET

```

```

I=1
TT=SIN(THETA)/COS(THETA)
A(I,1,1)=1.
A(I,1,2)=RCURV*SIN(THETA)
A(I,2,1)=0.
A(I,2,2)=1.
J=2
TT=SIN(THETA/2.)/COS(THETA/2.)
A(J,1,1)=1.-THETA*TT
A(J,1,2)=RCURV*THETA
A(J,2,1)=-2.*TT/RCURV+THETA*TT*TT/RCURV
A(J,2,2)=A(J,1,1)

```

```

3 CONTINUE

```

```

C PRINT 53,(((A(I,J,K),K=1,2),J=1,2),I=1,2)

```

```

53 FORMAT(4F)

```

```

RETURN

```

```

END

```

```

SUBROUTINE DRIFT(AL)
COMMON/BLOK4/A(2,2,2),B(2,2,2),D(2,2,2)
DO 1 I=1,2

```

```

A(I,1,1)=1.
A(I,1,2)=AL
A(I,2,1)=0.
A(I,2,2)=1.

```

```

1 CONTINUE

```

```

C PRINT 53,(((A(I,J,K),K=1,2),J=1,2),I=1,2)

```

```

C TYPE 53,(((A(I,J,K),K=1,2),J=1,2),I=1,2)

```

```

53 FORMAT(4F)

```

```

RETURN

```

```

END

```

SUBROUTINE MM

COMMON/BLOK4/A(2,2,2),B(2,2,2),D(2,2,2)

DO 1 I=1,2

DO 1 J=1,2

DO 1 K=1,2

D(I,J,K)=0.

DO 1 L=1,2

D(I,J,K)=D(I,J,K)+A(I,J,L)*B(I,L,K)

1 CONTINUE

DO 2 I=1,2

DO 2 J=1,2

DO 2 K=1,2

B(I,J,K)=D(I,J,K)

2 CONTINUE

C PRINT 53,(((B(I,J,K),K=1,2),J=1,2),I=1,2)

C TYPE 53,(((B(I,J,K),K=1,2),J=1,2),I=1,2)

53 FORMAT(' MM',4F)

RETURN

END

FUNCTION FCHISQ(VCI,VD)

COMMON/BLOK7/NV(6),NCR(5),NPARM,NCON

DIMENSION VD(6),VCI(6)

CHISQ=0.

DO 1 J=1,NCON

CHISQ=CHISQ+(VD(J)-VCI(J))**2

1 CONTINUE

FCHISQ=CHISQ

RETURN

END

SUBROUTINE GETCMD(AREA,DEVNM,CMD,ERR)

IMPLICIT INTEGER(A-Z)

DIMENSION STSBLK(10)

CALL JSEGS

CALL DSRGET(AREA,DEVNM,STSBLK,ERR)

CALL JSEGR

IF(ERR) RETURN

CMD=ISHIFT(STSBLK(4),-18).AND."7777

RETURN

END

```

SUBROUTINE CURFIT(CHISQ,CHISQO,FLAMDA)
COMMON/BLOK2/C(6),DELTAC(6),MZ
COMMON/BLOK4/A(2,2,2),B(2,2,2)
COMMON/BLOK6/X(8,2),VD(6),VC(6),E,VC1(6)
COMMON/BLOK7/NV(6),NCR(5),NPARM,NCON
DIMENSION BETA(6),ALPHA(6,6),DERIV(5,6),BB(6),BS(6)
DIMENSION ARRAY(6,6)
DO 34 K=1,NPARM
  BETA(K)=0.
DO 34 J=1,NPARM
  ALPHA(J,K)=0.
  ALPHA(K,J)=0.
34  CONTINUE
CALL FDERIV(DERIV)
C      PRINT 200
200  FORMAT(' S/S PROGRAM CURFIT')
DO 46 K=1,NPARM
DO 46 J=1,NCON
  BETA(K)=BETA(K)+(VD(J)-VC1(J))*DERIV(J,K)
46  CONTINUE
C      PRINT 555,(BETA(K),K=1,NPARM)
555  FORMAT(' BETA',6F)
DO 47 J=1,NPARM
DO 47 K=J,NPARM
DO 45 L=1,NCON
  ALPHA(J,K)=ALPHA(J,K)+DERIV(L,J)*DERIV(L,K)
45  CONTINUE
  ALPHA(K,J)=ALPHA(J,K)
47  CONTINUE
CHISQO=FCHISQ(VC1,VD)
C      PRINT 111,CHISQO
111  FORMAT(' CHISQO=',E15.8)
110  FORMAT(' CHISQ=',E15.8)
71  DO 74 J=1,NPARM
DO 73 K=1,NPARM
  ARRAY(J,K)=ALPHA(J,K)/SQRT(ALPHA(J,J)*ALPHA(K,K))
73  CONTINUE
  ARRAY(J,J)=1.+FLAMDA
74  CONTINUE
CALL SPINV(ARRAY,NPARM,DET)
C      PRINT 116,DET
116  FORMAT(' DET=',E15.8)
DO 84 J=1,NPARM
  JJ=NV(J)
  BB(JJ)=C(JJ)
  BS(JJ)=C(JJ)
DO 84 K=1,NPARM
  BB(JJ)=BB(JJ)+BETA(K)*ARRAY(J,K)/SQRT(ALPHA(J,J)*ALPHA(K,K))
84  CONTINUE
DO 85 J=1,NPARM
  JJ=NV(J)
  C(JJ)=BB(JJ)
85  CONTINUE

```

-26- AGS 77-9

```

C      PRINT 112,(C(J),J=1,6)
112    FORMAT(' C(J)',6(F,1X))
      CALL TTRANS(2)
C      PRINT 556,((B(LL,M,2),M=1,2),LL=1,2)
556    FORMAT(10X,' B(L,M,2)',4F)
C      TYPE 113,(VC(I),I=1,4)
      VC(1)=B(1,1,1)
      VC(2)=B(1,1,2)
      VC(3)=B(1,2,1)
      VC(4)=B(2,1,1)
      VC(5)=B(2,1,2)
      VC(6)=B(2,2,1)
      DO 146 J=1,NCON
      NR=NCR(J)
      VCI(J)=VC(NR)
146    CONTINUE
C      PRINT 113,(VCI(I),I=1,NCON)
113    FORMAT(' VCI(I)',6F)
      CHISQ=FCHISQ(VCI,VD)
C      PRINT 110,CHISQ
      IF(CHISQ-CHISQ)95,107,107
95     FLAMDA=10.*FLAMDA
C      PRINT 115,FLAMDA
115    FORMAT(' FLAMDAI=',F)
      DO 86 J=1,NPARM
      JJ=NV(J)
      C(JJ)=BS(JJ)
86     CONTINUE
      GO TO 71
107    FLAMDA=FLAMDA/10.
C      PRINT 114,FLAMDA
114    FORMAT(' FLAMDAE=',E15,8)
      RETURN
      END

```

```

SUBROUTINE FDERIV(DERIV)
COMMON/BLOK2/C(6),DELTAC(6),MZ
COMMON/BLOK4/A(2,2,2),B(2,2,2)
COMMON/BLOK7/NV(6),NCR(5),NPARM,NCON
DIMENSION DERIV(5,6),XP(4,2),VCC(6),VCD(6)
C      PRINT 200
200    FORMAT(' S/S PROGRAM FDERIV',/)
      DO 18 KK=1,NPARM
      NKK=NV(KK)
      CK=C(NKK)
      DELTA=DELTAC(NKK)
      C(NKK)=CK+DELTA
      CALL TTRANS(2)
      VCD(1)=B(1,1,1)
      VCD(2)=B(1,1,2)
      VCD(3)=B(1,2,1)
      VCD(4)=B(2,1,1)
      VCD(5)=B(2,1,2)
      VCD(6)=B(2,2,1)
C      PRINT 154,C(KK)
C      TYPE 154,C(KK)
154    FORMAT(F)
C      PRINT 155,(VCD(I),I=1,6)
155    FORMAT(' VCD',6F)
153    FORMAT(2F)
      C(NKK)=CK-DELTA
      CALL TTRANS(2)
      VCC(1)=B(1,1,1)
      VCC(2)=B(1,1,2)
      VCC(3)=B(1,2,1)
      VCC(4)=B(2,1,1)
      VCC(5)=B(2,1,2)
      VCC(6)=B(2,2,1)
C      TYPE 154,C(KK)
C      PRINT 154,C(KK)
C      PRINT 156,(VCC(I),I=1,6)
156    FORMAT(' VCC',6F)
      DO 25 J=1,NCON
      NR=NCR(J)
      DERIV(J,KK)=(VCD(NR)-VCC(NR))/(2,*DELTAC(NKK))
25    CONTINUE
      C(NKK)=CK
18    CONTINUE
C      PRINT 101,((DERIV(J,KK),J=1,NCON),KK=1,NPARM)
101    FORMAT(' DERIV(J,KK)',/,4F)
      RETURN
      END

```

```

SUBROUTINE SPINV(ARRAY,NPARM,DET)
DIMENSION ARRAY(6,6),IK(6),JK(6)
10  DET=1.
11  DO 100 K=1,NPARM
    AMAX=0.
21  DO 30 I=K,NPARM
    DO 30 J=K,NPARM
23  IF(ABS(AMAX)-ABS(ARRAY(I,J))) 24,24,30
24  AMAX=ARRAY(I,J)
    IK(K)=I
    JK(K)=J
30  CONTINUE
31  IF(AMAX) 41,32,41
32  DET=0.
    GO TO 140
41  I=IK(K)
    IF(I=K) 21,51,43
43  DO 50 J=1,NPARM
    SAVE=ARRAY(K,J)
    ARRAY(K,J)=ARRAY(I,J)
50  ARRAY(I,J)=-SAVE
51  J=JK(K)
    IF(J=K) 21,61,53
53  DO 60 I=1,NPARM
    SAVE=ARRAY(I,K)
    ARRAY(I,K)=ARRAY(I,J)
60  ARRAY(I,J)=-SAVE
61  DO 70 I=1,NPARM
    IF(I=K) 63,70,63
63  ARRAY(I,K)=-ARRAY(I,K)/AMAX
70  CONTINUE
71  DO 80 I=1,NPARM
    DO 80 J=1,NPARM
    IF(I=K) 74,80,74
74  IF(J=K) 75,80,75
75  ARRAY(I,J)=ARRAY(I,J)+ARRAY(I,K)*ARRAY(K,J)
80  CONTINUE
81  DO 90 J=1,NPARM
    IF(J=K) 83,90,83
83  ARRAY(K,J)=ARRAY(K,J)/AMAX
90  CONTINUE
    ARRAY(K,K)=1./AMAX
100 DET=DET*AMAX
101 DO 130 L=1,NPARM
    K=NPARM-L+1
    J=IK(K)
    IF(J=K) 111,111,105
105 DO 110 I=1,NPARM
    SAVE=ARRAY(I,K)
    ARRAY(I,K)=-ARRAY(I,J)
110 ARRAY(I,J)=SAVE
111 I=JK(K)
    IF(I=K) 130,130,113

113 DO 120 J=1,NPARM
    SAVE=ARRAY(K,J)
    ARRAY(K,J)=-ARRAY(I,J)
120 ARRAY(I,J)=SAVE
130 CONTINUE
140 RETURN
END

```