UCRL-51397

A CODE FOR VIEWING MAFCO CONDUCTORS FROM ANY ANGLE

T. N. Haratani
R. W. Moir

June 15, 1973

Prepared for U.S. Atomic Energy Commission under contract No. W-7405-Eng-48

LAWRENCE LIVERMORE LABORATORY

University of California/Livermore

Printed in the United States of America
Available from
National Technical Information Service
U. S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151
Price: Printed Copy $___*; Microfiche $0.95

*Pages	NTIS Selling Price
1-50	$4.00
51-150	$5.45
151-325	$7.60
326-500	$10.60
501-1000	$13.60

TID-4500, UC-32
Mathematics and
Computers

LAWRENCE LIVERMORE LABORATORY
University of California/Livermore, California/94550

UCRL-51397

A CODE FOR VIEWING MAFCO CONDUCTORS FROM ANY ANGLE

T. N. Haratani
R. W. Moir

MS. date: June 15, 1973

A CODE FOR VIEWING MAFCO CONDUCTORS FROM ANY ANGLE

Abstract

A code designated MPVER rotates MAFCO conductors so that they can be viewed from any angle. The code makes it easier to visualize conductors and serves as a check for MAFCO input. The equations describing the rotating process are given.

Introduction

A magnetic field code (MAFCO)[1] handles a variety of conductors in three dimensions. Because of the difficulty of visualizing these conductors, we have written the MPVER program (Appendix) to plot the projection of conductors used by MAFCO.

To design the MPVER program we had to transform the MAFCO input describing the conductor coordinates into a rotated coordinate system. This input consisted of parameters defining loops, arcs, helices, straight lines, and a series of straight line segments. Since helices are seldom used we did not include them in the program.

Figure 1 shows an example of loops and arcs in a primed coordinate system. Loops and arcs are specified by x_0, y_0, and z_0, the coordinates in the unprimed coordinate system at the center of curvature. Figure 1 also shows the radius A and the Eulerian angles α, β, and ϕ. The angle ϕ ranges from ϕ_1 to ϕ_2 for arcs and from 0° to 360° for circles.

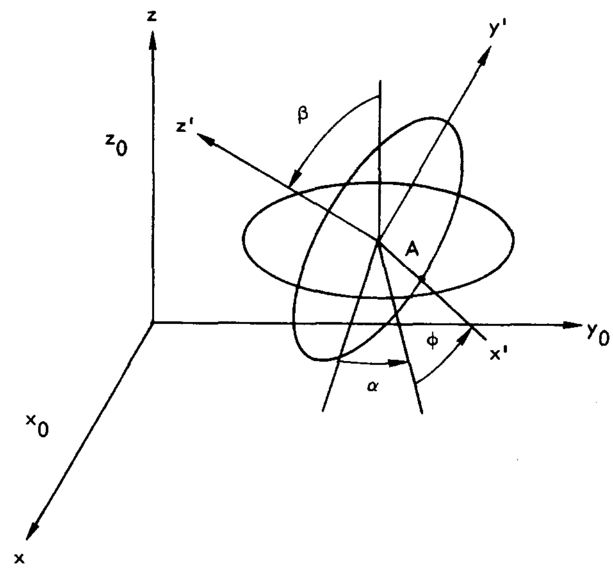


Fig. 1. Coordinate system showing the Eulerian angles.

The Rotation Process

A point on the arc in the prime coordinates is $x' = A$, $y' = 0$, $z' = 0$. In the unprimed coordinates the point is located by the vector $\vec{x}$.

$$\vec{x} = A^{-1}(\alpha, \beta, \phi)\, \vec{x}' + \vec{x}_0 \tag{1}$$

$$\vec{x}_0 = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} ; \vec{x}' = \begin{pmatrix} A \\ 0 \\ 0 \end{pmatrix} \tag{2}$$

In the double primed coordinates (Fig. 2) the point is

$$\vec{x}'' = A^{-1}(\zeta, \eta, \nu)\, \vec{x} , \tag{3}$$

where the transformation matrix A^{-1} is the standard rotation matrix.[2]

Finally, we want to look at the projection of the point onto the $x'' - y''$ plane.

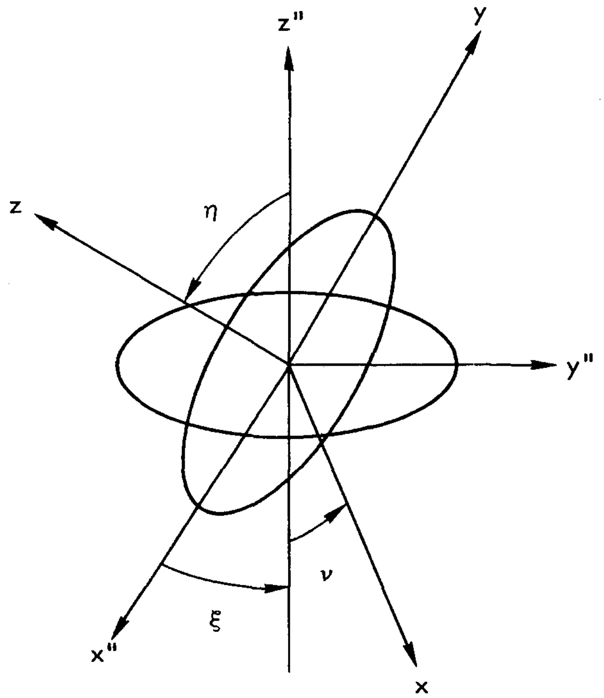


Fig. 2. Coordinate system showing the viewing angles.

For straight lines and straight line segments we specify the end points

$$x_1, y_1, z_1, x_2, y_2, z_2$$

in the unprimed coordinates and use the transformation

$$\vec{x}'' = A^{-1}(\zeta, \eta, \nu)\, x . \tag{4}$$

The matrix $A^{-1}(\alpha, \beta, \phi)$ is

$$\begin{pmatrix} j & k & l \\ m & n & o \\ p & q & r \end{pmatrix} , \tag{5}$$

where

$j = \cos\phi \cos\alpha - \cos\beta \sin\alpha \sin\phi$
$k = \sin\phi \cos\alpha - \cos\beta \sin\alpha \sin\phi$
$l = \sin\beta \sin\alpha$
$m = \cos\phi \sin\alpha + \cos\beta \cos\alpha \sin\phi$
$n = -\sin\phi \sin\alpha + \cos\beta \cos\alpha \cos\phi$
$o = -\sin\beta \cos\alpha$
$p = \sin\beta \sin\phi$
$q = \sin\beta \cos\phi$
$r = \cos\beta .$

The matrix

$$A^{-1}(\zeta, \eta, \nu) \text{ is } \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \tag{6}$$

where

$a = \cos\nu \cos\zeta - \cos\eta \sin\zeta \sin\nu$
$b = -\sin\nu \cos\zeta - \cos\eta \sin\zeta \cos\nu$
$c = \sin\eta \sin\zeta$
$d = \cos\nu \sin\zeta + \cos\eta \cos\zeta \sin\nu$
$e = -\sin\nu \sin\zeta + \cos\eta \cos\zeta \cos\nu$
$f = -\sin\eta \cos\zeta$
$g = \sin\eta \sin\nu$
$h = \sin\eta \cos\nu$
$i = \cos\eta .$

For arcs and loops we perform the matrix multiplications indicated by Eqs. 1 through 5.

$$\vec{x} = \begin{pmatrix} j & k & l \\ m & n & o \\ p & q & r \end{pmatrix} \begin{pmatrix} A \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} \qquad (7)$$

$$\vec{x}'' = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}. \qquad (8)$$

The x'' and y'' components are plotted as the projection of the conductor

$$\vec{x}'' = a(jA + x_0) + b(mA + y_0) + c(pA + z_0)$$

$$y'' = d(jA + x_0) + e(mA + y_0) + f(pA + z_0) . \qquad (9)$$

For straight lines and straight line segments

$$x'' = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \qquad (10)$$

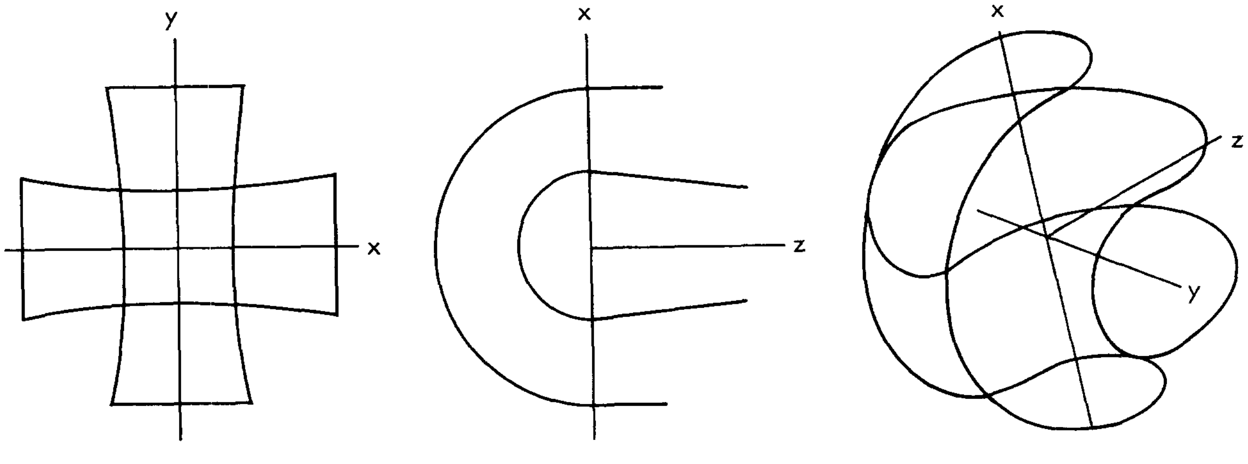
and

$$x'' = ax + by + cz$$

$$y'' = dx + ey + fz . \qquad (11)$$

The form of Eq. (13) allows us to plot straight lines and straight line segments by specifying the (x, y, z) values of the end points. To plot loops and arcs we must first break the arc into segments and plot straight lines (cords). The arc is specified by the angle ϕ which we arbitrarily break into equal angular segments of 2° between ϕ_1 and ϕ_2. The loops are broken into equal segments of 6°. Then we use Eq. (11) to plot these straight line segment approximations to arcs and loops.

The instructions for input data cards are shown in Table 1. To illustrate the use of the code three views of a simple conductor are shown in Fig. 3. The figure is based on input data given in Table 2. Figure 4 shows more complicated conductors used in the design of a direct energy conversion device.



Zeta = 0°, ϵ = 0°, η = 0°. Zeta = 90°, ϵ = 90°, η = 0°. Zeta = 120°, ϵ = 120°, η = 30°.

Fig. 3. Three views of the Yin-Yang coil.

Table 1. Input and output for MPVER (MAFCO plot on Versatec).

INPUT

Card one, format (4i5)

NLOOPP, NARCPL, NLINEP, NGCEPL, where each variable selects the conductors one wants plotted. For example, if NLOOPP = 1 then every loop will be plotted and if NLOOPP = 2, every other loop will be plotted.

Card two, format (i5)

NVIEWS = number of different views

Card three, format (3F10.1)

TZETA (i), TETA (i), TNU (i), i = 1, NVIEWS.

Zeta, eta, and nu determine the orientation of the conductors for each view.

Card four through the last input card are MAFCO data cards.[3]

For each set of conductors there is a maximum of 200 loops, 500 arcs, 50 helices, 400 straight lines, and 2400 straight line segments.

OUTPUT

The output file (TOX) can be plotted on the Versatec printer/plotter via the following teletype writer message:

DCNTLR/1 1

TTY responds TYPE DD80 FILE NAME, X OFFSET Y OFFSET, OR L

TOX

TTY responds ALL DONE

If a DD80 plot is desired, copy the file TOX to another ten character file starting with the letters DX, i.e., DXTOXTOXTO. Teletype the following instructions:

Give DXTOXTOXTO 999999 / 1 .2

TTY responds IK

END

Table 2. Sample input.

```
1     1     1     1
3
    0.0         0.0         0.0
    90.         90.         0.0
   120.        120.         30.
1
0     8     0     3     0     1     10     YIN YANG-5  6.15  LOBE ANGLE
     0.         30.          0.         65.          0.       83.85 -1000000.
   180.        360.
     0.        -30.          0.         65.          0.       96.15  1000000.
   180.        360.
    65.          0.          0.         30.         90.         90. -1000000.
     0.        180.
   -65.          0.          0.         30.         90.         90.  1000000.
     0.        180.
    30.          0.          0.         65.         90.       83.85 -1000000.
     0.        180.
   -30.          0.          0.         65.         90.       96.15  1000000.
     0.        180.
     0.         65.          0.         30.          0.         90. -1000000.
   180.        360.
     0.        -65.          0.         30.          0.         90.  1000000.
   180.        360.
   -80.         0.0         0.0         80.         0.0         0.0
    0.0        -40.         0.0         0.0        80.0         0.0
    0.0         0.0         0.0         0.0         0.0        80.0
```

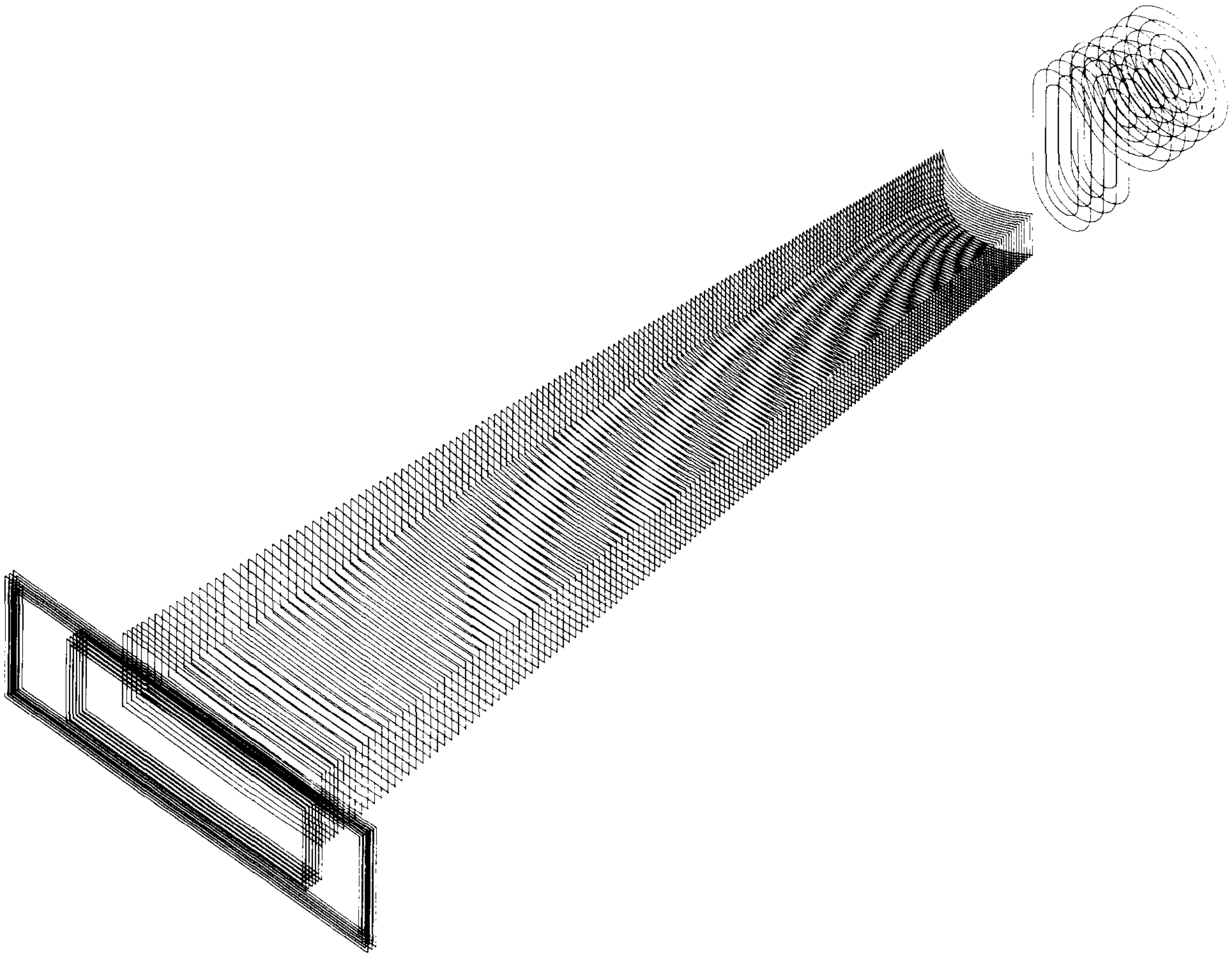


Fig. 4. Conductors for a direct energy conversion device.

References

1. W. A. Perkins and J. C. Brown, J. App. Phys. 35, 3337 (1964).
2. H. Goldstein, Classical Mechanics, (Addison-Wesley Publishing Co., Inc., Reading, Mass., 1959), p. 107.
3. W. A. Perkins and J. C. Brown, MAFCO—A Magnetic Field Code for Handling General Current Elements in Three Dimensions, Lawrence Livermore Laboratory, Rept. UCRL-7744 (1966).

Appendix — MPVER Code

```
*ID    164MMN              PLOT                       MPLOT                BOXU06
*      CONTROLLEE 361100 MVER
*      XEQ MVER
*      LIST8
*      DUMP ( DEC,OCT)
*      FORTRAN
       PROGRAM MPLOT (HSP,INPUT,OUTPUT)
       OPTIMIZE
       CALL KEEP80 (3RTOM)
       CALL ASSIGN(3,15)

       CLICHE STORAGE
       PARAMETER(L0=50, OCC=700, VM=5000, TEN=10, CD=400, CO=100,
      P      MM1VC=2400)
       COMMON TA, TB, TC, TD, TE, TF, TG, TH, TI, JAY, JAZ
       COMMON H(L0), THETA(L0), DELTA(L0), AA(L0), XR(L0), Z(L0),
      C      XM(OCC), YM(OCC), ZM(OCC), AM(OCC), AL(OCC), BE(OCC),
      C      XP(VM), YP(VM), ZP(VM), TZETA(TEN), TETA(TEN), TNU(TEN)
       ENDCLICHE
       USE STORAGE

       PARAMETER (PI=3.1415927,  OUT = 3,   IN=2, RAD=.01745329)
       DIMENSION P1(OCC), P2(OCC), BIG(4), BIGA(3), BIGR(TEN),
      D      X1(CD), Y1(CD), Z1(CD), X2(CD), Y2(CD), Z2(CD),
      D      GX(MM1VC), GY(MM1VC), GZ(MM1VC),NGN(CO),GI(CO),
      D      HA(L0), HD(L0), HP1(L0), HP2(L0), HI(L0), HXY(L0)
       DATA (DELFHI = 1.0 )
       DELPHI = DELPHI * RAD
       N = 0
C
        RIT IN, 885, NLOOPP, NARCPL, NLINEPL, NGCEPL, NVIEWS,
      R           (TZETA(I), TETA(I), TNU(I), I=1,NVIEWS)
       RIT IN, 919, NPROBS
       RIT IN, 919, NLOOP, NARC, NHELIX, NLINE, NGCE
C
 JOUT  FORMAT ( 1X, 5HZETA=, F7.1, 3X, 4HETA=, F7.1, 3X, 3HNU=, F7.1 )
 885   FORMAT (4I5 / I5 /(3F10.1))
   886 FORMAT ( 3F10.1)
   887 FORMAT (4F10.1, I5 )
 888   FORMAT (6F10.1 )
   889 FORMAT(6F10.1/2F10.1)
 919   FORMAT (5I5)
C                                           ********************************
C                                           READ IN THE MAFCO DATA CARDS.
C                                           ********************************
       IF ( NLOOF ) LOOPS, ARCS, LOOPS
C
C          ********************
C  ***  READ IN THE LOOPS  ***
C          ********************
C
LOOPS  CONTINUE
       RIT IN,888,(XM(I), YM(I), ZM(I), AM(I), AL (I), BE (I),I=1,NLOOP)
       DO LO I = 1, NLOOP
       AL(I) = AL(I) * RAD
 LO    BE(I) = BE(I) * RAD
C
       N = N + 1
       IFIRST = 1
       LAST = NLOOP
```

```
       JAY = LAST
       JAZ = JAY + NLOOP
       CALL BIGGE(IFIRST, LAST, BIG, N)
C
C
ARCS   IF ( NARC ) ARK, HELIX, ARK
C
C           ********************
C           READ IN THE ARCS
C           ********************
C
ARK    JZ = NLOOP + 1
       KZ = NLOOP + NARC
       RIT IN,889,(XM(I),YM(I),ZM(I),AM(I),AL(I),BE(I),P1(I),P2(I),I=JZ,
      1 KZ )
C
       JL = 0
       DO AR KP = JZ, KZ
       AL(KP) = AL(KP) * RAD
       BE(KP) = BE(KP)  * RAD
       P1(KP) = P1(KP) * RAD
       P2(KP) = P2(KP) * RAD
       JL = JL + 1
       PHIW = P1(KP)
       TA = 1.0
       TB = 0.0
       TC = 0.0
       TD = 0.0
       TE = 1.0
       TF = 0.0
       TG = 0.0
       TH = 0.0
       TI = 1.0
       CALL ROTATE ( PHIW, KP, JL )
       JL = JL + 1
       PHIW = P2(KP)
 AR    CALL ROTATE ( PHIW, KP, JL )
       DO ARR JG = 1, JL
       XP(JG) = ABSF(XP(JG))
       YP(JG) = ABSF(YP(JG))
 ARR   ZP(JG) = ABSF(ZP(JG))
       BIGA(1) = AMAXAF(XP,1,JL,1,M)
       BIGA(2) = AMAXAF(YP,1,JL,1,M)
       BIGA(3) = AMAXAF(ZP,1,JL,1,M)
       N = N + 1
       BIG(N) = AMAXAF ( BIGA,1,3,1,M )
C
HELIX IF ( NHELIX ) HELX, LINES, HELX
C
C  ***  THE HELICES CARDS ARE MERELY READ IN SO THAT THE MAFCC DATA CARD
C  ***  DECK CAN STAY INTACT. NO HELICES WILL BE PLOTTED.
C
HELX   CONTINUE
C
C           ********************
C  ***  READ IN THE HELICES  ***
C           ********************
C
         RIT IN,888,(HA(I),HD(I),HP1(I),HP2(I),HI(I),HXY(I),I=1,NHELIX)
C
```

```
LINES IF ( NLINE ) LINE, LI, LINE
C
C           ********************
C   ***  READ IN THE STRAIGHT LINES  ***
C           ********************
C
LINE  RIT IN, 888,(X1(I), Y1(I), Z1(I), X2(I), Y2(I), Z2(I), I=1,NLINE)
C
      N = N + 1
      JYY = NLINE
      JZZ = JYY + NLINE
      DO LSL J = 1, NLINE
      JAY = JYY + J
      JAZ = JZZ + J
      XP(J) = ABSF ( X2(J) - X1(J) )
      XP(JAY) = ABSF ( Y2(J)-Y1(J) )
 LSL  XP(JAZ) = ABSF ( Z2(J) - Z1(J) )
      BIG(N) = AMAXAF ( XP, 1, NLINE, 1, M )
C
 LI   IF ( NGCE ) GCE, PLOTT, GCE
C
C           ********************
C           READ IN THE GENERAL CURRENT ELEMENTS
C           ********************
C
 GCE  NGX = NGCE
      LMAX = 0
      DO LGC LG = 1, NGCE
      L = LMAX + 1
      RIT IN, 887, GX(L),GY(L),GZ(L),GI(LG),NGN(LG)
      LMIN = L + 1
      LMAX = L + NGN(LG) - 1
      RIT IN, 886, ( GX(L), GY(L), GZ(L), L = LMIN, LMAX )
 LGC  CONTINUE
C
      N = N + 1
      LMAX = 0
      DO LGCE J = 1, NGCE
      L = LMAX + 1
      LMAX = L + NGN(J) - 1
      CALL AMINMX ( GX,L,LMAX,1,AMN1,AMX1)
      CALL AMINMX ( GY,L,LMAX,1,AMN2,AMX2)
      CALL AMINMX ( GZ,L,LMAX,1,AMN3,AMX3)
      AX1 = ABSF (AMX1-AMN1 )
      AX2 = ABSF ( AMX2-AMN2 )
      AX3 = ABSF ( AMX3-AMN3 )
      IF ( AX1 .GE. AX2 ) , WEE
      IF ( AX1 .GE. AX3 ) , GROC
      XP(J) = AX1
      GO TO LGCE
 WEE  IF ( AX2 .GE. AX3 ) , GROC
      XP(J) = AX2
      GO TO LGCE
 GROC XP(J) = AX3
 LGCE CONTINUE
      BIG(N) = AMAXAF ( XP,1,NGCE,1,M )
C
C
C         ***************************
C         PREPARE TO DO SOME PLOTTING
```

```
C          ***************************
C
PLOTT CONTINUE
C
      CALL CRTIO ( 2HBN,1,1 )
      CALL DDERS(0)
C
      BIGGER = AMAXAF ( BIG,1,N,1,M )
      XMIN = -BIGGER
      YMIN = XMIN
      XMAX = BIGGER
      YMAX = XMAX
      WOT 3, BI,(BIG(I),I=1,N)
 BI   FORMAT (4E12.3)
C
      SFX = ( XMAX - XMIN ) / XMX
      SFY = ( YMAX - YMIN ) / YMX
      LEGX = XMX - XMN
      LEGY = YMX - YMN
C
C
C                                                    ************************
C                                                    SET UP FOR THE GRIDS.
C                                                    ************************
C
      DO TOP K = 1, NVIEWS
C
      PRINT JCUT,      TZETA(K), TETA(K), TNU(K)
C
      CALL FRAME
      CALL SETCRT ( XMIN,YMIN,1 )
      CALL MAPG ( XMIN,XMAX,YMIN,YMAX )
C
      ZETA = TZETA(K) * RAD
      ETA = TETA(K) * RAD
      UNU = TNU(K) * RAD
      SZE = SINF ( ZETA )
      CZE = COSF (ZETA)
      SET = SINF ( ETA )
      CET = COSF ( ETA )
      SNU=SINF( UNU )
      CNU=COSF( UNU )
      TA=CNU*CZE-CET*SZE*SNU
      TB=-SNU*CZE-CET*SZE*CNU
      TC=SET*SZE
      TD=CNU*SZE+CET*CZE*SNU
      TE=-SNU*SZE+CET*CZE*CNU
      TF=-SET*CZE
      TG=SET*SNU
      TH=SET*CNU
      TI=CET
C
      IF ( NLOOP ) PLOTL, PLT, PLOTL
C
C                                                    ****************
C                                                    PLOT THE LOOPS.
C                                                    ****************
PLOTL CONTINUE
C
      DO PLL JK = 1, NLOOP, NLOOPP
```

```
      DO PL JL = 1, 60
      XJ = JL
      PHIW = 2.0 * PI * XJ / 59. - 2.0 * PI / 59.
C
C *** COMPUTE ROTATED X AND Y FOR PLOTTING. CALL THEM XP AND YP.
C
      CALL ROTATE(PHIW, JK, JL)
C
 PL   CONTINUE
C
 PLL  CALL TRACE ( XP,YP,60,1,1 )
C
 PLT  IF ( NARC ) PLOTA, PLINE, PLOTA
C                                                  ****************
C                                                  PLOT THE ARCS.
C                                                  ****************
C
C ***  DO SOME CALCULATING FOR CALCOMP PLOTTING.
C
C *** J FOR A GIVEN ARC, CALL IT ARC NUMBER JK, I MUST CALCULATE JL POINTS.
C ***  ALONG THAT ARC, FROM PHI1 TO PHI2. DELPHI IS THE INCREMENT FOR PHI.
C  *** THE CARTESIAN PLOINTS ARE XP(JL) AND YP(JL), STORED CONSECUTIVELY
C
PLOTA CONTINUE
C
      DO PLI J = JZ, KZ, NARCPL+1
      JK = J
      JAY = J + 1
 PLA  JL = 0
      PHIW = P1(JK)
C
C *** COMPUTE ROTATED X AND Y FOR PLOTTING. CALL THEM XP AND YP.
C
 PLC  JL = JL + 1
C
      CALL ROTATE(PHIW, JK, JL)
C
      PHIW = PHIW + DELPHI
      FI = P2(JK) - PHIW
      IF ( FI ) PLD,PLC,PLC
 PLD  HDEL = DELPHI * .95
C
      IF (HDEL + FI ) PLF, PLF, PLE
 PLE  PHIW = P2(JK)
      GO TO PLC
C
 PLF  CALL TRACE ( XP,YP,JL,1,1 )
C
      IF ( JK-JAY) PLH,PLI,PLH
 PLH  JK = JK+1
      GO TO PLA
 PLI  CONTINUE
      WOT 3,100,(XM(I),YM(I),ZM(I),AM(I),XP(I),YP(I),ZP(I),I=1,JL)
 100  FORMAT (1X,7E15.3)
      WOT 3,101
 101  FORMAT(1H1)
C
PLINE IF ( NLINE ) PLOTS, PGC, PLOTS
C                                                  ****************
C                                                  PLOT THE LINES.
```

```
C                                                         ****************
PLOTS CONTINUE
      DO PLIN I = 1, NLINE, NLINEP+1
      J=I+1
      XP(1) = TA * X1(I) + TB * Y1(I) + TC * Z1(I)
      XP(2) = TA * X2(I) + TB * Y2(I) + TC * Z2(I)
      YP(1) = TD * X1(I) + TE * Y1(I) + TF * Z1(I)
      YP(2) = TD * X2(I) + TE * Y2(I) + TF * Z2(I)
      CALL TRACE ( XP,YP,2,1,1 )
      XP(1) = TA * X1(J) + TB * Y1(J) + TC * Z1(J)
      XP(2) = TA * X2(J) + TB * Y2(J) + TC * Z2(J)
      YP(1) = TD * X1(J) + TE * Y1(J) + TF * Z1(J)
      YP(2) = TD * X2(J) + TE * Y2(J) + TF * Z2(J)
 PLIN CALL TRACE ( XP,YP,2,1,1 )
C
C                                                         *********************
C                                                         PLOT THE GENERAL CE,S
C                                                         *********************
C
PGC   IF ( NGCE ) PLOTG, TOP, PLOTG
C
PLOTG KG= 0
      DO PGB J = 1, NGCE , NGCEPL
      DO PGA I = 1, NGN(J)
      KG= KG+ 1
      XP(I) = TA * GX(KG)+ TB * GY(KG)+ TC * GZ(KG)
 PGA  YP(I) = TD * GX(KG)+ TE * GY(KG)+ TF * GZ(KG)
      NNG = NGN(J)
 PGB  CALL TRACE ( XP,YP,NNG,1,1 )
C
 TOP  CONTINUE
C
      CALL PLCTE
C
      CALL EXIT
      END
      SUBROUTINE ROTATE(PHIW,JK,JL)
      OPTIMIZE
      USE STORAGE
      CPHI = COSF ( PHIW )
      SPHI = SINF ( PHIW )
      SAL = SINF ( AL(JK) )
      CAL = COSF ( AL(JK) )
      SBE = SINF ( BE(JK) )
      CBE = COSF ( BE(JK) )
      TJ = CPHI * CAL - CBE * SAL * SPHI
      TM = CPHI * SAL + CBE * CAL * SPHI
      TP = SBE * SPHI
      TS = TA * TJ + TB * TM + TC * TP
      TV = TD * TJ + TE * TM + TF * TP
      TY = TG * TJ + TH * TM + TI * TP
      XP(JL) = TS*AM(JK) + TA*XM(JK) + TB*YM(JK) + TC*ZM(JK)
      YP(JL) = TV*AM(JK) + TD*XM(JK) + TE*YM(JK) + TF*ZM(JK)
      ZP(JL) = TY*AM(JK)+TG*XM(JK)+TH*YM(JK)+TI*ZM(JK)
      RETURN
      END
```

```
*       FORTRAN
        SUBROUTINE BIGGE(IFIRST, LAST, BIG, N)
        OPTIMIZE
        USE STORAGE
        DIMENSION BIG(4)
C
C         THIS SUBROUTINE HELPS MAKE SURE THE PICTURE DOES NOT GO OFF SCALE.
C
        JAX = 0
        DO LASTA J = IFIRST, LAST
        JAX = JAX + 1
        JAY = JAY + 1
        JAZ = JAZ + 1
        XP(JAX) = ABSF (XM(J) + AM(J) )
        XP(JAY) = ABSF ( YM(J) + AM(J) )
LASTA   XP(JAZ) = ABSF(ZM(J)+AM(J))
        BIG(N) = AMAXAF ( XP,1,JAZ,1,M )
        WOT 3,10,(XP(K),K=1,JAZ)
 10     FORMAT (5X,10E11.2)
        RETURN
        END
```

Distribution

LLL Internal Distribution

Roger E. Batzel
T. K. Fowler
R. F. Post
J. L. Brady
R. H. Bulmer
R. P. Freis
T. N. Haratani 25
M. A. Harrison
B. C. Howard
S. L. Rompel
F. H. Coensgen
C. C. Damm
J. H. Foote
S. Fernbach
T. J. Fessenden
C. W. Hartman
E. B. Hooper
W. E. Nexsen
C. A. Larsen
R. W. Moir 25
W. L. Barr
A. R. Harvey
A. K. Chargin
C. E. Taylor
C. D. Henning
G. A. Carlson
B. McNamara
J. D. Lee
L. C. Pittenger
TID File 30

External Distribution

R. W. Bussard
U.S. Atomic Energy
Commission
Washington, D.C.

J. File
Princeton University
Princeton, New Jersey

H. Laquer
Los Alamos Scientific
Laboratory
Los Alamos, New Mexico

M. Lubell
Oak Ridge National Laboratory
Oak Ridge, Tennessee

L. M. Lidsky
Massachusetts Institute of
Technology
Cambridge, Massachusetts

TID-4500 Distribution, UC-32
Mathematics and Computers 207

MSG/lc/edas

92 734