DA-583

UCRL-52029

**ABCXYZ: VECTOR POTENTIAL (A) AND MAGNETIC FIELD (B)
CODE (C) FOR CARTESIAN (XYZ) GEOMETRY
USING GENERAL CURRENT ELEMENTS**

D. V. Anderson
J. Breazeal
C. H. Finan
B. M. Johnston

September 14, 1976

Prepared for U.S. Energy Research & Development
Administration under contract No. W-7405-Eng-48

**LAWRENCE
LIVERMORE
LABORATORY**
University of California / Livermore

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research & Development Administration, 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.

NOTICE

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Energy Research & Development Administration to the exclusion of others that may be suitable.

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22161

Price: Printed Copy \$ Microfiche \$2.25

<u>Page Range</u>	<u>Domestic Price</u>
001-025	\$3.50
026-050	\$4.00

LAWRENCE LIVERMORE LABORATORY
University of California, Livermore, California 94550

NOTICE
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, 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.

LCRL-52029

**ABCXYZ: VECTOR POTENTIAL (A) AND MAGNETIC FIELD (B)
CODE (C) FOR CARTESIAN (XYZ) GEOMETRY
USING GENERAL CURRENT ELEMENTS**

D. V. Anderson, J. Breazeal, C. H. Finan, and B. M. Johnston

MS. date: September 14, 1976

Contents

Abstract	1
Introduction	1
Grid, Data, and Flow Structure	2
Transformation to Local Preferred Coordinates	3
Fields from Straight Line Current Elements	6
Fields from Arc or Loop Current Elements	10
Sample Problem: Fields of 2X1IB	14
References	20
Appendix A: Data Structure	21
Appendix B: Timing Analysis	24
Appendix C: Source Code Listing for ABCXYZ	25
Appendix D: Input for 2X1IB Sample Problem	112

ABCXYZ: VECTOR POTENTIAL (A) AND MAGNETIC FIELD (B) CODE (C) FOR CARTESIAN (XYZ) GEOMETRY USING GENERAL CURRENT ELEMENTS

Abstract

ABCXYZ is a computer code for obtaining the Cartesian components of the vector potential and the magnetic field on an observed grid from an arrangement of current carrying wires. Arbitrary combinations of straight line segments, arcs, and loops are allowed in the specification of the currents. Arbitrary positions and orientations of the current carrying elements are also allowed. Specification of the wire diameter permits the computation of well-defined fields, even in the interiors of the conductors. An optional feature generates magnetic field lines. Extensive graphical and printed output is available to the user including contour, grid-line, and field-line plots.

Introduction

ABCXYZ is a MAFCO¹-like code for obtaining both vector potentials and magnetic fields. We do not intend to replace MAFCO but, for a certain class of problems in Cartesian geometry involving the vector potential and the magnetic field, we feel this new code is helpful. The helix option has been omitted from this code because helices can be represented more efficiently by general current elements. Other desirable features of ABCXYZ do not appear in the original MAFCO. Particularly, vector potentials are not computed by MAFCO although it could be revised to remedy this problem.²

To compute the fields, we translate and rotate the coordinate system into a preferred location and orientation centered at the current element, using a full three-angle Euler rotation. In comparison MAFCO uses only a two-angle Euler rotation and then only for certain current elements.

In this report, we cover the important aspects of the ABCXYZ code. The observer grid, data structure, and overall flow diagram are described and the Euler angle transformation is also reviewed. The fields of straight line and arc segments are discussed in separate sections. General current elements and loops are considered as special cases of the line and arc respectively. In addition a sample problem from the 2XIIB experiment at LLL, together with results, is provided. The data structure, timing results, and code listings are presented in the Appendices.

Grid, Data, and Flow Structure

The fields produced by the current elements are evaluated at observer points located on a rectangular x,y,z, grid. Figure 1 illustrates this grid and identifies some of the relevant iortran variables that are used. Here, we show uniform grid spacings but virtually any observer grid can be substituted so long as the grid points are given by their Cartesian coordinates. An option for generating field lines similar to MAFCO is also included.

We expect to apply this code to several coil geometries that have been created previously by MAFCO. Since many data cards are involved, we decided that as one option, the MAFCO option, the data format would be identical to the MAFCO input. Another option, the ABCXYZ normal option, would permit data to be read in a simplified structure. Under both options, the first segment of data is read in by NAMELIST. The current element data is read in next; the format of these data depends on the option chosen. Specific details of the data structure are discussed in Appendix A.

Storage of the main arrays has been allocated so that the seven 3-D arrays for the fields spanning the observed grid are stored in LCM (large core memory) on the CDC-7600. Because storage into LCM is a comparatively rare event, the values are sent there by a random access. Arrays that store the data card information are held in LCM and each of these arrays can have lengths up to several thousand elements. Appendix B gives a more detailed discussion of the code organization.

The flow structure is straightforward as the inner loop ranges over the entire set of current elements for each observer grid point. The code listing provided in Appendix C gives the detailed flow structure.

All quantities in this report are measured in "laboratory" units; length is in centimeters, angles are in degrees, current is in amperes, and magnetic field (B) is in gauss. Also, $\mu_0 = 0.4\pi$. These units are used in the MAFCO code and were adopted here so users of MAFCO could easily convert to this code.

Fig. 1. The rectangular, uniformly spaced grid used with ABCXY%.

Transformation to Local Preferred Coordinates

For a given current element and observer position, it is convenient to transform the coordinate system from that of the laboratory system to one that is positioned and oriented with the current element. In the case of loops or arcs, the origin must be shifted to the loop or arc center (XS, YS, ZS) . The system must also be rotated until the arc or loop lies in the XL, YL plane with the observer in the XL, ZL plane. Here XL, YL , and ZL denote the coordinates measured in the local preferred coordinate (LPC) system. For lines in the LPC system, the current element line extends from the origin (XS, YS, ZS) in the laboratory coordinates) along the positive Z axis while the observer is again located in the XL, ZL plane with coordinates $XLLOC, 0$, and $ZLOC$.

The data for the loops and arcs already specify the Euler angles α and β together with the arc radius A . For arcs, the angles θ_1 and θ_2 are also given to specify the beginning and the ending of the arc. For lines the

beginning and ending Cartesian coordinates of the line segment are given. The preferred coordinates have their origin at XS, YS, ZS while the angles α and β are solved from the X2, Y2, Z2 coordinates that specify the end of the line segment.

Figure 2 displays the Euler angle rotations.³ The angles θ_1 and θ_2 are specified along the ψ angular displacement. The coordinates XT, YT, ZT denote the result of translating the origin to the center of the current element.

(a)

(b)

(c)

Fig. 2. Euler angle rotations. The three separate rotations shown here are performed in sequence to transform from the laboratory system to the preferred local coordinate system. Also displayed are typical coil elements in the local preferred system, labeled by XP, YP, and ZP.

That is,

$$\begin{aligned}XTO &= XO - XS, \\XTO &= YO - YS, \\ZTO &= ZO - ZS,\end{aligned}\tag{1}$$

where the coordinates of the observer are relative to the current element position (XS, YS, ZS). This is followed by the rotation about the ZT axis, through the primary Euler angle α . These coordinates are,

$$\begin{aligned}XAO &= XTO \cos \alpha + YTO \sin \alpha, \\YAO &= -XTO \sin \alpha + YTO \cos \alpha, \\ZAO &= ZTO.\end{aligned}\tag{2}$$

The secondary Euler rotation through the angle β about the XA axis gives

$$\begin{aligned}XBO &= XAO, \\YBO &= YAO \cos \beta + ZAO \sin \beta, \\ZBO &= -YAO \sin \beta + ZAO \cos \beta.\end{aligned}\tag{3}$$

The final Euler rotation through the angle ψ about the ZB axis yields the local preferred coordinates,

$$\begin{aligned}XLOC &= XBO \cos \psi + YBO \sin \psi, \\YLOC &= -XBO \sin \psi + YBO \cos \psi, \\ZLOC &= ZBO.\end{aligned}\tag{4}$$

Because ψ is not given in the data, we define it by requiring $YLOC = 0$. Using Eq. (4), we find

$$\psi = \arctan (YBO/XBO).\tag{5}$$

In straight line current elements, neither α nor β are given but $X2$, $Y2$, and $Z2$ are provided in the data. Here we find α and β by requiring that the local preferred coordinate of the current line segment end be given by

$$XL2 = YL2 = 0; |ZL2| = \text{length of segment}.\tag{6}$$

From the definitions of the Euler transformations of $X2$, $Y2$, $Z2$, we find

$$\alpha = \arctan (-XT2/YT2),\tag{7}$$

$$\beta = \arctan (-YA2/ZA2),\tag{8}$$

where $XT2$, $YT2$, $YA2$, and $ZA2$ are defined in analogy to XTO , YTO , YAO , and ZAO as given in Eqs. (1) and (7).

Finally, ZL2 is given by the Euler rotations by the following sequence:

$$\begin{aligned}YA2 &= -XT2 \sin \alpha + YT2 \cos \alpha, \\ZA2 &= ZT2,\end{aligned}\tag{9}$$

$$ZB2 = YA2 \sin \beta + ZA2 \cos \beta,\tag{10}$$

$$ZL2 = ZB2.\tag{11}$$

The angles ϕ_1 and ϕ_2 , specifying the beginning and ending of the arc, must be shifted in the local preferred system by

$$\begin{aligned}PHI1L &= \phi_1 - \psi \\PHI2L &= \phi_2 - \psi.\end{aligned}\tag{12}$$

The details of obtaining the local Cartesian field components due to the current element as seen by the observer (AXLOC, AYLOC, AZLOC, BXLOC, BYLOC, BZLOC) are left to the following sections. Once these Cartesian components have been obtained, they must then be rotated back to the laboratory system. The algorithm for the A components is

$$\begin{aligned}AXB &= AXLOC \cos \psi - AYLOC \sin \psi, \\AYB &= AXLOC \sin \psi + AYLOC \cos \psi, \\AXB &= AZLOC;\end{aligned}\tag{13}$$

$$\begin{aligned}AXA &= AXB, \\AYA &= AYB \cos \beta - AZB \sin \beta, \\AZA &= AYB \sin \beta + AZB \cos \beta;\end{aligned}\tag{14}$$

$$\begin{aligned}AX &= AXA \cos \alpha - AYA \sin \alpha, \\AY &= AXA \sin \alpha + AYA \cos \alpha, \\AZ &= AZA.\end{aligned}\tag{15}$$

An identical formulation is used to obtain BX, BY, and BZ. The rotation is performed in the appropriate subroutine (LINES or LOOPARC) and the back rotation is implemented as a cliché. These subroutines are each invoked once for each observer point/current element pair.

Fields from Straight Line Currents Elements

Here we review the details of performing the straight line segment integrals to obtain A and B. Before calculating the special results from which we obtain the preferred local coordinates, we must evaluate the general expression involved. Figure 3 shows the basic arrangement of the problem and

illustrates the displacement vectors between the current element and the observer.

The equation for the line segment is

$$\underline{r} - \underline{\rho}_1 = s(\underline{\rho}_2 - \underline{\rho}_1), \quad (16)$$

where $0 < s < 1$ over the segment.

A vector increment along the segment can be written as

$$d\underline{\ell} = ds (\underline{\rho}_2 - \underline{\rho}_1), \quad (17)$$

and rearranging Eq. (16) gives

$$\underline{r} = \underline{\rho}_1 + s(\underline{\rho}_2 - \underline{\rho}_1). \quad (18)$$

To obtain the vector potential $\underline{A}$, we evaluate the form,²

$$\underline{A} = \frac{\mu_0 I}{4\pi} \int_1^2 \frac{d\underline{\ell}}{r}, \quad (19)$$

that is consistent with the Coulomb Gauge $\nabla \cdot \underline{A} = 0$.

The magnitude r is given as

$$r = \left\{ \underline{\rho}_1^2 + s \left[(2\underline{\rho}_2 \cdot \underline{\rho}_1) - (\underline{\rho}_2^2) \right] + s^2 \left[(\underline{\rho}_2^2 + \underline{\rho}_1^2) - (2\underline{\rho}_2 \cdot \underline{\rho}_1) \right] \right\}^{1/2}. \quad (20)$$

Fig. 3. Straight line current element and observer in preferred local coordinate system. (See Appendix A for definitions of variables.)

This is of the form

$$r = (a + bs + cs^2)^{1/2}, \quad (21)$$

where

$$\begin{aligned} a &= \rho_1^2, \\ b &= (2\rho_2 \cdot \rho_1) - 2\rho_1^2, \\ c &= (\rho_2^2 + \rho_1^2) - (2\rho_2 \cdot \rho_1). \end{aligned} \quad (22)$$

From Gradshteyn⁴ (integral 2.261) we obtain

$$\begin{aligned} \underline{A} &= (\rho_2 - \rho_1) \frac{\mu_0 I}{4\pi} \int_0^1 \left[\frac{ds}{(a + bs + cs^2)^{1/2}} \right] \\ &= (\rho_2 - \rho_1) \frac{\mu_0 I}{4\pi} \int_0^1 \left[\frac{1}{\sqrt{c}} \ln (2 cr + 2 cs + b) \right] ds. \end{aligned} \quad (23)$$

Working through the algebra, we find the component of $\underline{A}$ parallel to the wire where l is the length of the wire segment:

$$\underline{A} = \frac{\mu_0 I}{4\pi} \ln \left[\frac{(\ell\omega_2 + \rho_2^2) - (\rho_2 \cdot \rho_1)}{(\ell\omega_1 - \rho_1^2) + (\rho_2 \cdot \rho_1)} \right]. \quad (24)$$

Next, we specialize to the preferred local coordinate system, where

$$\begin{aligned} \rho_2 \cdot \rho_1 &= (x_0^2 + z_0^2) - (z_0 z_2), \\ \rho_2^2 &= x_0^2 + (z_0^2 + z_2^2) - 2(z_0 z_2), \\ \rho_1^2 &= x_0^2 + z_0^2, \end{aligned} \quad (25)$$

where $x_0 = XLOC$, $y_0 = YLOC$, and $z_0 = ZLOC$ as defined in Eq. (4). Finally we obtain the simple results

$$A_z = \frac{\mu_0 I}{4\pi} \ln \left(\frac{\rho_2 + z_2 - z_0}{\rho_1 - z_0} \right),$$

$$A_x = A_y = 0. \quad (26)$$

When the observer is co-linear with the current element but beyond the wire, $\rho_1 = z_0$ and A_z is not defined. Taking the limit of A_z as $x_0 \rightarrow 0$, we recover a well-defined result:

$$A_z = \frac{\mu_0 I}{4\pi} \ln \left| \frac{z_0}{z_0 - z_2} \right|, \quad (27)$$

when x_0 is zero or very small ($|x_0| < |z_0| \times 10^{-3}$).

For the magnetic fields, the form analogous to Eq. (19) is the Biot-Savart law;

$$\underline{B} = \frac{-\mu_0 I}{4\pi} \int_1^2 \frac{d\ell \times \underline{r}}{r^3}. \quad (28)$$

Making the substitutions into Eqs. (20) through (22), we find

$$\underline{B} = -\frac{\mu_0 I}{4\pi} (\underline{\rho}_2 \times \underline{\rho}_1) \int_0^1 \left[\frac{ds}{(a + bs + cs^2)^{3/2}} \right], \quad (29)$$

and referring to Gradshteyn⁴ (integral 2.264.5) we have

$$\underline{B} = \frac{\mu_0 I}{4\pi} (\underline{\rho}_2 \times \underline{\rho}_1) \left\{ \frac{2 \left[a^{1/2} 2(c+b) - (a+b+c)^{1/2} b \right]}{(4ac - b^2)(a+b+c)^{1/2} a^{1/2}} \right\}. \quad (30)$$

Using Eq. (22), we convert this to

$$\underline{B} = -\frac{\mu_0 I}{4\pi} (\rho_2 \times \rho_1) \left[\frac{(\rho_1 + \rho_2)}{(\rho_1 \rho_2) (\rho_1 \rho_2 + \underline{\rho}_1 \cdot \underline{\rho}_2)} \right]. \quad (31)$$

Taking the local coordinates into account, Eq. (31) can be written

$$B_y = \frac{\mu_0 I}{4\pi} (x_0 z_2) \frac{(\rho_1 + \rho_2)}{(\rho_1 \rho_2) (\rho_1 \rho_2 + \underline{\rho}_1 \cdot \underline{\rho}_2)}, \quad (32)$$

where $B_x = B_z = 0$. We can check this result by applying $\underline{B} = \nabla \times \underline{A}$ to Eq. (26).

To handle the singular case where the observer is on the wire, we have assigned a finite radius R_{wire} to all the wire current elements. This eliminates the infinite delta function that is normally assumed for current densities. For straight lines, the following algorithm replaces the above formulas when an observer is inside the wire:

$$B_y = 2 \frac{\mu_0 I}{4\pi} x_0 / (R_{\text{wire}}^2),$$

$$A_z = x_0 B_y,$$

$$A_x = A_y = B_x = B_z = 0.$$

(33)

The subroutine LINES implements all the formulas derived in this section.

Fields from Arc or Loop Current Elements

Evaluation of the field integrals for arc current element segments is not quite so simple as it is for straight lines. Rather than deriving the field components in a general coordinate system, we specialize to the local preferred system at the outset. For this integration, we use cylindrical coordinates locally. Because the observer is at the $Y_0 = 0$ plane, the cylindrical components A_r and A_ϕ are given as

$$A_r = A_x,$$

$$A_\phi = A_y.$$

(34)

By virtue of the third Euler rotation in angle ψ , the beginning and ending

Fig. 4. Arc segment current element and observer in preferred local coordinate system. (See Appendix A for definitions of variables.)

angles, ϕ_1 and ϕ_2 , are transformed to

$$\begin{aligned}\phi'_1 &= \phi_1 - \psi, \\ \phi'_2 &= \phi_2 - \psi.\end{aligned}\quad (35)$$

These are defined in the local preferred coordinate system and measure angles relative to the local YL = 0 plane about the ZL axis. In Fig. 4, the arc current element in these coordinates is shown.

To perform the integral of the form given in Eq. (19), we follow Jackson⁵ (Eqs. 5.35 and 5.36) and write

$$A_y = -\frac{\mu_0 I}{4\pi} a \int_{\phi_1}^{\phi_2} \left\{ \frac{\cos \phi (d\phi)}{\left[(a^2 + x_0^2 + z_0^2) - 2(ax_0 \cos \phi) \right]^{1/2}} \right\}. \quad (36)$$

Here a is the arc radius, $(x_0, 0, z_0)$ is the local observer position, and ϕ is the angle along the arc. (This formula is somewhat different from Jackson because we use cylindrical coordinates where he uses spherical coordinates.) Similarly,

$$A_x = -\left(\frac{\mu_0 I}{4\pi}\right) a \int_{\phi_1}^{\phi_2} \left\{ \frac{\sin \phi (d\phi)}{\left[(a^2 + x_0^2 + z_0^2) - 2(ax_0 \cos \phi) \right]^{1/2}} \right\}. \quad (37)$$

To relate Eq. (36) to the incomplete elliptic integrals, we transform the angles to

$$\omega = \frac{\phi' + \pi}{2}, \quad (38)$$

so that

$$\cos \phi' = \sin^2 \omega - \cos^2 \omega - 2 \sin^2 \omega - 1. \quad (39)$$

If we make the usual definition that

$$k^2 = \frac{4ax_0}{(a+x_0)^2 + z_0^2}, \quad (40)$$

then the integral for A_y can be reduced to

$$A_y = \left(\frac{\mu_0 I}{4\pi}\right) a \left\{ \int_{\omega_1}^{\omega_2} \left[\frac{1}{(2ax_0)^{1/2}} \right] \left(\frac{a^2 + x_0^2 + z_0^2}{ax_0} \right) \left(\frac{k^2}{2} \right)^{1/2} (1 - k^2 \sin^2 \omega)^{-1/2} d\omega \right. \\ \left. - \int_{\omega_1}^{\omega_2} \left[\frac{2}{(2ax_0)^{1/2}} \right] \left(\frac{2}{k^2} \right)^{1/2} (1 - k^2 \sin^2 \omega)^{1/2} d\omega \right\}. \quad (41)$$

A subroutine ILIPTIC⁶ in the code computes the extended incomplete elliptic integrals,

$$F^*(\omega_1, \omega_2, k) = F(\omega_2, k) - F(\omega_1, k), \\ E^*(\omega_1, \omega_2, k) = E(\omega_2, k) - E(\omega_1, k). \quad (42)$$

Alternatively, they are given by

$$F^*(\omega_1, \omega_2, k) = \int_{\omega_1}^{\omega_2} (1 - k^2 \sin^2 \omega)^{-1/2} d\omega, \quad (43) \\ E^*(\omega_1, \omega_2, k) = \int_{\omega_1}^{\omega_2} (1 - k^2 \sin^2 \omega)^{1/2} d\omega.$$

With these, Eq. (41) can be written as

$$A_y = 2 \left(\frac{\mu_0 I}{4\pi} \right) \frac{a}{\sqrt{(a + x_0)^2 + z_0^2}} \left[\left(\frac{2}{k^2} - 1 \right) F^*(\omega_1, \omega_2, k) \right. \\ \left. - \frac{2}{k^2} E^*(\omega_1, \omega_2, k) \right]. \quad (44)$$

The x component in Eq. (37) is obtained by an integration using the substitution $u = \cos \phi$. We find

$$A_x = - \left(\frac{\mu_0 I}{4\pi} \right) \left(\frac{1}{x_0} \right) \left[(a^2 + x_0^2 + z_0^2) - 2(ax_0 u) \right]^{1/2} \frac{u_2}{u_1}. \quad (45)$$

Finally, $Z_z = 0$ because the current element has no z component.

The B components may be derived in a similar fashion from the Biot-Savart law or we may operate the curl on the A vector to generate the B values.

Either of these methods generates the following formulas, identical to those found in the MAFCO report:

$$\begin{aligned} B_x &= \left(\frac{\mu_0 I}{4\pi} \right) z_0 (A\bar{Q}_1), \\ B_y &= \left(\frac{\mu_0 I}{4\pi} \right) z_0 (A\bar{Q}_3), \\ B_z &= \left(\frac{\mu_0 I}{4\pi} \right) [a(A\bar{Q}_0) - x_0 (A\bar{Q}_1)] , \end{aligned} \quad (46)$$

where

$$\begin{aligned} (A\bar{Q}_0) &= \left(\frac{k^2}{1 - k^2} \right) \left(\frac{1}{2x_0} \right) [(a + x_0)^2 + z_0^2]^{-1/2} \\ &\times [E(\omega_2, k) - E(\omega_1, k)] - \left[\frac{k^2 \sin \omega_2 \cos \omega_2}{(1 - k^2 \sin^2 \omega_2)^{1/2}} \right] + \left[\frac{k^2 \sin \omega_1 \cos \omega_1}{(1 - k^2 \sin^2 \omega_1)^{1/2}} \right] , \\ (A\bar{Q}_1) &= \frac{1}{x_0} (a + x_0^2 + z_0^2)^{-1/2} [F(\omega_1, k) - F(\omega_2, k)] + \left(\frac{2 - k^2}{k^2} \right) (A\bar{Q}_0), \\ (A\bar{Q}_3) &= \frac{1}{x_0} (a + x_0^2 + z_0^2)^{-1/2} \left[(1 - k^2 \sin^2 \omega_1)^{-1/2} - (1 - k^2 \sin^2 \omega_2)^{-1/2} \right] . \end{aligned} \quad (47)$$

In the limit of small k , the formulas in Eq. (47) may become undefined. Expanding these in a series expansion for small k gives the formulas used when, for example, $k^2 < 10^{-3}$. Again following MAFCO,

$$\begin{aligned} B_x &= \left(\frac{\mu_0 I}{4\pi} \right) \frac{a \{ z_0 [\sin(2\omega_1) - \sin(2\omega_2)] \}}{(a^2 + z_0^2)^{3/2}} , \\ B_y &= \left(\frac{\mu_0 I}{4\pi} \right) \frac{2a [z_0 (\sin^2 \omega_1 - \sin^2 \omega_2)]}{(a^2 + z_0^2)^{3/2}} , \\ B_z &= \left(\frac{\mu_0 I}{4\pi} \right) \left[\frac{2a^2}{(a^2 + z_0^2)^{3/2}} \right] (\omega_2 - \omega_1) . \end{aligned} \quad (48)$$

Similarly, Eqs. (44) and (45) have limiting forms to use for small k , $k^2 < 10^{-3}$. They are

$$\begin{aligned} A_y &= \left(\frac{\mu_0 I}{4\pi} \right) \left[\frac{a}{(a^2 + x_0^2 + z_0^2)^{1/2}} \right] [\sin(2\omega_1) - \sin(2\omega_2)], \\ A_x &= \left(\frac{\mu_0 I}{4\pi} \right) \left[\frac{a}{(a^2 + x_0^2 + z_0^2)^{1/2}} \right] [u_2 - u_1], \\ A_z &= 0. \end{aligned} \quad (49)$$

For the special case of a current loop, the above formulas reduce to

$$A_y = \left(\frac{\mu_0 I}{4\pi} \right) \left\{ \frac{4a}{[(a + x_0)^2 + z_0^2]^{1/2}} \right\} \left\{ \frac{[(2 - k^2) K(k)] - 2E(k)}{k^2} \right\}, \quad (50)$$

where $K(k)$ and $E(k)$ are the complete elliptic integrals of the first and second kind. We note here that $F^*(0, \pi, k) = 2K(k)$ and $E^*(0, \pi, k) = 2E(k)$. Also

$$A_x = A_z = 0. \quad (51)$$

For the magnetic field of a loop we find for $k > 10^{-3}$,

$$B_x = \left(\frac{\mu_0 I}{2\pi} \right) \left[\frac{z_0}{x_0 (a^2 + x_0^2 + z_0^2)^{1/2}} \right] \left[-K(k) + \left(\frac{a^2 + x_0^2 + z_0^2}{a^2 - x_0^2 + z_0^2} \right) E(k) \right], \quad (52)$$

or $B_x = 0$ for $k < 10^{-3}$. The other components for all k values are

$$\begin{aligned} B_y &= 0, \\ B_z &= \frac{\mu_0 I}{2\pi} (a^2 + x_0^2 + z_0^2)^{-1/2} \left[K(k) + \left(\frac{a^2 - x_0^2 - z_0^2}{a^2 - x_0^2 + z_0^2} \right) E(k) \right]. \end{aligned} \quad (53)$$

Sample Problem: Fields of 2XIIB

In Fig. 5, we show the coil arrangement of the 2XIIB plasma confinement device. These coils are represented by 80 arcs, 70 loops, and zero straight line segments.

Fig. 5. Coils for 2XIIB.

Using the data structure for MAFCO, we obtained the field on a $12 \times 12 \times 12$ grid comprising one octant of the device. Fields in the other octants can be obtained by symmetry relations so, in effect, we can determine the fields for the entire volume enclosed by the coils.

To show all of the diagnostic options available with this code we present a multitude of diagnostics. First, a print-out of the A_x array with our compressed exponential form given by the PRNT routine^{7,8} is presented in Table 1, a photo-reproduction of the actual output. (We can also output A_y , A_z , B_x , B_y , B_z , or $|B|$ by this PRNT routine.)

Contour plots can be obtained for any of the above arrays in any grid plane. In Fig. 6, we show $|B|$ contours at the $x = 0$, $y = 0$, and $z = 0$ planes. In plasma devices these contours when closed indicate that confinement of plasma is possible.

Grid line plots are useful in many applications and any of the above quantities can be plotted along a grid line in any of the three directions. In Fig. 7, plots of $|B|$ along the grid lines $x = 0$, $y = 0$; $x = 0$, $z = 0$; and $y = 0$, $z = 0$ are presented.

Table 1. The A_x array is displayed as an example of the compressed output from the PRNT routine. The three-digit mantissa is preceded by an understood decimal point. The single-digit exponent uses letters for exponents in the range 10 to 35, given by the letters A to Z, respectively.

```

1009
1010 OCPU,SYS,I/O UNWNTED REAL TIME FOR CASE(MILLISEC)
1011      51735      0      190
1012 THERE ARE FINAL RESULTS...
1013 I
1014 AXA
1015
1016 O K=      I
1017
1018      I= 1 I= 2 I= 3 I= 4 I= 5 I= 6 I= 7 I= 8 I= 9 I=10 I=11 I=12
1019 J= 1 196+5 196+5 196+5 196+5 196+5 195+5 194+5 193+5 192+5 190+5 189+5 187+5
1020 J= 2 205-3 202-3 205-3 213-3 225-3 242-3 268-3 304-3 340-3 374-3 405-3 432-3
1021 J= 3-196+5-196+5-196+5-196+5-196+5-195+5-194+5-193+5-192+5-190+5-189+5-187+5
1022 J= 4-392+5-392+5-392+5-391+5-390+5-389+5-387+5-385+5-383+5-380+5-377+5-373+5
1023 J= 5-586+5-587+5-586+5-585+5-584+5-582+5-579+5-576+5-572+5-568+5-563+5-557+5
1024 J= 6-779+5-779+5-779+5-777+5-775+5-772+5-769+5-767+5-759+5-753+5-746+5-739+5
1025 J= 7-969+5-969+5-969+5-967+5-964+5-961+5-956+5-950+5-943+5-935+5-927+5-917+5
1026 J= 8-116+6-116+6-116+6-115+6-115+6-115+6-114+6-113+6-112+6-111+6-110+6-109+6
1027 J= 9-134+6-134+6-134+6-134+6-133+6-133+6-132+6-131+6-130+6-129+6-127+6-126+6
1028 J=10-152+6-152+6-152+6-152+6-151+6-151+6-150+6-149+6-147+6-146+6-144+6-142+6
1029 J=11-170+6-170+6-170+6-169+6-169+6-168+6-167+6-166+6-164+6-162+6-160+6-158+6
1030 J=12-187+6-187+6-187+6-186+6-186+6-185+6-184+6-182+6-180+6-178+6-176+6-173+6
1031
1032 O K=      2
1033
1034      I= 1 I= 2 I= 3 I= 4 I= 5 I= 6 I= 7 I= 8 I= 9 I=10 I=11 I=12
1035 J= 1 182+5 182+5 182+5 182+5 191+5 181+5 180+5 179+5 177+5 176+5 174+5 173+5
1036 J= 2-170-3 117-6 220-3 440-3 656-3 867-3 107-2 126-2 144-2 159-2 173-2 185-2
1037 J= 3-182+5-182+5-182+5-182+5-181+5-180+5-179+5-177+5-176+5-174+5-173+5
1038 J= 4-364+5-364+5-364+5-363+5-362+5-361+5-359+5-357+5-354+5-351+5-348+5-344+5
1039 J= 5-544+5-544+5-544+5-543+5-541+5-539+5-536+5-533+5-529+5-525+5-520+5-515+5
1040 J= 6-722+5-723+5-722+5-721+5-719+5-716+5-712+5-707+5-702+5-696+5-689+5-682+5
1041 J= 7-899+5-899+5-899+5-897+5-894+5-890+5-885+5-879+5-872+5-864+5-856+5-846+5
1042 J= 8-107+6-107+6-107+6-107+6-107+6-106+6-105+6-105+6-104+6-103+6-102+6-101+6
1043 J= 9-124+6-124+6-124+6-124+6-123+6-123+6-122+6-121+6-120+6-119+6-118+6-116+6
1044 J=10-141+6-141+6-141+6-140+6-140+6-139+6-138+6-137+6-136+6-134+6-133+6-131+6
1045 J=11-157+6-157+6-157+6-157+6-156+6-155+6-154+6-153+6-151+6-149+6-147+6-145+6
1046 J=12-173+6-173+6-173+6-172+6-172+6-171+6-169+6-168+6-166+6-164+6-161+6-159+6
1047
1048 O K=      3
1049
1050      I= 1 I= 2 I= 3 I= 4 I= 5 I= 6 I= 7 I= 8 I= 9 I=10 I=11 I=12

```

Fig. 6. Contour plots of $|B|$ at the $x = 0$, $y = 0$, and $z = 0$ planes in (a), (b), and (c), respectively.

Field line plots are also very desirable from the point of view of the physics and are produced by a field line generator similar to the one used in the MAFCO code. The field lines are generated by a string of rectangular coordinates with parameters determined by the arc length s . A corresponding string of B_x , B_y , and B_z values is stored. Two kinds of plots are available. Projections of the field lines on the $x = 0$, $y = 0$, or $z = 0$ plane can be plotted with one or several such lines plotted per graph. Figure 8 shows the

Fig. 7. Plots of $|B|$ along the $x = 0$, $y = 0$; $x = 0$, $z = 0$; and $y = 0$, $z = 0$ grid lines in (a), (b), and (c), respectively.

projections of six field lines on the $y = 0$ plane all of which start at $y = 0$, $z = 0$ but at different x values.

Plots of field quantities vs arc length s along a field line are most useful diagnostics. Figure 9 plots $|B|$ vs s for the six field lines shown in Fig. 8. Figure 10 plots r vs z for these six field lines.

Plots extending through more than one octant can also be obtained by rerunning the code with a more extensive grid. For example, if we are interested in the $y = 0$ plane, we can use a $11 \times 3 \times 11$ grid. This way, plots will cover both positive and negative values of x and z . In Fig. 11 we plot some

Fig. 8. Projection of six field lines on the $y = 0$ plane, all originating from $y = 0$, $z = 0$ but from different x values.

Fig. 9. Plot of $|B|$ vs arc length s for the six field lines in Fig. 8.

Fig. 10. Plot of magnitude r vs z for the six field lines in Fig. 8.

Fig. 11. Plot of field lines in the $y = 0$ plane on an extended $11 \times 3 \times 11$ grid to show both positive and negative values of x and z .

of the field lines in the $y = 0$ plane on this extended grid. Figure 12 plots contours of $|B|$ and A_y in this plane, also on the $11 \times 3 \times 11$ grid. (Appendix D contains the listing of the input required to generate the output described in this sample problem.)

Fig. 12. Contour plots of $|B|$ (a) and A_y (b) in the $y = 0$ plane on an extended $11 \times 3 \times 11$ grid to show both negative and positive values of x and z .

References

1. W. A. Perkins and J. C. Brown, "MAFCO - A Magnetic Field Code for Handling General Current Elements in Three Dimensions," Lawrence Livermore Laboratory, Livermore, Rept. UCRL-7744 Rev. 2 (1966).
2. B. Johnston, Lawrence Livermore Laboratory, private communication (1975).
3. H. Goldstein, *Classical Mechanics* (Addison-Wesley, London, 1950), p. 107.
4. I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products* (Academic Press, New York, 1965), pp. 81-83.
5. J. D. Jackson, *Classical Electrodynamics* (John Wiley & Sons, New York, 1962), pp. 141-142.
6. The subroutines ILIPTIC and IEI (called by ILIPTIC) are taken from the MAFCO coding: see code listing in Appendix C for details.
7. R. Lee, Naval Research Laboratory, Washington D.C., private communication (1973).
8. J. Breazeal, Lawrence Livermore Laboratory, memorandum on "PRNT, a subroutine which prints an array in a compressed tabular format," (June 10, 1975).

Appendix A: Data Structure

Except for the data cards specifying the current elements, all the data is read in by NAMELIST.* The variables to be defined by the NAMELIST data are identified in the following list. All physical variables are in laboratory units and default values are zero unless noted otherwise.

IBOX	User's box number (e.g., 7HBOX582).
IOUTFIL	Name of binary file created that contains the vector potential and magnetic field (e.g., 5HTXIIV).
NDOPT	Data option = MAFCO or = ABCXYZ alphanumeric data (e.g., 5HMAFCO).
IAFLD	Compute vector potential when = 1.
IBFLD	Compute magnetic field when = 1.
DIA	Wire diameter for all current elements.
KI	(1,KI), (1,KJ), (1,KK) denote the dimension of the 3-D Cartesian observer grid.
KJ	
KK	
DELX	Cell spacing of the observer grid.
DELY	
DELZ	
YBOT	Position of the (1,1,1) grid point.
XBOT	
ZBOT	
MFIELD	<p>Number of field lines to follow = MFIELD </p> <p>If MFIELD < 0, field line plots are superimposed.</p> <p> MFIELD ≤ 6 is a restriction. Five different plots are made for each field line:</p> <ul style="list-style-type: none"> ● B vs s (the arc length), ● x vs y (x,y plane projection), ● y vs z (y,z plane projection), ● x vs z (x,z plane projection), ● r vs z (r,z "azimuthal" projection).
XBAR (6)	Coordinates of starting point on the field line; up to six lines are permitted
YBAR (6)	
ZBAR (6)	

* LTSS chapter 207, LRLTRAN language.

DELMAX (6) Increment in s to use in field line generation.
 STOTAL (6) Maximum length to follow a field line.
 NSKIP (6) Output and plot every NSKIP point along field line.

Contour plots and grid line plots for the field values $|B|$, $|A|$, A_x , A_y , A_z are requested by entering the NAMELIST input quantities that specify the maximum and minimum values of the contour lines. The particular contour plots and the grid line plots are not drawn if the limiting values are set equal to each other. The purpose of limiting values is to avoid bunching contour lines around current carrying elements. Twenty contour lines are placed between limiting values. The following variables specify limiting values for their respective parameters:

BMAXPLT, BMINPLT	$ B $,
AMAXPLT, AMINPLT	$ A $,
AXMAXPLT, AXMINPLT	A_x ,
AYMAXPLT, AYMINPLT	A_y ,
AZMAXPLT, AZMINPLT	A_z .

The plane of the contour plot of $|B|$, $|A|$, A_x , A_y , A_z is determined by the NGR array.

NGR (1) I (contour plot of y, z plane at $x(I)$; zero = "don't plot").
 NGR (2) J (contour plot of x, z plane at $y(J)$; zero = "don't plot").
 NGR (3) K (contour plot of x, y plane at $z(K)$; zero = "don't plot").

The six grid-line plots of the array (ordinate) are determined by input to the IGR, IXGR, IYGR, and IZGR arrays.

IGR 0, 1, 2, or 3 refer to no plot; x , y , or z grid-line plot, respectively.
 IXGR index of x coordinate of a point on the line.
 IYGR index of y coordinate of a point on the line.
 IZGR index of z coordinate of a point on the line.

The IGR array indicates the direction of the abscissa. The location of the line is given by IXGR, IYGR, and IZGR. The first item in each array pertains to the first line, the 2nd item pertains to the second line, etc.

The following NAMELIST values are needed only for MAFCO-type current element data.

NLOOP	Number of loops.
NARC	Number of arcs.
NLINE	Number of straight lines.

NGCL Number of general current element groups.
 STOP Terminates NAMELIST data.

For MAFCO-type data, the following formatted data is read in after the NAMELIST data to describe the current elements.

- CIRCULAR LOOP cards (one for each loop in the problem) containing XS, YX, ZX, A α , β , CUR: Format (7F10.3).
- ARC Cards (two for each arc in this problem) containing XS, YS, ZS, A, α , β , CIR: Format (7F10.3) on card one, and Phi1, Phi2: Format (2F10.3) on card two.
- STRAIGHT LINE cards (one for each straight line in the problem) containing XS, YS, ZS, X2, Y2, Z2, CUR (from 1 to 2): Format (7F10.3).
- GENERAL CURRENT ELEMENT cards (one group for each element in this problem) containing on the first card, X, Y, Z of the first point, current along points, number of points: Format (4F10.3, 15). On following cards, X, Y, Z of succeeding points: Format (3F10.3).

For ABCXYZ type data, STOP should be placed after the value for IZGR. Formatted data is then read as described below.

GENERAL DATA CARD FOR ABCXYZ CURRENT ELEMENT: Format (10F.3)

ITYPE Flags last card when negative; |ITYPE| = 1,2,3 for lines, loops, and arcs respectively.

CUR Current in amperes.

XS }
 YS } Position of current element in centimeters.
 ZS }

	Lines	Loops	Arcs
D1	X2	A	A
D2	Y2	*	ϕ_1
D3	Z2	*	ϕ_2
D4	*	α	α
D5	*	β	β

Last card in the ENTIRE DATA FILE: Format (A10).

* Unused blank value.

Appendix B: Timing Analysis

A timing comparison between program ABCXYZ and FMAFJ3D (a recent MAFCO version) on a $10 \times 10 \times 10$ grid produced the following results (times are for calculation of one grid point/current element pair).

	FMAFJ3D	ABCXYZ	ABCXYZ
	computed B-field (sec)	computed B-field (sec)	computed A-field and B-field (sec)
Loop element (100 elements processed)	6.75×10^{-5}	6.47×10^{-5}	7.19×10^{-5}
Line element (100 elements processed)	2.42×10^{-5}	3.43×10^{-5}	4.62×10^{-5}
Arc element (100 elements processed)	37.8×10^{-5}	29.3×10^{-5}	30.7×10^{-5}

The following features of the ABCXYZ code contribute significantly to the processing speed of the program.

- The data are stored in LCM and are read into SCM in blocks of 999. The trigonometric functions of α and β are computed in the initialization routine and are stored in LCM to remove redundant calls to the sin and cos functions.
- The loop and arc elements are processed by the LOOPARC routine and line elements are processed by the LINES routine. The rotation and backrotation functions are incorporated in-line to eliminate calls to subroutines. The subroutine ILIPTK and the cliché ELIPTK are borrowed from FMAFJ3D with certain annotated changes.
- We expect that the code will operate more efficiently when a large number of elements are processed because of the start-up time required to read the blocks from LCM.

Appendix C: Source Code Listing for ABCXYZ

```

1 $      CHAT ABC,,CRT$,LIST,S82,L,L
2 $      EXE ZOHCON BZOCHAR DR.
3 $      LOD S82 40000 STACKLIB, BZOCHAR, ORDERLIB, CRT$ ; 1 RUN+ 1* LODMAP
4 $      RUN TX!!!
5
6 C
7 C
8 C..PROGRAM ABCXYZ
9 C..VERSION 1 NOVEMBER 18, 1974
10 C..ORIGINATOR: DAVID ANDERSON
11 C..PROGRAMMER: JULIE BREAZEAL
12
13 C..SKETCH OF MAFCO'S SUCCESSOR
14 C...ROUTINES BORROWED FROM MAFCO: CLICHE ELIPTK, ILIPTK, IEI
15 C... A CARTESIAN VECTOR POTENTIAL AND MAGNETIC FIELD CODE FOR ARBITRARY
16 C...3-D CURRENT ELEMENTS
17
18 C..THE BINARY FILE BZOCHAR CONTAINS THE SUBROUTINE CONTUR WRITTEN BY
19 C      BRUCE LANGDON
20
21
22 C..MAX GRID SIZE DETERMINED BY IMAX,JMAX,KMAX.MUST RECOMPILE IF GRID
23 C      IS NOT SUFFICIENT. DYNAMIC DIMEN MODS WILL FIX THIS.
24
25 C..CHANGES NEED:
26 C      1.SET UP DYNAMIC DIMENSIONING ON ARRAYS BY INPUT OF GRID SIZES
27 C      2.DO TIMING ANALYSIS
28 C      3.ALLOW FOR ARBITRARY GRID BY FIX IN INITIALIZATION
29
30
31      CREATE MACROS (2048,20)
32
33      CLICHE DIMM (MACROS)
34      COMMON/KKK/ KI, KJ, KK
35      PARAMETER (IMAX = 11)
36      PARAMETER (JMAX = 3)
37      PARAMETER (KMAX = 11)
38      PARAMETER (MAXDM = 1MAX)
39      ENDCliche
40
41 C*****
42 C

```

```

43 C          CLICHE STORAGE
44 C
45 C.....
46          CLICHE STORAGE (MACROS)
47
48          OPTIMIZE
49
50          MACRO DIMM (MACROS)
51          USE DIMM
52
53          LCM (AB)
54
55          PARAMETER (IODEBUG=1)
56          PARAMETER (MFF = 20)
57          PARAMETER (MSTOP = 999)
58          PARAMETER (MAP=1)
59          PARAMETER (IMFLG = 0)
60
61
62          PARAMETER (ILINE = 1)
63          PARAMETER (ILOOP = 2)
64          PARAMETER (IARC = 3)
65
66          PARAMETER (JIN = 59)
67          PARAMETER (JOUT = 59)
68          PARAMETER (NI = 4)
69          PARAMETER (PI = 3.14159271)
70          PARAMETER (FAC4 = 1./(4.*PI))
71          PARAMETER (PIHALF = .5*PI)
72          PARAMETER (PIX2 = 2.*PI)
73          PARAMETER (FMU = 4.*PI*.E-1)
74
75          PARAMETER (NSIZE = 1300)
76          PARAMETER (NT = 1001)
77          PARAMETER (DEG2RAD=1./57.2957795131)
78          PARAMETER (FMUFAC4 = FMU*FAC4)
79
80          ABSOLUTE LOC2(2),LOC3(3)
81          ABSOLUTE LOC16(16B),LOC17(17B)
82          INTEGER ANAME
83          DIMENSION SUPER(IJMAX,JMAX,KMAX,7)
84

```

```

85     DIMENSION XZLC(INSIZE),ALC(INSIZE),YZLC(INSIZE),ZZLC(INSIZE),
86     .   ALPHALC(INSIZE),BETALC(INSIZE)
87     DIMENSION PH1ZLC(INSIZE),PH2ZLC(INSIZE),XTITLE(30)
88
89     DIMENSION ITV(INSIZE),ALS(INSIZE),ALCOS(INSIZE),BTS(INSIZE),BTC(INSIZE)
90
91     C=INSIZE*MAX. NUMBER OF DATA ELEMENTS INPUT
92     DIMENSION ALPHA(INSIZE),BETA(INSIZE),PPH1(INSIZE),PPH2(INSIZE),
93     .   XZINT(1),YZINT(1),ZZINT(1),A(1),PH1(INSIZE),PH2(INSIZE),
94     .   COMMON/AB/AYA(1:MAX,JMAX,KMAX),AYA(1:MAX,JMAX,KMAX),AZA(1:MAX,JMAX,KMAX),
95     .   BXA(1:MAX,JMAX,KMAX),BYA(1:MAX,JMAX,KMAX),BZA(1:MAX,JMAX,KMAX),
96     .   BMDA(1:MAX,JMAX,KMAX)
97
98     COMMON/BACKR/ BXLOC,BYLOC,BZLOC,SP,CP,AXLOC,AYLOC,AZLOC
99     COMMON/CAS/ AX,AY,AZ, BX,BY,BZ, ASC,ASC,ASC,CURASC,,
100     .   ASC, ITYPESC,XZSC,YZSC,ZZSC,
101     .   CB,SB,CA,SA
102
103
104     COMMON/DATSCM/ ITYPEINT,CURINT,XSINT,YSINT,ZSINT,
105     .   D1INT,D2INT,D3INT,ALSININT,ALCOSINT,BETASININT,
106     .   BETACOSINT,
107     .   SCAR1(INSIZE),SCAR2(INSIZE),SCAR3(INSIZE)
108
109     LCM (DATLC)
110     COMMON/DATLC/ ITYPELC(INSIZE), XSLC(INSIZE), YSLC(INSIZE), ZSLC(INSIZE),
111     .   D1LC(INSIZE), D2LC(INSIZE), D3LC(INSIZE),
112     .   D4LC(INSIZE),D5LC(INSIZE),CURLC(INSIZE),
113     .   ALSINLC(INSIZE),ALCOSLC(INSIZE),BETASINLC(INSIZE),
114     .   BETACOSLC(INSIZE)
115
116     COMMON/ILIP/OMEGA1,OMEGA2,CAY,CAYSQ,FINCMP,EINCMP,
117     .   FIRSTK, SCNOK
118
119     COMMON/INPT/ IDATA,NDOPT,DIA,XMU1,XMU1X2,
120     .   AXMU1X2,IC,MFIELD,MFACTIVE,SBX,SBY,SBZ,IAFLD,IBFLD
121     .   ,MFABS, ITITLE(30), IOUTFIL,
122     .   NLOOP,NARC,NLINE,NGCL
123
124     COMMON/MISC/ GAM(4), UZ(21,RWIRE,RWIRESQ,IBETA(4),IXPK(16)
125     .   ,IBOLD(4), RRWIRESQ,ISING, INFIL,ITDIM,JJDIM,IIMIN,IIMAX,
126     .   JUMIN,JUMAX,BMODMIN

```

```

127
128 COMMON/LAB/ ANAME(10)
129
130 COMMON/OB/IOB,JOB,KOB,XO,YO,ZO,X(IMAX),Y(JMAX),Z(KMAX),
131 . DELX,DELY,DELZ,XBOT,YBOT,ZBOT,IBUG
132
133 COMMON/ROT/ XLOC,YLOC,ZLOC,ZL2,PHI1L,PHI2L,PS1
134 COMMON /PLTCOM/ NPL(3),NGR(3),IXGR(6),IYGR(6),IZGR(6),IGR(6).
135 . BMINPLT,BMAXPLT,AMINPLT,AMAXPLT,AXMINPLT,AXMAXPLT,
136 . AYMINPLT,AYMAXPLT,AZMINPLT,AZMAXPLT
137
138 COMMON /FIELDLN/ XBAR(MFF),YBAR(MFF),ZB,R(MFF),
139 . DELMAX(MFF), STOTAL(MFF), NSKIP(MFF), XF(MSTOP), YF(MSTOP), ZF(MSTOP),
140 . SAPROX(MSTOP),BMAGF(MSTOP),BXFLD(MSTOP),BYFLD(MSTOP),
141 . BZFLD(MSTOP) ,
142 . DXRK(6),DYRK(6),DZRK(6),XRK(6),YRK(6),ZRK(6)
143
144 C..IBFLD CAN BE INPUT. NONZERO=COMPUTE MAGNETIC FIELD
145 C..IAFLD CAN BE INPUT. NONZERO =COMPUTE VECTOR POTENTIAL
146 DATA (IBFLD = 0)
147 DATA (IAFLD = 0)
148 DATA (NPL = 1,2,3)
149 DATA (NGR = 0,0,0)
150 DATA (IXGR = 0,0,1,5,1,5)
151 DATA (IYGR = 1,5,0,0,1,5)
152 DATA (IZGR = 1,5,1,5,0,0)
153 DATA (IGR = 0,0,0,0,0,0)
154
155 EQUIVALENCE (D1LC,X2LC,ALC)
156 EQUIVALENCE (D2LC,Y2LC,PHI1LC)
157 EQUIVALENCE (D3LC,Z2LC,PHI2LC)
158 EQUIVALENCE (D4LC,ALPHALC)
159 EQUIVALENCE (D5LC,BETALC)
160 EQUIVALENCE (D1,X2,A1)
161 EQUIVALENCE (D2,Y2,PHI1)
162 EQUIVALENCE (D3,Z2,PHI2)
163 EQUIVALENCE (SCAR1,ALPHA,BETA,PPH11,PPH12)
164 EQUIVALENCE (SCAR2,ALS,ALCS,BTS,BTC)
165 EQUIVALENCE (SCAR3,ITY1)
166 EQUIVALENCE (ITITLE,XTITLE)
167 EQUIVALENCE (SUPER,AXA)
168

```



```

169      EQUIVALENCE (AZB,AZLOC), (AXAT,AXB)
170      EQUIVALENCE (BZB,BZLOC), (BXAT,BXB)
171      EQUIVALENCE (BZ,BZAT), (AZ,AZAT)
172      ENDCLICHE
173
174 C*****
175 C
176 C      CLICHE ELIPTK CLICHE USED BY ELIPTIC INTEGRAL CALCULATIONS
177 C..INPUT: CAYSQ VIA COMMON/ILIP/
178 C..OUTPUT: FIRSTK VIA COMMON/ILIP/
179 C      SCNDK VIA COMMON/ILIP/
180 C
181 C*****
182      CLICHE ELIPTK (MACROS) (4,4)
183      ETA=1.-CAYSQ
184      IF (CAYSQ) ,{4}
185      IF(ETA) , {2},
186      {1} ETALN=LOGI(1./ETA)
187      FIRSTK=1.3862944+ETA*(.096663443 + ETA*(.035900924 + ETA*(.037425637
188      + ETA*(.014511962)))+(1.5+ETA*(.12498594+ETA*(.068802486
189      + ETA*(.033283553 +ETA*(.0044178701)))))*ETALN
190      SCNDK = 1.+ETA*(.44325141 +ETA*(.062606012 +ETA*(.047573835 +
191      ETA*(.017365065)))+ETALN*ETA*(.24998368 +ETA*(.0920018 +
192      ETA*(.040696975 +ETA*(.0052644964)))
193      GO TO {3}
194      {2} FIRSTK=1.0E39
195      SCNDK=1.0
196      GO TO {3}
197      {4} FIRSTK = 1.5707963      $$$ 12/19/74 BY JOB
198      SCNDK = 1.5707963      $$$ 12/19/74 BY JOB
199      {3} CONTINUE
200      ENDCLICHE
201 C*****
202 C
203 C      CLICHE BMAKE
204 C
205 C*****
206      CLICHE BMAKE (MACROS) (4,2)
207      DIMENSION SPARAY(IMAX)
208
209      WOT 3, {1H1}
210      MOUT = {2} (LD)

```

```

211      WRITE (3,MOUT)
212
213      DO (4) K = 1,KK,1
214      WRITE (3,{3}) K
215
216      CALL PRNT ((1),IIDIM,JJDIM,IIMIN,IIMAX,JJMIN,JJMAX)
217
218
219 141 CONTINUE
220
221 {3} FORMAT (1H0, 4H K= ,15, /)
222
223      ENDCLICHE
224
225 C*****
226
227      CLICHE BACKROTATE (MACROS) (6,6)
228
229 C..INVERT ALL THE ABOVE TRANSFORMATIONS EXCEPT
230 C..NOW WE WISH TO PRODUCE THE VECTOR FIELD
231 C..RESULTS IN THE LAB SYSTEM
232 C..BACKROTATE THRU TERTIARY EULAR ANGLE PSI
233
234      DIF (1BFLD),{1}
235      BXB = BXLOC*CP - BYLOC*SP
236      BYB = BXLOC*SP + BYLOC*CP
237 C.....BZB = BZLOC      EQUIVALENCED
238 {1} CONTINUE
239
240      IF (1AFLD),{2}
241      AXB = AXLOC*CP - AYLOC*SP
242      AYB = AXLOC*SP + AYLOC*CP
243 C.....AZB = AZLOC      EQUIVALENCED
244 {2} CONTINUE
245
246      DIF (1BFLD),{3}
247 C..BACKROTATE THRU SECONDARY EULAR ANGLE BETA
248 C.....BXAT = BXB      EQUIVALENCED
249      BYAT = BYB*CB - BZB*SB
250      BZAT = BYB*SB + BZB*CB
251 {3} CONTINUE
252

```

```

253          IF (IAFLD).14)
254 C.....AXAT = AXB          EQUIVALENCED
255          AYAT = AYB*CB - AZB*SB
256          AZAT = AYB*SB + AZB*CB
257 14)      CONTINUE
258
259 C..BACKROTATE THRU PRIMARY EULER ANGLE ALPHA
260
261          DIF (IBFLD).15)
262          BX = BXAT*CA - BYAT*SA
263          BY = BXAT*SA + BYAT*CA
264 C.....BZ = BZAT          EQUIVALENCED
265 15)      CONTINUE
266
267          IF (IAFLD).16)
268          AX = AXAT*CA - AYAT*SA
269          AY = AXAT*SA + AYAT*CA
270 C.....AZ = AZAT          EQUIVALENCED
271 16)      CONTINUE
272
273          ENDCLICHE
274 C.....
275 C          MAIN PROGRAM
276 C
277 C.....
278
279          PROGRAM CARD (OUTP, TAPE3=OUTP,TAPE4)
280
281          MACRO STORAGE (MACROS)
282          USE STORAGE
283
284
285          CALL GOB (2000B,1ER,0100B,INFIL)
286 C..SUBROUTINE INITIAL ALSO READS DATA AND INITIALIZES VALUES
287
288          CALL INITIAL
289
290 AGAIN      CONTINUE
291          MFABS = ABSF(MFIELD*1.0)
292          IF (MFIELD) CALL FLDLN
293
294

```

```

295
296      CALL GOB (2411B,IER,0300B,IBETA)
297
298      IBOLD(1) = IBETA(1)
299      IBOLD(2) = IBETA(2)
300      IBOLD(3) = IBETA(3)
301 C...LOOP OVERLOOP OVER OBSERVER POINTS
302
303      DIF (MAP) .LSK0
304 SK0      CALL BEGINMAP (MAP)
305
306
307      DO GRIDLOOP JOB = 1,K1
308      DO GRIDLOOP JOB = 1,K2
309      DO GRIDLOOP KOB = 1,K3
310
311
312      XO = X(JOB)
313      YO = Y(JOB)
314      ZO = Z(KOB)
315
316      DIF (DEBUG) .LSK1
317      IBUG = 0
318
319      IF (JOB.GT.4 .OR. JOB.GT.4 .OR. KOB.GT.4) GO TO SK1
320      WOT 3, (1H0, CPU,SYS,I/O UNWNTD REAL TIME FOR CASE(MILLISEC) )
321      WOT 3, (XO,YO,ZO= .3F10.2) ,XO,YO,ZO
322      IBUG = 1
323 SK1      CONTINUE
324
325      CALL FIELDS
326
327 GRIDLOOP CONTINUE
328
329
330      CALL GOB (2411B,IER,0300B,IBETA)
331      WOT 3, (1H0, CPU,SYS,I/O UNWNTD REAL TIME FOR CASE(MILLISEC) )
332      WOT 3, (3110) ,((IBETA(1)-IBOLD(1))/1000 I = 1,3)
333      IBOLD(1) = IBETA(1)
334      DIF (MAP) .LSK2
335 SK2      CALL ENDMAP(3)
336

```

```

337         DIF (IMFLG),SK4
338         CALL MEDIA (2)
339 SK4      CONTINUE
340
341 C..WRITE BINARY OUTPUT FILE IOUTFIL
342         IF (IOUTFIL.EQ.0) GO TO SK41
343
344         ITITLE(8) = IMAX
345         ITITLE(9) = JMAX
346         ITITLE(10) = KMAX
347
348         XTITLE(11) = DELX
349         XTITLE(12) = DELY
350         XTITLE(13) = DELZ
351         XTITLE(14) = BMODMIN
352
353
354         BUFFER OUT (5,1) (ITITLE(1),ITITLE(16),0
355 S1        IF (UNIT,5) S1,,ER1
356
357         BUFFER OUT (5,1) (AXA(1),BZA(IMAX+JMAX+KMAX),16)
358 S2        IF (UNIT,5) S2,,ER1
359
360
361
362
363 ER1      CONTINUE
364 SK41     CONTINUE
365
366
367         IF (IBFLD),SK5
368         CALL ABCPLOT(0,BMODA)
369         CALL EMPTY(3)
370
371 SK5      CONTINUE
372
373         IF (AMAXPLT.NE.AMINPLT),SK6
374
375         DO LPA I = 1,K1
376         DO LPA J = 1,KJ
377         DO LPA K = 1,KK
378         BMODA(I,J,K) = SQRTI(AXA(I,J,K)**2+AYA(I,J,K)**2+AZA(I,J,K)**2)

```

```

379 LP4 CONTINUE
380
381 CALL ABCPLOT(1,BMODA)
382
383 SK6 CONTINUE
384
385 IF (AXMAXPLT.NE.AXMINPLT),SK7
386 CALL ABCPLOT(2,AXA)
387 SK7 CONTINUE
388
389 IF (AYMINPLT.NE.AYMAXPLT),SK8
390 CALL ABCPLOT(3,AYA)
391 SK8 CONTINUE
392
393 IF (AZMINPLT.NE.AZMAXPLT),SK9
394 CALL ABCPLOT(4,AZA)
395 SK9 CONTINUE
396
397 CALL MOREDATA
398
399 GO TO AGAIN
400
401 END

```

```

402
403
404      SUBROUTINE INITIAL
405
406          CODE ANALYSIS
407      COMMON/GOBCOM/ INFO(50,
408
409      DIMENSION INCARD(8)
410      EQUIVALENCE (INFO(36),IBOX)
411
412      MACRO STORAGE (MACROS)$      USE STORAGE
413 C  GET CONTROLLEE LOD DATE AND RUN DATE
414
415      WOT JOUT, ( CONTROLLEE DATE: ,2A10) ,INFO(31),INFO(32)
416
417      INF1 = INFO(31)
418      INF2 = INFO(32)
419      CALL TIME
420      WOT JOUT, ( RUN DATE: ,2A10) ,INFO(31),INFO(32)
421
422 C  CREATE DROPFILE +ABCXYZ
423      IBETA(1) = 7R+ABCXYZ
424      IBETA(2) = LOC17 + 400B
425
426      CALL GOB(010)B,IER,0,IBETA)
427      IF(IER) CALL ERRMSG(1,IBETA,1)
428
429 C  EXPAND CONTROLLEE FOR LCM BY
430
431      IBETA(1) = LOC17
432      IBETA(2) = LOC16
433      CALL GOB (1300B,IER,0,IBETA)
434      ITEMP = 7R+ABCXYZ
435      IF(IER) CALL ERRMSG (2,ITEMP,1)
436
437
438
439 C  GET BOX NO. FOR HSP OUTPUT
440      WOT 3, ( PROGRAM ABCXYZ OUTPUT )
441      WOT 3, ( CONTROLLEE DATE: ,2A10) ,INF1,INF2
442      WOT 3, ( RUN DATE: ,2A10) ,INFO(31),INFO(32)
443

```

```

444 C ASSIGN TAPE 4 TO INPUT FILE NAME
445 40 CONTINUE
446 WOT 3, (// INFILE NAME: ,A10) ,INFIL
447
448 INFILEFT = INFIL
449
450 C RIGHT ADJUST FILENAME INFIL
451 IF (INFIL.EQ.0) CALL ERRMSG(3,INFIL,1)
452 42 IF((INFIL .INT. 778) 43,
453 INFIL = INFIL .SHL. 54
454 GO TO 42
455 43 CONTINUE
456
457 C OPEN FILE NAME INFIL
458 IBETA(1) = INFIL
459 IBETA(2) = 0
460 CALL GOB(0301B,IER,0,IBETA)
461 IF(IER) CALL ERRMSG(3,INFIL,1)
462
463 C WRITE OUT CARD INPUT LISTING TO MSP
464 CALL ASSIGN (4,INFIL)
465 WOT 3, (// INFILE CARD LISTING )
466 WOT 3, ( (5) (1) (5) (2) (5) (3) (5) (4) (5) )
467
468 45 CONTINUE
469 RIT 4, ( BA10) ,(INCARD(1),1=1,B)
470 WOT 3, (IH ,BA10) ,(INCARD(1),1=1,B)
471
472 DO 47 J = 1,B
473 DC 47 I = 1,10
474 INCARD(J) = INCARD(J) .SHL. 6
475 IF (INCARD(J) .INT. 778) ,47
476 IF ((INCARD(J) .INT. 778) .NE. IRE) GO TO 45
477 47 CONTINUE
478
479 C BACKSPACE TO BEGINNING TO PROCESS DATA
480 CALL REWIND(4)
481
482
483 LOC2 = 0400000000B + .LOC. ERDUMP
484 LOC2 = (LOC2 .SHL. 30) + (.LOC. IXPB)
485 LOC3 = .LOC. ERDUMP

```



```

486          GO TO 9
487 ERDUMP    CONTINUE
488
489 C PROGRAM COUNTER WHERE HARDWARE ERROR OCCURED AND PSD COUNTER
490     IPRG = (IXPK(1).SHR.36) .INT. 777777B
491     IPSD = (IXPK(4).SHR.36) .INT. 777777B
492     WOT JOUT ( HARDWARE ERROR...+ABCCYZ )
493     WOT JOUT. ( P-REG= .06,4X, PSD-REG= .06) .IPRG,IPSD
494
495     CALL MEDIA (1)
496     CALL EMPTY(3)
497     CALL GOB(0400B,IER,0,0)
498
499 9          CONTINUE
500
501
502 C..SET UP DOB0 INITIALIZATION
503     CALL CRTID (3HABC,1)
504     CALL KEEP80 (4RABCP)
505
506 C..READ IN LEAD DATA CARD WHICH DENOTES TYPE OF INPUT; MAFCO OR ABCXYZ
507     ENTRY MOREDATA
508
509     IOUTFIL = 0
510     NDOPT = 0
511     BMODMIN = 1.E+10
512
513     NAMELIST/CPLTCOM/NGR, IXGR, IYGR, IZGR,IBOX,NDOPT,IOUTFIL,IGR,
514     .      BMAXPLT, BMINPLT, AMINPLT, AMAXPLT, AXMINPLT,
515     .      AXMAXPLT, AYMINPLT, AYMAXPLT, AZMINPLT, AZMAXPLT
516     .      IAFLD,IBFLD,DIA,KI,KJ,KK,DELX,DELY,DELZ,XBOT,YBOT,ZBOT,
517     .      HFIELD,XBAR,YBAR,ZBAR,DELMAX,STOTAL,NSKIP,NLOOP,NARC,
518     .      NLINE,NGCL
519
520
521     INPUT DATA CPLTCOM,N1,3
522
523     IF (NDOPT.EQ.0) CALL EXIT
524     DO 65 I = 1,10
525     NDOPT = NDOPT.SHL.6
526     IF (NDOPT .INT. 77B) 66.
527 65 CONTINUE

```

```

528 66  CONTINUE
529 C  CREATE TAPE 5 TO OUTPUT FILE NAME
530
531      IF (IOUTFIL .EQ. 0) GO TO 67
532
533
534 C  ASSIGN TAPE 5 TO IOUTFIL
535      WOT 3, (  OUTFILE NAME:  ,A10) ,IOUTFIL
536
537
538      NLEN = (((IMAX*JMAX*KMAX)*6+20)/512)+1)*512
539
540
541      CALL DEVICE (6HCREATE,IOUTFIL,NLEN,IER)
542      IF (IER .NE. 0) CALL ERRMSG(4,IOUTFIL,I
543
544      CALL ASSIGN (5,IOUTFIL)
545      ITITLE(1) =  PROGRAM ABCXYZ OUTPUT
546      ITITLE(3) =  INF1
547      ITITLE(4) =  INF2
548      ITITLE(5) =  INFO(31)
549      ITITLE(6) =  INFO(32)
550      ITITLE(7) =  INFIL
551
552
553 C  LATER TRUNCATE FILE WHEN CLOSING IT
554
555 67  CONTINUE
556      IF ((NDOPT .INT. 77B).EQ.1RA) CALL DATAABCXYZ
557      IF ((NDOPT .INT. 77B).EQ.1RM) CALL DATAMAFCO
558
559
560 C...CHECK DIMENSION ALLOCATION TO WARN USE
561      IF (K1.GT.IMAX) WOT JOUT,FOR1,IMAX
562      IF (KJ.GT.JMAX) WOT JOUT,FOR2,JMAX
563      IF (KK.GT.KMAX) WOT JOUT,FOR3,KMAX
564
565 FOR1  FORMAT ( MAX GRID PTS IN I EXCEEDED.JMAX= ,I3)
566 FOR2  FORMAT ( MAX GRID PTS IN J EXCEEDED.JMAX= ,I3)
567 FOR3  FORMAT ( MAX GRID PTS IN K EXCEEDED.KMAX= ,I3)
568
569

```

```

570 C ZERO OUT LCM
571 DO 100 I = 1,IMAX*JMAX*KMAX*7
572 AXA(I) = 0
573 100 CONTINUE
574
575 C..INITIALILZE VALUES
576 RWIRE = .5*DIA
577 RWIRESQ = RWIRE**2
578 RRWIRESQ = 1./RWIRESQ
579
580
581 C..SET UP RECTANGULAR GRID
582 DO IGRID I = 1,KI
583 X(I) = (I-1)*DELX + XBOT
584 IGRID CONTINUE
585
586 DO JGRID J = 1,KJ
587 Y(J) = (J-1)*DELY + YBOT
588 JGRID CONTINUE
589
590 DO KGRID K = 1,KK
591 Z(K) = (K-1)*DELZ + ZBOT
592 KGRID CONTINUE
593
594
595 IIDIM = IMAX
596 JJDIM = JMAX
597 IIMIN = 1
598 IIMAX = KI
599 JJMIN = 1
600 JJMAX = KJ
601
602 BLOCKCOPY (ITYELC,ITY,IDATA)
603 BLOCKCOPY (ALPHALC,ALPHA,IDATA)
604
605 DO TRIGI I = 1,IDATA
606 IF (ITY(I).EQ.(LINE)) GO TO TRIGI
607 ALS(I) = SIN(ALPHA(I))*DEG2RAD)
608 TRIGI CONTINUE
609
610 BLOCKCOPY (ALS,ALSINLC,IDATA)
611

```

```

612      DO TRIG2 I = 1, IDATA
613      IF (ITY(I).EQ.1)LINE) GO TO TRIG2
614      ALCS(I)=COS(ALPHA(I)*DEG2RAD)
615 TRIG2 CONTINUE
616
617      BLOCKCOPY (ALCS,ALCOSLC,1DATA)
618      BLOCKCOPY (BETALC,BETA,1DATA)
619
620      DO TRIG3 I = 1, IDATA
621      IF (ITY(I).EQ.1)LINE) GO TO TRIG3
622      BTS(I)=SIN(BETA(I)*DEG2RAD)
623 TRIG3 CONTINUE
624
625      BLOCKCOPY (BTS,BETASINLC,1DATA)
626
627      DO TRIG4 I = 1, IDATA
628      IF (ITY(I).EQ.1)LINE) GO TO TRIG4
629      BTC(I) = COS(BETA(I)*DEG2RAD)
630 TRIG4 CONTINUE
631
632      BLOCKCOPY (BTC,BETACOSLC,1DATA)
633      BLOCKCOPY (PHI1LC,PPH11,1DATA)
634
635      DO TRIG5 I = 1, IDATA
636      IF (ITY(I).EQ.1)LINE) GO TO TRIG5
637      IF (ITY(I).EQ.1)LOOP) PPH11(I)=0.0
638      IF (ITY(I).EQ.1)ARC) PPH11(I)=PPH11(I)*DEG2RAD
639 TRIG5 CONTINUE
640
641      BLOCKCOPY (PPH11,PHI1LC,1DATA)
642      BLOCKCOPY (PHI2LC,PPH12,1DATA)
643
644      DO TRIG6 I = 1, IDATA
645      IF (ITY(I).EQ.1)LINE) GO TO TRIG6
646      IF (ITY(I).EQ.1)LOOP) PPH12(I)=PIX2
647      IF (ITY(I).EQ.1)ARC) PPH12(I)=PPH12(I)*DEG2RAD
648 TRIG6 CONTINUE
649
650      BLOCKCOPY (PPH12,PHI2LC,1DATA)
651
652
653      RETURN

```

654 END

```

655
656
657
658      SUBROUTINE DATAMAFCO
659
660          MACRO STORAGE (MACROS)
661          USE STORAGE
662
663          DIMENSION TITLE(10)
664
665
666 C...DATA CARD STRUCTURE
667 C...FIRST CARD: DIA, KI, KJ, KK, DELX, DELY, DELZ, XBOT, YBOT, ZBOT
668 C      THIS NEXT SECTION OF CARDS IS REPEATED FOR EACH PROBLEM
669 C
670 C * * * * *
671 C
672 C      FIRST CARD OF THIS SECTION CONTAINS IN COLS.
673 C
674 C          1- 5  NUMBER OF LOOPS
675 C          6-10  NUMBER OF ARCS
676 C          11-15  NUMBER OF HELICES
677 C          16-20  NUMBER OF STRAIGHT LINES
678 C          21-25  NUMBER OF GENERAL CURRENT ELEMENT GROUPS
679 C          26-30  NUMBER OF GRIDS
680 C          31-35  NUMBER OF CALCULATIONS OF FIELD LINES
681 C          36-72  COMMENTS CONCERNING THIS PROBLEM
682 C
683 C...UNITS: LENGTH IN CENTIMETER; ANGLES IN DGREE; CURRENT IN AMPERES
684 C
685 C      CIRCULAR LOOP CARDS- ONE FOR EACH LOOP IN THIS PROBLEM
686 C      CONTAINING
687 C          X,Y,Z COORDINATES,RADIUS, ALPHA, BETA, CURRENT  -- FORMAT (7F10.1)
688 C
689 C      ARC CARDS - TWO FOR EACH ARC IN THIS PROBLEM CONTAINING ON CARD -
690 C          ONE - X,Y,Z COORDINATES, RADIUS, ALPHA, BETA, CURRENT --FORMAT (7F
691 C          TWO - PH11, PH12
692 C
693 C      HELIX CARDS- ONE FOR EACH HELIX IN THIS PROBLEM
694 C      CONTAINING
695 C          RADIUS, Z-DISTANCE OF HALF-TURN, STARTING ANGLE, ENDING ANGLE,CUR
696 C          DISTANCE FROM XY PLANE -- FORMAT (6F10.1)

```

```

697 C
698 C   STRAIGHT LINE CARDS- ONE FOR EACH STRAIGHT LINE IN THIS PROBLEM
699 C   CONTAINING
700 C     X,Y,Z OF THE FIRST END, X,Y,Z OF THE OTHER END, CURRENT ALONG THE
701 C     -- FORMAT( 7F10.1 )
702 C
703 C   GENERAL CURRENT ELEMENT CARDS- ONE GROUP FOR EACH LINE IN THIS PROB
704 C   CONTAINING ON THE FIRST CARD
705 C     X,Y,Z OF THE FIRST ELEMENT, CURRENT ALONG POINTS, NUMBER OF POINT
706 C     -- FORMAT( 4F10.1, 15 )
707 C   ON FOLLOWING CARDS
708 C     X,Y,Z OF SUCCEEDING ELEMENTS
709 C
710 C .....
711 C
712 C..READ FIRST DATA CARD
713       WOT 3, (1H1, INPUT FOR THIS CASE )
714       WOT 3, (1H0, DIA,K1,KJ,KK,DELX,DELY,DELZ,XBOT,YBOT,ZBOT )
715
716       WOT 3,F1,DIA,K1,KJ,KK,DELX,DELY,DELZ,XBOT,YBOT,ZBOT
717 C
718 C   READ IN THE NUMBER OF LOOPS, HELICES, STRAIGHT LINES, GEN. CURRENT E
719 C   POINTS FOR FIELDS RELATIVE TO Z-AXIS, FIELD LINES AND COMMENTS
720
721       WOT 3, (1H0, NLOOP,NARC,NHELIX,NLINE,NGCL,MPNTS,MFIELD,TITLE )
722
723
724       WOT 3,2,NLOOP,NARC,NHELIX,NLINE,NGCL,
725           MPNTS,MFIELD,(TITLE(1),1=1,6)
726 C
727 C
728       KARCS = 0
729       KHEL = 0
730       KLINE = 0
731
732 C..READ IN VARIABLES FOR LOOPS
733       IF (NLOOP) ,49
734
735       WOT 3, (1H0, XS,YS,ZS,A,ALPHA,BETA,CUR )
736       RIT N1,25,(XSLC(1),YSLC(1),ZSLC(1),ALC(1),ALPHALC(1),BETALC(1),CURLC(1),
737           , 1=1,NLOOP)
738       WOT 3,25,(XSLC(1),YSLC(1),ZSLC(1),ALC(1),ALPHALC(1),BETALC(1),

```

```

739          CURLC(I),I=1,NLOOP)
740
741      IDATA = NLOOP
742
743
744
745 C..READ IN VARIABLES FOR ARCS
746 49  CONTINUE
747      LARC = NLOOP + 1
748      KARCS = NLOOP + NARC
749
750      IF (NARC) ,51,
751
752      WOT 3, (IH0, XS,YS,ZS,A,ALPHA,BETA,CUR,PHI1,PHI2 )
753
754      DO 50 I = LARC, KARCS
755          RIT N1,25,XSLC(I),YSLC(I),ZSLC(I),ALC(I),ALPHALC(I),BETALC(I),CURLC(I)
756          RIT N1,6,PHI1LC(I),PHI2LC(I)
757          WOT 3,3,XSLC(I),YSLC(I),ZSLC(I),ALC(I),ALPHALC(I),BETALC(I),CURLC(I),
758              PHI1LC(I),PHI2LC(I)
759 50  CONTINUE
760
761      IDATA = KARCS
762
763 C..READ IN VARIABLE FOR HELIXES
764 51  CONTINUE
765
766      LHEL = KARCS + 1
767      KHEL = KARCS + NHELIX
768
769      IF (NHELIX) ,59,
770
771      WOT 3, (IH0, XS,YS,ZS,A,CUR,Z2 )
772      RIT N1,4,(XSLC(I),YSLC(I),ZSLC(I),ALC(I),CURLC(I),Z2LC(I),I=LHEL,KHEL)
773      WOT 3,4,(XSLC(I),YSLC(I),ZSLC(I),ALC(I),CURLC(I),Z2LC(I),I=LHEL,KHEL)
774
775      IDATA = KHEL
776
777 C..READ IN VARIABLES FOR STRAIGHT LINE
778 59  CONTINUE
779      LLINE = KHEL + 1
780      KLINE = KHEL + NLINE

```

```

781      IF (NLINE) .69,
782      WOT 3, (1H0, XS,YX,ZX,X2,YX,Z2,CUR )
783
784
785      DO 65 I = LLINE,KLINE
786      RIT NI,25,XSLC(I),YSLC(I),ZSLC(I),X2LC(I),Y2LC(I),Z2LC(I),CURLC(I)
787      WOT 3,25,XSLC(I),YSLC(I),ZSLC(I),X2LC(I),Y2LC(I),Z2LC(I),CURLC(I)
788 65    CONTINUE
789
790      I DATA = KLINE
791
792 C..READ IN VARIABLE FOR GENERAL CURRENT ELEMENTS
793 69    IF (NGCL) .81,
794
795      I = KLINE
796
797      WOT 3, (1H0, XS,YS,ZS,X2,Y2,Z2,CUR )
798      DO 80 I1 = 1,NGCL
799      I = I+1
800      JJ = I
801      RIT NI,24,XSLC(I),YSLC(I),ZSLC(I),CURLC(JJ),NPT
802
803      DO 75 KM = 2,NPT
804      I = I+1
805      RIT NI,5,XSLC(I),YSLC(I),ZSLC(I)
806
807      X2LC(I-1) = XSLC(I)
808      Y2LC(I-1) = YSLC(I)
809      Z2LC(I-1) = ZSLC(I)
810      CURLC(I-1) = CURLC(JJ)
811      ITYPELC(I-1) = 1
812
813 75    CONTINUE
814      I = I-1
815      WOT 3,25,(XSLC(KM),YSLC(KM),ZSLC(KM),X2LC(KM),Y2LC(KM),Z2LC(KM),CURLC(KM),
816      , KM = JJ,JJ+NPT-1)
817
818 80    CONTINUE
819      I DATA = I
820
821 81    CONTINUE
822

```



```

823      IF (NLOOP) ,90,
824
825      DO IDLOOP I = 1,NLOOP
826          IYPELC(I) = 2
827      IDLOOP CONTINUE
828
829      90 CONTINUE
830      IF (KARCS) ,92,
831
832      DO IDARCS I = LARC,KARCS
833          IYPELC(I) = 3
834      IDARCS CONTINUE
835
836      92 CONTINUE
837      IF (KHEL) ,94,
838
839      DO IDHEL I = LHEL,KHEL
840          IYPELC(I) = 4
841      IDHEL CONTINUE
842
843
844      94 CONTINUE
845      IF (KLINE) ,96,
846
847      DO IDLINES I = LLINE,KLINE
848          IYPELC(I) = 1
849      IDLINES CONTINUE
850
851      96 CONTINUE
852
853      2  FORMAT (7I5, 6A6)
854      3  FORMAT (9F10.3)
855      4  FORMAT (6F10.3)
856      5  FORMAT (3F10.3)
857      6  FORMAT (2F10.3)
858      24 FORMAT (4F10.3,15)
859      25 FORMAT (7F10.3)
860      F1  FORMAT (F8.3,3I5,6F8.3)
861      RETURN
862
863      END

```

```

864      SUBROUTINE DATAABCKYZ
865          MACRO STORAGE (MACROS)
866          USE STORAGE
867      C..DATA CARD STRUCTURE
868      C..FIRST CARD: DIA, KI, KJ, KK, DELX, DELY, DELZ, XBOT, YBOT, ZBOT
869      C   DIA: DIAMETER OF WIRES TO PREVENT UNDEFINED FIELDS (CM)
870      C   KI,KJ,KK: NO. OF GRID POINTS IN EACH DIRECTION
871      C   DELX,DELY,DELZ: SPACING OF GRID (CM)
872      C   XBOT,YBOT,ZBOT: COORDINATES OF BOTTOM OF WIRE (CM)
873      C   MFIELD: NO. FIELD LINES TO CALC. AND PLOT
874
875      C..REMAINING CARDS:
876      C..TYPE, CUR, XS, YS, ZS, D1, D2, D3,D4,D5
877
878      C..TYPE: (1,2,3) FOR LINES, LOOPS, ARCS
879      C..CUR: WIRE CURRENT (AMPERES)
880      C   TYPE GT. 0 IF MORE CARDS TO FOLLOW; LT.0 IF THIS IS LAST
881      C..XS, YS,ZS: POSITION OF CURRENT ELEMENT IN (CM)
882
883      C..D1, D2, D3, D4, D5, : ORIENTATION AND SIZE OF CURRENT ELEMENT
884      C   X2, Y2, Z2, C, D FOR LINES (END POSITION OF LINE)
885      C   A, ALPHA, BETA, - , - FOR LOOPS
886      C   A, ALPHA, BETA, PH11, PH12 FOR ARCS
887
888      C..A: RADIUS OF LOOP OR ARC (CM)
889      C..ALPHA: EULERIAN ANGLE (DEGREES)
890      C..BETA: EULERIAN ANGLE (DEGREES)
891      C..PH11: ANGLE AT WHICH ARC BEGINS (DEGREES)
892      C..PH12: ANGLE AT WHICH ARC ENDS (DEGREES)
893
894      C*****
895
896
897          DIMENSION TYPELC(INSIZE)
898          EQUIVALENCE (TYPELC,TYPELC)
899      C..READ FIRST DATA CARD
900          WOT 3, (IH1, INPUT FOR THIS CASE )
901          WOT 3, (IHC, DIA,KI,KJ,KK,DELX,DELY,DELZ,XBOT,YBOT,ZBOT,MFIELD )
902
903          WOT 3,F2,DIA,KI,KJ,KK,DELX,DELY,DELZ,XBOT,YBOT,ZBOT,MFIELD
904
905          ICASE = 0

```

```

906
907      WOT 3, (1H0, TYPE,CUR,X5,Y5,Z5,D1,D2,D3,D4,D5 )
908 MOREDATA CONTINUE
909      ICASF = ICASE + 1
910
911      RIT NI,F1,TYPELC(ICASE),CURLC(ICASE),XSLC(ICASE),YSLC(ICASE),
912      . ZSLC(ICASE), D1LC(ICASE),D2LC(ICASE),D3LC(ICASE),D4LC(ICASE),
913      . D5LC(ICASE)
914      WOT 3,F1,TYPELC(ICASE),CURLC(ICASE),XSLC(ICASE),YSLC(ICASE),
915      . ZSLC(ICASE), D1LC(ICASE),D2LC(ICASE),D3LC(ICASE),D4LC(ICASE),
916      . D5LC(ICASE)
917      ITYPELC(ICASE) = TYPELC(ICASE)
918
919
920      IF (ITYPELC(ICASE).GT.0) GO TO MOREDATA
921
922      ITYPELC(ICASE) = -ITYPELC(ICASE)
923      IDATA = ICASE
924 START CONTINUE
925
926      RETURN
927
928 F1      FORMAT (10F8.3)
929 F2      FORMAT (F7.3,3I3,6F7.3,13)
930
931      END

```

```

932
933 SUBROUTINE FIELDS
934
935 MACRO STORAGE ( MACROS)
936 USE STORAGE
937
938 CODE ANALYSIS
939
940 C*****
941 IF (IAFLD),SK1
942 SAX = 0.0
943 SAY = 0.0
944 SAZ = 0.0
945 SK1 CONTINUE
946
947 IF (IBFLD),SK2
948 SBX = 0.0
949 SBY = 0.0
950 SBZ = 0.0
951 SK2 CONTINUE
952
953 C..ONE LOOP OVER ALL CURRENT ELEMENTS REGARDLESS OF TYPE
954
955 IPART = 999
956
957 DO AIELMS IBIG = 1, IDATA, 999
958
959 IF ( IDATA-IBIG).GE.999) GO TO IB10
960
961 IPART = IDATA-IBIG+1
962
963 IB10 CONTINUE
964
965 READ LCM (ITYPE ,ITYPELC(IBIG),IPART)
966 READ LCM (CUR ,CURLC(IBIG),IPART)
967 READ LCM (XS ,XSLC(IBIG),IPART)
968 READ LCM (YS ,YSLC(IBIG),IPART)
969 READ LCM (ZS ,ZSLC(IBIG),IPART)
970 READ LCM (D1 ,D1LC(IBIG),IPART)
971 READ LCM (D2 ,D2LC(IBIG),IPART)
972 READ LCM (D3 ,D3LC(IBIG),IPART)
973 READ LCM (ALSIN ,ALSINLC(IBIG),IPART)

```

```

974      READ LCM (ALCOS ,ALCOSLC(I BIG),I PART)
975      READ LCM (BETASIN ,BETASINLC(I BIG),I PART)
976      READ LCM (BETACOS ,BETACOSLC(I BIG),I PART)
977
978      DO PARTEL ICASE = 1,I PART
979
980          IC = ICASE
981
982          ITYPESC = ITYPE(ICASE)
983          XSC = XS(ICASE)
984          YSC = YS(ICASE)
985          ZSC = ZS(ICASE)
986          XMU1 = CUR(ICASE)*FMUFAC4
987          CURSC = CUR(ICASE)
988          XMU1X2 = XMU1*2
989
990
991          IF (ITYPESC-1) 2,
992
993      1      CONTINUE
994          X2SC = X2(ICASE)
995          Y2SC = Y2(ICASE)
996          Z2SC = Z2(ICASE)
997
998          CALL LINES
999          GO TO RESUME
1000
1001      2      CONTINUE
1002          ASC = A(ICASE)
1003          AXMU1X2 = XMU1X2*A(ICASE)
1004
1005          SA = ALSIN(ICASE)
1006          CA = ALCOS(ICASE)
1007          SB = BETASIN(ICASE)
1008          CB = BETACOS(ICASE)
1009
1010      3      CALL LOOPARC
1011
1012
1013      RESUME CONTINUE
1014
1015          IF (IAFLD),SK3

```

```

1016      SAX = SAX + AX
1017      SAY = SAY + AY
1018      SAZ = SAZ + AZ
1019 SK3    CONTINUE
1020
1021      DIF (IBFLD) ,SK4
1022      SBX = SBX + BX
1023      SBY = SBY + BY
1024      SBZ = SBZ + BZ
1025 SK4    CONTINUE
1026
1027      DIF (IDEBUG) ,STB
1028      IF (IBUG) ,STB
1029      WOT 3, ( XSC,YSC,ZSC= ,3F10.3) ,XSC,YSC,ZSC
1030      WOT 3, ( XMU1,ASC,ALPHASC,BETASC= ,4E12.3) ,XMU1,ASC,ALPHA(CASE),
1031      , BETA(CASE)
1032      WOT 3, ( PH11(CASE),PH12(CASE)= ,2E12.3) ,PH11(CASE),PH12(CASE)
1033      WOT 3, ( AX,AY,AZ= ,3E12.3) ,AX,AY,AZ
1034      WOT 3, ( BX,BY,BZ= ,3E12.3) ,BX,BY,BZ
1035 STB    CONTINUE
1036
1037      PARTEL CONTINUE
1038
1039      ALELMS CONTINUE
1040
1041 C..IF FLDLN ROUTINE ACTIVE LET IT SAVE SBX,SBY,SBZ
1042      IF (MFACTIVE) RETURN
1043
1044      IF (IAFLD) ,SK9
1045      AXA(10B,JOB,KOB) = SAX
1046      AYA(10B,JOB,KOB) = SAY
1047      AZA(10B,JOB,KOB) = SAZ
1048 SK9    CONTINUE
1049
1050      DIF (IBFLD) ,SK10
1051      BXA(10B,JOB,KOB) = SBX
1052      BYA(10B,JOB,KOB) = SBY
1053      BZA(10B,JOB,KOB) = SBZ
1054      BMODA(10B,JOB,KOB) = SQRT1(SBX**2+SBY**2+SBZ**2)
1055      BMODMIN = AMINI(BMODMIN,BMODA(10B,JOB,KOB))
1056 SK10   CONTINUE
1057
1058      RETURN
1059
1060      END

```

```

1061
1062
1063      SUBROUTINE JLIPTK
1064 C
1065      MACRO ELIPTK (MACROS)
1066 C
1067 C..INPUT VIA COMMON
1068 C      OMEGA1 (RADIAN)
1069 C      OMEGA2 (RADIAN)
1070 C      FIRSTK VIA ROUTINE IEI
1071 C      SCNDK VIA ROUTINE IEI
1072
1073 C..OUTPUT VIA COMMON
1074 C      FINCMP 1ST INCOMPLETE ELLIPTIC INTEGRAL (DIMENSIONLESS)
1075 C      EINCMP 2ND      DITTO
1076
1077 C..ROUTINE USED: IEI, CLICHE ELIPTK
1078
1079      OPTIMIZE
1080
1081      PARAMETER (PI = 3.1415927)
1082      PARAMETER (PI2 = PI/2.)
1083      PARAMETER (RECP1 = 1./PI)
1084
1085      COMMON/ILIP/OMEGA1,OMEGA2,CAY,CAYSQ,FINCMP,EINCMP,FIRSTK,SCNDK
1086      CODE ANALYSIS
1087
1088
1089 C*****
1090 C
1091      OMEG = OMEGA1
1092      IL = 1
1093      100 SIGN = 1.0
1094      IF ( OMEGA1 ) 150., .175
1095 C
1096      FINCMP = 0.0
1097      EINCMP = 0.0
1098      GO TO 600
1099 C
1100      150 SIGN = -SIGN
1101      OMEGA1 = -OMEGA1
1102 C

```

```

1103 175 IF ( P12 - OMEGA1 ) , 400, 500
1104 C OMEGA1 GR THAN P1/2
1105 IF ( OMEGA1 - P12 ) 400, 400,
1106 N = OMEGA1 * RECP1
1107 AN = N
1108 IF ( N ) , 250,
1109 C CALCULATE OMEGA
1110 OMEGA1 = OMEGA1 - AN * P1
1111 250 CONTINUE
1112 USE ELIPTK
1113 AN = AN + AN
1114 IF ( OMEGA1 - 0.01 ) , , 300
1115 C OMEGA APPROX P1
1116 FINCMP = AN * FIRSTK
1117 EINCMP = AN * SCNDK
1118 GO TO 600
1119 C
1120 300 IF ( OMEGA1 - P12 ) , 375, 350
1121 C OMEGA LESS THAN P1/2
1122 CALL IE1
1123 FINCMP = FINCMP + AN * FIRSTK
1124 EINCMP = EINCMP + AN * SCNDK
1125 GO TO 600
1126 C OMEGA GR THAN P1 /2
1127 350 IF ( OMEGA1 - P12 ) 375, 375,
1128 OMEGA1 = P1 - OMEGA1
1129 AN = AN + 2.0
1130 CALL IE1
1131 FINCMP = AN * FIRSTK - FINCMP
1132 EINCMP = AN * SCNDK - EINCMP
1133 GO TO 600
1134 C OMEGA = P1/2
1135 375 AN = AN + 1.0
1136 FINCMP = AN * FIRSTK
1137 EINCMP = AN * SCNDK
1138 GO TO 600
1139 C OMEGA1 = P1/2
1140 400 CONTINUE
1141 USE ELIPTK
1142 FINCMP = FIRSTK
1143 EINCMP = SCNDK
1144 GO TO 600

```



```

1145 C                                OMEGA1 LESS THAN  $\pi/2$ 
1146   500 CALL IEI
1147 C
1148   600 IF ( IL ) , 700,
1149         IL = IL - 1
1150         F = SIGN * F1NCMP
1151         E = SIGN * E1NCMP
1152         OMEGA1 = OMEGA2
1153         GO TO 100
1154 C.....SIGN CHANGE ON F1NCMP FROM ORIGINAL ....BY JOB 12/12/74
1155   700 F1NCMP = SIGN * F1NCMP - F
1156         E1NCMP = SIGN * E1NCMP - E
1157         OMEGA1 = OMEG
1158         RETURN
1159         END

```

```

1160      SUBROUTINE IEI
1161 C
1162 C
1163 C
1164 C..INPUT VIA COMMON
1165 C      OMEGA1 (RADIANS)
1166 C      OMEGA2 (RADIANS)
1167 C      CAY
1168 C      CAYSQ
1169
1170 C..OUTPUT VIA COMMON
1171 C      FINCMP 1ST INCOMPLETE ELLIPTIC INTEGRAL (DIMENSIONLESS)
1172 C      EINCMP 2ND DITTO
1173
1174 C
1175 C      CALCULATES INCOMPLETE ELLIPTIC INTEGRALS OF THE
1176 C      FIRST AND SECOND KINDS
1177
1178
1179      OPTIMIZE
1180
1181      CODE ANALYSIS
1182
1183      PARAMETER (PI=3.1415927)
1184      COMMON/ILIP/ OMEGA1,OMEGA2,CAY,CAYSQ,FINCMP,EINCMP,FIRSTK,
1185      SCNDK
1186
1187 C.....
1188 C
1189      DIMENSION SS(4),BE(4)
1190      S1 = SIN( OMEGA1 )
1191      C0 = COS( OMEGA1 ) / S1
1192      A=1.
1193      IF ( 0.9539 - CAY ) 1, 1,
1194      B = SQRT( 1.0 - CAYSQ )
1195      S1=0.
1196      S2=0.
1197      A1=B*B
1198      N=0
1199      J = 0
1200      3 J = J + 1
1201      AA=A*B

```

```

1202      C0=.5*(C0-AA/C0)
1203      IF (C0) , .5
1204      N=N+N+1
1205      GO TO 6
1206      5 N=N+N
1207      6 P=A*A
1208      A=.5*(A+B)
1209      P=P-A
1210      FN=-N*(N/4)*4
1211      TEM1=1.5+FN
1212      TEM2=RR2I(A*A+C0*C0)
1213      IF (TEM1) 7,7,
1214      S2=S2+P*TEM2
1215      GO TO 9
1216      7 S2=S2-P*TEM2
1217      9 S1=.5*S1+P
1218      A1=AA
1219      IF ( 4 - J ) , 16,
1220      B = SQRTI( AA )
1221      GO TO 3
1222      16 CONTINUE
1223      TEM3 = ATANF ( A/C0 )
1224      TEM3 = TEM3 + TEM3
1225      TEM4 = ( 0.25 - S1 ) / ( A + A )
1226      IF (C0) , .11
1227      N=N+1
1228      11 FN=N
1229      FP = P1 * FN + TEM3
1230      FINCMP = 0.03125 * FP / A
1231      EINCMP = 0.25 * ( S2 + FP * TEM4 )
1232      RETURN
1233      1 S1 = 1.0 - CAYSQ
1234      CZ = C0
1235      BE(1) = CAY
1236      SS(1)=S1
1237      BE(4) = 1.0 + BE(1)
1238      C0 = ( C0 + SQRTI( C0 * C0 + 1.0 - CAYSQ ) ) / BE(4)
1239      S1=C0*C0
1240      DO 13 J=1,3
1241      SS(J+1)=RR2I(1.+S1)
1242      AA=A*BE(J)
1243      A=.5*BE(4)

```

```

1244      BE(J+1)=SQRT1(AA)
1245      P=A*A
1246      SI=.5*SI+P-AA
1247      BE(4)=A+BE(J+1)
1248      C0=(A*C0+SQRT1(P*(SI+1.)-AA))/BE(4)
1249 13 SI=C0*C0
1250      SI=SQRT1(1.+SI)
1251      P=1./SI
1252      U=0.
1253      BE(4)=A
1254
1255      U = U+U+BE(4)*(P-SS(4))
1256      U = U+U+BE(3)*(P-SS(3))
1257      U = U+U+BE(2)*(P-SS(2))
1258      U = U+U+BE(1)*(P-SS(1))
1259
1260      EINCMP = 4.0 * SI * LOGF(( 1.0 + SI ) / C0 , / A + P + U
1261      A = 1.0
1262      B = CAY
1263      A1 = CAYSQ
1264      SI = CZ
1265      DO 12 I = 1,2
1266      AA = A * B
1267      SI = A * SI + SQRT1( A * A * ( SI * SI + 1.0 ) - A1 )
1268      A = 0.5 * ( A + B )
1269      SI = SI / ( A + A )
1270      B = SQRT1( AA )
1271 12 A1 = AA
1272      SI = A * SI + SQRT1( A * A * ( SI * SI + 1.0 ) - AA )
1273      A = 0.5 * ( A + B )
1274      SI = SI / ( A + A )
1275      FINCMP = LOGF( ( 1.0 + SQRT1( 1.0 + SI * SI ) ) / SI ) / A
1276      RETURN
1277      END

```

```

1278      SUBROUTINE LINES
1279
1280      MACRO BACKROTATE, STORAGE (MACROS)
1281      USE STORAGE
1282
1283      CODE ANALYSIS
1284
1285 C*****
1286
1287 C*****
1288 C..X0,Y0,Z0  DENOTE OBSERVER IN LAB FRAME
1289 C..XS,YS,ZS  DENOTE CURRENT ELEMENT IN LAB
1290
1291 C..TRANSLATE TO PUT ORIGIN AT CURRENT ELEMENT
1292
1293      XT0 = X0 - XSC
1294      YTO = Y0 - YSC
1295      ZTO = Z0 - ZSC
1296
1297 C..ROTATE ABOUT Z THROUGH PRIMARY EULER ANGLE ALPHA
1298
1299      XT2 = X2SC - XSC
1300      YT2 = Y2SC - YSC
1301      ZT2 = Z2SC - ZSC
1302
1303 C..ROTATE IN ALPHA UNTIL XA2 IS ZERO
1304
1305      SA = 0.0
1306      CA = 1.0
1307      IF ((XT2.EQ.0.0).AND.(YT2.EQ.0.0)) GO TO 10
1308      ARITHREG RTFAC
1309      RTFAC = RRT21(XT2**2+YT2**2)
1310      CA = YT2*RTFAC
1311      SA = -XT2*RTFAC
1312      RELEASE RTFAC
1313
1314 10      CONTINUE
1315
1316
1317 C..XA2 = XT2*CA + YT2*SA = 0.0
1318      YA2 = -XT2*SA + YT2*CA
1319      ZA2 = ZT2

```

```

1320
1321
1322      XAO = XTO*CA + YTO*SA
1323      YAO = -XTO*SA + YTO*CA
1324      ZAO = ZTO
1325
1326 C..ROTATE ABOUT XAO THRU SECONDARY EULER ANGLE BETA
1327
1328 C ROTATE IN BETA UNTIL YB2 IS ZERO
1329      SB = 0.0
1330      CB = 1.0
1331      IF ((YA2.EQ.0.0).AND.(ZA2.EQ.0.0)) GO TO 20
1332
1333      ARITHREG RTFAC
1334      RTFAC = RRT2I(YA2**2+ZA2**2)
1335      CB = ZA2*RTFAC
1336      SB = -YA2*RTFAC
1337      RELEASE RTFAC
1338
1339 20      CONTINUE
1340
1341 C.. XB2 = XA2 = 0.0
1342 C.. YB2 = YA2*CB + ZA2*SB = 0.0
1343
1344      ZB2 = -YA2*SB + ZA2*CB
1345
1346      XBO = XAO
1347      YBO = YAO *CB + ZAO*SB
1348      ZBO = -YAO*SB + ZAO*CB
1349
1350 C..TERTIARY EULER ANGLE PSI DEFINED BY YPSI=0; XPSI.GT.0
1351 C..FIRST ASSUME XPSI .GT.0
1352 C..ATAN2 RESULTS IN -PI TO +PI RADIANS
1353
1354      SP = 0.0
1355      CP = 1.0
1356      IF ((YBO.EQ.0.0).AND.(XBO.EQ.0.0)) GO TO 30
1357
1358      ARITHREG RTFAC
1359      RTFAC = RRT2I(YBO**2+XBO**2)
1360      CP = XBO*RTFAC
1361      SP = YBO*RTFAC

```

```

1362          RELEASE RTFAC
1363
1364 30          CONTINUE
1365
1366 C..FINALLY GET OBSERVER IN LOCAL PREFERRED COORDINATES
1367
1368          XLOC = XB0*CP+YB0*SP
1369          YLOC = -XB0*SP + YB0*CP
1370          IF (ABSF(YLOC).GT.1.E-3) PAUSE
1371
1372          ZLOC = ZB0
1373
1374
1375 C..XL2 = XB2*CP + YB2*SP = 0.0
1376 C..YL2 = -XB2*SP + YB2*CB = 0.0
1377
1378          ZL2 = ZB2
1379
1380
1381
1382 C  CHECK TO SEE IF WE ARE INSIDE WIRE
1383          IF ((ZLOC.LE.-RWIRE).OR.(ZLOC.GE.(ZL2+RWIRE))) GO TO OUTSIDE
1384          IF (XLOC.GE.RWIRE) GO TO OUTSIDE
1385
1386 C  HERE WE ARE INSIDE THE WIRE AND NOT SINGULAR
1387          BTLW = XMUIX2*XLOC/(RWIRE**2)
1388          DIF (1BFLD) ,SK2
1389          BZLOC = 0.0
1390          BXLOC = 0.0
1391          BYLOC = BTLW
1392  SK2          CONTINUE
1393
1394          IF (1AFLD) ,SK4
1395          AXLOC = 0.0
1396          AYLOC = 0.0
1397          AZLOC = XLOC * BTLW
1398  SK4          CONTINUE
1399
1400          USE BACKROTATE
1401          RETURN
1402
1403 OUTSIDE CONTINUE

```

```

1404      RHO1 = SQRT((XLOC**2+ZLOC**2)
1405      RHO2 = SQRT((XLOC**2 + (ZLOC-ZL2)**2)
1406      DIF (IBFLD),SK5
1407      BZLOC = 0.0
1408      BXLOC = 0.0
1409
1410      RHODOT = XLOC **2 - ZLOC*(ZL2-ZLOC)
1411
1412      G = (RHO1 + RHO2)/(RHO1*RHO2*(RHO1*RHO2+RHODOT))
1413
1414      BYLOC = XMU1 * G *(ZL2*XLOC)
1415 SK5      CONTINUE
1416
1417 C      NOW GET THE VECTOR POTENTIAL FIELDS LOCALLY.
1418      IF (IAFLD),SK12
1419      AXLOC = 0.0
1420      AYLOC = 0.0
1421      IF ((ABSF(XLOC).LT.ABSF(ZLOC*1.E-3)).AND.
1422      ((ZL2-ZLOC)*ZLOC).LT.0.0)) GO TO LOCDONF
1423      AZLOC = XMU1*LOG1((RHO2+ZL2-ZLOC)/(RHO1-ZLOC))
1424
1425      USE BACKROTATE
1426
1427      RETURN
1428 LOCDONF CONTINUE
1429      AZLOC = XMU1*LOG1(ABSF(ZLOC/(ZLOC-ZL2)))
1430
1431 SK12      CONTINUE
1432      USE BACKROTATE
1433
1434      RETURN
1435      END

```



```

1436
1437
1438
1439
1440           SUBROUTINE LOOPARC
1441
1442
1443           MACRO BACKROTATE, ELIPTK, STORAGE ( MACROS)
1444           USE STORAGE
1445           CODE ANALYSIS
1446
1447 C*****
1448
1449 C..GET INTO PREFERRED LOCAL COORDINATE SYSTEM OF ARC ELEMENT.
1450
1451
1452 C..GET INTO PREFERRED LOCAL COORDINATE SYSTEM OF ARC ELEMENT
1453 C....FOR ARCS AND LOOPS ONLY.
1454 C*****
1455 C..XO,YO,ZO  DENOTE OBSERVER IN LAB FRAME
1456 C..XS,YS,ZS  DENOTE CURRENT ELEMENT IN LAB
1457
1458 C..TRANSLATE TO PUT ORIG.N AT CURRENT ELEMENT
1459
1460           XTO = XO - XSC
1461           YTO = YO - YSC
1462           ZTO = ZO - ZSC
1463
1464 C..ROTATE ABOUT Z THROUGH PRIMARY EULER ANGLE ALPHA
1465
1466           XAO = XTO*CA + YTO*SA
1467           YAO = -XTO*SA + YTO*CA
1468           ZAO = ZTO
1469
1470 C..ROTATE ABOUT XAO THRU SECONDARY EULER ANGLE BETA
1471
1472           XBO = XAO
1473           YBO = YAO *CB + ZAO*SB
1474           ZBO = -YAO*SB + ZAO*CB
1475
1476 C..TERTIARY EULER ANGLE PSI DEFINED BY YPSI=0; XPSI.GT.0
1477 C..FIRST ASSUME XPSI .GT.0

```

```

1478 C..ATAN2 RESULTS IN -PI TO +PI RADIANS
1479
1480         PSI = ATAN2(YBO,XBO)
1481
1482         IF (PSI.LT.0.0) PSI = PSI + P[X2
1483         CP = COS(PSI)
1484         SP = SIN(PSI)
1485
1486 C..FINALLY GET OBSERVER IN LOCAL PREFERRED COORDINATES
1487
1488         XLOC = XBO*CP+YBO*SP
1489         YLOC = -XBO*SF + YBO*CP
1490
1491         ZLOC = ZBO
1492
1493
1494         PHI2L = PHI2(1C) - PSI
1495         PHI1L = PHI1(1C) - PSI
1496
1497
1498
1499
1500
1501         DIF (DEBUG),,ST1
1502         IF (1BUG) ,ST1
1503         WOT 3, (1H0)
1504         WOT 3, ( XLOC,YLOC,ZLOC= ,3F10.2) ,XLOC,YLOC,ZLOC
1505         WOT 3, ( PSI,SP,CP= ,3E12.3) ,PSI,SP,CP
1506         WOT 3, ( ZL2,PHI1L,PHI2L= ,3E12.3) ,ZL2,PHI1L,PHI2L
1507 ST1 CONTINUE
1508
1509 C..SOLVE ALL ARCS AND LOOPS AS ARCS
1510 C..USE LOGIC TO SPEED UP LOOPS CASE WHEN NEEDED.
1511
1512
1513 C..HERE WE CHECK TO SEE IF OBSERVER IS INSIDE WIRE.
1514
1515         IF (XLOC) ,EXTERIOR
1516         DELPHI = RWIRE/ASC
1517         PHI1BOT = PHI1L-DELPHI
1518         PHI1TOP = PHI2L+DELPHI
1519

```

```

1520
1521
1522 C..DOES ANGLE EXCLUDE OBSERVER FROM WIRE INTERIOR
1523     IF (PHIBOT.LE.0.0 .AND. PHITOP .GE.0.0) GO TO STAY
1524     IF (PHIBOT.LE.PIX2 .AND. PHITOP .GE.PIX2) GO TO STAY
1525     GO TO EXTERIOR
1526
1527 STAY      RADOBSQ = ZLOC**2 + (ASC-XLOC)**2
1528
1529     RADOBS = SQRT(RADOBSQ)
1530
1531     IF (RADOBSQ.GE.RWIRESQ) GO TO EXTERIOR
1532
1533 C..COMING THROUGH HERE PLACES OBSERVER INSIDE WIRE
1534
1535     BTLW = XMUIX2*RADOBS*RRWIRESQ
1536     DIF (IBFLD) ,SK2
1537     BZLOC = ((ASC-XLOC)/RADOBS)*BTLW
1538     BXLOC = (ZLOC/RADOBS)*BTLW
1539     BYLOC = 0.0
1540 SK2      CONTINUE
1541
1542     IF (IAFLD) ,SK4
1543     AZLOC = 0.0
1544     AYLOC = BTLW*RADOBS
1545     AXLOC = 0.0
1546 SK4      CONTINUE
1547
1548     DIF (IDEBUG) , ,ST2
1549     IF (IBUG) ,ST2
1550     WOT 3, ( BTLW,BXLOC,BZLOC,AYLOC= ,4E12.3) ,
1551     , BTLW,BXLOC,BZLOC,AYLOC
1552     WOT 3, ( PHIBOT,PHITOP= ,2F10.3) ,PHIBCT,PHITOP
1553 ST2      CONTINUE
1554
1555     USE BACKROTATE
1556     RETURN
1557
1558 EXTERIOR CONTINUE
1559
1560 C..HERE WE GET LOCAL VECTOR POTL.
1561

```

```

1562         PRECOEF = RRT2/((ASC+XLOC)**2+ZLOC**2)
1563         CAYSQ = 4.*ASC*XLOC/((ASC+XLOC)**2+ZLOC**2)
1564         CAY = SQRT1(CAYSQ)
1565
1566         DIF (IDEBU),,ST3
1567         IF (IBUG),ST3
1568         WOT 3, ( PRECOEF,CAYSQ,CAY= ,3E12.3) ,PRECOEF,CAYSQ,CAY
1569 ST3 CONTINUE
1570
1571         IF (ITYPESC .EQ. IARC) GO TO ARCTYPE
1572
1573 C...HERE WE DO COMPLETE LOOPS
1574
1575         USE ELIPTK
1576
1577
1578         DIF (IAFLD),SK6
1579         AYLOC = 2.*AXMU(X2*PRECOEF*(2.-CAYSQ)*FIRSTK-2.*SCNDK)/
1580             (CAYSQ+.1E-37)
1581         AXLOC = 0.0
1582         AZLOC = 0.0
1583
1584 SK6 CONTINUE
1585
1586
1587         DIF (IBFLD),SK8
1588         FCAYSQ = (2.-CAYSQ)/(2.*(1.-CAYSQ))
1589
1590         BXLOC = XMU(X2*ZLOC*PRECOEF*(FCAYSQ*SCNDK-FIRSTK)/
1591             (XLOC+.1E-37)
1592
1593         BYLOC = 0.0
1594         BZLOC = XMU(X2*PRECOEF*(FIRSTK+((ASC**2-XLOC**2-ZLOC**2)/
1595             ((ASC-XLOC)**2+ZLOC**2))*SCNDK)
1596 SK8 CONTINUE
1597
1598         DIF (IDEBU),,ST4
1599         IF (IBUG),ST4
1600         WOT 3, ( FIRSTK,SCNDK,FCAYSQ= ,3E12.3) ,FIRSTK,SCNDK,FCAYSQ
1601 ST4 CONTINUE
1602
1603         USE BACKROTATE

```

```

1604          RETURN
1605
1606 ARCTYPE  CONTINUE
1607
1608 C..HERE WE DO ARCS IE INCOMPLETE LOOPS
1609
1610          OMEGA1 = .5*(PHI1L*PI)
1611          OMEGA2 = .5*(PHI2L*PI)
1612          U2(1) = COS(PHI1L)
1613          U2(2) = COS(PHI2L)
1614
1615          DIF (IDEBUG),ST42
1616          IF (IBUG),ST42
1617          WOT 3, ( U2(1)= .2E12.3) ,U2(1),U2(2)
1618 ST42  CONTINUE
1619
1620
1621          IF (CAYSQ.GT.1.E-3) GO TO SLOW
1622          IF (1AFLD),SK10
1623
1624          SOMEG1 = SIN(2.*OMEGA1)
1625          SOMEG2 = SIN(2.*OMEGA2)
1626
1627          AYLOC = .5* AXMU1X2*(SOMEG1-SOMEG2)*RRT21*(XLOC+ASC)**2
1628                + ZLOC**2)
1629
1630          AXLOC = .5*AXMU1X2*(U2(2)-U2(1))*RRT21(ASC**2+XLOC**2
1631                + ZLOC**2)
1632
1633          AZLOC = 0.0
1634 SK10  CONTINUE
1635
1636          DIF (18FLD),SK12
1637          ASZS = ASC**2+ZLOC**2
1638          RTASZS = 1./(ASZS*SQRT1(ASZS))
1639
1640          BXLOC = .5*AXMU1X2*ZLOC*(SIN(2.*OMEGA1)-SIN(2.*OMEGA2))*RTASZS
1641
1642          BYLOC = AXMU1X2*ZLOC*((SIN(OMEGA1))**2
1643                -(SIN(OMEGA2))**2)*RTASZS
1644
1645          EZLOC = AXMU1X2*ASC*(OMEGA2-OMEGA1)*RTASZS

```

```

1646 SK12      CONTINUE
1647
1648          USE BACKROTATE
1649          RETURN
1650
1651 SLOW        CALL ILIPTK
1652
1653          DIF (IDBUG),ST4:
1654          IF (IBUG),ST41
1655          WOT 3, ( OMEGA1,OMEGA2,FINCMP,EINCMF= ,4E12.3) ,OMEGA1,
1656          , OMEGA2,FINCMF,EINCMF
1657 ST41 CONTINUE
1658
1659          IF (IAFLD),SK14
1660          AYLOC = AXMU1X2*PRECOEF*((2.-CAYSQ)*FINCMF
1661                  -2.*EINCMF)/(CAYSQ+1.E-37)
1662
1663          ARG1 = ASC**2+XLOC**2+ZLOC**2-2.*ASC*XLOC*U2(1)
1664
1665          ARG2 = ASC**2+XLOC**2+ZLOC**2-2.*ASC*XLOC*U2(2)
1666
1667          AXLOC = -XMU1*(SQRT1(ARG2)-SQRT1(ARG1))/(XLOC+1.E-37)
1668
1669          AZLOC = 0.0
1670 SK14        CONTINUE
1671
1672 C..NEXT WE GET MAGNETIC FIELDS
1673 C..HERE WE FOLLOW MAFLD ARCS
1674 C..FIRST GET INCOMPLETE ELLIPTIC INTEGRAL REQUIRED
1675
1676
1677
1678          DIF (IBFLD),SK16
1679          SW1 = SIN(OMEGA1)
1680          SW2 = SIN(OMEGA2)
1681          CW1 = COS(OMEGA1)
1682          CW2 = COS(OMEGA2)
1683
1684          RECR1 = RRT21(1.-CAYSQ*SW1**2)
1685          RECR2 = RRT21(1.-CAYSQ*SW2**2)
1686
1687          SCORT1 = CAYSQ*SW1*CW1*RECR1

```

```

1688          SCORT2 = CAYSQ*SW2*CW2*RECR2
1689
1690
1691          RECR1 = PRECOEF/(XLOC+1.E-37)
1692          DEN = (1.-CAYSQ)+1.E-37
1693
1694          AQBR0 = (CAYSQ/DEN)*.5*RECR1*(EINCMF-SCORT2
1695                +SCORT1)
1696          AQBR1 = (-F|NCMP)*RECR1+((2.-CAYSQ)*(CAYSQ+1.E-37))*AQBR0
1697
1698          AQBR3 = RECR1*(RECR1-RECR2)
1699
1700
1701
1702          BXLOC = XMU1*ZLOC*AQBR1
1703          BYLOC = XMU1*ZLOC*AQBR3
1704
1705          BZLOC = XMU1*(ASC*AQBR0-XLOC*AQBR1)
1706 SK16      CONTINUE
1707          USE BACKRGATE
1708          RETURN
1709
1710
1711 LOCDONE   CONTINUE
1712
1713          DIF (IDEBUG),,ST6
1714          IF (IBUG),ST6
1715          WOT 3, ( AXLOC,AYLOC,AZLOC= ,3E12.3),AXLOC,AYLOC,AZLOC
1716          WOT 3, ( BXLOC,BYLOC,BZLOC= ,3E12.3),BXLOC,BYLOC,BZLOC
1717 ST6      CONTINUE
1718
1719          USE BACKROTATE
1720          RETURN
1721
1722          END

```

```

1723      SUBROUTINE MEDIA (LIND)
1724
1725      MACRO BMAKE, STORAGE (MACROS)
1726      USE STORAGE
1727
1728      A1  FORMAT ( 'HERE ARE DEBUG RESULTS....' )
1729
1730      A2  FORMAT ( 'HERE ARE FINAL RESULTS...' )
1731
1732      VALUE ITAB, IJDIM, JJDIM, IJMIN, IJMAX, JJMIN, JJMAX
1733
1734      DIMENSION ITAB(10)
1735
1736      DATA (ITAB(1)=.LOC.A1, .LOC.A2)
1737
1738      DATA (ANAME(1)=10H( AXA ) )
1739      DATA (ANAME(2)=10H( AYA ) )
1740      DATA (ANAME(3)=10H( AZA ) )
1741      DATA (ANAME(4)=10H( BXA ) )
1742      DATA (ANAME(5)=10H( BYA ) )
1743      DATA (ANAME(6)=10H( BZA ) )
1744      DATA (ANAME(7)=10H( BMODA ) )
1745
1746      WRITE (3, ITAB(LIND))
1747
1748      IBEGIN = 1
1749      IEND = 7
1750      IF (IBFLD.EQ.0) IEND = 3
1751      IF (IAFLD.EQ.0) IBEGIN = 4
1752
1753      DO DIAG LD=IBEGIN, IEND
1754
1755      USE BMAKE (SUPER(1,1,K,LD)) (ANAME)
1756      DIAG CONTINUE
1757
1758      RETURN
1759
1760      END

```


1761
1762 SUBROUTINE HELP (IOPT)
1763
1764 MACRO STORAGE (MACROS)
1765 USE STORAGE
1766 WOT JOUT, (THIS IS THE HELP PACKAGE)
1767
1768 RETURN
1769 END

.

```

1770
1771     SUBROUTINE ERRMSG (MSGNO,MES,NOWRD)
1772
1773     DIMENSION MES(10)
1774
1775     MACRO STORAGE(MACROS)
1776     USE STORAGE
1777
1778
1779     VALUE MTAB
1780
1781     DIMENSION MTAB(20)
1782
1783     M1  FORMAT( 'CANNOT CREATE DROPPFILE ',4A10)
1784     M2  FORMAT( 'CANNOT EXPAND DROPPFILE FOR LCM ',4A10)
1785     M3  FORMAT( 'CANNOT OPEN FILE ',4A10)
1786     M4  FORMAT( 'CANNOT CREATE FILE ',4A10)
1787     M5  FORMAT( 'SEARCH FAILED WITH ',4A10)
1788     M6  FORMAT( 'PARITY/WD CNT ERROR ON ',A10)
1789
1790     DATA (MTAB(1) = .LOC. M1, .LOC. M2, .LOC. M3, .LOC. M4, .LOC. M5,
1791           .LOC. M6)
1792
1793     WRITE (JOUT, MTAB(MSGNO)) (MES(I),I=1,NOWRD)
1794
1795     IF(MSGNO .NE. 5) GO TO 100
1796 C   FORCE HARDWARE ERROR FOR DUMP
1797         A = 1./0.
1798         B = A + C
1799     100     CONTINUE
1800
1801
1802 C   EMPTY HSP BUFFER
1803         NOT 3, (1H)
1804         CALL EMPTY(3)
1805
1806 C.. QUIT BUT DO NOT DESTROY CONTROLLEE
1807         CALL GOB (D400B,IER,0,0)
1808
1809     RETURN
1810     END

```

```

1811      SUBROUTINE SPRNT (ARIN,IROUT,IMAX)
1812
1813          OPTIMIZE
1814
1815          LCM (ARIN)
1816
1817 C  INPUT
1818 C  ARIN:  BEGINNING LOCATION OF FLOATING POINT NUMBERS TO PROCESS
1819 C  IROUT: BEGINNING LOCATION TO STORE COMPACT BCD REPRESENTATION
1820 C         OF FLOATING POINT NUMBERS
1821 C  IMAX:  TOTAL NO. OF SUCCESSIVE FLOATING POINT NUMBERS TO PROCESS
1822
1823 C  PURPOSE
1824 C  TO CONVERT N=IMAX SUCCESSIVE FLOATING POINT NUMBERS INTO A COMPACT
1825 C  BCD REPRESENTATION IN THE E-FORMAT AND STORE THEM IN N SUCCESSIVE
1826 C  TEMPORARY LOCATIONS BEGINNING AT IROUT WHICH ARE LATER TO BE
1827 C  OUTPUT IN R6 FORMAT.
1828 C  FOR EXAMPLE THE FLOATING POINT NUMBER -1.7465E+25 IS PRINTED
1829 C  AS -175+Q WHERE THE DECIMAL POINT IS ASSUMED BETWEEN THE MINUS
1830 C  SIGN AND THE DIGIT 1.  THE 2 DIGIT EXPONENT IS CONVERTED TO A SINGLE
1831 C  CHARACTER BY A TABLE LOOKUP ON (0,1,...,9,A,B,...,Z)=(0,1,2,...,35)
1832
1833 C  IF AN I. OR R. OR *. IS RETURNED BY SUBROUTINE ETYPE, IT IS PRINTED
1834
1835 C  IF THE EXPONENT IS GREATER THAN +35 OR LESS THAN -35 A * IS
1836 C  PRINTED
1837
1838
1839 C.....
1840      DOUBLE ETYPE,RR
1841      GENERIC ETYPE
1842      DIMENSION IGET(2)
1843      EQUIVALENCE (IGET(1),RR)
1844      EQUIVALENCE (IGET(2),ISAVE)
1845
1846      DIMENSION ARIN(2), IROUT(2), IEX((0,71))
1847
1848      DATA (IEX(0)=2R *,2R-Z,2R-Y,2R-X,2R-W,2R-V,2R-U,2R-T,2R-S,
1849      .      2R-R,2R-Q,2R-P,2R-O,2R-N,2R-M,2R-L,2R-K,2R-J,2R-I,2R-H,
1850      .      2R-G,2R-F,2R-E,2R-D,2R-C,2R-B,2R-A,2R-9,2R-8)
1851      DATA (IEX(29)=2R-7,2R-6,2R-5,2R-4,2R-3,2R-2,2R-1,2R+0, 1,2R+2,
1852      .      2R+3,2R+4,2R+5,2R+6,2R+7,2R+8,2R+9,2R+A,2R+B,2R+C,2R+D,2R+E,

```

```

1853      2R+F,2R+G,2R+H,2R+I,2R+J,2R+K,2R+L,2R+M,2R+N,2R+O,2R+P,
1854      2R+Q,2R+R,2R+S,2R+T,2R+U,2R+V,2R+W,2R+X,2R+Y,2R+Z)
1855
1856
1857      DO 100 I = 1,IMAX
1858          IROUT(I) = 0
1859          TEMP = ARIN(I)
1860          IF (ARIN(I).NE.0) GO TO 40
1861          IROUT(I) = 6R 0
1862          GO TO 100
1863
1864 40      CONTINUE
1865          RR = ETYPE(TEMP,9,2)
1866
1867          IF (((SAVE.NE. 8R+.      ) .AND.
1868              (ISAVE.NE. 8R|.      ) .AND.
1869              (ISAVE.NE. 8RR.      ) ) GO TO 50
1870          IROUT(I) = ISAVE .SHR. 20
1871          GO TO 100
1872
1873 50      CONTINUE
1874          ISAVE = ISAVE .SHL. 6
1875          IROUT(I) = (((ISAVE.SHL.6).INT.77B).UN.IROUT(I)
1876
1877          IROUT(I) = (((ISAVE.SHL.12).INT.77B).UN.(IROUT(I).SHL.6)
1878
1879          IROUT(I) = (((ISAVE.SHL.24).INT.77B).UN.(IROUT(I).SHL.6)
1880
1881          IROUT(I) = (((ISAVE.SHL.30).INT.77B).UN.(IROUT(I).SHL.6)
1882
1883          IF (((ISAVE.SHL.36).INT.77B).EQ.1RE) GO TO 60
1884
1885          ILUK = 0
1886          GO TO 70
1887
1888 60      CONTINUE
1889          ILUK = (((ISAVE.SHL.48).INT.17B) * 10
1890
1891          ILUK = (((ISAVE.SHL.54).INT.17B) + ILUK
1892
1893          IF (((ISAVE.SHL.39).LT.0) ILUK = -ILUK
1894          ILUK = ILUK+36+1

```

```

1895
1896         IF (ILUK.GT.7) .OR. ILUK.LT.1) ILUK = 0
1897
1898 70  CONTINUE
1899         IROUT(1) = (IROUT(1).SHL.12).UN.IEX(ILUK)
1900
1901 100 CONTINUE
1902
1903         RETURN
1904
1905         END

```

```

1906      SUBROUTINE PRNT (ARRAY, IDIM, JDIM, IMIN, IMAX, JMIN, JMAX)
1907      LCM (ARRAY)
1908
1909 C..PROGRAMMER: J BREAZEAL
1910 C..DATE: 1/10/75
1911
1912 C..THINGS TO CONSIDER IN USING THIS SUBROUTINE
1913 C   SET VALUE OF ICR
1914 C   SET VALUE OF NOUT
1915 C   ADD DECLARATIVE FOR LCM IF ARRAY IS IN LCM
1916 C   ADD DECLARATIVE VALUE IDIM, JDIM, IMIN, IMAX, JMIN, JMAX
1917 C   THIS ROUTINE CALLS SPRNT (J BREAZEAL) AND ORDERLIB ROUTINES
1918
1919 C..THIS ROUTINE IS FASHIONED AFTER THE NRL PRNT ROUTINE (SEE D ANDERSON)
1920
1921 C..THIS ROUTINE WILL PRINT THE 2-D ARRAY SPECIFIED. A PARTIAL
1922 C..PRINTOUT OF THE ARRAY IS OBTAINED BY SETTING IMIN, IMAX, AND
1923 C..JMIN, JMAX. THE MAXIMUM POINTS PER LINE IS 19 IN THE I DIRECTION.
1924 C..THERE IS NO LIMIT IN THE J DIRECTION. IF MORE THAN 19 POINTS
1925 C..SPAN THE PRINTOUT IN THE I DIRECTION, THE PRINTOUT OCCURS IN
1926 C..PARTIAL BLOCKS, WHERE I=IMIN TO IMIN+19, FOR JMIN TO JMAX;
1927 C..      AND NEXT: I=IMIN+19 TO IMIN+39, FOR JMIN TO JMAX; ETC.
1928
1929 C..USAGE:
1930
1931 C   CALL PRNT (ARRAY(1,1), IDIM, JDIM, IMIN, IMAX, JMIN, JMAX)
1932
1933
1934 C..ARRAY: ARRAY TO BE PRINTED
1935 C..IDIM: DIMENSION IN I
1936 C..JDIM: DIMENSION IN J
1937 C..IMIN, IMAX: LOW AND HIGH RANGE OF INDEX I
1938 C..JMIN, JMAX: LOW AND HIGH RANGE OF INDEX J
1939 C..ICR: MAX NO. OF POINTS TO PRINT PER LINE
1940
1941      OPTIMIZE
1942
1943      VALUE IDIM, JDIM, IMIN, IMAX, JMIN, JMAX
1944
1945      PARAMETER (ICR = 19)          $$$ POINTS PRINTED PER LINE, MAX OF 19.....
1946      PARAMETER (NOUT = 3)          $$$ OUTPUT UNIT NO. ....
1947      DIMENSION ARRAY (IDIM, JDIM), SPARRAY(ICR)

```

```

1948
1949
1950      I2 = 0
1951      DO 25 IBLK = IMIN,IMAX,ICR
1952
1953          I2 = I2 + ICR + IMIN - 1
1954          IF (I2.GT.IMAX) I2 = IMAX
1955          INUM = I2 - IBLK + 1
1956          WRITE (NOUT,51) (INUM,(I,I=IBLK,I2))
1957
1958      DO 20 J = JMIN,JMAX
1959
1960          CALL SPRT (ARRAY(IBLK,J),SPARAY,INUM)
1961          WRITE (NOUT,50) (J,(SPARAY(I),I=1,INUM))
1962 20  CONTINUE
1963
1964 25  CONTINUE
1965
1966
1967 50  FORMAT (3H J=,I2, 1966)
1968 51  FORMAT (5H      ,*(      )=      ,I2))
1969
1970      RETURN
1971      END

```

```

1972
1973
1974
1975      SUBROUTINE HEAPSORT( KEYS, RECS, RECS2,RECS3,RECS4,RECS5,RECS6,RECS7, N )
1976 $
1977 $
1978 $      THIS SUBROUTINE USES A MODIFICATION OF THE -HEAPSORT- ALGORITHM
1979 $      DESCRIBED BY KNUTH IN -THE ART OF COMPUTER PROGRAMMING- VOL 3 SEC 5.2.3
1980 $      TO DO AN ASCENDING SORT OF -RECS-, BASED ON THE VALUES OF -KEYS-.
1981 $
1982 $      THUS WHEN -HEAPSORT- RETURNS, THE ARRAY -KEYS- IS IN ASCENDING
1983 $      ORDER, AND THE ARRAY -RECS- WILL HAVE BEEN IDENTICALLY PERMUTED.
1984 $
1985 $      -KEYS- AND -RECS- ARE FLOATING POINT ARRAYS.
1986 $
1987 $
1988 $ *****  D E C L A R A T I O N S  *****
1989 $
1990 $
1991      ALL INTEGER
1992      INTEGER N
1993      REAL KEY, REC, KEYS, RECS
1994 C..1 LINE ADDED 3/12/74
1995      REAL RECS2,RECS3,RECS4,REC2,REC3,REC4,RECS5,RECS6,RECS7,REC5,REC6,REC7
1996
1997      DIMENSION KEYS(2), RECS(2),RECS2(2),RECS3(2),RECS4(2),RECS5(2),RECS6(2),
1998              RECS7(2)
1999      CODE ANALYSIS
2000 $
2001      OPTIMIZE
2002
2003      ARITHREG KEY, REC
2004      INDEXREG I, J, L, R
2005 $
2006 $
2007 $
2008 $ *****  M A K E   A   H E A P  *****
2009 $
2010 $
2011 $      H1 - INITIALIZE
2012      L = N .SHR. 1
2013      R = N

```



```

2014 $
2015 $
2016 $ H2 - MAKE A HEAP BETWEEN L AND R
2017 H2 CONTINUE
2018 KEY = KEYS(L)
2019 REC = RECS(L)
2020 C.. 3 LINE ADDED 5/12/75
2021 REC2 = RECS2(L)
2022 REC3 = RECS3(L)
2023 REC4 = RECS4(L)
2024 REC5 = RECS5(L)
2025 REC6 = RECS6(L)
2026 REC7 = RECS7(L)
2027
2028 $
2029 $
2030 $ H3 - PREPARE FOR SIFTUP
2031 J = L
2032 $
2033 $
2034 $ H4 - ADVANCE DOWNWARD
2035 H4 CONTINUE
2036 I = J
2037 J = J + J
2038 IF (J .EQ. R) GO TO H6
2039 IF (J .GT. R) GO TO H6
2040 $
2041 $
2042 $ H5 - FIND LARGER SON
2043 IF (KEYS(J) .LT. KEYS(J+1)) J = J + 1
2044 $
2045 $
2046 $ H6 - IS LARGER SON LARGER THAN KEY
2047 H6 IF (KEY .GE. KEYS(J)) GO TO H8
2048 $
2049 $
2050 $ H7 - MOVE IT UP
2051 KEYS(I) = KEYS(J)
2052 RECS(I) = RECS(J)
2053 C..6 LINES ADDED 5/12/75
2054 RECS2(I) = RECS2(J)
2055 RECS3(I) = RECS3(J)

```

```

2056      RECS4(I) = RECS4(J)
2057      RECS5(I) = RECS5(J)
2058      RECS6(I) = RECS6(J)
2059      RECS7(I) = RECS7(J)
2060
2061      GO TO H4
2062 $
2063 $
2064 $   HB - STORE KEY, REC
2065 HB   CONTINUE
2066      KEYS(I) = KEY
2067      RECS(I) = REC
2068 C..6 LINES ADDED 5/12/75
2069      RECS2(I) = REC2
2070      RECS3(I) = REC3
2071      RECS4(I) = REC4
2072      RECS5(I) = REC5
2073      RECS6(I) = REC6
2074      RECS7(I) = REC7
2075      L = L - 1
2076      IF (L .GT. 0) GO TO H2
2077 $
2078 $
2079 $
2080 $ ***** DO THE SORT *****
2081 $
2082 $
2083 $   I1 - INITIALIZE FOR EACH SELECTION PASS THRU THE HEAP
2084 I1   CONTINUE
2085      KEY = KEYS(R)
2086      REC = RECS(R)
2087 C.. 3 LINES ADDED 5/12/75
2088      REC2 = RECS2(R)
2089      REC3 = RECS3(R)
2090      REC4 = RECS4(R)
2091      REC5 = RECS5(R)
2092      REC6 = RECS6(R)
2093      REC7 = RECS7(R)
2094
2095
2096      KEYS(R) = KEYS(I)
2097      RECS(R) = RECS(I)

```

```

2098 C..6 LINES ADDED 5/12/75
2099     RECS2(R) = RECS2(I)
2100     RECS3(R) = RECS3(I)
2101     RECS4(R) = RECS4(I)
2102     RECS5(R) = RECS5(I)
2103     RECS6(R) = RECS6(I)
2104     RECS7(R) = RECS7(I)
2105
2106     I = I
2107     R = R - I
2108 $
2109 $
2110 $ 12 - FIND LARGER SON OF I-TH NODE
2111 12 CONTINUE
2112     J = I + I
2113     IF (J .GT. R) GO TO 14
2114     IF (J .EQ. R) GO TO 13
2115     IF (KEYS(J) .LT. KEYS(J+1)) J = J + 1
2116 $
2117 $
2118 $ 13 - MOVE LARGER SON OF I-TH NODE INTO NODE I
2119 13 CONTINUE
2120     KEYS(I) = KEYS(J)
2121     RECS(I) = RECS(J)
2122 C..6 LINES ADDED 5/12/75
2123     RECS2(I) = RECS2(J)
2124     RECS3(I) = RECS3(J)
2125     RECS4(I) = RECS4(J)
2126     RECS5(I) = RECS5(J)
2127     RECS6(I) = RECS6(J)
2128     RECS7(I) = RECS7(J)
2129
2130     I = J
2131     GO TO 12
2132 $
2133 $
2134 $ 14 - FIND THE APPROPRIATE LEVEL IN THE HEAP FOR -KEY-
2135 14 CONTINUE
2136     J = J .SHR. 1
2137 15 CONTINUE
2138     IF (J .EQ. 1) GO TO 16
2139     I = J .SHR. 1

```

```

2140      IF (KEYS(I) .GT. KEY) GO TO 16
2141      KEYS(J) = KEYS(I)
2142      RECS(J) = RECS(I)
2143 C..6 LINES ADDED 5/12/75
2144      RECS2(J) = RECS2(I)
2145      RECS3(J) = RECS3(I)
2146      RECS4(J) = RECS4(I)
2147      RECS5(J) = RECS5(I)
2148      RECS6(J) = RECS6(I)
2149      RECS7(J) = RECS7(I)
2150
2151      J = I
2152      GO TO 15
2153 $
2154 $
2155 $ 16 - INSERT -KEY- AT THE APPROPRIATE LEVEL (WHICH WE HAVE JUST DETERMINED)
2156 16 CONTINUE
2157      KEYS(J) = KEY
2158      RECS(J) = REC
2159 C..6 LINES ADDED 5/12/75
2160      RECS2(J) = REC2
2161      RECS3(J) = REC3
2162      RECS4(J) = REC4
2163      RECS5(J) = REC5
2164      RECS6(J) = REC6
2165      RECS7(J) = REC7
2166
2167      IF (R .GT. I) GO TO 11
2168      RETURN
2169 $
2170 $
2171      RELEASE KEY, REC, I, J, L, R
2172 $
2173 $
2174      END

```

```

2175      SUBROUTINE ABCPLOT (IKIND,ARRAY)
2176
2177
2178 C..PURPOSE:  TO DRIVE PLOT ROUTINE LINEPLT AND CRPLT
2179
2180      MACRO STORAGE (MACROS)
2181      USE STORAGE
2182      CODE ANALYSIS
2183
2184
2185
2186 C..CONTOUR PLOT FOR ARRAY CONTAINING BMOD OR AMOD
2187 C..IKIND = 0 FOR BMOD; =1,2,3,4 FOR AMOD
2188
2189
2190      DIMENSION IABSC(B,3),IORDIN(B),ARRAY(IMAX,UMAX,KMAX)
2191
2192
2193      COMMON/CPLT/ SCAL(4),NFR, DOWNER, UPPER, LEVL
2194
2195
2196 C..FOR LINE PLOTS
2197      BYTE IGRID(2),IDUM1(40),IXGRID(6),IYGRID(6),IZGRID(6)
2198      STRUCTURE (IH1,IGRID/IDUM1/IXGRID/IYGRID/IZGRID)
2199
2200 C..FOR CONTOUR PLOTS
2201      BYTE NPLANE(2),IDUM2(52),NGRID(6)
2202      STRUCTURE (IH2,NPLANE/IDUM2/NGRID)
2203
2204
2205      SCAL(1)=SCAL(2)=0.0
2206      SCAL(3)=SCAL(4)=0.0
2207
2208      LEVL = 20
2209      IF (IKIND.NE.0) 2,
2210      UPPER = BMAXPLT
2211      DOWNER = BMINPLT
2212      GO TO 10
2213
2214 2      CONTINUE
2215      IF (IKIND.NE.1) 4,
2216      UPPER = AMAXPLT

```

```

2217     DOWNER = AMINPLT
2218     GO TO 10
2219
2220 4     CONTINUE
2221     IF (IKIND.NE.2) 6,
2222     UPPER = AXMAXPLT
2223     DOWNER = AXMINPLT
2224     GO TO 10
2225
2226 6     CONTINUE
2227     IF (IKIND.NE.3) 8,
2228     UPPER = AYMAXPLT
2229     DOWNER = AYMINPLT
2230     GO TO 10
2231
2232 8     CONTINUE
2233     UPPER = AZMAXPLT
2234     DOWNER = AZMINPLT
2235 10    CONTINUE
2236
2237
2238
2239 C..DO CONTOUR PLOTS
2240 100   CONTINUE
2241
2242
2243
2244
2245
2246 C..CONTOUR PLOT NO.1
2247     IABSC(1) = Y AXIS (CM) ...CONTOURS OF BMOD AT PLANE...
2248     IF (IKIND.EQ.0) GO TO 150
2249     IF (IKIND.EQ.1) IABSC(3) = NTOURS OF AMOD AT PLANE...
2250     IF (IKIND.EQ.2) IABSC(3) = NTOURS OF AX AT PLANE...
2251     IF (IKIND.EQ.3) IABSC(3) = NTOURS OF AY AT PLANE...
2252     IF (IKIND.EQ.4) IABSC(3) = NTOURS OF AZ AT PLANE...
2253 150   CONTINUE
2254     IORDIN(1) = Z AXIS (CM)
2255     IF (NGR(1)) ,250
2256     NPLANE = NPL(1)
2257     NGRID = NGR(1)
2258

```

```

2259      CALL CRPLOT (ARRAY, IH2, IABSC, IORDIN, X, Y, Z)
2260
2261 C...CONTOUR PLOT NO.2
2262 250 CONTINUE
2263      IF (NGR(2)) ,350
2264      NPLANE = NPL(2)
2265      NGRID = NGR(2)
2266
2267      IABSC(1) = X AXIS (C)
2268
2269      CALL CRPLOT (ARRAY, IH2, IABSC, IORDIN, X, Y, Z)
2270
2271 C...CONTOUR PLOT NO.3
2272 350 CONTINUE
2273      IF (NGR(3)) ,450
2274      NPLANE = NPL(3)
2275      NGRID = NGR(3)
2276
2277      IORDIN(1) = Y AXIS (CM)
2278      IABSC(1) = X AXIS (C)
2279
2280      CALL CRPLOT (ARRAY, IH2, IABSC, IORDIN, X, Y, Z)
2281
2282 C...LINE PLOT NO.1
2283 450 CONTINUE
2284      NFR = 1
2285      IABSC(1,1) = X COORDINATE (CM)
2286      IABSC(1,2) = Y COORDINATE (CM)
2287      IABSC(1,3) = Z COORDINATE (CM)
2288      IORDIN(1) = BMOD (GAUSS)
2289      IF (IKIND.EQ.0) GO TO 451
2290      IF (IKIND.EQ.1) IORDIN(1) = AMOD (GAU
2291      IF (IKIND.EQ.2) IORDIN(1) = AX (GAU
2292      IF (IKIND.EQ.3) IORDIN(1) = AY (GAU
2293      IF (IKIND.EQ.4) IORDIN(1) = AZ (GAU
2294 451 CONTINUE
2295
2296      DO 750 L = 1,6
2297      IF (IGR(L)) ,750
2298      IXGRID = IXGR(L)
2299      IYGRID = IYGR(L)
2300      IZGRID = IZGR(L)

```

```

2301      IGRID = IGR(L)
2302      CALL LINEPLT (ARRAY,IH1,IABSC(1,IGR(L)),IORDIN,X,Y,Z)
2303
2304
2305 750  CONTINUE
2306
2307      CALL PLOTEA
2308
2309
2310      RETURN
2311      END

```



```

2312      SUBROUTINE LINEPLT (ARRAY, IHOLD,IABSC,IORDIN,X,Y,Z)
2313
2314      LCM ARRAY
2315
2316      CODE ANALYSIS
2317
2318      OPTIMIZE
2319
2320 C..PURPOSE: THIS ROUTINE WILL DO A LINE PLOT FROM A 3D INPUT ARRAY
2321 C..THE PARTICULAR LINE PLOTTED IS DETERMINED BY THE INPUT PARAMETER
2322 C..IHOLD. THE TITLES FOR THE ABSCISSA AND ORDINATE ARE SPECIFIED
2323 C..IN THE ARRAYS IABSC AND IORDIN. ONE OR MORE LINES WILL BE PLOTTED
2324 C..PER FRAME BY SUPPRESSING THE FRAME ADVANCE WHICH ALSO INHIBITS
2325 C..THE PRINTING OF TITLES AND DOING A MAPPING.
2326
2327 C..INPUT:
2328 C..ARRAY: 3-D ARRAY INPUT
2329 C..IHOLD: BITS (58-59) 1=X GRID FOR ABSCISSA,2=Y GRID FOR ABSCISSA
2330 C..      3=Z GRID FOR ABSCISSA
2331 C..BIT (12-17) X GRID NUMBER TO FIX
2332 C..BIT (6-11) Y GRID NUMBER TO FIX
2333 C..BIT (0-5) Z GRID NUMBER TO FIX
2334 C..IABSC AND IORDIN DEFINED AS FOLLOWS FOR TITLE PRINTING WITH MAX 80 CHAR.
2335 C   DATA IABSC/ ABSCISSA TITLE /
2336
2337 C..INPUT SPECIFIED IN COMMON
2338 C..SCAL(1) MIN ORDINATE
2339 C..SCAL(2) MAX ORDINATE (IF MIN=MAX=0 USE FUNCTION MAX AND MIN)
2340 C..SCAL(3) MIN ABSCISSA
2341 C..SCAL(4) MAX ABSCISSA (IF MIN=MAX=0 USE ARGUMENT MAX AND MIN)
2342 C..NFR 1= DO FRAME ADVANCE, PLOT TITLES, AND DO MAPPING; 0=SUPERIMPOSE PLOTS
2343 C..X,Y,Z 1-D ARRAYS OF GRID
2344
2345
2346 C..PARAMETER DECLARATIONS
2347
2348
2349
2350      MACRO DIMM (MACROS)
2351      USE DIMM
2352
2353      COMMON /CPLT/ SCAL(4) , NFR, DOWNER, UPPER, LEVL

```

```

2354
2355     DIMENSION ARRAY (IMAX,JMAX,KMAX),IABSC(2),IORDIN(2),ABSC(MAXDM),
2356                   ORDIN(MAXDM),X(2),Y(2),Z(2)
2357
2358     BYTE IGRID(2),IDUM(40),IXGRID(6),IYGRID(6),IZGRID(6)
2359     STRUCTURE (IHOLD,IGRID/IDUM/IXGRID/IYGRID/IZGRID)
2360
2361 C..CHECK FOR FRAME ADVANCE FLAG
2362     IF (NFR) CALL FRAME
2363
2364 C..WRITE TITLES IF JUST HAD FRAME ADVANCE
2365     IF (NFR) ,99
2366
2367     CALL SETCH (1.0,30.,0,0,1,1,0)
2368     WOT 100, (8A10) ,((IORDIN(1),I=1,8)
2369
2370 C..SET UP ABSC AND ORDIN ARRAYS
2371 99  CONTINUE
2372     GO TO (100,200,300),IGRID
2373 100 CONTINUE
2374     IF (NFR) ,109
2375     IABSC(4) = AT Y(CM) AND Z(CM)=
2376     CALL SETCH (10.,1.0,0,0,1,0,0)
2377     WOT 100, (6A10,2F6.2) ,(IABSC(1),I=1,6),Y(IYGRID),Z(IZGRID)
2378 109 CONTINUE
2379
2380     DO 110 I = 1,KI
2381     ABSC(I) = X(I)
2382 110 ORDIN(I) = ARRAY(I,IYGRID,IZGRID)
2383
2384     MAXGRID = KI
2385     GO TO 400
2386
2387 200 CONTINUE
2388     IF (NFR) ,209
2389     IABSC(4) = AT X(CM) AND Z(CM)=
2390     CALL SETCH (10.,1.0,0,0,1,0,0)
2391     WOT 100, (6A10,2F6.2) ,(IABSC(1),I=1,6),X(IXGRID),Z(IZGRID)
2392 209 CONTINUE
2393
2394     DO 210 J = 1,KJ
2395     ABSC(J) = Y(J)

```

```

2396 210 ORD(N(J) = ARRAY(I\XGRID,J,I\ZGRID)
2397
2398 MAXGRID = KJ
2399 GO TO 400
2400
2401 300 CONTINUE
2402 IF (NFR).309
2403 IABSC(4) = AT X(CM) AND Y(CM)=
2404 CALL SETCH (10.,1.0,0.0,1.0,0)
2405 WOT 100, (6A10,2F6.2) ,(IABSC(I),I=1,6),X(I\XGRID),Y(I\YGRID)
2406 309 CONTINUE
2407
2408 DO 310 K = 1,KK
2409 ABSC(K) = Z(K)
2410 310 ORD(N(K) = ARRAY(I\XGRID,I\YGRID,K)
2411
2412 MAXGRID = KK
2413
2414 C..SET MAPPING PARAMETERS
2415 400 CONTINUE
2416 IF (NFR) ,500
2417 IF (SCAL(1).UN.SCAL(2)) GO TO 410
2418 C..FIND MAX AND MIN OF ORDINATE
2419 CALL AMINMX (ORDIN,1,MAXGRID,1,SCAL1,SCAL2)
2420 GO TO 420
2421
2422 410 CONTINUE
2423 SCAL1 = SCAL(1)
2424 SCAL2 = SCAL(2)
2425
2426 420 CONTINUE
2427 IF (SCAL(3).UN.SCAL(4)) GO TO 430
2428 SCAL3 = ABSC(1)
2429 SCAL4 = ABSC(MAXGRID)
2430 GO TO 440
2431
2432 430 CONTINUE
2433 SCAL3 = SCAL(3)
2434 SCAL4 = SCAL(4)
2435
2436 440 CONTINUE
2437

```

```
2438      CALL MAPS (SCAL3,SCAL4,SCAL1,SCAL2,0.1,.99,0.1,.99)
2439
2440 530    CONTINUE
2441
2442      CALL SETCRT (ABSC,ORDIN,1)
2443      CALL TRACE (ABSC,ORDIN,MAXGRID,1,1)
2444
2445      RETURN
2446      END
```

```

2447
2448
2449
2450
2451      SUBROUTINE FLDLN
2452
2453      MACRO STORAGE (MACROS)
2454      USE STORAGE
2455
2456
2457 C..PURPOSE:  TO CALCULATE THE N FIELDS LINES AND PLOT FOUR COMPOSITE
2458 C  GRAPHS OF THE N FIELD LINES GENERATED.  THE FOUR GRAPHS ARE:
2459 C  1. BMOD VS DISTANCE S ALONG THE LINE
2460 C  2. X VS Y AT Z=0
2461 C  3. Y VS Z AT X=0
2462 C  4. X VS Z AT Y=0
2463
2464 C..METHOD:  THE INITIAL POINT AND DELTA S IS DEFINED FOR EACH FIELD LINE.
2465 C  THE CODE USES THE RUNGE KUTTA ROUTINE TO CALCULATE 5 STARTING
2466 C  POINTS ALONG THE FIELD LINE AT DELTA S INCREMENTS AND THEN USES
2467 C  THE HAMMING PREDICTOR-CORRECTOR THEREAFTER TILL THE TOTAL LENGTH
2468 C  OF THE FIELD LINE HAS BEEN DEFINED.  THE PROCEDURE
2469 C  CALCULATES  $S=2*(XMAX-XMIN+YMAX-YMIN+ZMAX-ZMIN)$ 
2470 C  THEN IT CALCULATES  $\Delta S = S/1000$  THEN THE FIELD IS COMPUTED
2471 C  ALONG THE FIELD LINE TILL ONE OF THE CONDITIONS OCCURS:
2472 C  1. IF EXCEED 1000 POINTS ON THE LINE
2473 C  2. IF GO OUT OF BOX DEFINED BY THE GRID FOR X,Y,Z (THIS CHECK FOR
2474 C    PLOT 2,3,4 ONLY AND THEN THE CURRENT VALUE OF SAPROX REPLACES
2475 C    ORIGINAL S CALCULATED FOR THE FIELD LINE)
2476 C  3. IF EXCEED LENGTH S
2477 C  NSKIP IS USED IF INPUT, OR CALCULATED TO KEEP AT MOST 100 POINTS
2478 C  FOR THE PRINT AND PLOT.  THE ARRAY IS MOVED TO LCM AND POINTERS TO THE
2479 C  BEGINNING ADDRESS AND NO. OF ITEMS FOR THIS LINE IS SAVED
2480 C  AFTER ALL THE FIELD LINES ARE OBTAINED, THE COMPOSITE PLOT IS MADE
2481 C  IF ONLY ONE LINE IS PLOTTED PER FRAME, THE MAPPING IS DONE INDIVID
2482 C  OTHERWISE ALL THE FIELDS LINES ARE PLOTTED ON ONE FRAME.
2483 C  IN PLOTS 2,3,4 THE + AND - DIRECTIONS ARE PLOTTED FROM SYMMETRY
2484
2485 C*****
2486      PARAMETER (ANINE = 9.)
2487      PARAMETER (FOURTHIRD = 4./3.)
2488      PARAMETER (EIGHTH = 1./8.)

```

```

2489      PARAMETER (HALF = 1./2.)
2490      PARAMETER (OTT = 112.0/121.0)
2491      PARAMETER (OTO = 121.0/112.0)
2492      PARAMETER (OTN = 9.0/121.0)
2493      PARAMETER (SIXTH = 1.0/6.0)
2494
2495 C..CONTIGUOUS DUE TO QVSET..XFLC,YFLC,ZFLC,BFLC,SFLC
2496 C..CONTIGUOUS DUE TO QVSET..XF,YF,ZF,SAPROX,BMAGF,BXFLD,BYFLD,BZFLD
2497
2498      LCM (FLDLCM)
2499      COMMON/FLDLCM/ XFLC(MSTOP),YFLC(MSTOP),ZFLC(MSTOP),BFLC(MSTOP),SFLC(MSTOP)
2500
2501      DIMENSION LCFWA(10),LCNUM(10),R(101)
2502      DIMENSION CAXX(4),ELXX(4),EMXX(4),COEF(4)
2503      DATA (COEF = HALF,HALF,1.0)
2504
2505
2506
2507 C*****
2508
2509 C..SET FLDLN ROUTINE ACTIVE FLAG
2510      MFACTIVE = 1
2511
2512      IBFLD = MFABS
2513
2514      CALL FLDLNIN
2515
2516 C..INITIALIZE FIELD LINE INDEX
2517
2518      MF = 0
2519
2520 C..INITIALIZE INDEX TO SAVE CURRENT FIELD LINE ARRAYS IN LCM
2521      LCFWA(1) = 1
2522
2523 /
2524 C..CLEAR LCM STORAGE FOR XFLC,YFLC,ZFLC,BFLC,SFLC
2525
2526      CALL QVSET (0.0,XFLC(1),5*MSTOP)
2527
2528      PASS = 1.0
2529
2530 C..INITIALIZE FIELD LINE TO FOLLOW

```

```

2531 725 CONTINUE
2532 IF (PASS .LT. 0.0) GO TO 740
2533
2534 MF = MF + 1
2535
2536
2537 CALL QVSET (0.0,XF(1),8*MSTOP)
2538
2539 WOT 3,F4,MF,DELMAX(MF)*PASS,STOTAL(MF)*PASS
2540 F4 FORMAT ( IFIELD LINE INDEX = ,I5, DELTA S= ,F10.3,
2541 S LENGTH(CM) = ,F10.3)
2542
2543 C..CALCULATE S
2544
2545 S = 2.0*(X(K1)-X(1)+Y(KJ)-Y(1))+Z(KK)-Z(1))
2546
2547 IF (STOTAL(MF),740
2548 IF (STOTAL(MF).LT.S) S = STOTAL(MF)
2549
2550 C..RESET INTEGRATION JUMP INDEX
2551 740 CONTINUE
2552 NIX = 1
2553 S = S*PASS
2554
2555
2556 C..RESET 1ST TIME FLAG TO GET DELTA S
2557 NDX3 = 1
2558
2559 C..RESET FLAG FOR HAMMING PREDICTOR CORRECTOR
2560 ISKIP = 0
2561
2562 C..RESET DATA INDEX
2563 IF (PASS.LT.0.0) GO TO 741
2564
2565
2566 INDEX3 = 0
2567
2568 741 CONTINUE
2569 C..RESET DELTA FOR CURRENT LINE
2570 DELTA = DELMAX(MF)*PASS
2571 C..CALCULATE WHEN WITHIN END OF FIELD LINE
2572 TEST = S - .5*DELTA

```

```

2573 C..INITIALIZE FIELD LINE WITH STARTING VALUES
2574     XF(INDEX3+1) = XBAR(MF)
2575     YF(INDEX3+1) = YBAR(MF)
2576     ZF(INDEX3+1) = ZBAR(MF)
2577     SAPROX(INDEX3+1) = 0.0
2578
2579
2580 C..NOW FOLLOW FIELD LINE
2581     CALL FOLLOW
2582
2583     PASS = PASS*(-1.0)
2584     IF (PASS.LT.0.0) GO TO 725
2585
2586 C..EXTRACT EVERY NTH DATA POINT TO FORM COMPRESSED ARRAY FOR PRINT
2587 C   AND PLOT AND SAVE PARTIAL ARRAY IN LCM
2588
2589     CALL HEAPSORT (SAPROX, BMAGF,XF,YF,ZF,BXFLD,BYFLD,BZFLD,INDEX3)
2590
2591
2592     CALL COMPRESS
2593
2594 C..PRINT FIELD LINE (COMPRESSED ARRAY)
2595     CALL LINEOUT
2596
2597 C..PLOT BLOCK OF DATA OBTAINED FOR FIELD LINE IF SINGLE PLOT REQUESTED
2598     IF (MFIELD.GT.0) CALL LINCRT1
2599
2600
2601 C..CHECK IF DONE ALL FIELD LINES
2602
2603
2604     IF (MFABS - MF) .,725
2605
2606     IF (MFIELD.LT.0) CALL LINCRTALL
2607 C..SET FLAG THAT FLDLN ROUTINE INACTIVE
2608     MFACTIVE = 0
2609     RETURN FLDLN
2610
2611     ENTRY FOLLOW
2612
2613 C   PURPOSE::TO INTEGRATE DELTA S ALONG FIELD LINE IN THE DIRECTION
2614 C   OF THE FIELD UNTIL THE QUIT CRITERIA ARE MET

```



```

2615
2616 C*****
2617 C  INLINE FUNCTIONS:
2618     PRED1(XRK1,DXRK2,DXRK3,DXRK4) = XRK1 + FTDS*(2.0*DXRK4-DXRK3
2619     .                               +2.0*DXRK2)
2620
2621     CORR1(XRK5,PRED) = OT0* (XRK5-OTN*PRED)
2622
2623     PRED2(DXRK4,DXRK5,EM) = THRED* (EM+2.0*DXRK5-DXRK4)
2624
2625     CORR2(XRK3,XRK5,AC) = EIGHTH* (ANINE*XRK5-XRK3+AC)
2626
2627     PRED3(XRK2,DXRK5,DXRK4,DXRK3) = XRK2+FTDS (2.0*DXRK5
2628     .                               -DXRK4+2.0*DXRK3)
2629
2630     CORR3(PJ,CO,PO) = PJ+OTI*(CO-PO)
2631 C*****
2632
2633 729  CONTINUE
2634     INDEX3 = INDEX3 + 1
2635
2636     KEY = 0
2637     XO = XF(INDEX3)
2638     YO = YF(INDEX3)
2639     ZO = ZF(INDEX3)
2640
2641 C...GET  BXFLD(INDEX3),BYFLD(INDEX3),BZFLD(INDEX3),BMAGF(INDEX3)
2642     CALL FIELDS
2643
2644 C...TO  STORE BXFLD,BYFLD,BZFLD,BMAGF
2645     BXFLD(INDEX3) = SBX
2646     BYFLD(INDEX3) = SBY
2647     BZFLD(INDEX3) = SBZ
2648     BMAGF(INDEX3) = SQRT((SBX**2+SBY**2+SBZ**2)
2649
2650     IF (BMAGF(INDEX3)) 731,
2651     INDEX3 = INDEX3-1
2652     RETURN
2653
2654 731  CONTINUE
2655     BMAGF = 1./BMAGF(INDEX3)
2656

```

```

2657         IF (NDX3) ,B10,
2658         NDX3 = 0
2659
2660 809 CONTINUE
2661         THRED = 3.0*DELTA
2662         FIDS = FOURTHIRO * DELTA
2663         DELTA1 = 1./DELTA
2664 810 CONTINUE
2665         GO TO (B15,B15,315,B15,B15,B16,B17),NIX
2666 815 CONTINUE
2667         CALL RUGKUT (DELTA)
2668         GO TO B19
2669
2670 C..START OF HAMMING METHOD
2671 C..SOURCE: A. RALSTON, FIRST COURSE IN NUMERICAL ANALYSIS, MCGRAW-
2672 C HILL, P186-215
2673
2674 816 CONTINUE
2675         NIX = 7
2676         DO 1670 I = 1,5
2677             DXRK(I) = DXRK(I)*DELTA1
2678             DYRK(I) = DYRK(I)*DELTA1
2679             DZRK(I) = DZRK(I)*DELTA1
2680 1670 CONTINUE
2681
2682         PX0 = PRED1(XRK(1),DXRK(2),DXRK(3),DXRK(4))
2683         PY0 = PRED1(YRK(1),DYRK(2),DYRK(3),DYRK(4))
2684         PZ0 = PRED1(ZRK(1),DZRK(2),DZRK(3),DZRK(4))
2685
2686         CX0 = CORR1(XRK(5),PX0)
2687         CY0 = CORR1(YRK(5),PY0)
2688         CZ0 = CORR1(ZRK(5),PZ0)
2689         GO TO B18
2690
2691 817 CONTINUE
2692         EMX = BXFLD(INDEX3)*BMAGF1
2693         EMY = BYFLD(INDEX3)*BMAGF1
2694         EMZ = BZFLD(INDEX3)*BMAGF1
2695
2696         IF (ISKIP) 1800,,
2697 1750 CONTINUE
2698         ACX = PRED2(DXRK(4),DXRK(5),EMX)

```

```

2699      ACY = PRED2(DYRK(4),DYRK(5),EMY)
2700      ACZ = PRED2(DZRK(4),DZRK(5),EMZ)
2701
2702      CXJ = CORR2(XRK(3),XRK(5),ACX)
2703      CYJ = CORR2(YRK(3),YRK(5),ACY)
2704      CZJ = CORR2(ZRK(3),ZRK(5),ACZ)
2705
2706      C...ROTATE THINGS
2707      XRK(2) = XRK(3)
2708      YRK(2) = YRK(3)
2709      ZRK(2) = ZRK(3)
2710
2711      XRK(3) = XRK(4)
2712      YRK(3) = YRK(4)
2713      ZRK(3) = ZRK(4)
2714
2715      XRK(4) = XRK(5)
2716      YRK(4) = YRK(5)
2717      ZRK(4) = ZRK(5)
2718
2719      DXRK(3) = DXRK(4)
2720      DYRK(3) = DYRK(4)
2721      DZRK(3) = DZRK(4)
2722
2723      DXRK(4) = DXRK(5)
2724      DYRK(4) = DYRK(5)
2725      DZRK(4) = DZRK(5)
2726
2727      XRK(5) = CXJ - OTN*(CXJ - PXJ)
2728      YRK(5) = CYJ - OTN*(CYJ - PYJ)
2729      ZRK(5) = CZJ - OTN*(CZJ - PZJ)
2730
2731      PXO = PXJ
2732      PYO = PYJ
2733      PZO = PZJ
2734
2735      CXO = CXJ
2736      CYO = CYJ
2737      CZO = CZJ
2738
2739      IF (ISKIP) 816,.
2740      INDEX3 = INDEX3 - 1

```

```

2741       15KIP = 1
2742       IF GE * XPR, XPR, ZPR AT 0 = 1
2743       AMX = XPR * 5
2744       AMY = XPR * 5
2745       AMZ = ZPR * 5
2746       GO TO 1770
2747       IF GE * XPR, XPR, ZPR AT 0 = 1
2748       1800 CONTINUE
2749       DXPR * 5 = EMX
2750       DYPR * 5 = EMY
2751       DZPR * 5 = EMZ
2752       819 CONTINUE
2753       PXJ = PRED3 * PR * 1, D * PR * 1, D * PR * 1
2754       PYJ = PRED3 * PR * 2, D * PR * 1, D * PR * 1
2755       PZJ = PRED3 * PR * 3, D * PR * 1, D * PR * 1
2756
2757       AMX = CORR3 * PXJ, D * PR
2758       AMY = CORR3 * PYJ, D * PR
2759       AMZ = CORR3 * PZJ, D * PR
2760
2761       15KIP = 1
2762       1770 CONTINUE
2763       XF(INDEX3) = 11 + AMX
2764       YF(INDEX3) = 11 + AMY
2765       ZF(INDEX3) = 11 + AMZ
2766
2767       819 CONTINUE
2768
2769       IF (NI * 6) 1845, 1845, 1
2770       IF 15KIP, 1846, 1846
2771
2772       1845 CONTINUE
2773       SAPROX(INDEX3+1) = SAPROX(INDEX3) + DELTA
2774       IF (MSTOP .LE. INDEX3+1) GO TO 750
2775
2776       C..CHECK IF GO OUT OF BOX
2777
2778       IF (XF(INDEX3).GE.XIKJ) GO TO 750
2779       IF (YF(INDEX3).LE.XIKJ) GO TO 750
2780       IF (YF(INDEX3).GE.YIKJ) GO TO 750
2781       IF (YF(INDEX3).LE.YIKJ) GO TO 750
2782       IF (ZF(INDEX3).GE.ZIKJ) GO TO 750

```

```

2783      IF (ZF(INDEX#3,ILE,Z) .EQ. 0) GO TO 760
2784      GO TO 730
2785
2786      750 CONTINUE
2787      S = SAPROX(INDEX#3)
2788      GO TO 845
2789
2790      000001: IF EXCEED S LENGTH
2791
2792      730 CONTINUE
2793      IF (PASS .GT. 0) GO TO 729
2794      IF (SAPROX(INDEX#3) .GT. 5) GO TO 729
2795      GO TO 1846
2796      732 CONTINUE
2797      IF (S.LE. SAPROX(INDEX#3)) GO TO 845
2798
2799
2800      1846 CONTINUE
2801      IF (ISKIP) GO TO 729,729
2802      IF (PASS.GT.0) GO TO 845
2803      IF (SAPROX(INDEX#3) .GT. TEST) GO TO 729
2804      GO TO 845
2805      840 CONTINUE
2806
2807      IF (TEST-SAPROX(INDEX#3) .LT. 0)
2808
2809      845 CONTINUE
2810
2811      RETURN FOLLOW
2812
2813      ENTRY LINEOUT
2814
2815 C PURPOSE:
2816 C TO PRINT THE COMPRESSED VALUES FOR A FIELD LINE
2817
2818      NOT 3,F9
2819 F9 FORMAT (SAPROX      X      Y      Z
2820      , BXFLD      BYFLD      BZFLO      BMAGF
2821      , DELTA      )
2822
2823      NOT 3, (9E12,3) , (SAPROX(I),XF(I),YF(I),ZF(I),BXFLD(I),BYFLD(I),BZFLO(I),
2824      , BMAGF(I),DELMAX(MFI),J=1,NSTORE)

```

```

2825
2826         RETURN LINEOUT
2827
2828         ENTRY FLOLNIN
2829
2830
2831
2832 C  PURPOSE:  TO READ  INFO FOR FIELD LINE CALCULATION AND PLOTTING
2833
2834 C  INPUT:
2835 C  MFIELD: NO. OF FIELD LINES TO PROCESS. MAX OF '0
2836 C          INPUT ON FIRST DATA CARD; SEE DATAMAFCO OR DATAABCOXYZ
2837 C          SUBROUTINES
2838 C  XBAR,YBAR,ZBAR:  3-D ARRAYS CONTAINING INITIAL COORDINATES OF FIELD LINE
2839 C  DELMAX:  1-D ARRAY CONTAINING DELTA S TO USE FOR EACH FIELD LINE
2840 C  STOTAL:  1-D ARRAY CONTAINING S LENGTH FOR EACH FIELD LINE,OPTIONAL
2841 C  NSKIP:  1-D ARRAY CONTAINING NO. POINTS OT SKIP TO CREATE COMPRESSED
2842 C          ARRAY TO BE SAVED IN LCM;COMPRESSED SHOULD HAVE .LE. 100 POINTS
2843 C          PER FIELD LINE
2844
2845
2846         WOT 3, ( 1NO. OF FIELD LINES TO PLOT AND COMPUTE= ,15) ,MFIELD
2847
2848         WOT 3,F10
2849 F10  FORMAT( 0XBAR          YBAR          ZBAR          DELMAX          STOTAL          NSKIP )
2850
2851         WOT 3, (5F10.3,110) ,(XBAR(1),YBAR(1),ZBAR(1),DELMAX(1),
2852         .      STOTAL(1),NSKIP(1),)=1,MFABS)
2853
2854         RETURN FLOLNIN
2855
2856         ENTRY RUGKUT(DIZ)
2857
2858
2859         XIN = XF(INDEX3)
2860         YIN = YF(INDEX3)
2861         ZIN = ZF(INDEX3)
2862
2863         BXFTEMP = 0XFLO(INDEX3)
2864         BYFTEMP = 0YFLO(INDEX3)
2865         BZFTEMP = 0ZFLO(INDEX3)
2866         BMAGTEMP = 0MAGF(INDEX3)

```

```

2867
2868 10 KEY = KEY + 1
2869
2870 DBMAG1 = DIZ/BMAGTEMP
2871
2872 CAXX(KEY) = DBMAG1*BXFTEMP
2873 ELXX(KEY) = DBMAG1*BYFTEMP
2874 EMXX(KEY) = DBMAG1*BZFTEMP
2875
2876 IF (KEY.GE 4) GO TO 150
2877
2878 XO = XIN + CAXX(KEY) * COEF1KEY
2879 YO = YIN + ELXX(KEY) * COEF1KEY
2880 ZO = ZIN + EMXX(KEY) * COEF1KEY
2881
2882 90 CONTINUE
2883
2884 CALL FIELDS
2885
2886 SBXTEMP = SBX
2887 SBYTEMP = SBY
2888 SBZTEMP = SBZ
2889 BMAGTEMP = SQRT((SBX**2+SBY**2+SBZ**2)
2890
2891 801 CONTINUE
2892 IF (KEY.LT.4) GO TO 10
2893
2894 150 CONTINUE
2895 DELX = SIXTH*(CAXX(1)+2.0*CAXX(2)+2.0*CAXX(3)+CAXX(4))
2896 DELY = SIXTH*(ELXX(1)+2.0*ELXX(2)+2.0*ELXX(3)+ELXX(4))
2897 DELZ = SIXTH*(EMXX(1)+2.0*EMXX(2)+2.0*EMXX(3)+EMXX(4))
2898
2899 XRK(NIX) = XF(INDEX3+1) = XIN + DELX
2900 YRK(NIX) = YF(INDEX3+1) = YIN + DELY
2901 ZRK(NIX) = ZF(INDEX3+1) = ZIN + DELZ
2902
2903 DXRK(NIX) = DELX
2904 DYRK(NIX) = DELY
2905 DZRK(NIX) = DELZ
2906
2907 NIX = NIX + 1
2908

```

```

2909      RETURN RUGKUT
2910
2911      ENTRY COMPRESS
2912
2913 C..PURPOSE: TO COMPRESS THE NUMBER OF POINTS CALCULATED TO LELE100
2914 C POINTS AND SAVE THEM IN LCM FOR PLOTTING. IF
2915 C NSKIP(MF) IS LE 100 USE IT'S VALUE
2916
2917      NSTORE = INDEX*3
2918
2919      IF INDEX*3 LE 100 GO TO 500
2920
2921      NPASS = INDEX*3/100 + 1
2922
2923      IF NSKIP(MF) GE NPASS NPASS = NSKIP(MF)
2924      NSTORE = 0
2925      NGET = 1
2926
2927 100 NGET = NGET + NPASS
2928      IF (NGET*LEINDEX)31 GO TO 151
2929      NGET = INDEX*3
2930
2931 151 CONTINUE
2932      NSTORE = NSTORE + 1
2933      XF(NSTORE) = XF(NGET)
2934      YF(NSTORE) = YF(NGET)
2935      ZF(NSTORE) = ZF(NGET)
2936      BMAGF(NSTORE) = BMAGF(NGET)
2937      SAPROX(NSTORE) = SAPROX(NGET)
2938      BXFLD(NSTORE) = BXFLD(NGET)
2939      BYFLD(NSTORE) = BYFLD(NGET)
2940      BZFLD(NSTORE) = BZFLD(NGET)
2941      IF (NGET*LTINDEX)31 GO TO 100
2942
2943 500 CONTINUE
2944      WRITE LCM (XF(1),YFLC(LCFWA(MF)),NSTORE)
2945      WRITE LCM (YF(1),YFLC(LCFWA(MF)),NSTORE)
2946      WRITE LCM (ZF(1),ZFLC(LCFWA(MF)),NSTORE)
2947      WRITE LCM (BMAGF(1),BFLC(LCFWA(MF)),NSTORE)
2948      WRITE LCM (SAPROX(1),SFLC(LCFWA(MF)),NSTORE)
2949
2950      LCFWA(MF*1) = LCFWA(MF) + NSTORE

```



```

2951      LCNUM(MF) = NSTORE
2952
2953
2954      RETURN COMPRESS
2955
2956
2957      ENTRY LINCRTALL
2958
2959 C...TO SUPERIMPOSE THE 4 LINE PLOTS FOR ALL THE FIELD LINES THAT ARE
2960 C...  CALCULATED
2961
2962      LSTWD = LCFWA(MF+1) - 1
2963
2964      READ LCM (XF(1),XF(2),LSTWD)
2965      READ LCM (YF(1),YF(2),LSTWD)
2966      READ LCM (ZF(1),ZF(2),LSTWD)
2967      READ LCM (BMAGF(1),BF(1),LSTWD)
2968      READ LCM (SAPROX(1),SF(1),LSTWD)
2969
2970      CALL AMINMX (BMAGF,1,LSTWD,1,BMIN,BMAX)
2971      CALL AMINMX (SAPROX,1,LSTWD,1,SMIN,SMAX)
2972      IMF = 1
2973      GO TO 52
2974
2975      ENTRY LINCRTI
2976
2977 C  PURPOSE: TO MAKE FOUR PLOTS FOR THE CURRENT FIELD LINE, IN THE
2978 C  SINGLE LINE PER FRAME MODE AS REQUESTED BY THE MFIELD FLAG
2979
2980 C  1.  X VS Y AT Z = 0
2981 C  2.  Y VS Z AT X = 0
2982 C  3.  X VS Z AT Y = 0
2983 C  4.  S VS B
2984
2985
2986 5      FORMAT (              FIELD LINE PLOT )
2987 6      FORMAT (              FIELD LINE PLOT AT Z = 0 )
2988 7      FORMAT (              FIELD LINE PLOT AT Y = 0 )
2989 6      FORMAT (              FIELD LINE PLOT AT X = 0 )
2990
2991 110      FORMAT (              MOD B ALONG FIELD LINE PLOT )
2992 111      FORMAT ( ARC LENGTH FROM THE STARTING POSITION )

```

```

2993 112 FORMAT ( STARTING POSITION OF THE FIELD LINE :
2994
2995 20 FORMAT ( X(CM) )
2996 30 FORMAT ( Y(CM) )
2997 40 FORMAT ( Z(CM) )
2998 50 FORMAT ( X Y Z DELTA S NSKIP :
2999 51 FORMAT (5F10.3,15)
3000
3001 IMF = MF
3002
3003 CALL AMINMX (BMAGF,1,NSTORE,1,BMIN,BMAX)
3004 CALL AMINMX (SAPROX,1,NSTORE,1,SMIN,SMAX)
3005 52 CONTINUE
3006
3007 C..DO BMOD ALONG FIELD LINE PLOT
3008 CALL FRAME
3009
3010 CALL SETCH (25.0,1,0,1,0,1,0,0)
3011 WOT 100,111
3012 CALL SETCH (16.0,9,0,1,0,0,0,0)
3013 WOT 100,112
3014
3015 CALL SETCH (16.0,7,0,1,0,0,0,0)
3016 WOT 100,50
3017 CALL SETCH (20.0,6,0,1,0,0,0,0)
3018 WOT 100,51,(XBAR(1),YBAR(1),ZBAR(1),DELMAX(1),
3019 . STOTAL(1),NSKIP(1),I=IMF,MF)
3020
3021 CALL SETCH (2.0,60,0,1,0,1,0,0)
3022 WOT 100,5
3023 CALL SETCH (2.0,25,0,1,0,1,1,0)
3024 WOT 100,110
3025
3026 CALL MAPS (SMIN,SMAX,BMIN,BMAX,.12,.90,.30,.90)
3027
3028 DO 56 J = IMF,MF
3029 56 CALL TRACE (SAPROX(LCFWA(J)),BMAGF(LCFWA(J)),LCNUM(J))
3030
3031 C..DO X VS Y PLOT AT Z = 0
3032 CALL FRAME
3033
3034 CALL SETCH (16.0,7,0,1,0,0,0,0)

```

```

3035      WOT 100,50
3036      CALL SETCH (20.0,6.0,1.0,0.0,0)
3037      WOT 100,51,(XBAR(1),YBAR(1),ZBAR(1),DELMAX(1),
3038      . STOTAL(1),NSKIP(1),I=IMF,MF)
3039      CALL SETCH (2.0,60.0,1.0,1.0,0)
3040      WOT 100,6
3041      CALL SETCH (30.0,11.0,1.0,1.0,0)
3042      WOT 100,20
3043      CALL SETCH (16.0,9.0,1.0,0.0,0)
3044      WOT 100,112
3045      CALL SETCH (2.0,30.0,1.0,1.1,0)
3046      WOT 100,30
3047
3048      CALL MAPS (X(1),X(KJ),Y(1),Y(KJ),,12,.90,.30,.90)
3049
3050      DO 57 I = IMF,MF
3051 57    CALL TRACE (XF(LCFWA(I)),YF(LCFWA(I)),LCNUM(I))
3052
3053 C...DO Y VS Z AT X = J
3054      CALL FRAME
3055
3056      CALL SETCH (16.0,7.0,1.0,0.0,0)
3057      WOT 100,50
3058      CALL SETCH (20.0,6.0,1.0,0.0,0)
3059      WOT 100,51,(XBAR(1),YBAR(1),ZBAR(1),DELMAX(1),
3060      . STOTAL(1),NSKIP(1),I=IMF,MF)
3061
3062      CALL SETCH (2.0,60.0,1.0,1.0,0)
3063      WOT 100,8
3064      CALL SETCH (30.0,11.0,1.0,1.0,0)
3065      WOT 100,20
3066      CALL SETCH (16.0,9.0,1.0,0.0,0)
3067      WOT 100,112
3068      CALL SETCH (2.0,30.0,1.0,1.1,0)
3069      WOT 100,40
3070
3071      CALL MAPS (Y(1),Y(KJ),Z(1),Z(KJ),,12,.90,.30,.90)
3072
3073      DO 58 I = IMF,MF
3074 58    CALL TRACE (YF(LCFWA(I)),ZF(LCFWA(I)),LCNUM(I))
3075
3076 C...DO X VS Z AT Y = 0 PLOT

```

```

3077
3078      CALL FRAME
3079
3080      CALL SETCH (16.0,7.0,1,0,0,0,0)
3081      WOT 100,50
3082      CALL SETCH (20.0,6.0,1,0,0,0,0)
3083      WOT 100,51,(XBAR(1),YBAR(1),ZBAR(1),DELMAX(1),
3084      STOTAL(1),NSKIP(1),I=IMF,MF)
3085
3086      CALL SETCH (2.0,60.0,1,0,1,0,0)
3087      WOT 100,7
3088      CALL SETCH (30.0,11.0,1,0,1,0,0)
3089      WOT 100,20
3090      CALL SETCH (16.0,9.0,1,0,0,0,0)
3091      WOT 100,112
3092      CALL SETCH (2.0,30.0,1,0,1,1,0)
3093      WOT 100,40
3094
3095      CALL MAPS (X(1),X(K1),Z(1),Z(KK),.12,.90,.30,.90)
3096
3097      DO 59 I = IMF,MF
3098 59      CALL TRACE (XF(LCFWA(I)),ZF(LCFWA(I)),LCNUM(I))
3099
3100 C..DO R VS Z PLOT
3101      CALL FRAME
3102
3103      CALL SETCH (16.0,7.0,1,0,0,0,0)
3104      WOT 100,50
3105      CALL SETCH (20.0,6.0,1,0,0,0,0)
3106      WOT 100,51,(XBAR(1),YBAR(1),ZBAR(1),DELMAX(1),
3107      STOTAL(1),NSKIP(1),I=IMF,MF)
3108
3109      CALL SETCH (2.0,60.0,1,0,1,0,0)
3110      WOT 100, ( FIELD LINE PLOT OF R VS Z )
3111
3112      CALL SETCH (30.0,11.0,1,0,1,0,0)
3113      WOT 100, ( Z(CM) )
3114
3115      CALL SETCH (16.0,9.0,1,0,0,0,0)
3116      WOT 100,112
3117
3118

```

```

3119      CALL SETCH (2.0,30.0,1, 0, 1, 1, 0)
3120      WDT 100, (          R(CM) )
3121
3122      CALL AMINMX (XF,1,LSTWD,1,XMIN,XMAX)
3123      CALL AMINMX (YF,1,LSTWD,1,YMIN,YMAX)
3124      RMAX = AMAX1 (XMAX-XMIN, YMAX-YMIN)
3125
3126      CALL MAPS (Z(1),Z(KK),0.0,RMAX,.12,.90,.30,.90)
3127
3128      DO 79 I = 1,MF,MF
3129      DO 78 J = 1,LCNUM(1)
3130      R(J) = SQRT1 (XF(LCFWA(1)+J-1)**2+YF(LCFWA(1)+J-1)**2
3131      ,
3132      +ZF(LCFWA(1)+J-1)**2)
3132 78      CONTINUE
3133
3134      CALL TRACE (ZF(LCFWA(1)),R(1),LCNUM(1))
3135 79      CONTINUE
3136
3137      CALL PLOTEA
3138
3139
3140      RETURN
3141
3142
3143      END

```

```

3144      SUBROUTINE CRPLOT (ARRAY,IHOLD, IABSC,IORDIN,X,Y,Z)
3145
3146      LCM ARRAY
3147
3148      CODE ANALYSIS
3149
3150      OPTIMIZE
3151
3152 C..PURPOSE: THIS ROUTINE WILL DO CONTOUR PLOTS OF THE 3D ARRAY IN
3153 C..THE PLANE SPECIFIED BY THE PARAMETER IHOLD. THE CONTOURS PLOTTED
3154 C..AND THE ARRAY CPLT. THE TITLES FOR THE ABSCISSA AND ORDINATE ARE
3155 C..SPECIFIED IN THE ARRAYS IABSC AND IORDIN.
3156
3157 C..INPUT: IHOLD BIT (58-59) 1=PLOT YZ PLANE,2=PLOT XZ PLANE 3=PLOT XY PLANE
3158 C          BIT (0-5) INDEX VALUE OF CONSTANT PLANE TO BE PLOTTED
3159 C..          IN LTSS 304 FOR CONTOUR PLOTTING
3160 C..IABSC,IORDIN DEFINED AS FOLLOWS FOR TITLES (80 CHAR)
3161 C          DATA IABSC/ ABSCISSA TITLE / DATA IORDIN/ ORDINATE TITLE /
3162
3163 C..COMMON BLOCK INPUT:
3164 C..X,Y,Z GRID ARRAYS
3165 C..SCAL(1) ORDINATE MIN.
3166 C..SCAL(2) ORDINATE MAX. (IF MIN=MAX=0 USE FUNCTION MAX AND MIN)
3167 C..SCAL(3) ABSCISSA MIN.
3168 C..SCAL(4) ABSCISSA MAX. (IF MIN=MAX=0 USE ARGUMENT MAX AND MIN)
3169 C..NFR 1=DO FRAME ADVANCE,PLOT TITLES,DO MAPPING; 0=SUPERIMPOSE PLOTS
3170
3171
3172 C..PARAMETER DECLARATIONS
3173
3174
3175
3176      MACRO DIMM (MACROS)
3177      USE DIMM
3178
3179      COMMON /CPLOT/ SCAL(4), NFR, DOWNER, UPPER, LEVL
3180
3181
3182      DIMENSION ARRAY (IMAX,JMAX,KMAX),CPLT(40),IABSC(2),IORDIN(2),
3183      . PLOTT(MAXDM,MAXDM),GRIDABS(1),GRIDORD(1),
3184      . X(2),Y(2),Z(2)
3185

```

```

3186     POINTER (VA,GRIDABS)
3187     POINTER (VO,GRIDORD)
3188
3189     BYTE NPLANE(2),IDUM(52),NGRID(6)
3190     STRUCTURE (IHOLD,NPLANE/IDUM/NGRID)
3191
3192     CALL FRAME
3193
3194 C..SET UP JUMP FOR PLANE TO PLOT
3195
3196 C..STORE SUB-ARRAY FOR CONTOUR PLOT
3197     GO TO (100,200,300),NPLANE
3198 100  CONTINUE
3199     TEMP = X(NGRID)
3200     IABSC(6) = X(CM)=
3201
3202
3203     DO 110 K = 1,KK
3204     DO 110 J = 1,KJ
3205     PLOTT(J,K) = ARRAY(NGRID,J,K)
3206
3207
3208 110  CONTINUE
3209     GO TO 400
3210
3211 200  CONTINUE
3212     TEMP = Y(NGRID)
3213     IABSC(6) = Y(CM)=
3214
3215
3216     DO 210 K = 1,KK
3217     DO 210 I = 1,KI
3218     PLOTT(I,K) = ARRAY(I,NGRID,K)
3219
3220
3221 210  CONTINUE
3222     GO TO 400
3223
3224 300  CONTINUE
3225     TEMP = Z(NGRID)
3226     IABSC(6) = Z(CM)=
3227

```

```

3228
3229
3230      DO 310 J = 1,KJ
3231      DO 300 I = 1,KI
3232      PLOT(I,J) = ARRAY(I,J,NGRID)
3233
3234
3235 310  CONTINUE
3236
3237 C..SET MAPPING PARAMETERS
3238 400  CONTINUE
3239
3240 C..SET CONTOUR LEVELS
3241      IF (UPPER.EQ.DOWNER) RETURN CRPLOT
3242      IF (ABS(UPPER-DOWNER).LT.1.E-12*(ABS(UPPER)+ABS(DOWNER))) RETURN
3243      CRPLOT
3244      FAC = (UPPER-DOWNER)/LEVL
3245      NA = 0
3246      NB = LEVL
3247
3248      IF (DOWNER .GE. 0.0 ) GO TO 410
3249      NA = -LEV /2
3250      NB = LEVL/2 + 1
3251
3252 410  CONTINUE
3253
3254      IF ((SCAL(1).NE.0.0).OR.(SCAL(2).NE.0.0)) GO TO 440
3255      IF (NPLANE -1) 430,
3256 420  CONTINUE
3257      SCAL1 = Y(I)
3258      SCAL2 = Y(KJ)
3259      GO TO 450
3260
3261 430  CONTINUE
3262      SCAL1 = X(I)
3263      SCAL2 = X(KI)
3264      GO TO 450
3265
3266 440  CONTINUE
3267      SCAL1 = SCAL(1)
3268      SCAL2 = SCAL(2)
3269

```



```

3312 520 CONTINUE
3313 IF (NPLANE-3) ,540
3314 VO = .LOC. Z
3315
3316 MAXORD = KK
3317 GO TO 550
3318
3319 540 CONTINUE
3320 VO = .LOC. Y
3321
3322 MAXORD = KJ
3323 DELABS = GRIDABS(2) - GRIDABS(1)
3324 DELORD = GRIDORD(2) - GRIDORD(1)
3325
3326 C..DO CONTOUR PLOT
3327 550 CONTINUE
3328 DELABS = GRIDABS(2) - GRIDABS(1)
3329 DELORD = GRIDORD(2) - GRIDORD(1)
3330
3331 CALL SETCH (1.0, 40., 0, 0, 0, 0, 0)
3332 WOT 100, ( CONTOUR LEVEL )
3333
3334
3335 JCON = 0
3336 IF (DOWNER.LT.0) JCON=10
3337
3338 DO CP J = NA,NB
3339 WOT 100, (14,E12.2) ,J,(J+JCON*1.E-6)*FAC+DOWNER
3340 CP CONTINUE
3341
3342 SETPCH(1,0,0,0,200)
3343
3344 DO CP1 J = NA,NB
3345 CP1 CALL CONTUR (PLOT(1,1),1,MAXABS,1,MAXORD,MAXDM,(J+JCON*1.E-6)
3346 , *FAC+DOWNER,PLOTCP)
3347 RETURN CRPLOT
3348
3349 ENTRY PLOTCP (XCP,YCP,NCP)
3350 REAL XCP(NCP),YCP(NCP)
3351
3352 DO CP2 JJ = 1,NCP
3353 XCP(JJ) = XCP(JJ)*DELABS + GRIDABS(1) -DELABS

```

```

3354 CP2 YCP(JJ) = YCP(JJ)*DELORD + GRIDORD(I) - DELORD
3355
3356 IF (J.LT.0) GO TO CP3
3357 TRACE (XCP,YCP,NCP)
3358 RETURN PLOTCP
3359
3360 CP3 CONTINUE
3361 TRACEP ( XCP,YCP,NCP)
3362 RETURN PLOTCP
3363
3364 END

```

Appendix D: Input for 2XIIB Sample Problem

```

1 1BOX = 74BOX 582 1OUTFIL = 5HTX11V
2 NDOPT = 5HMAFCO 1AFLO = 1 18FLD = 1
3 DIA = .01 KI = 11 KJ = 3 KK = 11 DELX = 8. DELY = 2. DELZ = 8.
4 XBOT = -40. YBOT = -2. ZBOT = -40.
5 NLOOP = 70 NARC = 80
6 BMAXPLT = .2E+5 AymaxPLT = .5E+6
7 BMINPLT = .9E+4 AymINPLT = -.5E+6
8 IGR = 0 0 0 0 0 MF1ELO = -3
9 XBAR = 4. 8. 12. YBAR = 0. 0. 0. ZBAR = 0. 0. 0. DELMAX = .1 .1 .1
10 NGR = 0 2 0 STOP
11 0. 0. -390.000 96.143 0. 0. 22171.000 1
12 0. 0. -390.000 104.589 0. 0. 22171.000 2
13 0. 0. -390.000 113.035 0. 0. 22171.000 3
14 0. 0. -390.000 121.481 0. 0. 22171.000 4
15 0. 0. -390.000 129.927 0. 0. 22171.000 5
16 0. 0. -370.000 96.143 0. 0. 22171.000 6
17 0. 0. -370.000 104.589 0. 0. 22171.000 7
18 0. 0. -370.000 113.035 0. 0. 22171.000 8
19 0. 0. -370.000 121.481 0. 0. 22171.000 9
20 0. 0. -370.000 129.927 0. 0. 22171.000 10
21 0. 0. -350.000 96.143 0. 0. 22171.000 11
22 0. 0. -350.000 104.589 0. 0. 22171.000 12
23 0. 0. -350.000 113.035 0. 0. 22171.000 13
24 0. 0. -350.000 121.481 0. 0. 22171.000 14
25 0. 0. -350.000 129.927 0. 0. 22171.000 15
26 0. 0. -285.000 96.143 0. 0. 34887.000 16
27 0. 0. -285.000 104.589 0. 0. 34887.000 17
28 0. 0. -285.000 113.035 0. 0. 34887.000 18
29 0. 0. -285.000 121.481 0. 0. 34887.000 19
30 0. 0. -285.000 129.927 0. 0. 34887.000 20
31 0. 0. -210.000 96.143 0. 0. 34328.000 21
32 0. 0. -210.000 104.589 0. 0. 34328.000 22
33 0. 0. -210.000 113.035 0. 0. 34328.000 23
34 0. 0. -210.000 121.481 0. 0. 34328.000 24
35 0. 0. -210.000 129.927 0. 0. 34328.000 25
36 0. 0. -140.000 96.143 0. 0. 17164.000 26
37 0. 0. -140.000 104.589 0. 0. 17164.000 27
38 0. 0. -140.000 113.035 0. 0. 17164.000 28
39 0. 0. -140.000 121.481 0. 0. 17164.000 29
40 0. 0. -140.000 129.927 0. 0. 17164.000 30
41 0. 0. -70.000 133.653 0. 0. 76285.000 31
42 0. 0. -70.000 142.099 0. 0. 76285.000 32

```

43	0.	0.	-70.000	150.545	0.	0.	76285.00C	33
44	0.	0.	-70.000	158.991	0.	0.	76285.00C	34
45	0.	0.	-70.000	167.437	0.	0.	76285.00C	35
46	0.	0.	70.000	133.653	0.	0.	76285.00C	36
47	0.	0.	70.000	142.099	0.	0.	76285.00C	37
48	0.	0.	70.000	150.545	0.	0.	76285.00C	38
49	0.	0.	70.000	158.991	0.	0.	76285.00C	39
50	0.	0.	70.000	167.437	0.	0.	76285.00C	40
51	0.	0.	140.000	96.143	0.	0.	17164.00C	41
52	0.	0.	140.000	104.589	0.	0.	17164.00C	42
53	0.	0.	140.000	113.035	0.	0.	17164.00C	43
54	0.	0.	140.000	121.481	0.	0.	17164.00C	44
55	0.	0.	140.000	129.927	0.	0.	17164.00C	45
56	0.	0.	210.000	96.143	0.	0.	34328.00C	46
57	0.	0.	210.000	104.589	0.	0.	34328.00C	47
58	0.	0.	210.000	113.035	0.	0.	34328.00C	48
59	0.	0.	210.000	121.481	0.	0.	34328.00C	49
60	0.	0.	210.000	129.927	0.	0.	34328.00C	50
61	0.	0.	285.000	96.143	0.	0.	34887.00C	51
62	0.	0.	285.000	104.589	0.	0.	34887.00C	52
63	0.	0.	285.000	113.035	0.	0.	34887.00C	53
64	0.	0.	285.000	121.481	0.	0.	34887.00C	54
65	0.	0.	285.000	129.927	0.	0.	34887.00C	55
66	0.	0.	350.000	96.143	0.	0.	22171.00C	56
67	0.	0.	350.000	104.589	0.	0.	22171.00C	57
68	0.	0.	350.000	113.035	0.	0.	22171.00C	58
69	0.	0.	350.000	121.481	0.	0.	22171.00C	59
70	0.	0.	350.000	129.927	0.	0.	22171.00C	60
71	0.	0.	370.000	96.143	0.	0.	22171.00C	61
72	0.	0.	370.000	104.589	0.	0.	22171.00C	62
73	0.	0.	370.000	113.035	0.	0.	22171.00C	63
74	0.	0.	370.000	121.481	0.	0.	22171.00C	64
75	0.	0.	370.000	129.927	0.	0.	22171.00C	65
76	0.	0.	390.000	96.143	0.	0.	22171.00C	66
77	0.	0.	390.000	104.589	0.	0.	22171.00C	67
78	0.	0.	390.000	113.035	0.	0.	22171.00C	68
79	0.	0.	390.000	121.481	0.	0.	22171.00C	69
80	0.	0.	390.000	129.927	0.	0.	22171.00C	70
81	41.043	0.	8.915	59.425	90.000	77.745	-100000.0Y6	3
82	0.	180.000	8.915	61.775	90.000	77.745	-100000.0Y6	4
83	41.043	0.	0.	0.	0.	0.	0.	5
84	0.	0.	0.	0.	0.	0.	0.	6

85	41.043	0.	8.915	64.125	90.000	77.745	-100000.0YG	7
86	0.	180.000						8
87	41.043	0.	8.915	66.475	90.000	77.745	-100000.0YG	9
88	0.	180.000						10
89	41.043	0.	8.915	68.825	90.000	77.745	-100000.0YG	11
90	0.	180.000						12
91	41.043	0.	8.915	71.175	90.000	77.745	-100000.0YG	13
92	0.	180.000						14
93	41.043	0.	8.915	73.525	90.000	77.745	-100000.0YG	15
94	0.	180.000						16
95	41.043	0.	8.915	75.875	90.000	77.745	-100000.0YG	17
96	0.	180.000						18
97	41.043	0.	8.915	78.225	90.000	77.745	-100000.0YG	19
98	0.	180.000						20
99	41.043	0.	8.915	80.575	90.000	77.745	-100000.0YG	21
100	0.	180.000						22
101	-41.043	0.	8.915	53.425	90.000	102.255	100000.0YG	23
102	0.	180.000						24
103	-41.043	0.	8.915	61.775	90.000	102.255	100000.0YG	25
104	0.	180.000						26
105	-41.043	0.	8.915	64.125	90.000	102.255	100000.0YG	27
106	0.	180.000						28
107	-41.043	0.	8.915	66.475	90.000	102.255	100000.0YG	29
108	0.	180.000						30
109	-41.043	0.	8.915	68.825	90.000	102.255	100000.0YG	31
110	0.	180.000						32
111	-41.043	0.	8.915	71.175	90.000	102.255	100000.0YG	33
112	0.	180.000						34
113	-41.043	0.	8.915	73.525	90.000	102.255	100000.0YG	35
114	0.	180.000						36
115	-41.043	0.	8.915	75.875	90.000	102.255	100000.0YG	37
116	0.	180.000						38
117	-41.043	0.	8.915	78.225	90.000	102.255	100000.0YG	39
118	0.	180.000						40
119	-41.043	0.	8.915	80.575	90.000	102.255	100000.0YG	41
120	0.	180.000						42
121	0.	59.425	0.	42.000	0.	90.000	-100000.0YG	43
122	167.745	372.255						44
123	0.	61.775	0.	42.000	0.	90.000	-100000.0YG	45
124	167.745	372.255						46
125	0.	64.125	0.	42.000	0.	90.000	-100000.0YG	47
126	167.745	372.255						48

127	0.	66.475	0.	42.000	0.	90.000	-100000.0YG	49
128	167.745	372.255						50
129	0	68.825	0.	42.000	0.	90.000	-100000.0YG	51
130	167.745	372.255						52
131	0.	71.175	0.	42.000	0.	90.000	-100000.0YG	53
132	167.745	372.255						54
133	0.	73.525	0.	42.000	0.	90.000	-100000.0YG	55
134	167.745	372.255						56
135	0.	75.875	0.	42.000	0.	90.000	-100000.0YG	57
136	167.745	372.255						58
137	0.	78.225	0.	42.000	0.	90.000	-100000.0YG	59
138	167.745	372.255						60
139	0.	80.575	0.	42.000	0.	90.000	-100000.0YG	61
140	167.745	372.255						62
141	0.	-59.425	0.	42.000	0.	90.000	100000.0YG	63
142	167.745	372.255						64
143	0.	-61.775	0.	42.000	0.	90.000	100000.0YG	65
144	167.745	372.255						66
145	0.	-64.125	0.	42.000	0.	90.000	100000.0YG	67
146	167.745	372.255						68
147	0.	-66.475	0.	42.000	0.	90.000	100000.0YG	69
148	167.745	372.255						70
149	0.	-68.825	0.	42.000	0.	90.000	100000.0YG	71
150	167.745	372.255						72
151	0.	-71.175	0.	42.000	0.	90.000	100000.0YG	73
152	167.745	372.255						74
153	0.	-73.525	0.	42.000	0.	90.000	100000.0YG	75
154	167.745	372.255						76
155	0.	-75.875	0.	42.000	0.	90.000	100000.0YG	77
156	167.745	372.255						78
157	0.	-78.225	0.	42.000	0.	90.000	100000.0YG	79
158	167.745	372.255						80
159	0.	-80.575	0.	42.000	0.	90.000	100000.0YG	81
160	167.745	372.255						82
161	0.	-41.043	-8.915	59.425	0.	77.745	-100000.0YN	83
162	180.000	360.000						84
163	0.	-41.043	-8.915	61.775	0.	77.745	-100000.0YN	85
164	180.000	360.000						86
165	0.	-41.043	-8.915	64.125	0.	77.745	-100000.0YN	87
166	180.000	360.000						88
167	0.	-41.043	-8.915	66.475	0.	77.745	-100000.0YN	89
168	180.000	360.000						90

169	0.	41.043	-8.915	68.825	0.	77.745	-100000.0YN	91
170	180.000	360.000						92
171	0.	41.043	-8.915	71.175	0.	77.745	-100000.0YN	93
172	180.000	360.000						94
173	0.	41.043	-8.915	73.525	0.	77.745	-100000.0YN	95
174	180.000	360.000						96
175	0.	41.043	-8.915	75.875	0.	77.745	-100000.0YN	97
176	180.000	360.000						98
177	0.	41.043	-8.915	78.225	0.	77.745	-100000.0YN	99
178	180.000	360.000						100
179	0.	41.043	-8.915	80.575	0.	77.745	-100000.0YN	101
180	180.000	360.000						102
181	0.	41.043	-8.915	82.925	0.	102.255	-100000.0YN	103
182	180.000	360.000						104
183	0.	41.043	-8.915	85.275	0.	102.255	-100000.0YN	105
184	180.000	360.000						106
185	0.	41.043	-8.915	87.625	0.	102.255	-100000.0YN	107
186	180.000	360.000						108
187	0.	41.043	-8.915	89.975	0.	102.255	-100000.0YN	109
188	180.000	360.000						110
189	0.	41.043	-8.915	92.325	0.	102.255	-100000.0YN	111
190	180.000	360.000						112
191	0.	41.043	-8.915	94.675	0.	102.255	-100000.0YN	113
192	180.000	360.000						114
193	0.	41.043	-8.915	97.025	0.	102.255	-100000.0YN	115
194	180.000	360.000						116
195	0.	41.043	-8.915	99.375	0.	102.255	-100000.0YN	117
196	180.000	360.000						118
197	0.	41.043	-8.915	101.725	0.	102.255	-100000.0YN	119
198	180.000	360.000						120
199	0.	41.043	-8.915	104.075	0.	102.255	-100000.0YN	121
200	180.000	360.000						122
201	59.425	0.	-0.	42.000	90.000	90.000	-100000.0YN	123
202	-12.255	192.255						124
203	61.775	0.	-0.	42.000	90.000	90.000	-100000.0YN	125
204	-12.255	192.255						126
205	64.125	0.	-0.	42.000	90.000	90.000	-100000.0YN	127
206	-12.255	192.255						128
207	66.475	0.	-0.	42.000	90.000	90.000	-100000.0YN	129
208	-12.255	192.255						130
209	68.825	0.	-0.	42.000	90.000	90.000	-100000.0YN	131
210	-12.255	192.255						132

211	71.175	0.	-0.	42.000	90.000	90.000	-100000.0YN	133
212	-12.255	192.255						134
213	73.525	0.	-0.	42.000	90.000	90.000	-100000.0YN	135
214	-12.255	192.255						136
215	75.875	0.	-0.	42.000	90.000	90.000	-100000.0YN	137
216	-12.255	192.255						138
217	78.225	0.	-0.	42.000	90.000	90.000	-100000.0YN	139
219	-12.255	192.255						140
219	80.575	0.	-0.	42.000	90.000	90.000	-100000.0YN	141
220	-12.255	192.255						142
221	-59.425	0.	-0.	42.000	90.000	90.000	100000.0YN	143
222	-12.255	192.255						144
223	-61.775	0.	-0.	42.000	90.000	90.000	100000.0YN	145
224	-12.255	192.255						146
225	-64.125	0.	-0.	42.000	90.000	90.000	100000.0YN	147
226	-12.255	192.255						148
227	-66.475	0.	-0.	42.000	90.000	90.000	100000.0YN	149
228	-12.255	192.255						150
229	-68.825	0.	-0.	42.000	90.000	90.000	100000.0YN	151
230	-12.255	192.255						152
231	-71.175	0.	-0.	42.000	90.000	90.000	100000.0YN	153
232	-12.255	192.255						154
233	-73.525	0.	-0.	42.000	90.000	90.000	100000.0YN	155
234	-12.255	192.255						156
235	-75.875	0.	-0.	42.000	90.000	90.000	100000.0YN	157
236	-12.255	192.255						158
237	-78.225	0.	-0.	42.000	90.000	90.000	100000.0YN	159
238	-12.255	192.255						160
239	-80.575	0.	-0.	42.000	90.000	90.000	100000.0YN	161
240	-12.255	192.255						162
241 END								