X-640-66-313

NASA TM X-55615

A PROPOSAL FOR TRAPPED PARTICLE FLUX MAPPING IN THE OUTER MAGNETOSPHERE

BY

JUAN G. ROEDERER

JUNE 1966

GODDARD SPACE FLIGHT CENTER

GREENBELT, MARYLAND

Rq/N2349

X-640-66-313

A PROPOSAL FOR TRAPPED PARTICLE FLUX MAPPING IN THE OUTER MAGNETOSPHERE

by

Juan G. Roederer*

June 1966

*National Academy of Sciences–National Research Council Senior Post Doctoral Research Associate

Present address: Centro Nacional de Radiacion Cosmica, Peru 272, Buenos Aires, Argentina

A PROPOSAL FOR TRAPPED PARTICLE FLUX MAPPING
IN THE OUTER MAGNETOSPHERE

by

Juan G. Roederer

N67-13163

SUMMARY

A computer code is described which can be used for a two dimensional mapping of directional fluxes of energetic particles trapped in the outer magnetosphere. G. Mead's model of the magnetospheric field is used. The main function of the code is to find, on a fixed reference meridian, the field line, equatorial pitch angle and other parameters of the shell generated by the particles in their longitudinal drift, provided they are stably trapped.

A series of curves is given, which can be used for a first rough, graphical attempt of flux mapping.

The code is described in more detail in the Appendix.

author

CONTENTS

A PROPOSAL FOR TRAPPED PARTICLE FLUX MAPPING IN THE OUTER MAGNETOSPHERE

I. INTRODUCTION

Several years ago, a considerable improvement in the study of the inner radiation belt was achieved by the introduction of a convenient, two dimensional mapping of trapped particles fluxes.[1] Two main features of the geomagnetic field in the inner magnetosphere made this possible:

1. Time variations of the field within 2-3 earth radii are small enough, so that a world-wide adoption of a static field model is possible.

2. The azimuthal asymmetry of the field is small enough, so that particle shells can be considered degenerate:[2] particles mirroring on a given field line populate identical shells, irrespective of their equatorial pitch angles, or their mirror points. Therefore, in a steady state, not only the directional flux $j_\perp$, but also the omnidirectional flux J_0 of trapped particles is independent of longitude, being only a function of the mirror point field B_m and the local L-value (and the energy): $J_0 = J_0(B_m, L, E)$

In the outer magnetosphere, none of these two characteristic features is true. The field configuration is strongly time dependent, driven by variations of the solar wind, by the ring current and by the diurnal wabbling of the earth's dipole. It further shows a strong day-night asymmetry, which causes considerable shell splitting[3]. The fact, however, that some general features of the field configuration still seem to be preserved throughout all minor time variations may suffice to justify the use of a static model in a first attempt to organize particle flux data, measured at different positions in the outer magnetosphere during geomagnetically quiet epochs. In any case, such a procedure would be far better than using dipole quantities such as L-values, invariant latitudes, etc.

Shell splitting does not allow a two dimensional description of omnidirectional particle fluxes. However, directional fluxes of trapped particles still can be mapped in a two dimensional way, independent of the local time at the point of measurement, if one makes use of a model for the magnetospheric field and computes theoretical shell configurations. Consider such a model, and take a directional flux measurement made at a point of geomagnetic coordinates R_p, λ_p, φ_p, of particles incident from a given direction (mean direction of viewing of the detector). With the use of the model, one can determine the particles' pitch angle α_p at the point of measurement, their mirror points and mirror field B_m, their I-value

$$I = \int_m^{m'} \sqrt{1 - B(s)/B_m}\, ds,$$

equatorial pitch angle α_e and radial distance to the equatorial crossing R_e (intersection of the field line with the equatorial plane). One now takes a fixed reference meridian (for instance, the noon meridian), and computes with the aid of the model (and the conservation theorems of the adiabatic invariants) the equatorial pitch angle α_0 and the equatorial crossing R_0 of these particles, when they drift through this reference meridian, provided, of course, that they are stably trapped. The directional flux of the given class of particles at this reference meridian will be the same as at the point of measurement, according to Liouville's theorem. This procedure leads to an analog to B-L plots of directional fluxes, in which L is now replaced by the equatorial crossing R_0 in the reference meridian.

Magnetospheric models are not yet ripe for world-wide adoption; nevertheless it seemed useful to propose here a computer code, which for a given model, allows two-dimensional mappings of directional fluxes of trapped particles, measured anywhere in the magnetosphere, and giving proper diagnostics if the particles were on open field lines, or if they were unable to reach the reference meridian in their longitudinal drift (pseudo-trapped particles[4]).

We want to emphasize again the limitations of such a procedure. However, we suggest that the adoption of such a "recipe" for flux mapping would be far more legitimate than the forced use of dipole relations in a region of the magnetosphere which has little in common with a dipole field.

II. THE COMPUTER CODE "ORGAN"

This computer code, consists of two parts: the first one finds field line and shell parameters at the point of measurement for the particles in question; the second one finds the field line on which the particles will drift through the reference meridian (provided they are stably trapped).

The first part performs field line tracings to the conjugate point (in order to compute I), to the equator (in order to find α_e and R_e) and to the intersection with the earth's surface. The second part locates the point of prefixed I, B_m

values on the reference meridian, using a fast iteration process (subroutine SEARCH), and then finds α_0, R_0 and the intersection of the corresponding field line with the earth's surface.

The magnetic field configuration adopted in the present code is that given by Mead [5] (subroutine MODMAG). This model has a severe limitation in that it takes the dipole axis perpendicular to the ecliptic. However, the problem of the interaction of a steady flow of plasma with a tilted dipole has not yet been solved.

There are four parameters which have to be defined in this model: the minimum distance from the center of the earth to the magnetopause, in the noon meridian (called FRONT in the code); the distances from the center of the earth to the close and to the far limits of the neutral sheet in the midnight meridian (called SHEET1 and SHEET2, respectively), and the field intensity near the neutral sheet (called FSHEET). Suggested values [6] for these parameters are FRONT = 10 earth radii, SHEET1 = 8 earth radii,* SHEET2 = 200 earth radii and FSHEET = 15 gammas. Time variations can be simulated by adequate changes of these parameters.[3] The general field configuration is mainly determined by the value of the parameter FRONT; the trapping boundary and the limits of the pseudo-trapping regions,[3] however, are strongly influenced by the tail parameters.

Electric fields are not taken into account in this paper. This imposes the restriction of what follows, to energetic particles only.

A description of the complete code is given in the Appendix. Input to the program are the geomagnetic coordinates of the point of measurement, and the local pitch angle (the code can easily be modified to read in a direction in space, computing the local pitch angle for the model in question). Output are the geomagnetic coordinates of the mirror point, equatorial crossing and intersection with the earth's surface of the field line going through the point of observation, and the same quantities at the reference meridian, as well as the mirror point field intensity (in gauss) and I (in earth radii). Diagnostics are given if particles are initially on an open field line, if they are trapped in secondary, off-equator field minima (called "minimum-B pockets" in the code), if they precipitate into the ionosphere, or if they are pseudo-trapped, i.e., unable to complete a 180° drift around the earth. Computer time is in general of the order of a few seconds per case, on an IBM 7094. However, a case of pseudo-trapping may take as long as 15-20 seconds (subroutine SEARCH must make sure that there exists

*This does not necessarily mean that the neutral sheet actually starts at 8 earth radii; field lines are still closed at 15–18 R_e in the equatorial plane in the tail, in this case.

no point on the reference meridian with the wanted pair of I, B_m values). An option is given, which repeats all computations, now taking the midnight meridian as reference (this is necessary if one wants to know whether the particles are stably trapped).

III. GRAPHICAL RESULTS

Although the computer code presented here is designed to be applied individually for each point of measurement and each direction in space (i.e., pitch angle), we now present a series of graphs which may be used for a first, very rough attempt to organize directional flux data in the outer magnetosphere.

All graphs shown in Figure 1 display equatorial pitch angle α_0 vs. equatorial crossing R_0 in the reference meridian (noon); each graph corresponds to a fixed local time at the point of measurement. There are three groups of curves in each graph, corresponding to three different values of initial pitch angles at the point of measurement (full lines: $\alpha_p = 90°$; broken lines: $\alpha_p = 60°$; dotted lines: $\alpha_p = 30°$). The curves shown are of constant latitude and of constant radial distance of the point of measurement. Points of intersection of equal latitude and radial distance in each group are joined by thin lines; these thin lines therefore represent the relation of α_0 and R_0 for particles passing through the <u>same</u> point, situated at a given local time, with <u>different</u> pitch angles. There are regions in these graphs which are "inaccessible," i.e., for which no stably trapped particles exist.

An example is now given on how to use these graphs. Suppose that a directional particle flux measurement was made at a point at 7 earth radii and 20° geomagnetic latitude, at 0400 hours local time. The local average pitch angle was 60°. In order to find the equatorial pitch angle α_0 of these particles at the reference meridian, and their equatorial crossing R_0, one goes to the 0400 LT graph, sorts out the 60° pitch angle set (broken lines), and finds the intersection of the $\alpha_p = 20°$, $R_p = 7$ earth radii curves. For intermediate values, an interpolation scheme can be used.

In order to facilitate conversion of these α_0 - R_0 plots into B_m - R_0 plots (equivalent to the B-L coordinates), we present in Figure 2 the relation of mirror point field vs. equatorial pitch angle α_0, for different equatorial crossing R_0 in the noon meridian.

We leave it up to the reader to find out the main features revealed by these graphs, in particular, to inspect how and where the azimuthal asymmetry causes the greatest deformation of particle shells. For a more physical discussion of all these effects, see Ref. 3.

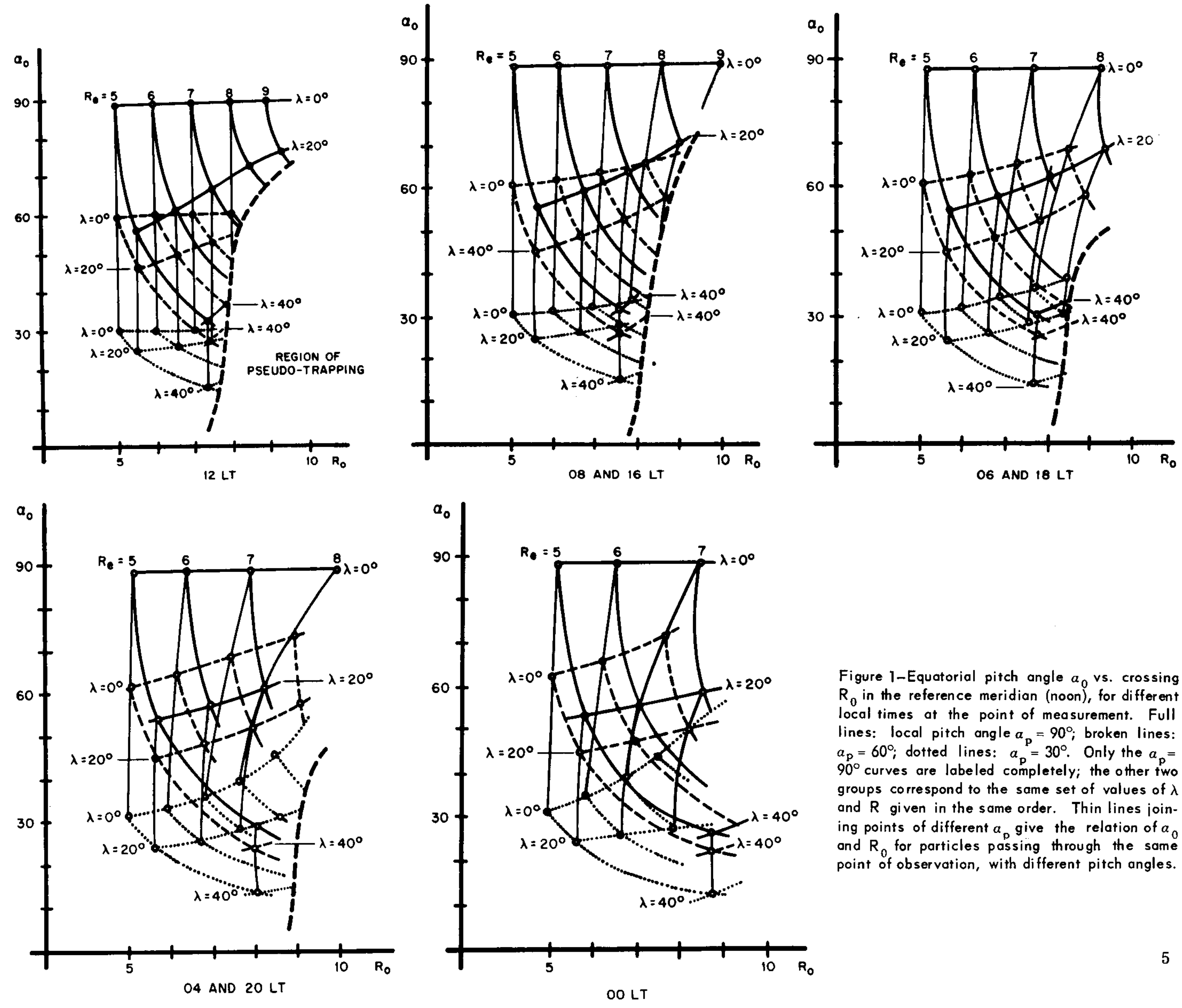

Figure 1—Equatorial pitch angle α_0 vs. crossing R_0 in the reference meridian (noon), for different local times at the point of measurement. Full lines: local pitch angle $\alpha_p = 90°$; broken lines: $\alpha_p = 60°$; dotted lines: $\alpha_p = 30°$. Only the $\alpha_p = 90°$ curves are labeled completely; the other two groups correspond to the same set of values of λ and R given in the same order. Thin lines joining points of different α_p give the relation of α_0 and R_0 for particles passing through the same point of observation, with different pitch angles.

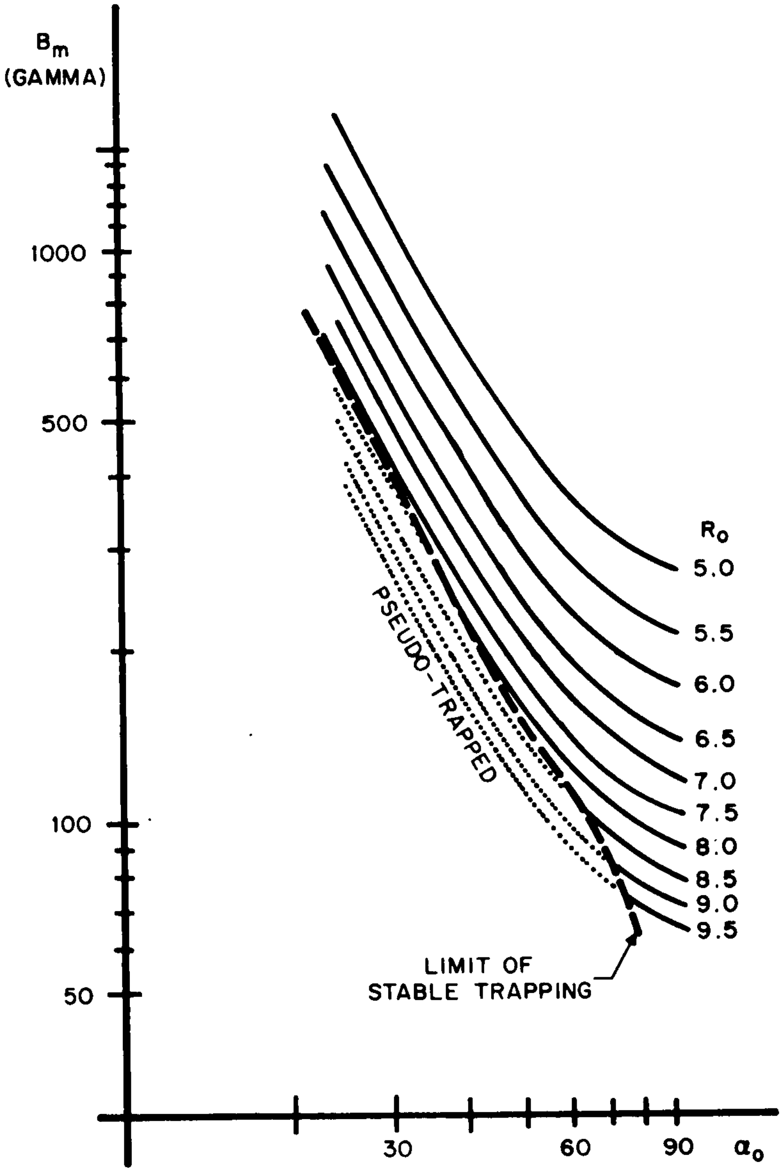

Figure 2—Mirror point field B_m vs equatorial pitch angle α_0, for different equatorial crossings R_0 in the noon meridian

ACKNOWLEDGMENTS

The author expresses his gratitude to Dr. Gilbert Mead for furnishing the code for magnetic field computations, and for valuable discussions.

APPENDIX

Main Program ORGAN

The main features of this program were explained in the text (Section II). Figure 3 shows a simplified flow chart. More information can be obtained from the comment cards in the listing given at the end of the Appendix, in particular regarding input.

Diagnostics are, in general, self-explanatory. "Minimum-B pocket" is the name of a region off the equatorial plane, in which the absolute field intensity passes through a secondary minimum along a field line. Two of such regions at positive and negative latitudes in the front side, near the boundary, are always present in a magnetospheric model; they also can show up in the tail, depending on the particular value given to the tail parameters. Whether they really occur in the magnetosphere is still an experimentally unsolved question.

Subroutine SEARCH

This subroutine consists of an iteration process which finds a point P with prefixed B and I values at a specified longitude. In addition, this subroutine gives detailed information about the field line which passes through the point P. Input for the subroutine are the geomagnetic coordinates of a starting point for the iteration process, the specified B and I values, and the prefixed tolerances for each of B and I. Output are the coordinates of the desired B-I point, and the coordinates of a series of points along the field line between the point P and its conjugate.

There are two basic processes in this subroutine: one varies the radial distance searching along a constant latitude and longitude line for a point of given B; the other searches along a nearly constant-B line in a given meridian plane, for a point of prefixed I. The first of these processes is very fast; it only involves calling the subroutine MODMAG which computes the magnetic field at each point. The second process is much more complicated and involves the use of subroutine INVAR, which traces a field line through each point at which it is called and evaluates the integral I. The program was constructed in such a way as to minimize the number of times INVAR is called.

Figure 4 shows a simplified flow chart for subroutine SEARCH. Geometrically, the procedure is as follows: starting at the input point, the program iterates the radial distance to a point Q where the prefixed B-value (called SB) is reached within a prefixed, gross tolerance designated by ERRB. The I value is then determined for this point Q. If it is not equal to the prefixed I-value (called SI) within the tolerance designated by ERRI, the latitude is changed by a small, prefixed amount DV in the correct direction. At this new latitude, a point R is found with the prefixed B-value; I is then calculated at this point. Points Q and R lie, approximately, on a constant B line in the meridian plane. A linear extrapolation is then used to find the point on this line which would have the specified value SI. This extrapolation is made as though the I-dependence were linear along the constant B line. This extrapolated point is now taken as a second approximation and is used as the starting point of the next iteration cycle. If the new starting point is close enough to the points Q and R, the step size in latitude DV is reduced by one-tenth. The iteration is continued until a point is found whose B and I values lie in the interval SB ± ERRB and SI ± ERRI. When this is accomplished, all tolerances are reduced two orders of magnitude and the entire procedure is repeated once more until a point is found whose B and I values agree with the prefixed ones within the new, reduced tolerance intervals.

If iterations exceed a given number of cycles (set = 20 in this code), a cutoff is introduced and a diagnostic is printed, telling that the code is unable to find a point with the specified I,B values. It should be noted that under normal conditions, it never takes more than 2-3 cycles in each iteration.

Notice that subroutine SEARCH can be used to determine B-I rings in the outer magnetosphere.

Subroutines INVAR, START, LINES, INTEG and MODMAG

The first four subroutines are slightly modified and implemented versions of McIlwain's program for L-computations.[7] Their purpose is to trace a field line (START and LINES with MODMAG) and to determine the second invariant I (INTEG). Input for INVAR is a point P; output are the B and I values at P, and the coordinates of a sequence of points along the field line beginning at P and progressing to its conjugate.

INVAR controls the rest of the above listed programs. When called from INVAR, subroutine START picks the first three points of a sequence along the field line, which are such that the second point is P and the B-values of the three are in decreasing order. LINES is a tracing routine which continues the sequence of points along the field line in the direction defined by START. This sequence progresses to the first point for which the B-value is again equal or exceeds that of P. The arc length between consecutive points in the sequence is,

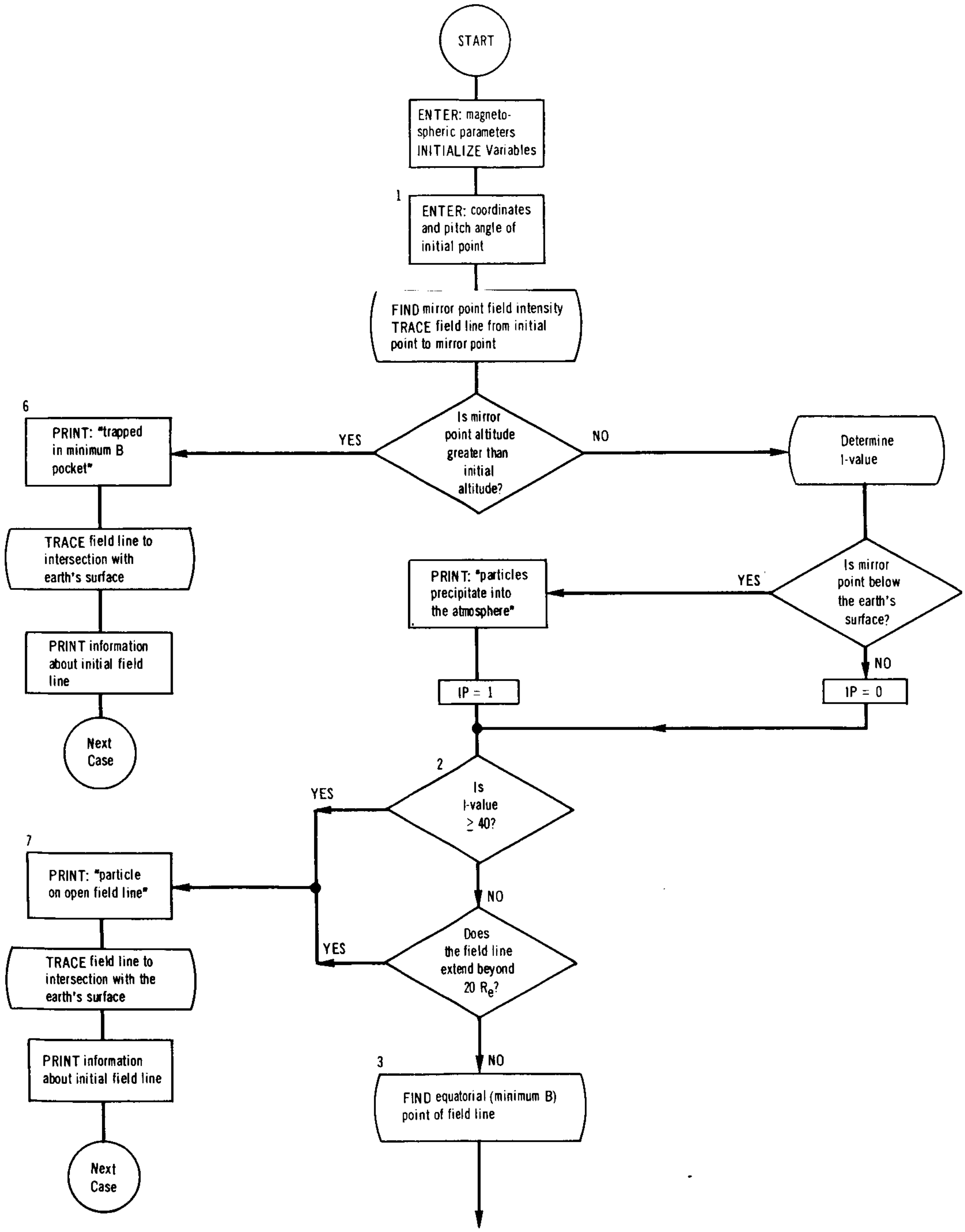

Figure 3–Flow chart for program ORGAN.

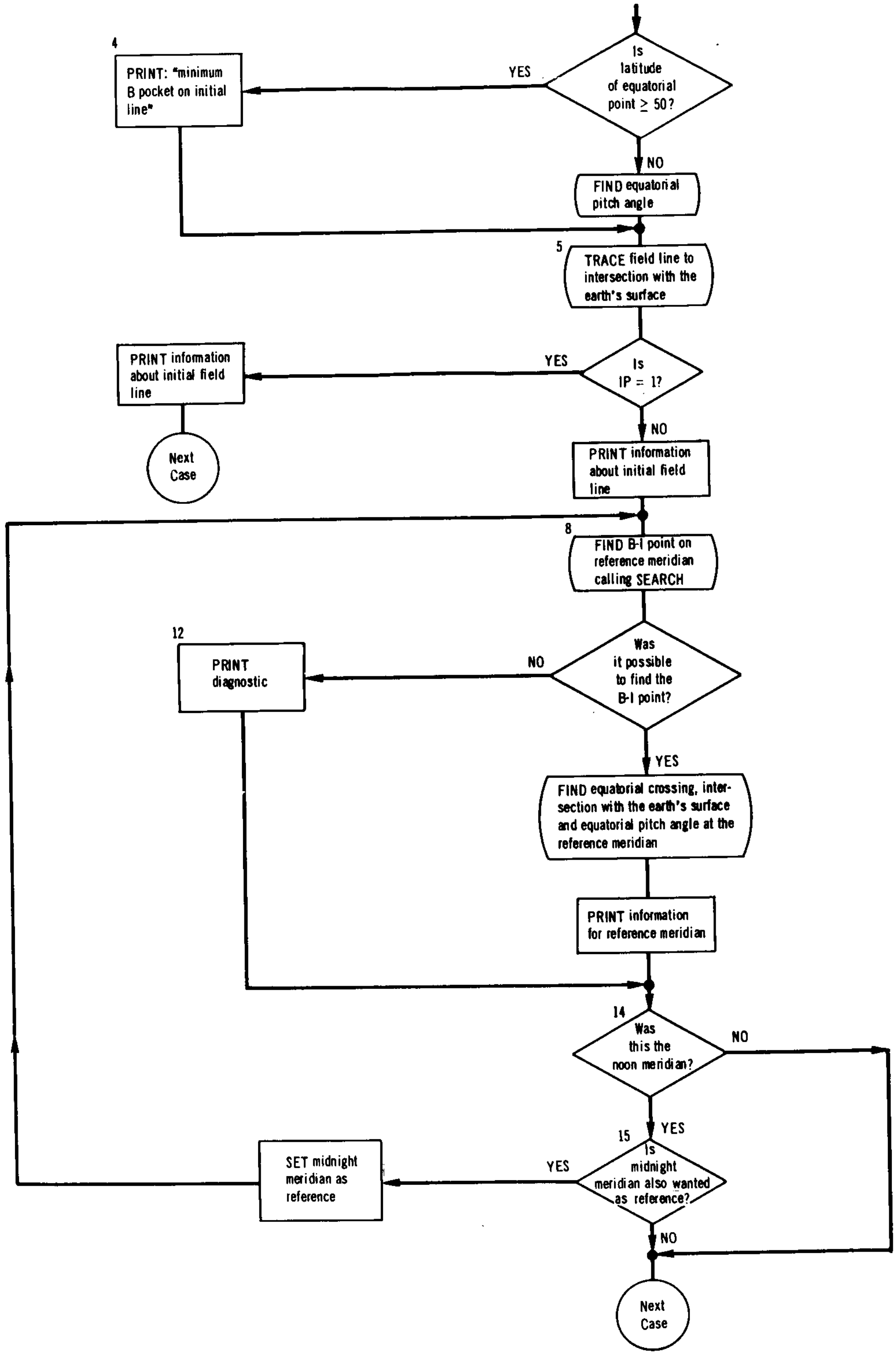

Figure 3(continued)–Flow chart for program ORGAN.

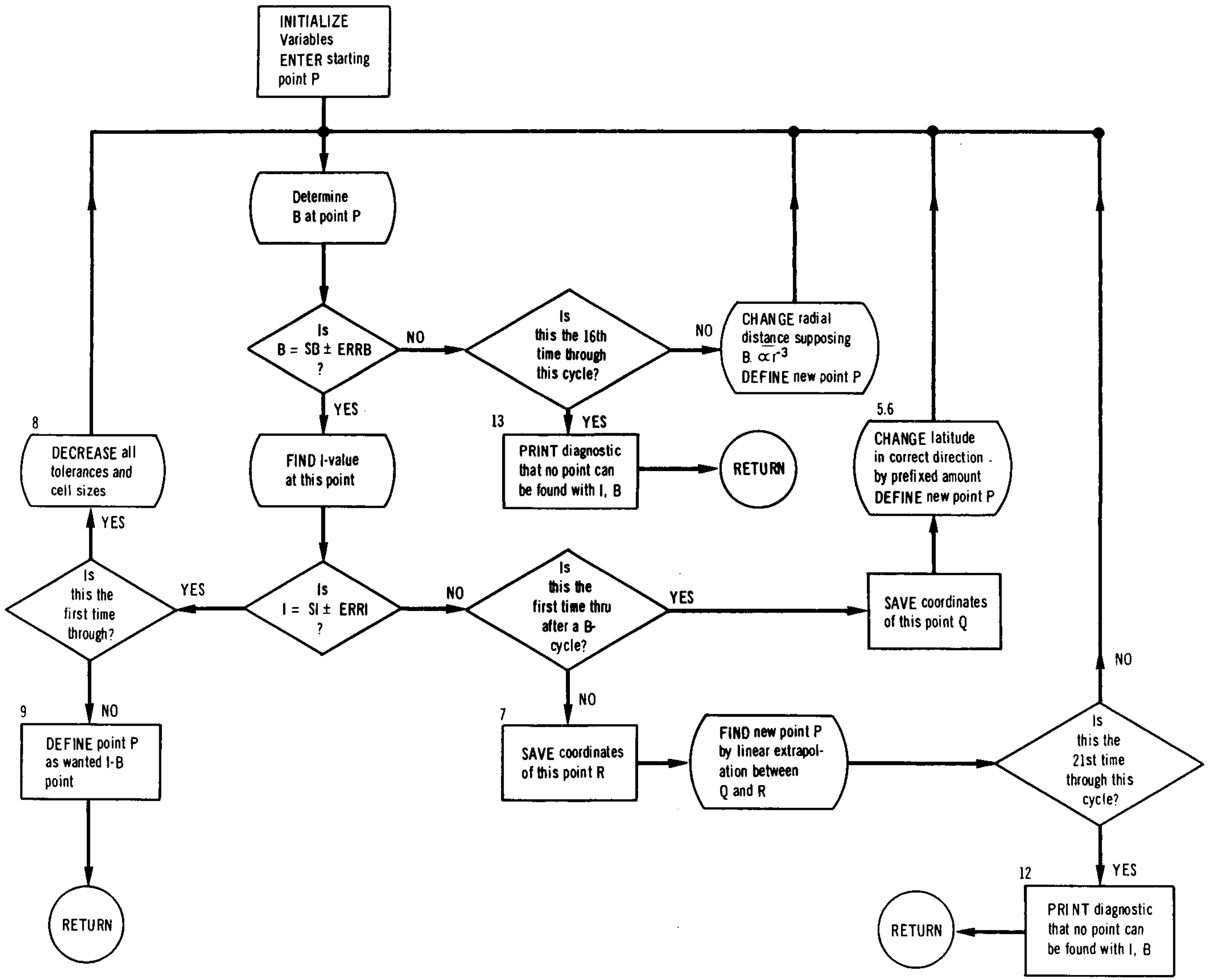

Figure 4–Flow chart for subroutine SEARCH

in general, a linear function of the geocentric distance of the points, initially being directly controlled by the prefixed value of the parameter called ERR. During the tracing, the arc length keeps doubling until an error control (dependent on ERR) levels it off at a value which is usually eight or ten times greater than the initial size. The result is speed and accuracy at the expense of a nonconstant and sometimes unpredictable cell size. The main merit of McIlwain's tracing routine clearly shows up when it is used for a magnetospheric model, where there are sudden, sharp bends of field lines near the boundary, and across the neutral sheet: in these regions, the automatic error control in LINES drastically reduces the cell size to ensure proper tracing.

Subroutine INTEG was left intact from McIlwain's code. Subroute MODMAG was written by G. Mead[8] and corresponds to his magnetospheric model.[5] This model is symmetric with respect to the equatorial plane, and field line tracings to determine I values could be performed from the point in question to the equator only. This time saving modification, however, was not introduced in the present code, in order to leave it flexible enough to be used with unsymmetric models.

Subroutine EQUAT

This subroutine commands field line tracing from a given point to the geomagnetic equator, i.e., to the point with minimum B value. Usually, the starting point is one defined in a previous INVAR call and is already close to the equator. EQUAT is always used for a more accurate determination of the equatorial point and is called with a much smaller value of ERR. When called from EQUAT, the cell-doubling mechanism in LINES is by-passed to ensure highest precision.

Subroutine BESECT

This subroutine operates similarly to EQUAT, but traces a field line downwards, i.e., in the direction opposite to the equatorial point, until a prefixed B-value is reached.

Subroutine INSECT

This subroutine traces a field line from a given point downwards to the intersection with a fixed altitude (the earth's surface).

All operations mentioned in connection with subroutines EQUAT, BESECT and INSECT are actually executed in START, LINES and MODMAG. Information about the particular type of tracing is transferred by a parameter called MMM.

REFERENCES

1. C. E. McIlwain, "Coordinates for Mapping the Distribution of Magnetically Trapped Particles," Journal of Geophys. Res. 66, 3681 - 3691 (1961).

2. E. C. Stone,, "The Physical Significance and Application of L, B_0 and R_0 to Geomagnetically Trapped Particles," Journal of Geophys. Res. 68, 4157 - 4166 (1963).

3. J. G. Roederer, "On the Adiabatic Motion of Energetic Particles in a Model Magnetosphere," NASA X-640-66-304 (1966).

4. J. G. Roederer, "Magnetospheric Phenomena," presented at the International Conference on Cosmic Rays, London 1965, to be published.

5. G. D. Mead, "Deformation of the Geomagnetic Field by the Solar Wind," Journal of Geophys. Res. 69, 1181 - 1196 (1964).

6. N. F. Ness and D. J. Williams, "Correlated Magnetic Tail and Radiation Belt Observations," Journal of Geophys. Res. 71, 322 - 325 (1966).

7. Distributed by Dr. C. E. McIlwain, University of California, San Diego.

8. G. Mead, private communication.

```
      PROGRAM ORGAN
      COMMON FRONT,SHEET1,SHEET2,FSHEET
      COMMON B(200),VN1(200),VN2(200),VN3(200),ARC(200),VNEAR(3),VBO(3),
     1VSAVE(3),BO,BNEAR,JUP,MMM
      COMMON VCONJ(3),ALTO
C
C     FOR ENQUIRIES ABOUT THIS CODE, WRITE TO JUAN G. ROEDERER
C     FACULTAD DE CIENCIAS EXACTAS, PERU 272 , BUENOS AIRES ARGENTINA
C
      DIMENSION V(3)
C
C     IF LESS ACCURACY BUT HIGHER SPEED IS WANTED, MULTIPLY ALL ERRORS
C     BY A COMMON FACTOR (NOT EXCEEDING 40-50)
C
      ERR=0.0001
      ERRB=0.004
      ERRI=0.008
      ERR1=ERR*0.01
      ERR2=ERR*40.
      ERR3=ERR*0.05
      ZERO=0.
      RAD=57.2957795
      RADIN=1./RAD
      ALTO=1.
C
C     FRONT IS DISTANCE IN EARTH RADII TO MAGNETOSPHERIC BOUNDARY IN THE
C     EARTH SUN DIRECTION (SUGGESTED VALUE = 10 )
C     SHEET1 IS DISTANCE TO INNER EDGE OF NEUTRAL SHEET IN MIDNIGHT
C     MERIDIAN (SUGGESTED VALUE = 8-10)
C     SHEET2 IS DISTANCE TO OUTER EDGE OF NEUTRAL SHEET IN MIDNIGHT
C     MERIDIAN (SUGGESTED VALUE = 200)
C     FSHEET IS MAGNETIC FIELD INTENSITY IN GAMMA NEAR NEUTRAL SHEET
C     (SUGGESTED VALUE = 15 GAMMA)
C
      READ (5,17) FRONT,SHEET1,SHEET2,FSHEET
      WRITE (6,17) FRONT,SHEET1,SHEET2,FSHEET
C
C     IF BOTH NOON AND MIDNIGHT MERIDIANS ARE WANTED AS REFERENCE PLANES
C     SET NREF = 1 . IF ONLY NOON MERIDIAN IS WANTED AS REFERENCE, SET
C     NREF = 0
C
      READ (5,18) NREF
C
C     ALT = GEOCENTRIC DISTANCE IN EARTH RADII TO POINT OF OBSERVATION
C     FLAT = GEOMAGNETIC LATITUDE OF SAME POINT
C     FLONG = LOCAL TIME OF SAME POINT, IN DEGREES EAST OF MIDNIGHT
C     (E.G. DAWN MERIDIAN HAS +90 DEG. , DUSK MERIDIAN -90 OR +270 )
C     ALPHA = LOCAL PITCH ANGLE IN DEGREES. CODE MAY BE EASILY IMPLEM=
C     ENTED SUCH AS TO READ  A DIRECTION IN A GEOMAGNETIC SYSTEM,
C     COMPUTING THE THEORETICAL PITCH ANGLE ASSOCIATED TO THIS DIRECTION
C     IN PLACE INDICATED IN COMMENT CARDS BELOW
C
1     READ (5,17) ALT,FLAT,FLONG,ALPHA
      IF (ALT.LT.0.5) GO TO 16
      WRITE (6,23) ALT,FLAT,FLONG,ALPHA
      REFLON=180.
      IP=0
      IF (ABS(FLAT).LT.1.) FLAT=0.1
      V(1)=ALT
```

```
      V(2)=(90.-FLAT)*RADIN
      V(3)=FLONG*RADIN
      SIT=ABS(SIN(V(2)))
C
C     THE FOLLOWING PART FINDS THE MAGNETOSPHERIC GEOMETRIC PARAMETERS
C     FOR THE FIELD LINE THROUGH THE POINT OF OBSERVATION
C
      CALL MODMAG (ALT,SIT,V(3),BR,BT,BP,BB,V(2))
      IF (ALPHA.LT.0.5) ALPHA=0.5
      SINA=SIN(ALPHA*RADIN)
      SSQA=SINA*SINA
      BMIR=BB/SSQA
      BO=BMIR
C
C     BESECT FINDS MIRROR POINT ON INITIAL FIELD LINE
C
      CALL BESECT (V(1),V(2),V(3),BALT,BLAT,BLONG,ERR)
      SALT=BALT
      SLAT=BLAT
      SLONG=BLONG
      IF (SALT.GT.ALT) GO TO 6
C
C     INVAR DETERMINES I VALUE
C
      CALL INVAR (VBO(1),VBO(2),VBO(3),ERR,BMIR,FI)
C     IF PRINTOUT OF ALL FIELD LINE POINTS BETWEEN MIRROR POINTS IS
C     WANTED, INSERT FOLLOWING CARDS
C     DO 500 J=2,JUP
C     OLAT=90.-VN2(J)*RAD
C     OLONG=VN3(J)*RAD
C     IF (OLONG.GT.180.) OLONG=OLONG-360.
C     IF (OLONG.LT.-180.) OLONG=OLONG+360.
C     500 WRITE(6,501) J,VN1(J),OLAT,OLONG
C     501 FORMAT(I5,3F15.3,E20.4)
      IF (VBO(1).GT.1.) GO TO 2
      WRITE (6,20)
      IP=1
      VBO(1)=V(1)
      VBO(2)=V(2)
      VBO(3)=V(3)
2     IF (FI.GT.40.) GO TO 7
      DO 3 J=1,JUP
      IF (VN1(J).GT.20.) GO TO 7
3     CONTINUE
C
C     EQUAT FINDS EQUATORIAL POINT ON INITIAL FIELD LINE
C
      CALL EQUAT (VNEAR(1),VNEAR(2),VNEAR(3),EQB,EALT,ELAT,ELONG,ERR1)
      IF (ABS(ELAT).GT.5.) GO TO 4
      RATIO=EQB/BMIR
      TGT=SQRT(RATIO/(1.-RATIO))
      PITCH=ATAN(TGT)*RAD
      WRITE (6,24) EALT,ELONG,EQB,PITCH
      GO TO 5
4     WRITE (6,19) EALT,ELAT,ELONG
5     CALL INSECT (VBO(1),VBO(2),VBO(3),CALT,CLAT,CLONG,ERR3)
      IF (IP.EQ.0) WRITE (6,25) SALT,SLAT,SLONG,BMIR,FI
      WRITE (6,26) CLAT,CLONG
      IF (IP.EQ.1) GO TO 1
      KOUNT=1
```

```
      GO TO 8
6     WRITE (6,31) SALT,SLAT,SLONG,BMIR
      CALL INSECT (VBO(1),VBO(2),VBO(3),CALT,CLAT,CLONG,ERR3)
      WRITE (6,26) CLAT,CLONG
      GO TO 1
7     CALL INSECT (VBO(1),VBO(2),VBO(3),CALT,CLAT,CLONG,ERR3)
      WRITE (6,22)
      IF (IP.EQ.0) WRITE (6,25) SALT,SLAT,SLONG,BMIR,FI
      WRITE (6,26) CLAT,CLONG
      GO TO 1
C
C     THE FOLLOWING PART FINDS THE MAGNETIC PARAMETERS ON THE NOON
C     AND/OR THE MIDNIGHT MERIDIAN
C
8     COOR1=BALT
      COOR2=BLAT
      COOR3=REFLON
C
C     SEARCH LOOKS FOR A PREFIXED B-L POINT
C
      CALL SEARCH (COOR1,COOR2,COOR3,BMIR,FI,BBB,FFI,ERR2,ERRB,ERRI,CHEC
     1K)
      IF ((CHECK.GT.0.5).OR.((KOUNT.EQ.1).AND.(VNEAR(1).GT.10.5))) GO TO
     1 12
C     IF PRINTOUT OF ALL FIELD LINE POINTS BETWEEN MIRROR POINTS IS
C     WANTED, INSERT FOLLOWING CARDS
C     DO 502 J=2,JUP
C     OLAT=90.-VN2(J)*RAD
C     OLONG=VN3(J)*RAD
C     IF (OLONG.GT.180.) OLONG=OLONG-360.
C     IF (OLONG.LT.-180.) OLONG=OLONG+360.
C     502 WRITE(6,501) J,VN1(J),OLAT,OLONG
      CALL EQUAT (VNEAR(1),VNEAR(2),VNEAR(3),REQB,REALT,RELAT,RELONG,ERR
     11)
      CALL INSECT (VSAVE(1),VSAVE(2),VSAVE(3),CALT,CLAT,CLONG,ERR3)
      IF (ABS(RELAT).GT.5.) GO TO 10
      RATIO=REQB/BMIR
      TGT=SQRT(RATIO/(1.-RATIO))
      RALPHA=ATAN(TGT)*RAD
      IF (KOUNT.EQ.2) GO TO 9
      WRITE (6,27) REALT,REQB,RALPHA,COOR1,COOR2,CLAT
      GO TO 14
9     WRITE (6,28) REALT,REQB,RALPHA,COOR1,COOR2,CLAT
      GO TO 14
10    WRITE (6,21) REALT,RELAT,RELONG
      IF (KOUNT.EQ.2) GO TO 11
      WRITE (6,27) ZERO,ZERO,ZERO,COOR1,COOR2,CLAT
      GO TO 14
11    WRITE (6,28) ZERO,ZERO,ZERO,COOR1,COOR2,CLAT
      GO TO 14
12    IF (KOUNT.EQ.1) GO TO 13
      WRITE (6,29)
      GO TO 1
13    WRITE (6,30)
      GO TO 15
14    IF (KOUNT.EQ.2) GO TO 1
15    IF (NREF.EQ.0) GO TO 1
      KOUNT=2
      REFLON=0.
      GO TO 8
```

```
16    CONTINUE
      RETURN
C
17    FORMAT (4F10.3)
18    FORMAT (I5)
19    FORMAT (1X/37H  MINIMUM-B POCKET IN INITIAL LINE AT,5X,4HRE =,F7.3
     1,5X,4HLAT=,F6.2,5X,5HLONG=,F7.2)
20    FORMAT (1X/39H  PARTICLES PRECIPITATE INTO ATMOSPHERE)
21    FORMAT (1X/35H  MINIMUM-B POCKET IN FINAL LINE AT,7X,4HRE =,F7.3,5
     1X,4HLAT=,F6.2,5X,5HLONG=,F7.2)
22    FORMAT (1X/30H  PARTICLES ON OPEN FIELD LINE/)
23    FORMAT (1X////10H  NEW CASE,9X,4HRE =,F7.3,5X,4HLAT=,F6.2,5X,5HLON
     1G=,F7.2,5X,12HPITCH ANGLE=,F5.2///1X)
24    FORMAT (9H  EQUATOR,10X,4HRE =,F7.3,5X,5HLONG=,F7.2,5X,3H B=,F7.6,
     15X,6HPITCH=,F6.2)
25    FORMAT (14H  MIRROR POINT,5X,4HRE =,F7.3,5X,4HLAT=,F6.2,5X,5HLONG=
     1,F7.2,5X,3HBM=,F7.6,5X2HI=,F6.2)
26    FORMAT (27H  INTERSECTION WITH SURFACE,8X,4HLAT=,F6.2,5X,5HLONG=,F
     17.2)
27    FORMAT (1X//18H  ON NOON MERIDIAN//8H EQUATOR,12X,2HR=,F7.3,5X,2HB
     1=,F7.6,7X,6HPITCH=,F6.2/13H MIRROR POINT,5X,4HRE =,F7.3,5X,4HLAT=,
     2F6.2/13H INTERSECTION,5X,4HLAT=,F6.2)
28    FORMAT (1X/22H  ON MIDNIGHT MERIDIAN//8H EQUATOR,12X,2HR=,F7.3,5X,
     12HB=,F7.6,7X,6HPITCH=,F6.2/13H MIRROR POINT,5X,4HRE =,F7.3,5X,4HLA
     2T=,F6.2/13H INTERSECTION,5X,4HLAT=,F6.2)
29    FORMAT (1X/37H  PSEUDO-TRAPPED, LEAVES THROUGH TAIL/)
30    FORMAT (1X/45H  PSEUDO-TRAPPED, LEAVES THROUGH MAGNETOPAUSE/)
31    FORMAT (1X/29H  TRAPPED IN MINIMUM-B POCKET/14H  MIRROR POINT,5X,4
     1HRE =,F7.3,5X,4HLAT=,F6.2,5X,5HLONG=,F7.2,5X,3HBM=,F7.6)
      END
```

```
      SUBROUTINE SEARCH (ALT,FLAT,FLONG,SB,SI,BB,FI,ERR,ERRB,ERRI,RLOST)
      COMMON FRONT,SHEET1,SHEET2,FSHEET
      COMMON B(200),VN1(200),VN2(200),VN3(200),ARC(200),VNEAR(3),VBO(3),
     1VSAVE(3),BO,BNEAR,JUP,MMM
      COMMON VCONJ(3),ALTO
C
C     SUBROUTINE DEFINES FIELD LINE GOING THROUGH POINT OF PREFIXED
C     B AND SECOND INVARIANT I , AT A GIVEN LONGITUDE
C
      DIMENSION V(3), V1(3), V2(3)
      DV=0.02
      RLOST=0.
      MCHECK=0
      ICHECK=0
      SERR=ERR
      SERRB=ERRB
      SERRI=ERRI
      V(1)=ALT
      IF (V(1).LT.0.5) GO TO 11
      V(2)=(90.-FLAT)/57.2957795
      V(3)=FLONG/57.2957795
      DCLT=1.5708
      ICON=1
1     ILIT=1
      DELV2=DV
      ICHECK=ICHECK+1
      IF (ICHECK.GT.20) GO TO 12
2     SIT=ABS(SIN(V(2)))
      CALL MODMAG (V(1),SIT,V(3),BR,BT,BP,BB,V(2))
3     FAC=1.-(SB-BB)/(3.*SB)
      IF (FAC.GT.1.5) FAC=1.5
      IF (FAC.LT.0.666) FAC=0.666
      V(1)=V(1)*FAC
      IF ((V(1).GT.100.).OR.(V(1).LT.0.5)) GO TO 11
      V1(1)=V(1)
      V1(2)=V(2)
      MCHECK=MCHECK+1
      IF (MCHECK.GT.15) GO TO 13
      CALL MODMAG (V(1),SIT,V(3),BR,BT,BP,BB,V(2))
      IF (ABS((BB-SB)/SB).GT.ERRB) GO TO 3
      MCHECK=0
      IF (ILIT.NE.1) GO TO 7
      ILIT=2
      CALL INVAR (V(1),V(2),V(3),ERR,BB,FI)
      IF (JUP.LT.0) GO TO 14
      V2(1)=V(1)
      V2(2)=V(2)
      B2=BB
      FI2=FI
      IF (ABS((FI-SI)/SI).LE.ERRI) GO TO 8
      IF (ABS(V(2)-DCLT).LT.0.1) GO TO 6
      SGN=SIGN(1.,(FI-SI))
      IF (V(2).LT.DCLT) GO TO 4
      DELV2=-SGN*DELV2
      GO TO 5
4     DELV2=SGN*DELV2
5     V(2)=V(2)+DELV2
      GO TO 2
6     V(2)=V(2)+DELV2
```

```
      CALL INVAR (V(1),V(2),V(3),ERR,BB,FI)
      IF (JUP.LT.0) GO TO 14
      IF ((FI-SI)*(FI-FI2).LE.0.) GO TO 2
      V(2)=V(2)-2.*DELV2
      GO TO 2
7     B1=BB
      CALL INVAR (V(1),V(2),V(3),ERR,BB,FI)
      IF (JUP.LT.0) GO TO 14
      IF (ABS((FI-SI)/SI).LE.ERRI) GO TO 8
      FACT=(SI-FI)/(FI2-FI)
      IF (ABS(FACT).GT.3.) FACT=3.*SIGN(1.,FACT)
      V(1)=V1(1)+(V2(1)-V1(1))*FACT
      V(2)=V1(2)+(V2(2)-V1(2))*FACT
      Y=AMIN1(ABS(V(2)-V1(2)),ABS(V(2)-V2(2)))
      IF (Y.GT.ABS(V1(2)-V2(2))) GO TO 1
      DV=Y
      GO TO 1
8     CONTINUE
      IF (ICON.EQ.2) GO TO 9
      ICON=2
      DV=DV*0.1
      ERR=ERR*0.025
      ERRB=ERRB*0.05
      ERRI=ERRI*0.04
      GO TO 1
9     ALT=V(1)
      FLAT=90.-V(2)*57.2957795
      FLONG=V(3)*57.2957795
      IF (FLONG.GT.180.) FLONG=FLONG-360.
      IF (FLONG.LT.(-180.)) FLONG=FLONG+360.
      DO 10 I=1,3
10    VSAVE(I)=V(I)
      GO TO 16
11    WRITE (6,17) V(1),FLAT,FLONG
      GO TO 15
12    WRITE (6,18) ICHECK
      GO TO 15
13    WRITE (6,19) ICHECK,MCHECK
      GO TO 15
14    WRITE (6,20) ICHECK,MCHECK,ALT,FLAT,FLONG
      JUP=1
15    RLOST=1.
16    ERR=SERR
      ERRB=SERRB
      ERRI=SERRI
      RETURN
C
C
C
17    FORMAT (75X,19H  ALT OUT OF LIMITS/75X,3F10.3)
18    FORMAT (69X,51H  SORRY,BUT I CANNOT FIND THAT DAMN POINT IN ICHECK
     1/69X,2I10)
19    FORMAT (69X,51H  SORRY,BUT I CANNOT FIND THAT DAMN POINT IN MCHECK
     1/69X,2I10)
20    FORMAT (75X,40H  SORRY,BUT POINT IS IN THAT DAMN POCKET/55X,2I10,3
     1E15.5)
      END
```

```
      SUBROUTINE EQUAT (DUM1,DUM2,DUM3,EB,EALT,ELAT,ELONG,ERR)
      COMMON FRONT,SHEET1,SHEET2,FSHEET
      COMMON B(200),VN1(200),VN2(200),VN3(200),ARC(200),VNEAR(3),VBO(3),
     1VSAVE(3),BO,BNEAR,JUP,MMM
      COMMON VCONJ(3),ALTO
C
C     SUBROUTINE TRACES FIELD LINE FROM A GIVEN POINT TO MINIMUM B
C
      DIMENSION V(3,3), VN(3), VP(3), R1(3), R2(3), R3(3)
      ERR1=ERR
      MMM=1
      JUP=1
      V(1,2)=DUM1
      V(2,2)=DUM2
      V(3,2)=DUM3
      ARC(1)=0.
      DCLT=1.5708
1     ARC(2)=V(1,2)*SQRT(ERR)*0.3
      IF (V(2,2)-DCLT) 2,3,3
2     ARC(2)=-ARC(2)
3     CALL START (R1,R2,R3,B,ARC,ERR,V)
      IF (JUP.LT.0) GO TO 5
      DO 4 I=1,3
      VP(I)=V(I,2)
4     VN(I)=V(I,3)
      CALL LINES (R1,R2,R3,B,ARC,ERR,J,VP,VN)
      IF (J.LT.200) GO TO 7
      ERR=4.*ERR
      GO TO 6
5     JUP=1
      ERR=ERR*0.1
6     CONTINUE
      WRITE (6,8) ERR
      GO TO 1
7     ERR=ERR1
      EB=BNEAR
      EALT=VNEAR(1)
      ELAT=90.-VNEAR(2)*57.2957795
      ELONG=VNEAR(3)*57.2957795
      IF (ELONG.GT.180.) ELONG=ELONG-360.
      IF (ELONG.LT.(-180.)) ELONG=ELONG+360.
      RETURN
C
C
C
8     FORMAT (81X,24H ERROR CHANGED  IN EQUAT,E15.4)
      END
```

```
      SUBROUTINE INSECT (P1,P2,P3,CALT,CLAT,CLONG,ERR)
      COMMON FRONT,SHEET1,SHEET2,FSHEET
      COMMON B(200),VN1(200),VN2(200),VN3(200),ARC(200),VNEAR(3),VBO(3),
     1VSAVE(3),BO,BNEAR,JUP,MMM
      COMMON VCONJ(3),ALTO
C
C     SUBROUTINE TRACES FIELD LINE DOWNWARDS UNTIL INTERSECTION WITH
C     ALTO (= 1. RE) IS REACHED
C
      DIMENSION V(3,3), VN(3), VP(3), R1(3), R2(3), R3(3)
      ERR1=ERR
      J=0
      SIT=ABS(SIN(P2))
      MMM=2
      V(1,2)=P1
      V(2,2)=P2
      V(3,2)=P3
      ARC(1)=0.
      DCLT=1.5708
1     ARC(2)=V(1,2)*SQRT(ERR)*0.3
      IF (V(2,2)-DCLT) 3,3,2
2     ARC(2)=-ARC(2)
3     CALL START (R1,R2,R3,B,ARC,ERR,V)
      DO 4 I=1,3
      VP(I)=V(I,2)
4     VN(I)=V(I,3)
      CALL LINES (R1,R2,R3,B,ARC,ERR,J,VP,VN)
      IF (J.LT.200) GO TO 5
      ERR=4.*ERR
      WRITE (6,6) ERR
      GO TO 1
5     ERR=ERR1
      JUP=J
      CALT=VCONJ(1)
      CLAT=90.-VCONJ(2)*57.2957795
      CLONG=VCONJ(3)*57.2957795
      IF (CLONG.GT.180.) CLONG=CLONG-360.
      IF (CLONG.LT.(-180.)) CLONG=CLONG+360.
      RETURN
C
C
6     FORMAT (81X,23HERROR CHANGED IN INSECT,E16.4)
      END
```

```
      SUBROUTINE BESECT (RUM1,RUM2,RUM3,BALT,BLAT,BLONG,ERR)
      COMMON FRONT,SHEET1,SHEET2,FSHEET
      COMMON B(200),VN1(200),VN2(200),VN3(200),ARC(200),VNEAR(3),VBO(3),
     1VSAVE(3),BO,BNEAR,JUP,MMM
      COMMON VCONJ(3),ALTO
C
C     SUBROUTINE TRACES FIELD LINE DOWNWARDS FROM A GIVEN POINT UNTIL
C     A PREFIXED B-VALUE IS REACHED
C
      DIMENSION V(3,3), VN(3), VP(3), R1(3), R2(3), R3(3)
      ERR1=ERR
      MMM=3
      JUP=1
      V(1,2)=RUM1
      V(2,2)=RUM2
      V(3,2)=RUM3
      ARC(1)=0.
      DCLT=1.5708
1     ARC(2)=V(1,2)*SQRT(ERR)*0.3
      IF (V(2,2)-DCLT) 3,3,2
2     ARC(2)=-ARC(2)
3     CALL START (R1,R2,R3,B,ARC,ERR,V)
      IF (JUP.LT.0) GO TO 5
      DO 4 I=1,3
      VP(I)=V(I,2)
4     VN(I)=V(I,3)
      CALL LINES (R1,R2,R3,B,ARC,ERR,J,VP,VN)
      IF (J.LT.200) GO TO 7
      ERR=4.*ERR
      GO TO 6
5     JUP=1
      ERR=ERR*0.1
6     CONTINUE
      WRITE (6,8) ERR
      GO TO 1
7     ERR=ERR1
      JUP=J
      BALT=VBO(1)
      BLAT=90.-VBO(2)*57.2957795
      BLONG=VBO(3)*57.2957795
      IF (BLONG.GT.180.) BLONG=BLONG-360.
      IF (BLONG.LT.(-180.)) BLONG=BLONG+360.
      RETURN
C
C
C
8     FORMAT (81X,24H ERROR CHANGED IN BESECT,E15.4)
      END
```

```
      SUBROUTINE INVAR (DUM1,DUM2,DUM3,ERR,BB,FI)
      COMMON FRONT,SHEET1,SHEET2,FSHEET
      COMMON B(200),VN1(200),VN2(200),VN3(200),ARC(200),VNEAR(3),VBO(3),
     1VSAVE(3),BO,BNEAR,JUP,MMM
      COMMON VCONJ(3),ALTO
C
C     SUBROUTINES INVAR,START,LINES AND INTEG ARE BASED ON MCILWAINS
C     INVAR CODE, CONVENIENTLY IMPLEMENTED
C
      DIMENSION V(3,3), VN(3), VP(3), BEG(200), BEND(200), BLOG(200), EC
     1O(200), R1(3), R2(3), R3(3)
      MMM=0
      JUP=1
      V(1,2)=DUM1
      V(2,2)=DUM2
      V(3,2)=DUM3
      ERR1=ERR
      ARC(1)=0.
1     ARC(2)=V(1,2)*SQRT(ERR)*0.3
      DCLT=1.5708
      IF (V(2,2)-DCLT) 2,3,3
2     ARC(2)=-ARC(2)
3     CALL START (R1,R2,R3,B,ARC,ERR,V)
      IF (JUP.LT.0) GO TO 8
      DO 4 I=1,3
      VP(I)=V(I,2)
4     VN(I)=V(I,3)
      CALL LINES (R1,R2,R3,B,ARC,ERR,J,VP,VN)
      IF (J.LT.200) GO TO 5
      ERR=ERR*4.
      WRITE (6,9) ERR
      GO TO 1
5     ERR=ERR1
      JUP=J
      DO 6 J=1,JUP
      ARC(J)=ABS(ARC(J))
6     BLOG(J)=ALOG(B(J))
      JEP=JUP-1
      DO 7 J=2,JEP
      ASUM=ARC(J)+ARC(J+1)
      DX=BLOG(J-1)-BLOG(J)
      DN=ASUM*ARC(J)*ARC(J+1)
      BCO=((BLOG(J-1)-BLOG(J+1))*ARC(J)**2-DX*ASUM**2)/DN
      CCO=(DX*ARC(J+1)-(BLOG(J)-BLOG(J+1))*ARC(J))/DN
      SA=.75*ARC(J)
      SC=SA+.25*ASUM
      DCO=BLOG(J-1)-CCO*SA*SC
      ECO(J)=BCO+CCO*(SA+SC)
      BEG(J)=EXP(DCO+ECO(J)*.5*ARC(J))
7     BEND(J)=EXP(DCO+ECO(J)*.5*(ASUM+ARC(J)))
      BEG(JUP)=BEND(JEP)
      BEND(JUP)=B(JUP)
      ECO(JUP)=(2.0/ARC(JUP))*ALOG(BEND(JUP)/BEG(JUP))
      CALL INTEG (ARC,BEG,BEND,B,JEP,ECO,FLINT)
      FI=FLINT
      BB=B(2)
8     CONTINUE
      RETURN
C
C
C
9     FORMAT (79X,26H ERROR INCREASED IN  INVAR,E15.4)
      END
```

```
      SUBROUTINE START (R1,R2,R3,B,ARC,ERR,V)
      COMMON FRONT,SHEET1,SHEET2,FSHEET
      COMMON B(200),VN1(200),VN2(200),VN3(200),ARC(200),VNEAR(3),VBO(3),
     1VSAVE(3),BO,BNEAR,JUP,MMM
      COMMON VCONJ(3),ALTO
      DIMENSION V(3,3), R1(3), R2(3), R3(3)
      LOOP=1
      SIT=ABS(SIN(V(2,2)))
1     IF (V(3,2)) 2,3,3
2     V(3,2)=V(3,2)+6.283185307
      GO TO 1
3     CALL MODMAG (V(1,2),SIT,V(3,2),BR,BT,BP,B(2),V(2,2))
      R2(1)=BR/B(2)
      DN=B(2)*V(1,2)
      R2(2)=BT/DN
      R2(3)=BP/(DN*SIT)
      IS=0
4     DO 5 I=1,3
5     V(I,1)=V(I,2)-ARC(2)*R2(I)
      SIT=ABS(SIN(V(2,1)))
6     CALL MODMAG (V,SIT,V(3,1),BR,BT,BP,B(1),V(2,1))
      IF (MMM.LT.2) GO TO 7
      IF (MMM.EQ.2) GO TO 8
      IF (B(1)-B(2)) 10,10,9
7     IF (B(1)-B(2)) 9,10,10
8     IF (V(1,1)-V(1,2)) 9,9,10
9     ARC(2)=-ARC(2)
      IF (LOOP.EQ.2) GO TO 14
      LOOP=2
      GO TO 4
10    R1(1)=BR/B(1)
      ARC(3)=ARC(2)
      DN=B(1)*V(1,1)
      R1(2)=BT/DN
      R1(3)=BP/(DN*SIT)
      DO 11 I=1,3
11    V(I,1)=V(I,2)-ARC(2)*(R1(I)+R2(I))/2.
      SIT=ABS(SIN(V(2,1)))
      IS=IS+1
      GO TO (6,12), IS
12    DO 13 I=1,3
13    V(I,3)=V(I,2)+ARC(3)*((1.5)*R2(I)-.5*R1(I))
      GO TO 15
14    JUP=-1
15    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LINES (R1,R2,R3,B,ARC,ERR,J,VP,VN)
      COMMON FRONT,SHEET1,SHEET2,FSHEET
      COMMON B(200),VN1(200),VN2(200),VN3(200),ARC(200),VNEAR(3),VBO(3),
     1VSAVE(3),BO,BNEAR,JUP,MMM
      COMMON VCONJ(3),ALTO
      INTEGER FLAG1,FLAG2
      DIMENSION R1(3), R2(3), R3(3), VN(3), VP(3), RA(3)
      M=MMM
      FLAG1=0
      DEL=0.01
      CRE=0.25
      IF (ERR-0.15625) 1,2,2
1     CRE=(ERR**0.333333333)
2     A3=ARC(3)
      AAB=ABS(A3)
      SNA=A3/AAB
      A1=ARC(1)
      A2=ARC(2)
      A06=A3*A3/6.0
      VN1(2)=VP(1)
      VN2(2)=VP(2)
      VN3(2)=VP(3)
      J=3
      ILP=1
      IS=1
      GO TO 8
3     IS=1
      J=J+1
      A06=A3*A3/6.0
      ARCJ=A1+A2+A3
      AD=(ASUM+A1)/AA
      BD=ASUM/BB
      CD=A1/CC
4     DO 7 I=1,3
      DD=R1(I)/AA-R2(I)/BB+R3(I)/CC
      GO TO (5,6), IS
5     RT=R1(I)-(AD*R1(I)-BD*R2(I)+CD*R3(I)-DD*ARCJ)*ARCJ
      RA(I)=R1(I)
      R1(I)=R2(I)
      R2(I)=R3(I)
      R3(I)=RT
      VP(I)=VN(I)
6     RBAR=(R2(I)+R3(I))/2.-DD*A06
7     VN(I)=VP(I)+A3*RBAR
8     IF (VN(2)) 9,10,10
9     VN(2)=-VN(2)
10    IF (VN(2)-3.141592653) 12,12,11
11    VN(2)=6.283185307-VN(2)
      GO TO 10
12    IF (VN(3)) 13,14,14
13    VN(3)=VN(3)+6.283185307
      GO TO 12
14    IF (VN(3)-6.283185307) 16,16,15
15    VN(3)=VN(3)-6.283185307
      GO TO 14
16    GO TO (17,18), IS
17    SIT=ABS(SIN(VN(2)))
      PRE1=VN(1)
      PRE2=PRE1*VN(2)
```

```
      PRE3=PRE1*SIT*VN(3)
      CALL MODMAG (VN,SIT,VN(3),BR,BT,BP,B(J),VN(2))
      R3(1)=BR/B(J)
      DN=B(J)*VN(1)
      R3(2)=BT/DN
      R3(3)=BP/(DN*SIT)
      ASUM=A3+A2
      AA=ASUM*A2
      BB=A3*A2
      CC=ASUM*A3
      IS=2
      GO TO 4
18    SIT=ABS(SIN(VN(2)))
      IF (VN(1).GT.8.) GO TO 19
      B(J)=B(J)*((PRE1/VN(1))**3)
19    IF (M.EQ.1) GO TO 23
      QRT=.5*ABS(R3(1))/(.1+ABS(R3(2)*VN(1)))
      X=(ABS(VN(1)-PRE1)+QRT*ABS(VN(1)*VN(2)-PRE2)+ABS(VN(1)*SIT*VN(3)-P
     1RE3))/(AAB*ERR*SQRT(1.+QRT*QRT))
      GO TO (23,20,23), ILP
20    IF (X-3.3) 23,21,21
21    A3=A3*0.2*(8.0+X)/(0.8+X)
      J=J-1
      ILP=3
      ASUM=A2+A1
      AA=ASUM*A1
      BB=A2*A1
      CC=ASUM*A2
      DO 22 I=1,3
      VN(I)=VP(I)
      R3(I)=R2(I)
      R2(I)=R1(I)
22    R1(I)=RA(I)
      GO TO 37
23    VN1(J)=VN(1)
      VN2(J)=VN(2)
      VN3(J)=VN(3)
      IF (M.EQ.2) GO TO 27
      IF (M.EQ.3) GO TO 29
      IF (B(J-1).GT.B(J)) GO TO 31
      IF (M.EQ.1) GO TO 25

      IF (FLAG1.EQ.1) GO TO 31
      FLAG1=1
      DO 24 I=1,3
24    VNEAR(I)=VN(I)
      BNEAR=B(J)
      GO TO 31
25    BNEAR=B(J-1)
      DO 26 I=1,3
26    VNEAR(I)=VP(I)
      GO TO 39
27    IF (VN(1).GT.ALTO) GO TO 31
      FAC=(ALTO-VP(1))/(VN(1)-VP(1))
      DO 28 I=1,3
28    VCONJ(I)=VP(I)+(VN(I)-VP(I))*FAC
      ARC(J)=ARC(J)*FAC
      VN1(J)=VCONJ(1)
      VN2(J)=VCONJ(2)
      VN3(J)=VCONJ(3)
```

```
      GO TO 39
29    IF (B(J).LT.B0) GO TO 31
      FAC=(B0-B(J-1))/(B(J)-B(J-1))
      DO 30 I=1,3
30    VB0(I)=VP(I)+(VN(I)-VP(I))*FAC
      ARC(J)=ARC(J)*FAC
      B(J)=B0
      VN1(J)=VB0(1)
      VN2(J)=VB0(2)
      VN3(J)=VB0(3)
      GO TO 39
31    IF (J.GE.200) GO TO 39
      A1=A2
      IF (M.NE.0) GO TO 32
      IF (B(J)-B(2)) 32,32,38
32    ILP=2
      A2=A3
      IF (M.EQ.1) GO TO 37
      A3=A3*.2*(8.+X)/(.8+X)
      AM=(2.-R3(2)*VN(1))*VN(1)*CRE
      IF (ABS(A3)-AM) 34,34,33
33    A3=SNA*AM
34    IF (SNA*R3(1)+.5) 35,35,37
35    AM=-.5*SNA*VN(1)/R3(1)
      IF (ABS(A3)-AM) 37,37,36
36    A3=SNA*AM
37    ARC(J+1)=A3
      AAB=ABS(A3)
      GO TO 3
38    CONTINUE
39    RETURN
      END
```

```
      SUBROUTINE INTEG (ARC,BEG,BEND,B,JEP,ECO,FI)
      COMMON FRONT,SHEET1,SHEET2,FSHEET
      COMMON B(200),VN1(200),VN2(200),VN3(200),ARC(200),VNEAR(3),VBO(3),
     1VSAVE(3),BO,BNEAR,JUP,MMM
      COMMON VCONJ(3),ALTO
      DIMENSION BEG(200), BEND(200), ECO(200)
    1 KK=JEP
      IF (KK-4) 3,2,4
    2 KK=KK-1
    3 A=B(KK-1)/B(2)
      X2=B(KK)/B(2)
      X3=B(KK+1)/B(2)
      ASUM=ARC(KK)+ARC(KK+1)
      DN=ARC(KK)*ARC(KK+1)*ASUM
      BB=(-A*ARC(KK+1)*(ARC(KK)+ASUM)+X2*ASUM**2-X3*ARC(KK)**2)/DN
      C=(A*ARC(KK+1)-X2*ASUM+X3*ARC(KK))/DN
      FI=1.570796326*(1.-A+BB*BB/(4.*C))/SQRT(ABS(C))
      RETURN
    4 T=SQRT(1.-BEND(2)/B(2))
      FI=(2.*T-ALOG((1.+T)/(1.-T)))/ECO(2)
      IF (B(2)-BEND(KK)) 6,6,5
    5 KK=KK+1
    6 T=SQRT(ABS(1.0-BEG(KK)/B(2)))
      FI=FI-(2.*T-ALOG((1.+T)/(1.-T)))/ECO(KK)
      KK=KK-1
      DO 15 I=3,KK
      ARG1=1.-BEND(I)/B(2)
      IF (ARG1) 7,7,8
    7 TE=1.E-5
      GO TO 9
    8 TE=SQRT(ARG1)
    9 ARG1=1.-BEG(I)/B(2)
      IF (ARG1) 11,11,10
   10 TB=SQRT(ARG1)
      GO TO 12
   11 TB=1.E-5
   12 IF (ABS(ECO(I))-2.E-5) 13,13,14
   13 FI=FI+((TE+TB)*(ARC(I)+ARC(I+1)))/4.
      GO TO 15
   14 FI=FI+(2.*(TE-TB)-ALOG((1.+TE)*(1.-TB)/((1.-TE)*(1.+TB))))/ECO(I)
   15 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MODMAG (RR,SINTH,PPHI,BR,BTHETA,BPHI,BB,THET)
      COMMON FRONT,SHEET1,SHEET2,FSHEET
C
C     SUBROUTINE ASSEMBLES MAGNETIC FIELD FROM TAIL, MAGNETOPAUSE AND
C     INTERNAL DIPOLE - FOR ENQUIRIES WRITE TO GILBERT MEAD, GODDARD
C     SPACE FLIGHT CENTER, GREENBELT MARYLAND 20771
C
      DIMENSION GG(7,7)
      DIMENSION G(7,7), CONST(7,7), P(7,7), DP(7,7), SP(7), CP(7)
      COSTH=COS(THET)
      SINPHI=-SIN(PPHI)
      COSPHI=-COS(PPHI)
      R0=FRONT
      R1=SHEET1
      R2=SHEET2
      BCS=FSHEET
      IF (JFIRST-13) 1,5,1
1     JFIRST=13
C
C     SET UP INITIAL CONSTANTS THE FIRST TIME AROUND
C
      DO 2 N=1,7
      DO 2 M=1,7
2     GG(N,M)=0.
      NMAX=7
C
C     THE FOLLOWING COEFFICIENTS ARE SCHMIDT-NORMALIZED
C
      GG(2,1)=-0.25111E5
      GG(3,2)=-0.12424E5
      GG(4,1)=-0.00716E5
      GG(4,3)=-0.02333E5
      GG(5,2)=-0.02397E5
      GG(5,4)=-0.00163E5
      GG(6,1)=0.00569E5
      GG(6,3)=-0.01078E5
      GG(6,5)=-0.00103E5
      GG(7,2)=0.00126E5
      GG(7,4)=-0.00187E5
      GG(7,6)=-0.00041E5
      P(1,1)=1.0
      DP(1,1)=0.
      SP(1)=0.
      CP(1)=1.
      DO 3 N=3,NMAX
      FN=N
      N2=N-2
      DO 3 M=1,N2
      FM=M
3     CONST(N,M)=((FN-2.)**2-(FM-1.)**2)/((2.*FN-3.)*(2.*FN-5.))
      DIMENSION SHMIDT(7,7)
      SHMIDT(1,1)=1.0
      DO 4 N=2,7
      FN=N
      SHMIDT(N,1)=SHMIDT(N-1,1)*(FN+FN-3.0)/(FN-1.0)
      FACT=2.0
      DO 4 M=2,N
      FM=M
      SHMIDT(N,M)=SHMIDT(N,M-1)*SQRT((FN-FM+1.0)*FACT/(FN+FM-2.0))
```

```
4       FACT=1.0
5       IF (RO-ROOLD) 6,9,6
6       ROOLD=RO
        DIMENSION FAC(7)
        FAC(2)=RO**3
        DO 7 N=3,NMAX
7       FAC(N)=RO*FAC(N-1)
        DO 8 N=2,NMAX
        DO 8 M=1,N
8       G(N,M)=SHMIDT(N,M)*GG(N,M)/FAC(N)
9       CONTINUE
C       BEGIN CALCULATION FOR SPECIFIED INPUT
C
        CT=COSTH
        ST=SINTH
        SP(2)=SINPHI
        CP(2)=COSPHI
C
C       CALCULATE SIN(M*PHI) AND COS(M*PHI)
C
        DO 10 M=3,NMAX
        SP(M)=SP(2)*CP(M-1)+CP(2)*SP(M-1)
10      CP(M)=CP(2)*CP(M-1)-SP(2)*SP(M-1)
        R=RR
        A=1.
        RA=R/A
        ROA=1.
        BR=0.
        BTHETA=0.
        BPHI=0.
        FN=1.
C
C       CALCULATE SPHERICAL HARMONICS FOR CAVITY FIELD
C
        DO 16 N=2,NMAX
        SUMR=0.
        SUMT=0.
        SUMP=0.
        FM=0.
C
C       DEVELOP LEGENDRE FUNCTIONS AND THEIR DERIVATIVES BY RECURSION FORM
C
        DO 15 M=1,N
        IF (N-M-1) 13,12,11
11      P(N,M)=CT*P(N-1,M)-CONST(N,M)*P(N-2,M)
        DP(N,M)=CT*DP(N-1,M)-ST*P(N-1,M)-CONST(N,M)*DP(N-2,M)
        GO TO 14
12      P(N,M)=CT*P(N-1,M)
        DP(N,M)=CT*DP(N-1,M)-ST*P(N-1,M)
        GO TO 14
13      P(N,N)=ST*P(N-1,N-1)
        DP(N,N)=ST*DP(N-1,N-1)+CT*P(N-1,N-1)
14      CONTINUE
        TS=G(N,M)*CP(M)
        SUMR=SUMR+P(N,M)*TS
        SUMT=SUMT+DP(N,M)*TS
        SUMP=SUMP+FM*P(N,M)*G(N,M)*SP(M)
        FM=FM+1.
15      CONTINUE
        BR=BR-ROA*FN*SUMR
```

```
      BTHETA=BTHETA-ROA*SUMT
      BPHI=BPHI+ROA*SUMP
      ROA=ROA*RA
      FN=FN+1.
16    CONTINUE
      BPHI=BPHI/ST
C
C     CALCULATE TAIL FIELD
C
      RCT=R*CT
      RCT2=RCT**2
      RSC=R*ST*CP(2)
      TOP=R2+RSC
      BOT=R1+RSC
      BX=-BCS*(ATAN(TOP/RCT)-ATAN(BOT/RCT))/3.14159265
      BPHI=BPHI+BX*SP(2)
      BRHO=-BX*CP(2)
      BY=BCS*ALOG((RCT2+TOP**2)/(RCT2+BOT**2))/6.28318531
      BR=BR+BRHO*ST-BY*CT
      BTHETA=BTHETA+BRHO*CT+BY*ST
C
C     ADD DIPOLE FIELD TO CAVITY FIELD
C
      R3=R**3
      BR=BR-62000.*COSTH/R3
      BTHETA=BTHETA-31000.*SINTH/R3
      BR=BR*1.E-05
      BTHETA=BTHETA*1.E-05
      BPHI=BPHI*1.E-05
      BB=SQRT(BR*BR+BTHETA*BTHETA+BPHI*BPHI)
      RETURN
      END
```