AFWL-TR-66-78

AFWL-TR
66-78

639436

A COMPUTATIONAL MODEL FOR GEOMAGNETICALLY TRAPPED PARTICLE SHELLS AND KINEMATIC PARAMETERS

Juan G. Roederer
Goddard Space Flight Center
Greenbelt, Maryland

James V. Herod, Captain, USA
Air Force Weapons Laboratory
Kirtland Air Force Base, New Mexico

TECHNICAL REPORT NO. AFWL-TR-66-78

September 1966

AIR FORCE WEAPONS LABORATORY
Research and Technology Division
Air Force Systems Command
Kirtland Air Force Base
New Mexico

AFWL-TR-66-78

Research and Technology Division
AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base
New Mexico

AFWL-TR-66-78

A COMPUTATIONAL MODEL FOR GEOMAGNETICALLY TRAPPED PARTICLE SHELLS AND KINEMATIC PARAMETERS

Juan G. Roederer*
Goddard Space Flight Center
Greenbelt, Maryland

James V. Herod, Captain, USA
Air Force Weapons Laboratory
Kirtland Air Force Base, New Mexico

TECHNICAL REPORT NO. AFWL-TR-66-78

*National Academy of Sciences-National Research Council, Senior Post-Doctoral Research Associate, on leave from Facultad de Ciencias Exactas y Naturales and Centro Nacional de Radiacion Cosmica, Buenos Aires, Argentina.

FOREWORD

This research was performed under Program Element 6.16.46.01.D, Project 5710, Subtask 07.047, and was funded by the Defense Atomic Support Agency (DASA).

Inclusive dates of research were October 1965 to June 1966. The report was submitted 14 July 1966 by the AFWL Project Officer, Captain James V. Herod, U.S. Army, (WLRTH).

This report has been reviewed and is approved.

JAMES V. HEROD
Captain USA
Project Officer

RALPH H. PENNINGTON
Colonel USAF
Chief, Theoretical Branch

CLAUDE K. STAMBAUGH
Colonel USAF
Acting Chief, Research Division

ABSTRACT

A computer program has been constructed which calculates a model for geomagnetically trapped particle shells and the associated kinematic parameters. Included is the calculation of longitude-dependent drift velocities and bounce paths. This document explains the methods used in constructing the program as well as the input and output of the code.

This page intentionally left blank.

CONTENTS

ILLUSTRATIONS

SECTION I

INTRODUCTION

The detailed study of time and longitude dependence of energetic particle fluxes injected at low altitudes into geomagnetically trapped orbits requires an accurate description of the geometry of the magnetic shells populated by these particles, as well as the determination of the longitude dependence of all intervening kinetic variables, such as drift velocities and bounce periods.

For particles mirroring at low altitudes in the South American Anomaly, where interactions with the atmosphere are the dominant processes governing particle diffusion and loss, changes of a few kilometers in altitude of magnetic shells may induce quite appreciable changes in the numerical results of theoretical diffusion and loss calculations. Considerable accuracy is therefore required. Of course, there will always be a "natural" limitation imposed by the degree of accuracy in the geomagnetic field description. However, there are other more or less controllable sources of errors, both systematic and random, which are inherent to the numerical methods used in shell geometry and particle kinematics computations, and which must be reduced to a minimum.

Particle shell calculations require lengthy computations, consisting mainly of field line tracings, calculations of the second adiabatic invariant, and search for B-L points. This last procedure, in particular, is the most critical one for the geometric determination of a particle shell, and is where the problem of precision is most acute. Furthermore, exact determination of drift velocities and bounce periods, and their longitude dependence, also implies lengthy computations. All this makes the problem of speed a most critical one.

In the present paper, a computer code is described which meets to some degree all these requirements of accuracy and speed, furnishing complete information about particle shells, drift velocities, and bounce periods, and which is flexible enough to be easily used under varying initial conditions or output requirements.

SECTION II

PARTICLE SHELL GEOMETRY

A charged particle moves in a time independent, trapping magnetic field conserving two invariants (references 1 and 2): the mirror point field intensity B_m, and the value of the integral I between conjugate mirror points; i.e.,

$$B_m = \text{const.} \tag{1a}$$

$$I = \int \sqrt{1 - \frac{B(s)}{B_m}}\, ds = \text{const.} \tag{1b}$$

To each point in space, a pair of values, B_m and I, can be attached, such that a particle mirroring there has the value I for the integral (equation 1b), B_m being the field intensity at that point. In a trapping field configuration, like the geomagnetic field, we can trace constant I and constant B surfaces (see sketch in figure 1). A given particle will therefore always spiral along the field lines passing through the ring-shaped intersections of such a pair of B_m, I surfaces, mirroring on the intersection and drifting in azimuth.

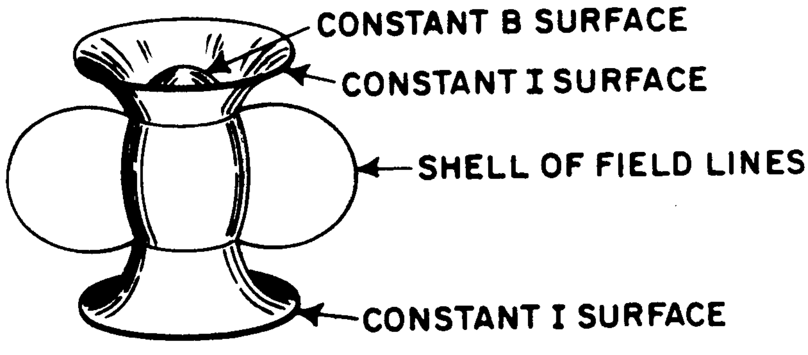

Figure 1. Geometric Definition of a Particle Shell.

McIlwain (reference 2) has shown that, in the real geomagnetic field, azimuthal asymmetry is small enough so that, to a fair approximation, particles initially mirroring on a common field line will always spiral along the same field lines at other longitudes, irrespective of their initial mirror points (or equatorial pitch angles). In other words, they will populate one and the same shell. McIlwain has found a relation $L = L(I,B_m)$ such that it has nearly the same value for all points along a given field line, and, therefore, for all field lines belonging to the shell generated by particles mirroring on the initial line. This parameter identifies this shell, and is taken as its proper geometric label. Physically, it gives the average distance in earth radii of the equatorial (minimum B) points of the shell to the geomagnetic center.

When there is a strong azimuthal asymmetry, as in the outer magnetosphere (reference 3), this shell "degeneracy" is removed (references 4 and 5), and particles initially mirroring on the same field line will populate different shells (shell splitting) according to their different mirror points or equatorial pitch angles. This shell splitting, of course, also does occur in the inner magnetosphere; however, it amounts to only a few kilometers at the equator, for L-shells of the order of 1.2. Nevertheless, following the discussion above, such a difference of a few kilometers may be of importance to mechanisms of atmospheric interactions, where it may compete with the scale height of atmospheric density. Special care must be taken, therefore, when a particle shell is to be defined for atmospheric scatter studies. For instance, if one is interested in the evolution of a group of particles mirroring far away from the equator, i.e., at high B-values or low altitudes in the Anomaly, the field lines on which these particles will mirror must be defined by the proper B_m, L (or B_m, I) ring, and not, for example, by the equatorial ring of the corresponding L-shell. The differences which result from picking shells by their equatorial points instead of some higher B-value points may be as much as 10-20 kilometers in the region of the Anomaly. The corresponding differences in the longitude dependence of drift velocities and bounce periods are, percentage wise, even higher (6 to 10 percent for $L = 1.25$); finally, the equatorial B value of a shell picked at a high B value becomes quite appreciably longitude-dependent (up to 6 to 7 percent for $L = 1.25$), as does the equatorial pitch angle of a given particle (although this is constant on a dipole shell).

Most of the longitude variations occur in the South American Anomaly (reference 6), where most of the atmospheric interactions also take place. Unfortunately, this is the region where the higher multipoles of the field expansion are felt most intensely, being therefore more exposed to the systematic errors in the particular field description used.

SECTION III

DETERMINATION OF PARTICLE SHELLS AND KINEMATIC PARAMETERS

The following procedure was adopted in the computer code to determine a given L-shell. First, a convenient B-value was chosen, which was centered in the B region to be explored. In general, for the study of atmospheric interactions of trapped particles, it is recommended to take a B value such that the corresponding southern B-L ring dips to about 250 to 300 kilometers in the Anomaly (for $L = 1.25$, choose $B = 0.22$).

The main numerical problem then is to find a point of prefixed B and L values, at any given longitude. An iteration method must be used, for in the case of a nondipole expansion there is no known analytical expression of L as a function of space (nor as an expansion in terms of the Gauss coefficients). The determination of L (or, really, I) always implies an integration along the field line; therefore, this is the critical part of the entire program with regard to speed and accuracy.

Once a B-L point is determined at a given longitude, information about the field line going through it must be provided. For particular atmospheric scattering problems, this field line must be traced to the intersection with a prefixed altitude level, say 100 kilometers, below which no trapping is possible. Coordinates of points of this field line may be needed either at a prefixed mesh of B-values, or at prefixed altitudes on the lower end of the field line (and their conjugate points on the other side). All this is usually wanted at many different longitudes until a grid of field lines of the shell is completed around the world. The present code was designed to provide this information.

As to the drift velocity and bounce period calculations, the following procedures were adopted. The most general expression for the average equatorial drift velocity u_o was given by Northrop and Teller (reference 7). Using a slightly different notation, we have

$$u_o = \frac{m \cdot v}{e \cdot B_o} \cdot \frac{\nabla_o I}{S_b} \tag{2}$$

where m is the particle's mass, e its electric charge, and v its velocity. B_o is the field intensity at the equatorial point of the shell, and $\nabla_o I$ is the gradient of I (equation 1b) calculated for a fixed mirror point field B_m. S_b is the rectified path of the particle between mirror points:

$$S_b = \int \frac{ds}{\sqrt{1 - \frac{B(s)}{B_m}}}$$

For the angular equatorial drift velocity $\dot{\phi}_o$, we have

$$\dot{\phi}_o = \frac{\partial \phi}{\partial x} \cdot u'_o \tag{3}$$

where $\partial\phi/\partial x$ is computed along the equatorial ring (dx is the element of arc of this ring). In the real geomagnetic field, all quantities B_o, $\nabla_o I$, and S_b are longitude dependent. Though S_b is defined by an integration (with an integrand which diverges at the limits of integration), one can easily show, by taking the derivative of equation (1b) with respect to B_m, that

$$S_b = I + 2\, B_m \frac{\partial I}{\partial B_m} \tag{4}$$

The derivative is to be taken along a given field line. This relation is far more convenient for computational purposes than the integral expression for S_b.

For particles mirroring on a given field line, drift velocities and bounce paths depend on their mirror points. It was found that, within the degree of accuracy wanted, this dependence can always be expressed in the form

$$u'_o = d(L,\phi) + \sigma_u(L,\phi)\left[1 - \sqrt{\frac{B_o}{B_m}}\right] \tag{5}$$

$$S_b = p(L,\phi) + \sigma_s(L,\phi)\left[1 - \sqrt{\frac{B_o}{B_m}}\right]$$

provided that B_o/B_m is greater than about 0.5. The coefficients d, p, σ_u, and σ_s are all longitude-dependent. However, their longitude averages are remarkably close to the corresponding dipole values (reference 8). Care must be taken in interpreting d and p as drift and bounce path values for particles mirroring close to the equator. If the field lines of the shell for which the quantities in equation (5) are computed are picked at a point with a high B value, d and p would <u>not</u> represent exactly the drift and bounce parameters of equatorial particles, because these, even for the same L value, would be drifting along a slightly different shell. This slight difference in altitude may result in quite considerable longitude variations of u_o and S_b.

SECTION IV

EXPLANATION OF THE PROGRAM

In what follows, the computer code will be briefly described in a logical order, starting with the key subroutine, rather than with the main program. A listing is given in the appendix.

1. SEARCH

Subroutine SEARCH provides the foundation for the entire program. It consists of an iteration process which finds a point P with a prefixed B and L value 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 geodetic coordinates of the starting point for the iteration process, the specified B and L values, and the prefixed tolerances for each of B and L. Outputs are the geodetic coordinates of the desired B-L point, and the geocentric coordinates of a series of points along the field line between the point P and its conjugate. The entire calculation is done in geocentric coordinates.

There are two basic processes in this subroutine: one varies altitude 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 L. The first of these processes is very fast; it involves only calling the subroutine NEWMAG which computes the intensity of 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. The program was constructed in such a manner as to minimize the number of times INVAR is used.

Figure 2 shows a simplified flow chart for subroutine SEARCH. Geometrically, the procedure used is as follows (see figure 3): starting at the input point P, the program iterates the altitude to a point S where the prefixed B-value (called SB) is reached within a prefixed, gross tolerance designated by ERRB. The L value is then determined for this point S. If it is not equal to the prefixed L-value (called SL) within the tolerance designated by ERRL, the latitude is changed by a small, prefixed amount DV in the correct direction. At this new latitude, a point T is found with the prefixed B-value; L is calculated at the point T.

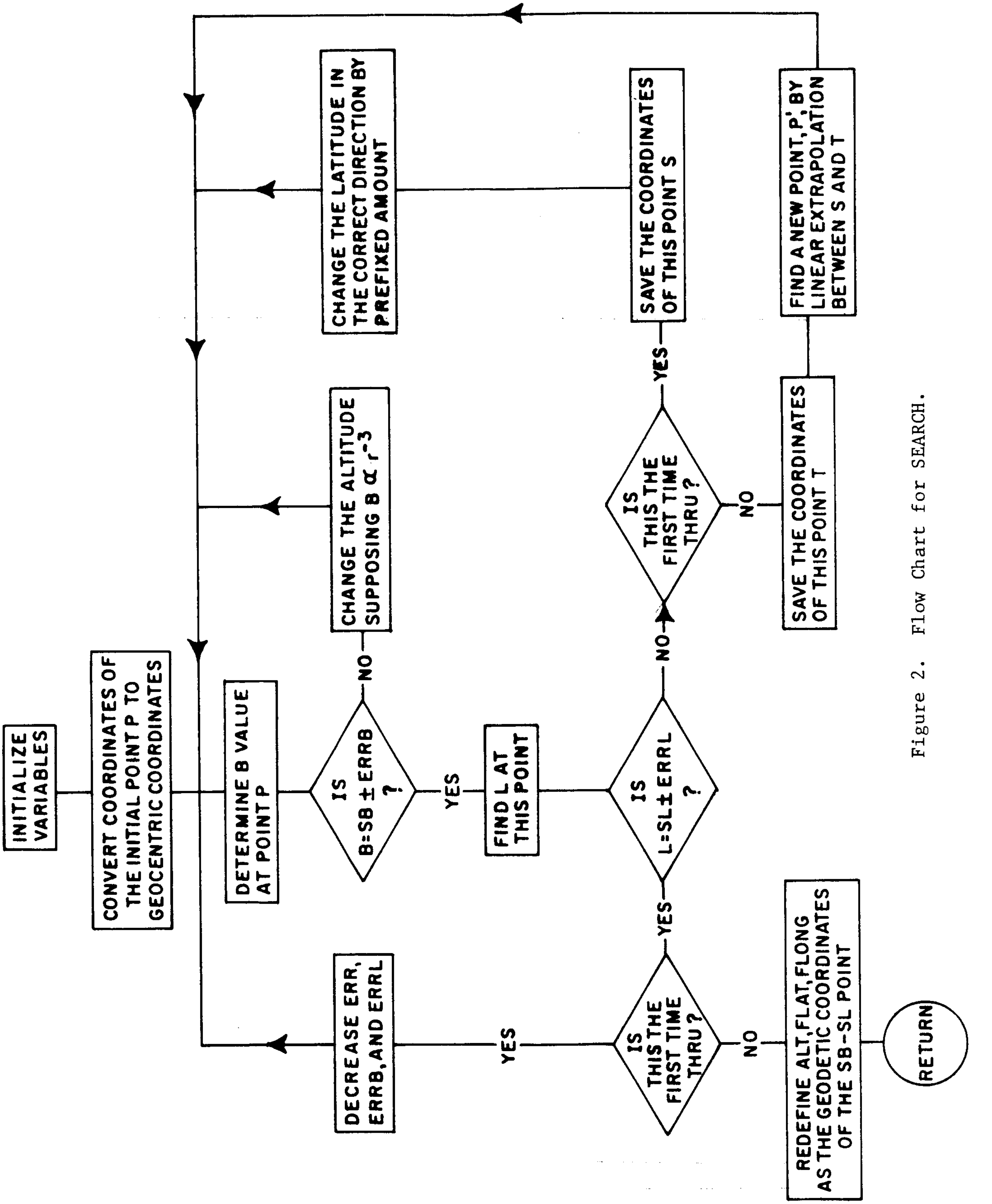

Figure 2. Flow Chart for SEARCH.

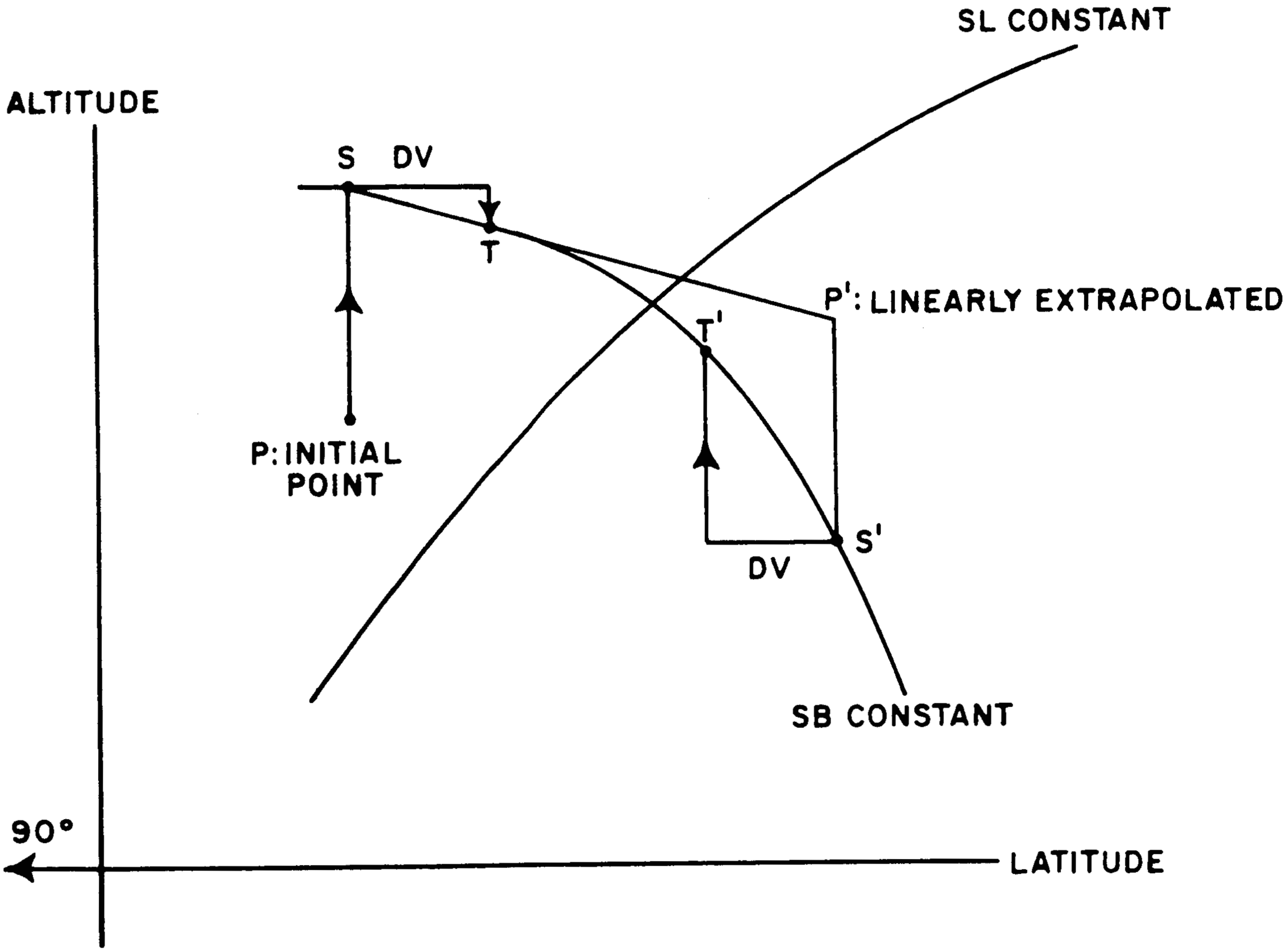

Figure 3. Iteration Procedure Followed in Subroutine SEARCH.

Points S and T 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 SL. This extrapolation is made as though the L-dependence were linear along the constant B line. This extrapolated point P' is now taken as a second approximation and is used as the starting point of the next iteration cycle. If the new point P' is close enough to the points S and T, the step size in latitude is reduced by one-tenth. The iteration is continued until a point is found which lies in the interval SB±ERRB and SL±ERRL. When this is accomplished, all tolerances are reduced two orders of magnitude and the entire procedure is repeated until a point is found whose B and L values agree with the prefixed ones within the new, reduced tolerance intervals.

To avoid the possibility of infinite cycling in case the tolerances were too small, a cutoff is introduced and a diagnostic is printed. The process is begun again at a slightly different value of SB (see section IV,11).

It should be noted that the input coordinates of the starting point P determine the longitude of the desired B-L point as well as the hemisphere in which it will lie. Indeed, the value of the longitude is never changed in this subroutine. It is recommended, in general, that the initial guess or starting point for the iteration process should not differ in latitude by more than 10 or 15 degrees from the real B-L point. This may be accomplished by consulting the usual B-L maps (references 9 and 10).

2. INVAR, START, LINES, INTEG, CARMEL, AND NEWMAG

This group of subroutines is slightly modified and implemented versions of McIlwain's program INVAR (reference 11). Their purpose is to trace a field line (START, LINES, and NEWMAG) and to determine the second invariant (equation (1b)) (INTEG) and the associated L-value (CARMEL). Input for INVAR is a point P in space given by geocentric coordinates; output are the B and L values at P and the geocentric 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-value 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 equals or exceeds that at P.

The arc length between consecutive points in the sequence is, in general, a linear function of the geocentric altitude of the two 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 value. The result is speed and accuracy at the expense of a nonconstant and sometimes unpredictable cell size.

INTEG and CARMEL are straightforward programs which were left intact from McIlwain's code. NEWMAG is also practically unchanged; it contains the latest Cain et al. (reference 12) field coefficients (truncated to 48) with an option for January 1960 or November 1966.

3. EQUAT

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

4. BESECT

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

5. INSECT

This subroutine traces a field line from a given point downwards to the intersection with a geodetic altitude. It should be noted that this subroutine can be used for the determination of geomagnetically conjugate points.

All operations mentioned in connection with subroutines EQUAT, BESECT, and INSECT are actually executed by START, LINES, and NEWMAG. Information about the particular type of tracing wanted is transferred by a parameter called MMM which is defined to be 0 in INVAR, 1 in EQUAT, 2 in BESECT, and 3 in INSECT.

6. BLRING

This subroutine finds the B-L point for a given longitude, gives the geocentric coordinates of a sequence of points along the field line between this point and its conjugate, and computes all geometric factors required for the determination of the longitude and B dependence of the particle drift velocity and the bounce period.

To accomplish this, the subroutine actually determines four field lines m, n, o, and p going through the points M, N, O, and P labeled $M(B',L + \delta L)$, $N(B',L)$, $O(B,L + \delta L)$, and $P(B,L)$, where $B' < B$ and $\delta L \ll L$ (see figure 4). The last point P is the true B-L ring point; the other three are needed in the calculation of the drift and bounce factors given in equations (2), (4), and (5). The procedure is as follows: for a given longitude, the field line through the point M is determined by calling SEARCH and EQUAT. Then the field line through the point N is determined, as well as the value of $\partial I/\partial B$, by calling INVAR at a point of the field line very close to N. Exactly the same procedure is repeated for the pair of points O and P. (Of course, the field line pairs m,o and n,p will be almost superposed.) The gradient of the second invariant is then computed for the pairs of field lines m,n and o,p. A correction for the longitude displacement of the equatorial point at the two different L shells is made to obtain a better estimate of the perpendicular distance between the equatorial rings of the two shells L and $L + \delta L$. Finally, the geomagnetic quantities appearing in equations (2) and (4) are calculated for the B-values B and B', and the factors d, p, σ_u, and σ_s, giving the mirror dependence (5), from these.

7. TRACE

This subroutine completes the field line tracing done in BLRING down to point R at a prefixed geodetic altitude level called ALTO, in one hemisphere (see figure 5). In the other hemisphere, it continues the tracing to the point C which is conjugate to R (i.e., which has the same B-value).

After calling BLRING, the first operation in TRACE is to insert the equatorial point which has been accurately determined in EQUAT. If P and Q are the end points of the field line traced in BLRING, TRACE chooses the one, say P, with the lowest altitude and traces the field line from there on downwards by calling INSECT. Special provision is made for the case that the lower of P and Q should fall below ALTO. Subroutine BESECT is then called at the point Q at the other end of the field line. The field line is then traced to a point C where the field intensity

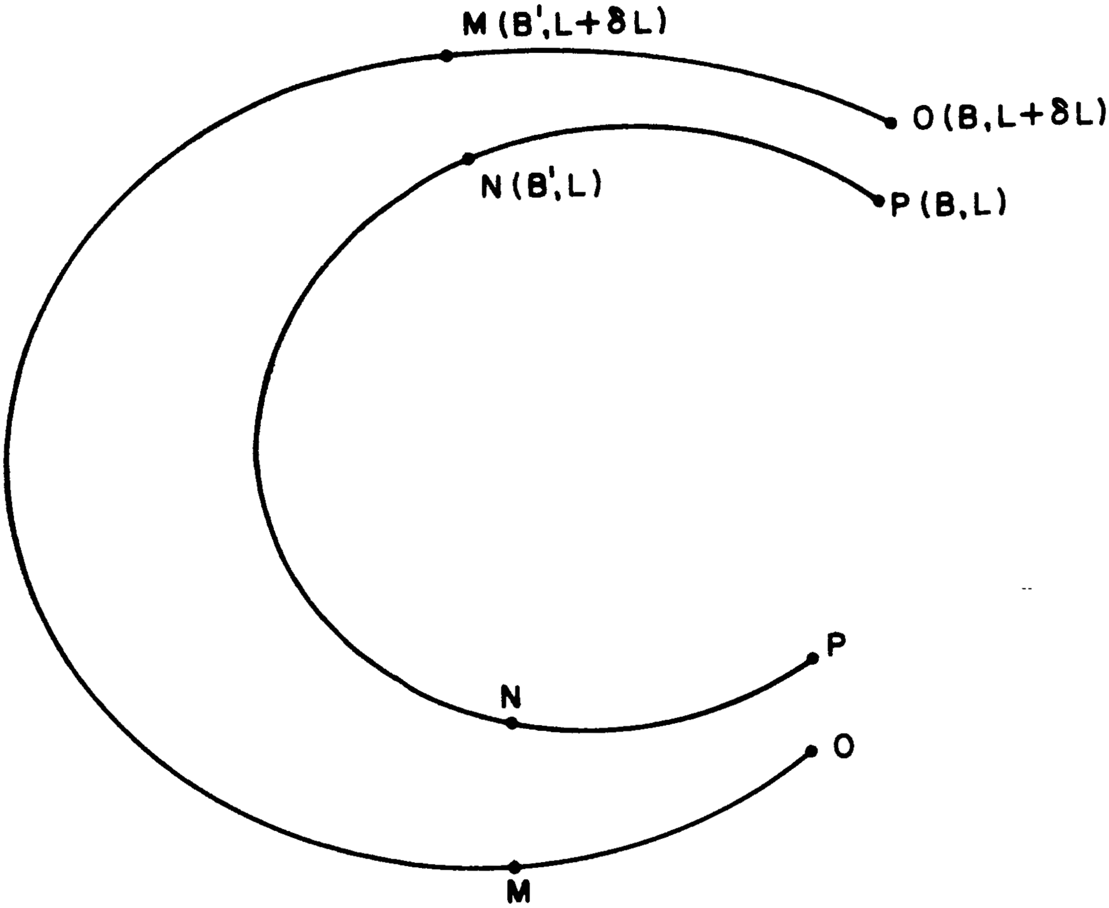

Figure 4. Pairs of Field Lines Traced for the Computation of Drift Velocity and Bounce Path.

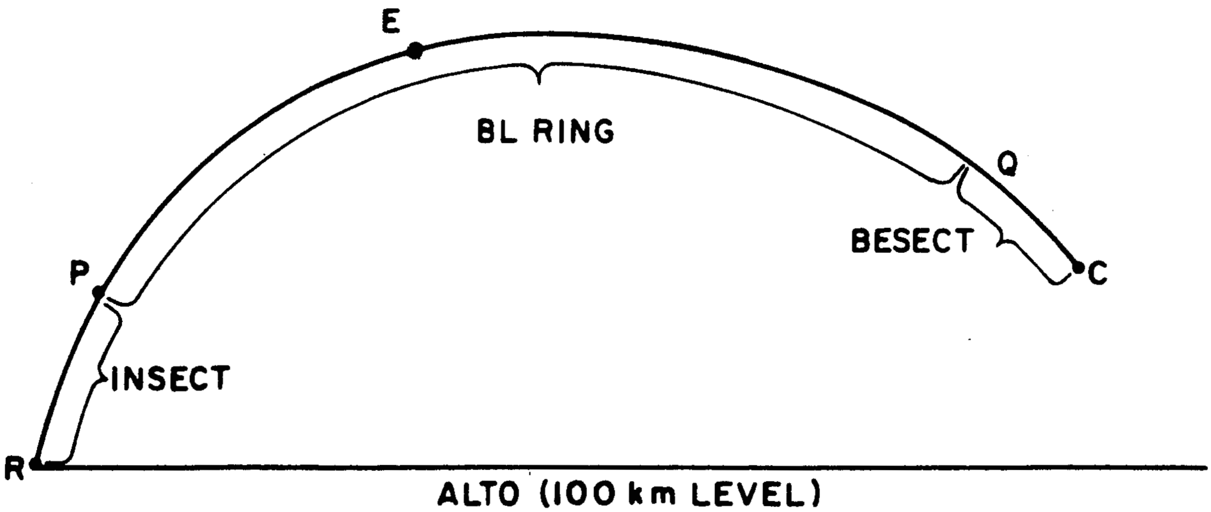

Figure 5. Portions of the Field Line Assembled by Subroutine Trace.

is the same as at R. In some cases, the point C may fall below ALTO; in that case, the points from which INSECT and BESECT trace are reversed, and the entire procedure is repeated.

In TRACE, all field line point coordinates are converted into geodetic coordinates. A flow chart for this subroutine is given in figure 6.

This subroutine can be modified conveniently to obtain tracing of the field line to the ALTO level in both the northern and the southern hemisphere (reference 14).

8. SHELL

SHELL is the first subroutine in the program and has five primary purposes: (1) read data and initialize the necessary parameters, (2) serialize the three portions of the field line data obtained in TRACE, (3) define the sequence in B (called the X-mesh) at which the field line data is desired, (4) interpolate the field line data onto this X-mesh, and (5) after editing the results, advance in longitude to the next field line until the shell is completed.

Data which are read into the program are the B-L coordinates (called SB and SL, respectively) of the point through which the field lines of the shell are to be traced, the altitude, latitude, and longitude of the point used as the starting point in the iteration technique of subroutine SEARCH, the step in longitude between successive field lines, the frequency of printing a detailed edit, and the controls for the B-mesh to be used. A full explanation of the input data may be found in section IV, 10.

Subroutine TRACE is called and the field line geometry is computed. Included in this geometry are the three sequences of points along three distinct portions of the field line (see figure 5). These three sequences do not advance in the same direction along the field line. Subroutine SHELL rearranges these three sequences of points into one monotonic sequence from point R to point C.

Two options for the X-mesh are provided. One defines a mesh of prefixed B-values, equally spaced. The other, though more complicated in conception, is specifically designed for studies of longitude dependence of trapped particles interacting with the atmosphere. In this option, the X-mesh is defined in such a way as to give B-values which correspond to points of prefixed altitudes, roughly proportional to the scale height, on the lower portion of the field line (portion E-P-R in figure 5). However, due to the longitude dependence of the

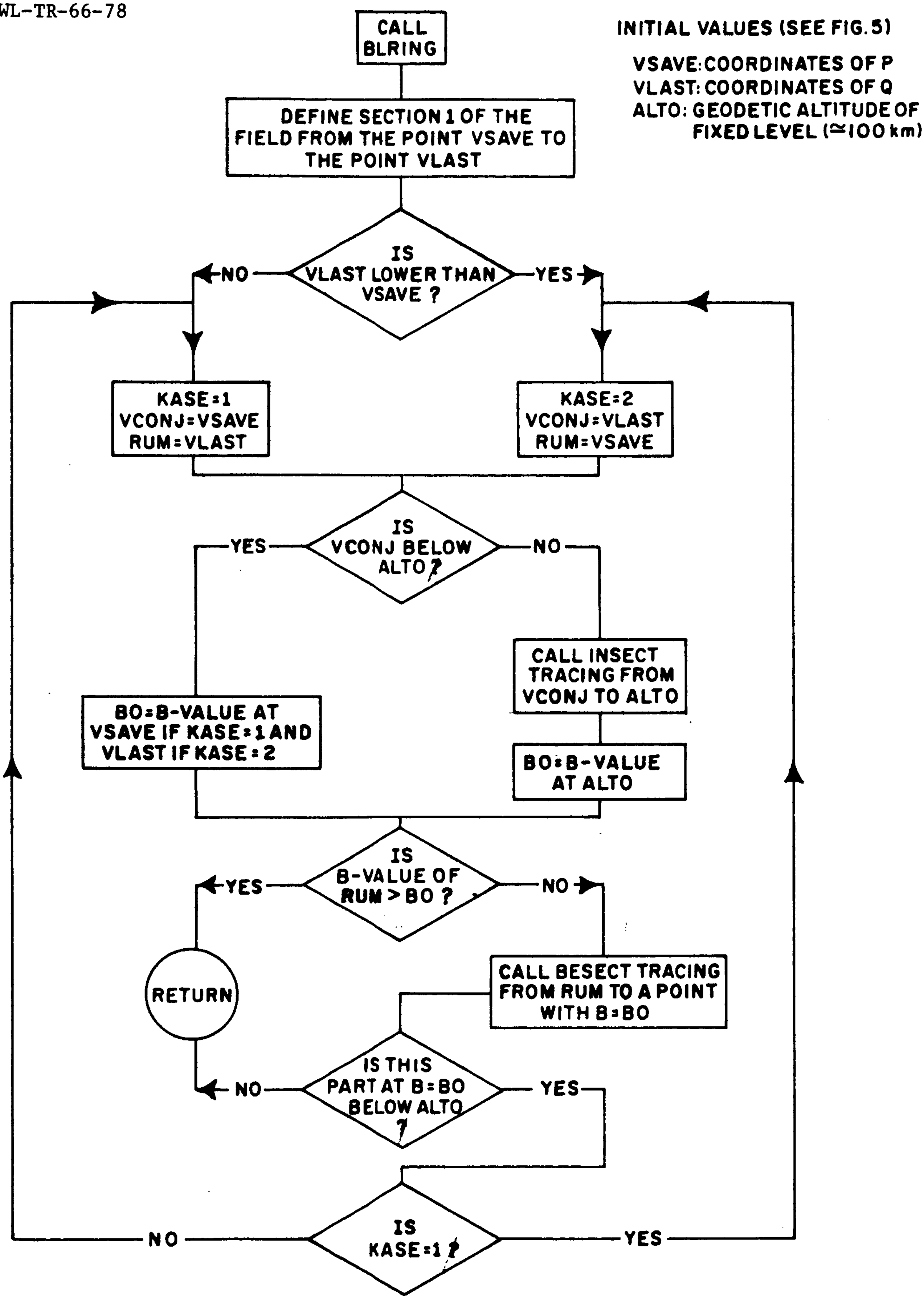

Figure 6. Flowchart for TRACE.

shell geometry, not all points of the mesh can be fixed in this way. Some mesh points have to be defined in a different manner. In the present code, this is done as follows: the first mesh point is always the equatorial point.* The third through twelfth points have prefixed, longitudinally independent B-values which are conveniently arranged as a function of the shell parameter L; the second point is inserted according to the actual value of the equatorial point. For the section of the X-mesh with prefixed altitudes, the user can select the number of points and the maximum and minimum altitudes between which these points should be placed. All remaining mesh points will be filled in between the twelfth point and the highest of the prefixed altitude points (see figure 7). This latter section of the mesh varies with longitude; its points have an accordion-like behavior, expanding and contracting as a function of longitude. The total number of mesh points is constant for the entire shell. Special care should be taken to avoid too big, sudden changes in step size. It should be noted that, for atmospheric scattering problems, a change in step size is less harmful if it occurs at higher altitudes.

9. SPLINE and YSPLN

These subroutines provide a very fast and accurate interpolation scheme (reference 13).

Let (x_i,y_i), $i=1,2,...,n$ be a sequence of number pairs having the property that $x_1 < x_2 < ... < x_n$. Subroutines SPLINE and YSPLN determine a function f defined on the number interval $[x_1,x_n]$ such that $f(x_i) = y_i$ and, on the interval $[x_{i-1},x_i]$, the function f is a cubic polynomial. Moreover, each of the derivatives f' and f'' exists and is continuous on $[x_1,x_n]$. Input for SPLINE is the sequence x, the sequence y, and the number of points in these sequences. SPLINE returns the sequence Y2DOT which is f'' evaluated at each value of x_i. If z is a number, subroutine YSPLN evaluates f(z). Input for YSPLN is the sequence x, the sequence y, the number of points, the sequence Y2DOT, and the number z. Subroutine YSPLN finds an integer j such that $x_{j-1} < z \leq x_j$ and then determines f(z) by using the cubic equation defining f on the interval $[x_{j-1},x_j]$.

*Remember that in the real field, the B-value at the equator of a shell defined by a B-L ring off the equator depends on the longitude.

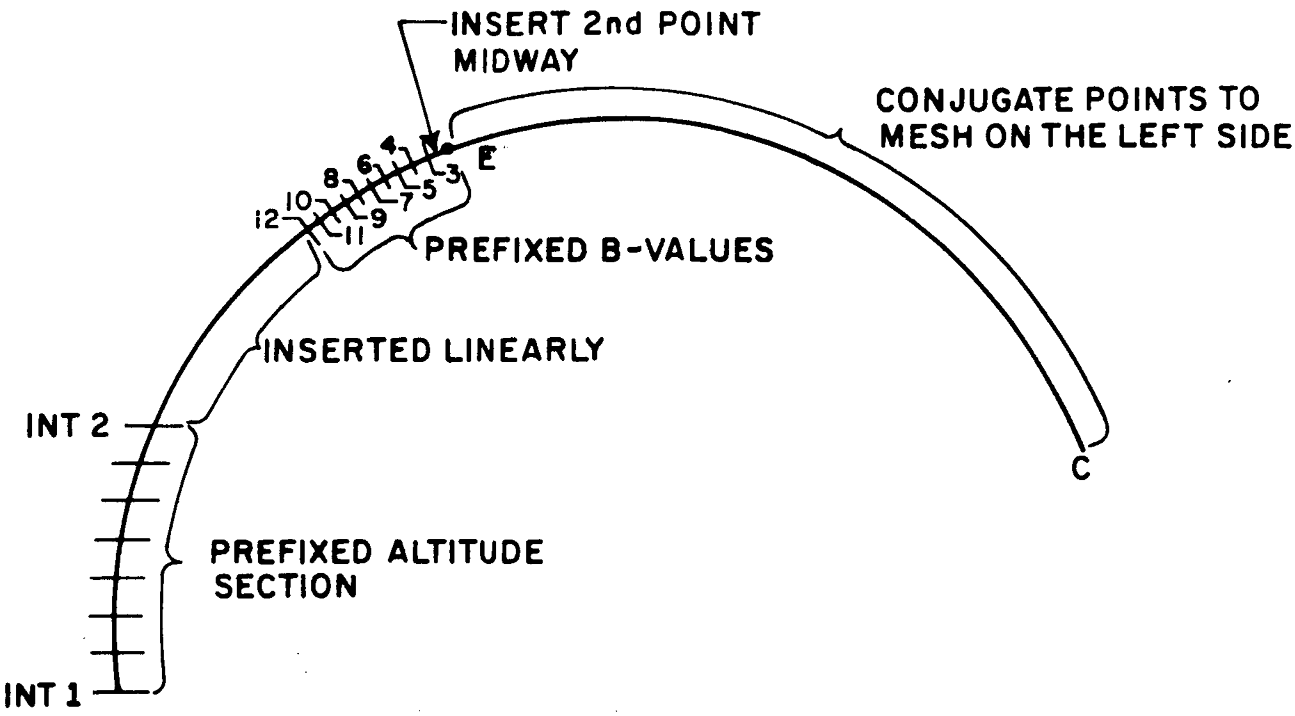

Figure 7. Definition of the X-mesh When Altitudes are Prefixed.

10. <u>DATA</u>

Below is a list of the input data together with a brief explanation of their functions and limitations:

SB, SL — The field line which is to be traced shall go through a point whose field intensity is SB and shall belong to the shell for which the McIlwain parameter has the value SL. The units for SB are gauss and the units for SL are Earth radii. In principle there are no restrictions for the values of SB and SL, except that the SB-SL point should lie above the altitude of ALTO. Of course, beyond a value of about 5 for SL, external sources of the field must be taken into consideration, and shell splitting invalidates the entire program (see section II).

ALT, FLAT, FLONG — These are the geodetic coordinates of a point, in kilometers and degrees, respectively. The positive direction for longitude is east of Greenwich. Subroutine SEARCH will use this point as the initial value in the iteration technique to find a point in the meridian FLONG for which B = SB and L = SL. The value of FLAT determines the hemisphere in which the B-L point shall lie. It was found that, in view of the considerable field distortions in the South American Anomaly, it is convenient to define the L-shells by the B-L ring points in the Northern hemisphere. FLAT should have absolute value different from 90 degrees; it should also be more than 5 to 10 degrees away from the geomagnetic equator. In general, it should not differ by more than 10 to 15 degrees from the real SB-SL point. Standard B-L maps (reference 9 and 10) can aid the user of the programer in selecting the initial value of ALT and FLAT.

SLONG1, SLONG2, SLONG3, SLONG4 — These are the longitudes at which the longitude step size is changed.

DLONG1, DLONG2, DLONG3 — These are the increments for the longitude step in degrees. After the geometry for a field line is calculated, the new field line is determined by changing the value of FLONG. The increment for FLONG is DLONG1 between SLONG2 and SLONG3, DLONG2 in the region between SLONG1 and SLONG2 and between SLONG3 and SLONG4, and DLONG3 elsewhere.

NPR — NPR is a positive integer which controls the frequency of printing. Detailed edit is given every NPR field lines of the shell.

MESHB — MESHB is used as a control for the choice of mesh of B-values to be used. If MESHB = 1, the mesh at which the field line data are to be interpolated is given as B-values of prefixed altitude points at one end of the field line. If MESHB = 0, then the mesh is prefixed with a constant step size in B.

BIN,DELB — If MESHB = 0, then the mesh of prefixed B-values begins at BIN, i.e., X(1) = BIN. Furthermore, the step size for this mesh is DELB. Of course, BIN must not be so large that the point on the field line with B-value equal to BIN is below ALTO, nor so small that BIN + 99* DELB is less than the equatorial value. Care must be taken to ensure that these latter restrictions are true for all longitudes.

INT1,INT2 ALT1,ALT2 — If MESHB = 1, then the altitudes of the points whose field intensities are X(INT1) and X(INT2) on the lower end of the field line are ALT1 and ALT2 kilometers, respectively. INT1 must not be greater than

100, which is the dimension of X, H, etc. INT2 should not be less than 12 for X(j), j = 3,4,...,12 is prefixed in B. ALT1 is the altitude of the lowest point on the field line. It can be equal to, but never less than, ALTO. ALT2 should be chosen so that the variations of the X-mesh size should be as small as possible (see figure 7).

11. DIAGNOSTICS

Below is a list of the possible diagnostics followed by an explanation of why they were printed:

SORRY, BUT I CANNOT FIND THAT POINT IN ICHECK

If this diagnostic is given, then subroutine SEARCH has completed 15 iterations trying to find the point at longitude FLONG with L value SL and B value SB. In the printout, the diagnostic is followed by the number of iterations performed to find the correct value of L, the number of iterations performed to find the correct value of B, the altitude, latitude, and longitude of the point at which the iteration process began, and the desired B,L value. The program chooses a new value for B; namely, SB becomes SB + .005, and the search process begins again.

There are three primary reasons why this diagnostic can be given. The first is that an error was made by the user: the initial value for the altitude, latitude, and longitude (ALT, FLAT, FLONG) of the desired B,L point (SB,SL) was too far from the correct value. The usual B,L maps can be consulted to prevent this problem.

The second reason that this diagnostic can occur appears to be caused by an inherent error in INVAR. Take a point P whose conjugate point P' lies below the surface of the earth. To calculate the value of L at P, INVAR uses subroutine LINES to trace the field line from P to the point P'. As has been described in section IV, b, LINES begins at P and determines a sequence of points along the field line through P until some point of the sequence has a B-value at least as large as that of P. However, the geometric field expansion used in LINES is strictly valid only for points above the surface of the earth, and a peculiar behavior must be expected in a forced tracing well below sea level. A region for which such peculiar behavior was found to affect the calculated value of L is shown in figure 8. When field lines are traced from this European region, one finds a B-dependence along the field line as sketched in figure 9. It is quickly realized, then, that whenever I (or L) is computed in INVAR for this field line,

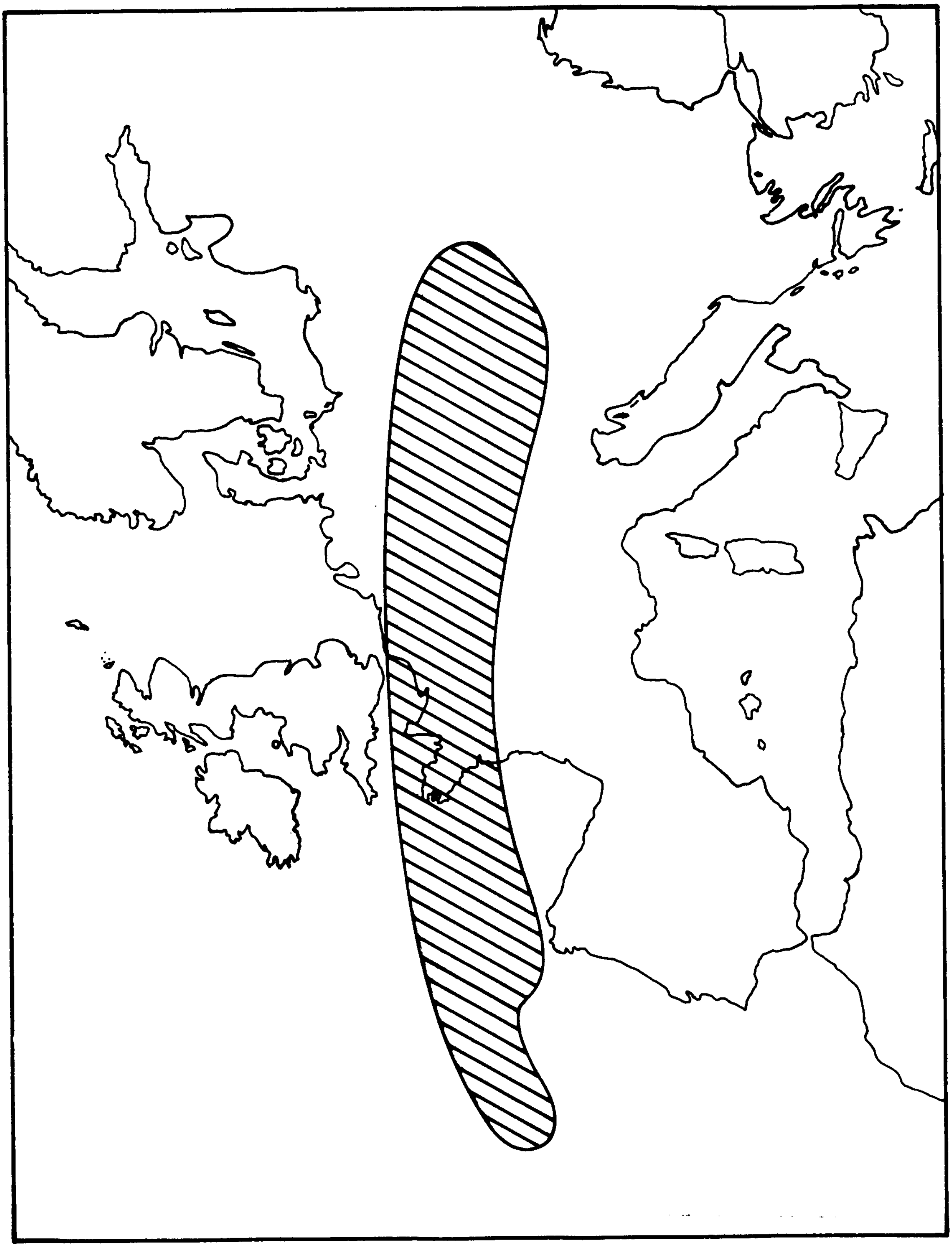

Figure 8. Region Over Europe for Which L Values Calculated in INVAR are not Accurate.

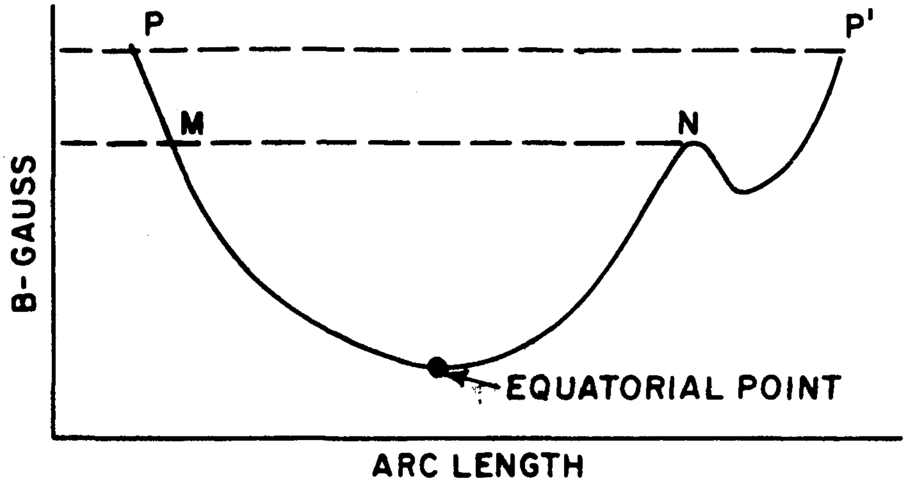

Figure 9. Sketch of B-dependence Along Field Lines Traced from the Area Shown in Figure 8.

only the section between M and N yields correct values of L. The value of L at M may differ by as much as 3.5 percent from that of a point on the same field line, but at a slightly higher B value. It is easily seen that the points iterated by the technique used in SEARCH may fail to converge to a point with the correct value of L because of the discontinuity. There, an inherently incorrect value of L may be computed.

The third reason why it is possible for this diagnostic to occur is that the tolerance ERRL may be changed to a value too small with respect to ERR (or ERR may be increased to a value too big). In this case, the iterated points may never fall in the interval SL±ERRL.

SORRY, BUT I CANNOT FIND THAT POINT IN MCHECK

If this diagnostic is given in the output, then subroutine SEARCH has completed 15 iterations trying to find the point at longitude FLONG with B-value SB. It is not expected that this diagnostic will ever occur for the geomagnetic field within the magnetosphere; it is programmed into the subroutine to prevent any possibility of an infinite iteration. The printout which follows the diagnostic, as well as the action taken by the program after printing the diagnostic, is exactly the same as that described in the discussion above.

INCREASE OF ERR IN EQUAT, or
INCREASE OF ERR IN BESECT, or
INCREASE OF ERR IN INSECT, or
INCREASE OF ERR IN INVAR

This diagnostic is given if, after calling LINES for the subroutine indicated in the diagnostic, more than 200 points are found in the tracing of the field line. The diagnostic is followed by the increased value of ERR which will be used in the recall of LINES. If, after increasing ERR and recalling LINES, it is found that less than 200 points were required to complete the section of the field line trace in question, the program will reset ERR to its original small value and proceed. This check on the number of points permitted in the trace is necessary to prevent an overflow of the dimension of B, VN1, VN2, VN3, BX, etc. Since ERR controls the maximum step size of the arc length along the field line, then an increase of ERR will allow a larger step size and, consequently, a smaller number of points in the determination of the field line.

X SHOULD BE INCREASING IN SPLINE

If this diagnostic is given, then the independent variable sequence with which the interpolation subroutine SPLINE is called is not an increasing sequence. Following the diagnostic is a list of the pair of sequences X and Y--the independent variable and the dependent variable. The subroutine continues its calculation but the results in the neighborhood of the section where the values of X are not increasing will likely not be correct. It is very unlikely that this diagnostic will ever be given when SPLINE is used as a subroutine to SHELL. The check is included mostly for users of SPLINE in other interpolation problems.

INCREASE THE DIMENSION OF ALL DIMENSIONED VARIABLES IN SPLINE AND YSPLN

This diagnostic occurs if the number of points in the sequence X is greater than the maximum dimension of the dimensioned variables in these two subroutines. The program will stop and the user must increase the dimension of X, Y, D, P, E, A, B, and Y2DOT in subroutines SPLINE and YSPLN. They are presented in this document as dimensioned 400 places merely to save on storage requirements.

AT LEAST FOUR POINTS ARE NEEDED IN THIS ROUTINE

The technique used in SPLINE requires at least four points. If there are not four values of the independent variable, the program stops. Should this diagnostic occur, the user can either decrease the value of ALTO or decrease the value of ERR. In either case, the number of points along the field line will increase.

PROGRAM HALT IN SUBROUTINE YSPLN----SEARCH LIST CONTAINS LESS THAN TWO POINTS

The technique used in YSPLN requires at least two points. Since a call of this subroutine logically follows a call of SPLINE, the diagnostic should never occur and is included only for logical completeness.

SECTION V

RESULTS

Tables I and II show typical printouts for the L = 1.25 shell. In the first table, the X-mesh was one of prefixed B-value; in the second table, the X-mesh was defined according to prefixed altitudes (see section IV,B).

Computer time for a complete run around the world, defining 72 field lines on the shell, was 2.56 minutes on a Control Data Corporation 6600 computer, and roughly four times more on an IBM 7094, for L = 1.25. The same run, without the calculation of drift velocity and bounce period takes 0.73 minute on the CDC 6600. Tracing of a given B-L ring, without any additional information other than giving the coordinates of the B-L point on 72 different longitudes around the world, requires only 20 seconds on the 6600, for L = 1.25. All these computer times increase with increasing L.

In figures 10 and 11 the longitude dependence of angular drift velocities and bounce paths, respectively, is shown, for L = 1.25 and B = 0.18 and 0.22. Notice that the bounce path (and, therefore, the bounce period) varies considerably less with longitude than does the drift velocity. This is to be expected according to expression (4), in which I and B_m are invariant, and should vary only a little with longitude.

When the curves in figures 10 and 11 are to be compared with those previously calculated, one has to bear in mind that longitude dependence of drift velocities and bounce paths depends very critically on how the shell was originally defined. The correct calculation requires picking the L-shell at precisely the B_m points for which these quantities are to be determined.

Table I

AT LONG= -35.00 THE COORDINATES OF THE POINT AT L= 1.25 AND B = .2200 ARE ALT = 787.46 AND LAT = .37
THE EQUATORIAL POINT OF THE FIELD LINE IS AT EALT = 1118.84 ELAT =-14.94 ELONG = -52.30 AND HAS B VALUE .1616
THE DRIFT PARAMETERS ARE DRIFT = 1.149 SLOPE = -.612 PATH = 1.915 PSLOPE = 1.628 DPHIDX = 8.693E-01

	MESH POINTS (GAUSS)	ALTITUDES(KM) NORTH	ALTITUDES(KM) SOUTH	ARC LENGTH FROM EQUATOR(KM) NORTH	ARC LENGTH FROM EQUATOR(KM) SOUTH	LATITUDES NORTH	LATITUDES SOUTH	LONGITUDES NORTH	LONGITUDES SOUTH
17	.17000	1114.5	935.4	865.7	892.5	-8.48	-21.64	-53.39	-51.19
18	.18000	1052.6	784.8	1256.9	1328.8	-5.56	-24.84	-53.87	-50.65
19	.19000	985.5	649.5	1539.4	1655.6	-3.47	-27.21	-54.23	-50.25
20	.20000	918.2	523.7	1767.3	1928.1	-1.79	-29.18	-54.53	-49.92
21	.21000	851.9	405.6	1961.7	2165.5	-.37	-30.88	-54.73	-49.63
22	.22000	787.5	293.3	2132.1	2379.0	.87	-32.41	-55.00	-49.38
23	.23000	724.9	186.9	2284.4	2573.0	1.96	-33.80	-55.20	-49.14

Table II

AT LONG= -55.00 THE COORDINATES OF THE POINT AT L= 1.25 AND B = .2200 ARE ALT = 787.46 AND LAT = .87
THE EQUATORIAL POINT OF THE FIELD LINE IS AT EALT = 1118.84 ELAT =-14.94 ELONG = -52.30 AND HAS B VALUE .1616
THE DRIFT PARAMETERS ARE DRIFT = 1.149 SLOPE = -.612 PATH = 1.915 PSLOPE = 1.628 DPHIDX = 8.683E-01

	MESH POINTS	ALTITUDES(KM)		ARC LENGTH FROM EQUATOR(KM)		LATITUDES		LONGITUDES	
	(GAUSS)	NORTH	SOUTH	NORTH	SOUTH	NORTH	SOUTH	NORTH	SOUTH
1	.16156	1126.4	1108.4	90.0	87.1	-14.31	-15.64	-52.40	-52.18
2	.16356	1139.4	1052.2	434.0	431.7	-11.73	-18.22	-52.84	-51.76
3	.16557	1135.1	1012.6	603.1	609.4	-10.45	-19.54	-53.05	-51.54
4	.16757	1126.8	976.0	735.6	751.2	-9.46	-20.59	-53.21	-51.36
5	.16957	1116.8	942.4	844.1	869.0	-8.64	-21.47	-53.35	-51.22
6	.17157	1105.6	910.3	939.7	973.3	-7.93	-22.23	-53.47	-51.09
7	.17357	1093.7	879.2	1025.3	1068.1	-7.29	-22.93	-53.58	-50.97
8	.17557	1081.2	849.0	1103.3	1155.5	-6.70	-23.57	-53.68	-50.86
9	.17757	1068.5	819.6	1175.6	1236.7	-6.17	-24.17	-53.77	-50.76
10	.17957	1055.4	790.9	1242.9	1312.9	-5.66	-24.72	-53.86	-50.67
11	.18157	1042.2	762.8	1306.0	1385.0	-5.19	-25.25	-53.94	-50.58
12	.18357	1028.9	735.2	1365.8	1453.7	-4.75	-25.75	-54.01	-50.50
13	.18497	1019.5	716.2	1405.9	1500.0	-4.45	-26.08	-54.06	-50.44
14	.18636	1010.0	697.4	1444.6	1544.8	-4.17	-26.41	-54.11	-50.39
15	.18776	1000.6	678.8	1482.0	1588.4	-3.89	-26.73	-54.16	-50.33
16	.18916	991.1	660.4	1518.2	1630.8	-3.62	-27.03	-54.21	-50.28
17	.19056	981.7	642.2	1553.3	1672.1	-3.36	-27.33	-54.25	-50.23
18	.19196	972.2	624.1	1587.4	1712.5	-3.11	-27.62	-54.30	-50.18
19	.19336	962.8	606.3	1620.6	1752.0	-2.87	-27.91	-54.34	-50.13
20	.19476	953.4	588.6	1652.9	1790.6	-2.63	-28.19	-54.38	-50.09
21	.19616	944.0	571.0	1684.4	1828.4	-2.40	-28.46	-54.42	-50.04
22	.19756	934.6	553.7	1715.2	1865.4	-2.17	-28.73	-54.46	-49.99
23	.19896	925.2	536.4	1745.3	1901.6	-1.95	-28.99	-54.50	-49.95
24	.20036	915.8	519.4	1774.7	1937.1	-1.73	-29.24	-54.54	-49.91
25	.20175	906.5	502.5	1803.5	1971.9	-1.52	-29.49	-54.57	-49.87
26	.20315	897.1	485.7	1831.7	2006.1	-1.32	-29.74	-54.61	-49.82
27	.20455	887.8	469.1	1859.3	2039.7	-1.12	-29.98	-54.65	-49.78
28	.20595	878.6	452.6	1886.4	2072.8	-.92	-30.22	-54.68	-49.74
29	.20735	869.3	436.2	1912.9	2105.3	-.72	-30.45	-54.71	-49.70
30	.20875	860.1	420.0	1938.9	2137.3	-.54	-30.68	-54.75	-49.67
31	.21015	851.0	403.9	1964.4	2168.8	-.35	-30.91	-54.78	-49.63
32	.21146	842.4	388.9	1987.9	2197.9	-.18	-31.11	-54.81	-49.59
33	.21269	834.4	374.9	2009.6	2224.9	-.02	-31.31	-54.84	-49.56
34	.21385	826.9	361.8	2029.7	2250.0	.12	-31.49	-54.87	-49.53
35	.21494	819.9	349.5	2048.4	2273.4	.26	-31.66	-54.89	-49.50
36	.21597	813.2	338.0	2065.9	2295.3	.39	-31.81	-54.91	-49.48

Table II (cont'd)

37	.21694	807.0	327.1	2082.1	2315.8	.50	-31.96	-54.93	-49.45
38	.21786	801.1	316.9	2097.4	2335.0	.62	-32.10	-54.95	-49.43
39	.21874	795.5	307.2	2111.7	2353.1	.72	-32.23	-54.97	-49.41
40	.21957	790.2	298.1	2125.2	2370.1	.82	-32.35	-54.99	-49.39
41	.22036	785.2	289.4	2137.8	2386.2	.91	-32.46	-55.01	-49.37
42	.22111	780.4	281.2	2149.8	2401.4	.99	-32.57	-55.02	-49.35
43	.22183	775.9	273.5	2161.2	2415.8	1.08	-32.67	-55.04	-49.33
44	.22251	771.6	266.1	2171.9	2429.4	1.15	-32.77	-55.05	-49.31
45	.22317	767.4	259.0	2182.1	2442.3	1.23	-32.86	-55.07	-49.30
46	.22379	763.5	252.3	2191.8	2454.6	1.30	-32.95	-55.08	-49.28
47	.22439	759.7	245.9	2201.0	2466.2	1.36	-33.04	-55.09	-49.27
48	.22497	756.1	239.9	2209.8	2477.4	1.43	-33.12	-55.10	-49.26
49	.22552	752.7	234.0	2218.2	2488.0	1.49	-33.19	-55.11	-49.25
50	.22604	749.4	228.5	2226.1	2498.1	1.54	-33.26	-55.12	-49.23
51	.22655	746.3	223.1	2233.6	2507.8	1.60	-33.33	-55.13	-49.22
52	.22703	743.3	218.0	2240.9	2517.0	1.65	-33.40	-55.14	-49.21
53	.22750	740.4	213.1	2247.8	2525.8	1.70	-33.46	-55.15	-49.20
54	.22794	737.6	208.4	2254.4	2534.3	1.75	-33.52	-55.16	-49.19
55	.22837	735.0	203.9	2260.7	2542.4	1.79	-33.58	-55.17	-49.18
56	.22879	732.4	199.6	2266.8	2550.2	1.84	-33.64	-55.18	-49.17
57	.22918	730.0	195.4	2272.6	2557.7	1.88	-33.69	-55.18	-49.16
58	.22957	727.6	191.4	2278.2	2564.9	1.92	-33.74	-55.19	-49.15
59	.22994	725.3	187.6	2283.6	2571.8	1.96	-33.79	-55.20	-49.15
60	.23030	723.1	183.8	2288.7	2578.5	1.99	-33.84	-55.21	-49.14
61	.23064	721.0	180.3	2293.7	2584.9	2.03	-33.89	-55.21	-49.13
62	.23098	718.9	176.8	2298.5	2591.1	2.06	-33.93	-55.22	-49.12
63	.23130	717.0	173.5	2303.1	2597.1	2.10	-33.97	-55.22	-49.12
64	.23161	715.0	170.2	2307.6	2602.8	2.13	-34.01	-55.23	-49.11
65	.23192	713.2	167.1	2311.9	2608.4	2.16	-34.05	-55.24	-49.10
66	.23221	711.4	164.1	2316.0	2613.7	2.19	-34.09	-55.24	-49.10
67	.23249	709.7	161.2	2320.1	2618.9	2.22	-34.13	-55.25	-49.09
68	.23277	708.0	158.4	2323.9	2623.9	2.25	-34.17	-55.25	-49.08
69	.23303	706.4	155.6	2327.7	2628.8	2.27	-34.20	-55.26	-49.08
70	.23329	704.8	153.0	2331.3	2633.4	2.30	-34.23	-55.26	-49.07
71	.23354	703.3	150.4	2334.8	2638.0	2.32	-34.27	-55.27	-49.07
72	.23379	701.8	147.9	2338.2	2642.4	2.35	-34.30	-55.27	-49.06
73	.23402	700.4	145.5	2341.5	2646.7	2.37	-34.33	-55.28	-49.06
74	.23425	699.0	143.2	2344.7	2650.8	2.40	-34.36	-55.28	-49.05
75	.23447	697.6	140.9	2347.8	2654.8	2.42	-34.39	-55.28	-49.05
76	.23469	696.3	138.7	2350.8	2658.7	2.44	-34.42	-55.29	-49.04
77	.23490	695.0	136.6	2353.7	2662.5	2.46	-34.44	-55.29	-49.04
78	.23511	693.8	134.5	2356.5	2666.2	2.48	-34.47	-55.30	-49.03
79	.23531	692.6	132.4	2359.2	2669.7	2.50	-34.49	-55.30	-49.03
80	.23550	691.4	130.5	2361.9	2673.2	2.52	-34.52	-55.30	-49.03

Table II (cont'd)

81	.23569	690.3	128.6	2364.5	2676.6	2.54	-34.54	-55.31	-49.02
82	.23588	689.2	126.7	2367.0	2679.9	2.56	-34.57	-55.31	-49.02
83	.23606	688.1	124.9	2369.5	2683.1	2.57	-34.59	-55.31	-49.02
84	.23623	687.0	123.1	2371.9	2686.2	2.59	-34.61	-55.32	-49.01
85	.23640	686.0	121.4	2374.2	2689.2	2.61	-34.63	-55.32	-49.01
86	.23657	685.0	119.7	2376.4	2692.2	2.62	-34.66	-55.32	-49.00
87	.23673	684.0	118.1	2378.6	2695.0	2.64	-34.68	-55.32	-49.00
88	.23689	683.1	116.5	2380.8	2697.8	2.65	-34.70	-55.33	-49.00
89	.23704	682.1	114.9	2382.9	2700.6	2.67	-34.72	-55.33	-48.99
90	.23720	681.2	113.4	2384.9	2703.2	2.68	-34.73	-55.33	-48.99
91	.23734	680.4	111.9	2386.9	2705.8	2.70	-34.75	-55.34	-48.99
92	.23749	679.5	110.4	2388.8	2708.4	2.71	-34.77	-55.34	-48.99
93	.23763	678.6	109.0	2390.7	2710.9	2.72	-34.79	-55.34	-48.98
94	.23777	677.8	107.7	2392.5	2713.3	2.74	-34.81	-55.34	-48.98
95	.23790	677.0	106.3	2394.3	2715.6	2.75	-34.82	-55.35	-48.98
96	.23803	676.2	105.0	2396.1	2717.9	2.76	-34.84	-55.35	-48.97
97	.23816	675.5	103.7	2397.8	2720.2	2.78	-34.86	-55.35	-48.97
98	.23829	674.7	102.4	2399.5	2722.4	2.79	-34.87	-55.35	-48.97
99	.23841	674.0	101.2	2401.1	2724.5	2.80	-34.89	-55.35	-48.97
100	.23853	673.3	100.0	2402.7	2726.6	2.81	-34.90	-55.36	-48.96

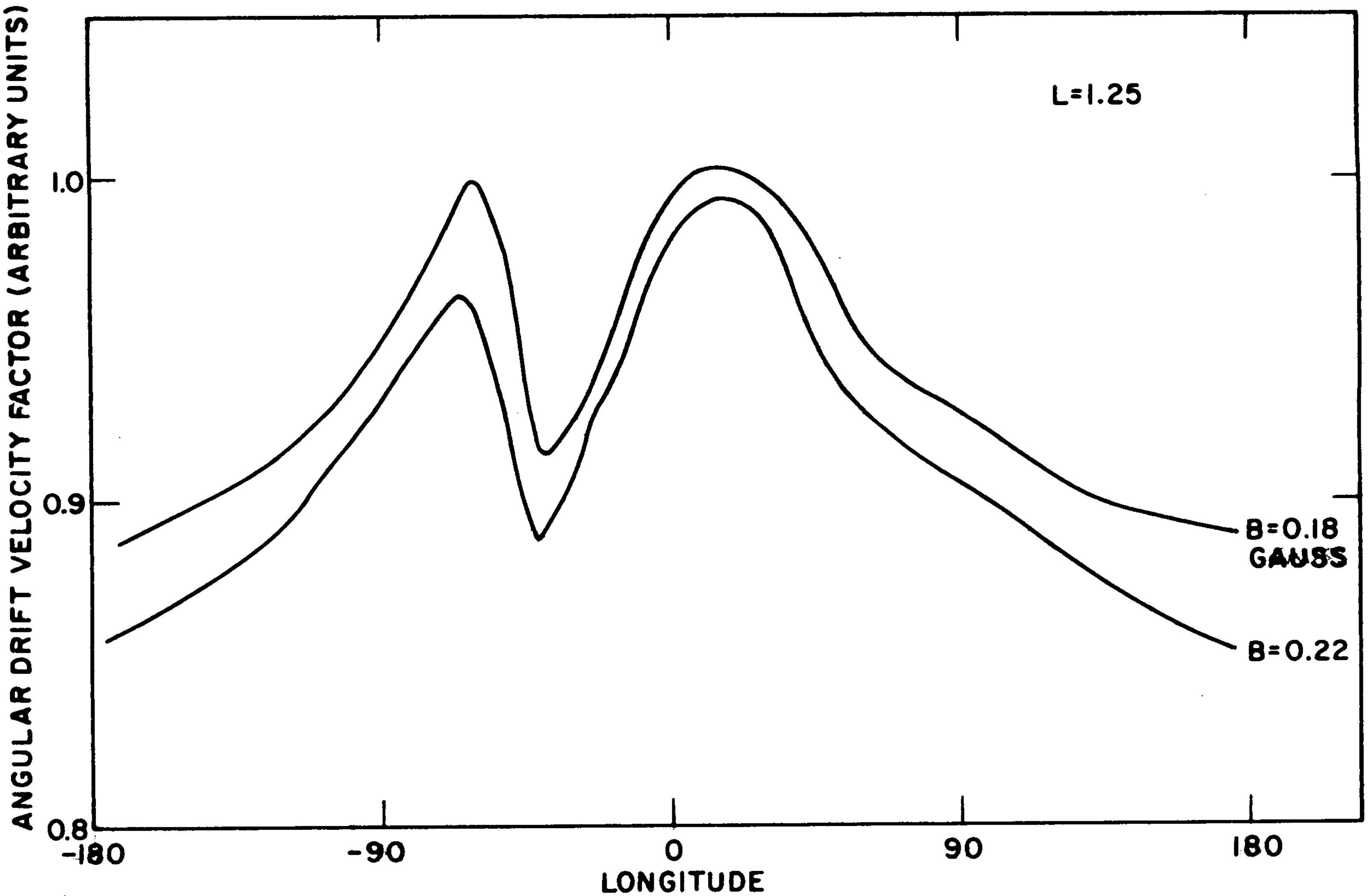

Figure 10. Longitude Dependence of Angular Drift Factor for B = .18 and B = .22 at L = 1.25.

In order to obtain the angular drift velocity in degrees/sec, multiply by $L * \gamma * \beta^2 * 2.31759 \times 10^{-2} *$ (mass in electron masses) where γ = total energy and $\beta = v/c$.

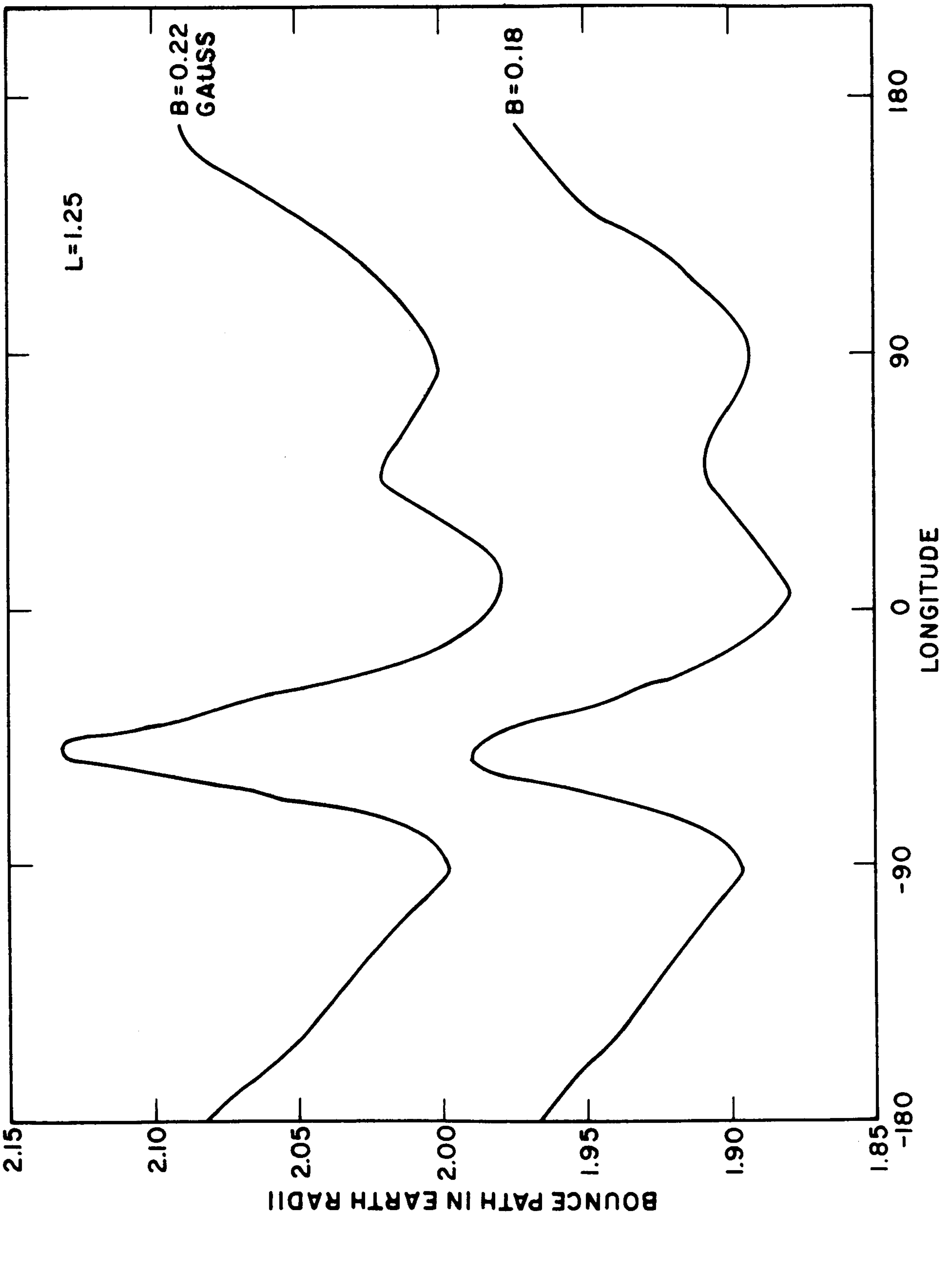

Figure 11. Bounce Path in Earth Radii for B = .22 and B = .18 at L = 1.25

APPENDIX I

PROGRAM DICTIONARY

In this appendix, some of the most important variables are defined for the principal subroutines.

Variable	DICTIONARY FOR SHELL	Units
ABX	arc length to the preceding point in the mesh along the field line.	earth radii
ALT	initial altitude value for the iteration process used by subroutine SEARCH. After TRACE is called, ALT is the altitude of the B-L point through which the field line has been traced and will be used as the initial value for the iteration at the next field line.	km
ALTO	minimum altitude level to which the field line will be traced.	km
ALT1	altitude of the lowest point in the prefixed altitude section of the field line. This is used only in case MESHB = 1.	km
ALT2	altitude of the highest point in the prefixed altitude section of the field line. This is used only in case MESHB = 1.	km
BIN	B-value of the first point of the sequence of prefixed B's. This is used only in case MESHB = 0.	gauss
BX	B-value of the sequence of points along the field line.	gauss
CLONG	longitude of the point on the field line at ALTO km.	degrees
DARC	arc length between successive points along the field line. DARC(j,1) is computed in BLRING, DARC(j,2) in INSECT, and DARC(j,3) in BESECT.	earth radii
DB	B-value of points along the field line. DB(j,1), DB(j,2), and DB(j,3) are calculated in BLRING, INSECT, and BESECT, respectively.	gauss

DELB	increment in B-value for the sequence of prefixed B's. This is used only in case MESHB = 0.	gauss
DEQ	distance between the equatorial points of successive field lines.	earth radii
DH	altitude of points along the field line. The calculation is done in the same order as that of DARC and DB.	km
DLAT	latitude of points along the field line. The calculation is done in the same order as that of DARC and DB.	degrees
DLONG	longitude of points along the field line. The calculation is done in the same order as that of DARC and DB.	degrees
DLONG1	increment in longitude in the region containing the anomaly for the SB-SL point through which the field line is to be traced.	degrees
DLONG2	same as DLONG1 except the region is outside the anomaly, but still not far from it.	degrees
DLONG3	same as DLONG1 except the region is away from the anomaly.	degrees
DPHIDX	change of equatorial longitude per unit equatorial distance.	radian/ earth radii
DRIFT	see dictionary for BLRING.	
EALT	altitude of the equatorial point.	km
ELAT	latitude of the equatorial point.	degrees
ELONG	longitude of the equatorial point.	degrees
ENC	increment for the X mesh for the small region of prefixed B (approximately X(3) through X(12)). This is used only in case MESHB = 1.	gauss
ERR	see dictionary for SEARCH.	
ERRB	see dictionary for SEARCH.	
ERRL	see dictionary for SEARCH.	
EVN	geocentric coordinates of the equatorial point.	
FLAT	initial value of latitude for the iteration used by SEARCH. See definition of ALT.	degrees

FLAX	latitude of the sequence of points along the field line.	degrees
FLONG	longitude of the B-L point through which the field is to be traced.	degrees
FLOX'	longitude of the sequence of points along the field line.	degrees
H	prefixed altitude sequence.	km
HXN	altitude of points in the final X-mesh for which the arc length to the equator has been defined to be negative.	km
HXP	altitude of points in the final X-mesh for which the arc length to the equator has been defined to be positive.	km
INT1	number of points in the X-mesh if MESHB = 1.	
INT2	subscript for the highest point in the section of prefixed altitude section of the X-mesh.	
JLAST	number of points in the sequences BX, HBX, FLAT, FLOX, ABX, and S.	
JUP	(JUP(1) - 1) is the number of points calculated by BLRING; (JUP(2) - 1) is the number of points calculated by INSECT; and (JUP(3) - 1) is the number of points calculated by BESECT.	
KASE	see dictionary for TRACE.	
KEQ	subscript for the equatorial point.	
LATN	latitude of points in the final X-mesh for which the the arc length to the equator has been defined to be positive.	degrees
LINE	number of field line calculations completed.	
LONGN	longitude of the points indicated in the definition of LATP.	degrees
MESHB	control for the type of X-mesh to be used (see section IV,8).	
NPR	edit control. Detailed printing occurs every NPR cycles.	
S	arc length from the equatorial point to a point in the sequence along the field line.	earth radii
SB	B-value of the point through which the field line will be traced.	gauss

SFLONG	longitude of the point through which the initial field line was traced.	degrees
SL	L value of the shell to be determined.	earth radii
SLONG1, SLONG2, etc.	longitude at which the longitude step size changes.	degrees
SLOPE	see dictionary for BLRING.	
SXN	arc length to the equator of points in the X-mesh as indicated in the definition of LATN.	earth radii
SXP	arc length to the equator of points in the X-mesh as indicated in the definition of LATP.	earth radii
X	B-value of points in the final mesh.	gauss

DICTIONARY FOR TRACE

ALT	see dictionary for SHELL.	
BALT	altitude of the last point calculated in BESECT. The B-value on the field line at this altitude will equal or be slightly higher than BO.	km
BO	B-value on the field line at the ALTO km level.	gauss
BCONJ	B-value to which BESECT must trace.	gauss
CLONG	longitude of the lowest point in the final X-mesh.	degrees
DARC	see dictionary for SHELL.	
DB	see dictionary for SHELL.	
DH	see dictionary for SHELL.	
DLONG	see dictionary for SHELL.	
DLAT	see dictionary for SHELL.	
DRIFT	see dictionary for BLRING.	
EALT	altitude of the equatorial point.	km
EB	B-value of the equatorial point.	gauss
ERR	see dictionary for SEARCH.	
EVN	geodetic coordinates of the equatorial point.	
FLAT	see dictionary for SEARCH.	degrees
FLONG	see dictionary for SEARCH.	
JNEAR	see dictionary for BLRING.	
JUP	see dictionary for SHELL.	
KASE	this has value 1 or 2 depending on which end of the field line is lower in the tracing of BLRING.	
KEQ	see dictionary for SHELL.	
PATH	see dictionary for BLRING.	

RUM	geodetic coordinates of the point at which BESECT begins to trace the field line.	
SB	see dictionary for SEARCH.	
SL	see dictionary for SEARCH.	
SLOPE	see dictionary for BLRING.	
VCONJ	after the call of INSECT, this is the geodetic coordinates of the intersection of the field line with ALTO.	
VLAST	geodetic coordinates of the last point in the section of the field line traced in SEARCH.	
VNEAR	see dictionary for BLRING.	
VN1	geodetic altitude of points in the section of field line traced in the last call of LINES.	earth radii
VN2	geodetic latitude of points in the section of the field line traced in the last call of LINES.	co-latitude
VN3	geodetic longitude of points in the section of the field line traced in the last call of LINES.	radians
VSAVE	geodetic coordinates of the first point in the section of the field line traced by SEARCH.	

DICTIONARY FOR BLRING

ALT	see dictionary for SEARCH.	
ARC	arc length between consecutive points of the field line.	earth radii
BB	see dictionary for SEARCH.	
BZERO	equatorial field intensity for a pure dipole at the given L-shell.	gauss
CORR	longitude correction for a parallel displacement of field lines.	radians
DRIFT	drift factor--see SLOPE.	
EB	equatorial B-value.	gauss
ERR	see dictionary for SEARCH.	
EALT	altitude of the equatorial point.	km
FLAT	see dictionary for SEARCH.	
FLONG	see dictionary for SEARCH.	
JU	(JU - 1) is the total number of points in the field as traced by BLRING.	
JNEAR	see VNEAR.	

PATH	rectified trajectory of a particle from one mirror point to its conjugate.	earth radii
PSLOPE	factor which gives the B-dependence of PATH in the expression (5).	
SB	see dictionary for SEARCH.	
SL	see dictionary for SEARCH.	
SLOPE	factor which gives the B-dependence of the drift velocity in the expression (5). To obtain the drift velocity in cm/sec, see comment cards.	
VINEAR	when determined in SEARCH, VNEAR is the geodetic coordinates of the first point in the field line tracing for which B increases again, i.e., VNEAR is the geodetic coordinates of the first point after the equatorial point. JNEAR - 1 is the number of points between the origin of the tracing and VNEAR. When determined in EQUAT, VNEAR is the geocentric coordinates of the equatorial point.	
VN1	see dictionary for SEARCH.	
VSAVE	see dictionary for SEARCH.	

DICTIONARY FOR SEARCH

ALT	geodetic altitude of the initial point in the iteration process--becomes the geodetic altitude of the SB-SL point.	km
BB	magnitude of the geomagnetic field intensity.	gauss
BP	east component of the geomagnetic field.	
BR	radial component of the geomagnetic field.	
BT	south component of the geomagnetic field.	
B1	field intensity at point T (see figure 3).	gauss
B2	field intensity at point S (see figure 3).	gauss
DV	initial increment in latitude.	radians
ERR	control parameter for the step size in line tracing.	
ERRB	relative gross tolerance in B; final tolerance is 1/20 of the initial value.	
ERRL	relative gross tolerance in L; final tolerance is 1/100 of the initial value.	

FLAT	geodetic latitude of the initial starting point in the iteration process--becomes the geodetic latitude of the SB-SL point.	
FLONG	geodetic longitude of the initial point in the iteration process.	
JUP	(JUP - 1) is the total number of points in the field line tracing.	
SB	prefixed B-Value of the point sought in SEARCH.	gauss
SL	prefixed L-value of the point sought in SEARCH.	earth radii

APPENDIX II

PROGRAM LISTING

```
      PROGRAM SHELL
      COMMON B(200),VN1(200),VN2(200),ARC(200),VNEAR(3),VCONJ(3),VLAST(3
     1),VPLAST(3),VBO(3),VSAVE(3),BCONJ,VO,BO,ALTO,MESHB,JNEAR,BNEAR,BSAVE
     2,BRSAVE,BPSAVE,BTSAVE,VN3(200),JSTOP
      DIMENSION ABX(600), BX(600), HBX(600), H(100), S(600), SKN(300)
      DIMENSION X(100), SXN(100), SXP(100), HXN(100), HXP(100), BKN(300)
      DIMENSION DARC(200,3), DB(200,3), DH(200,3), JUP(3), Y2DOT(600)
      DIMENSION DLONG(200,3), DLAT(200,3), FLAX(600), FLOX(600)
      DIMENSION LATN(100), LONGN(100), LATP(100), LONGP(100)
      DIMENSION EVN(3), EVN1(3)
      REAL LATN,LONGN,LATP,LONGP
      ERR=.02
      ERRB=.01
      ERRL=.01
      LINE=0
      DEQ=0.
C
C     ALTO IS THE MINIMUM ALTITUDE IN KM TO WHICH FIELD LINES SHOULD
C     BE TRACED
C
      ALTO=100.
C
C     FLONG IS THE PREFIXED LONGITUDE OF THE B-L POINT WHICH DETERMINES
C     THE FIELD LINE
C     ALT AND FLAT ARE APPROXIMATE COORDINATES OF THE B-L POINT
C     SL IS THE L-VALUE OF THE SHELL
C     SB (GAUSS) IS THE B-VALUE OF THE POINT WHICH DETERMINES EACH
C     FIELD LINE OF THAT SHELL
C     ALTITUDE IN KM ABOVE SEA LEVEL
C     EAST LONGITUDE IS POSITIVE
C
      READ (5,52) ALT,FLAT,FLONG
      SFLONG=FLONG
      READ (5,52) SB,SL
      WRITE (6,56) ALT,FLAT,FLONG,ERR,ERRB,ERRL,SB,SL
C
C     DLONG1 - - LONGITUDE STEP BETWEEN SLONG2 AND SLONG3
C     DLONG2 - - LONGITUDE STEP BETWEEN SLONG1 AND SLONG2 AND
C                BETWEEN SLONG3 AND SLONG4
C     DLONG3 - - LONGITUDE STEP ELSEWHERE
C     (NPR-1) - - NUMBER OF LINES SKIPPED BETWEEN DETAILED PRINTING
C
      READ (5,53) SLONG1,SLONG2,SLONG3,SLONG4
      WRITE (6,53) SLONG1,SLONG2,SLONG3,SLONG4
      READ (5,52) DLONG1,DLONG2,DLONG3,NPR
      WRITE (6,52) DLONG1,DLONG2,DLONG3,NPR
C
C     MESHB CONTROLS THE MESH FOR THE MAGNETIC FIELD ALONG THE LINE
C     SET MESHB EQUAL TO ONE IF THE MESH SHOULD BE DEFINED AT
C     FIXED ALTITUDES. IN THAT CASE, ALT1 AND ALT2 ARE END POINTS
C     OF THE PREFIXED ALTITUDES. SET MESHB EQUAL TO ZERO IF THE
C     MESH SHOULD BE DEFINED AT PREFIXED B-VALUES. IN THAT CASE, BIN
C     IS THE MINIMUM B-VALUE OF THE MESH, DELB IS THE INCREMENT. FIELD
C     MESH POINTS ARE CALLED X(J).
C     INT2 SHOULD BE LESS THAN 12.
C     INT1  MUST ALWAYS BE NOT GREATER THAN 100.
C
```

```
C     DEFINE PREFIXED X-MESH
C
      READ (5,54) MESHB,BIN,DELB
      IF (MESHB.EQ.1) GO TO 2
      X(1)=BIN
      DO 1 J=2,100
1     X(J)=X(J-1)+DELB
      GO TO 4
C
C     DEFINE  PREFIXED ALTITUDE MESH
C
2     READ (5,55) INT1,INT2
      READ (5,52) ALT1,ALT2
      ENT1=INT1
      ENT2=INT2
      WRITE (6,52) ENT1,ENT2
      WRITE (6,52) ALT1,ALT2
      JMIN=1
      JMAX=INT1
      E=ALOG(ALT1/ALT2)/ALOG(ENT1/ENT2)
      EN=-E
      A=ALT2*ENT2**EN
      DO 3 I=INT2,INT1
      FI=I
3     H(I)=A*FI**E
C
C     DEFINE FIELD LINE GEOMETRY
C
4     CALL TRACE (DB,DARC,DH,KEQ,ALT,FLAT,FLONG,ERR,ERRB,ERRL,SB,SL,KASE
     1,JUP,DRIFT,CLONG,SLOPE,EVN,DLONG,DLAT,PATH,PSLOPE)
      IF (LINE.EQ.0) GO TO 5
      DR=ABS(EVN1(1)-EVN(1))
      DTHE=ABS(EVN1(2)-EVN(2))
      DPHI=ABS(EVN1(3)-EVN(3))
      RO=(EVN1(1)+EVN(1))*0.5
      SIT=ABS(SIN((EVN1(2)+EVN(2))*0.5))
      SSQ=SIT*SIT
      DEQ=SQRT(DR*DR+RO*RO*(DTHE*DTHE+DPHI*DPHI*SSQ))
      DPHIDX=DPHI/DEQ
5     CONTINUE
      SIT=ABS(SIN(EVN(2)))
      SSQ=SIT*SIT
      OER=(6356.912+SSQ*(21.3677+.108*SSQ))/6371.2
      AER=EVN(1)-OER
      EALT=AER*6371.2
      ELAT=90.-EVN(2)*57.2957795
      ELONG=EVN(3)*57.2957795
      IF (ELONG.GT.180.) ELONG=ELONG-360.
      IF (ELONG.LT.-180.) ELONG=ELONG+360.
C
C     EQUATORIAL DRIFT VELOCITY IN DEGREES PER SECOND IS GIVEN BY
C     DRIFT VELOCITY (IN CM/SEC) * DPHIDX
C
      WRITE (6,57) FLONG,SL,SB,ALT,FLAT,EALT,ELAT,ELONG,DB(KEQ,1),DRIFT,
     1SLOPE,PATH,PSLOPE,DPHIDX
C
C     ARRANGE IN ORDER ALL PORTIONS OF FIELD LINE
```

```
C
      ABX(1)=0.0
      IF (JUP(2).EQ.0) GO TO 10
      JEND=JUP(2)-1
      DO 6 J=2,JEND
      JJ=2-J+JUP(2)
      JM1=J-1
      BX(JM1)=DB(JJ,2)
      FLOX(JM1)=DLONG(JJ,2)
      FLAX(JM1)=DLAT(JJ,2)
      HBX(JM1)=DH(JJ,2)
6     ABX(J)=DARC(JJ,2)
      IF (KASE.EQ.2) GO TO 8
      JEND=JUP(1)-1
      DO 7 J=2,JEND
      JPJ=JUP(2)-3+J
      BX(JPJ)=DB(J,1)
      FLAX(JPJ)=DLAT(J,1)
      FLOX(JPJ)=DLONG(J,1)
      HBX(JPJ)=DH(J,1)
7     ABX(JPJ+1)=DARC(J+1,1)
      KEQ=JUP(2)-3+KEQ
      GO TO 14
8     JEND=JUP(1)-1
      DO 9 J=2,JEND
      JPJ=JUP(2)-3+J
      JJ=2-J+JUP(1)
      BX(JPJ)=DB(JJ,1)
      FLAX(JPJ)=DLAT(JJ,1)
      FLOX(JPJ)=DLONG(JJ,1)
      HBX(JPJ)=DH(JJ,1)
9     ABX(JPJ+1)=DARC(JJ,1)
      KEQ=JUP(1)+JUP(2)-1-KEQ
      GO TO 14
10    IF (KASE.EQ.2) GO TO 12
      JEND=JUP(1)-1
      DO 11 J=2,JEND
      JM1=J-1
      BX(JM1)=DB(J,1)
      FLAX(JM1)=DLAT(J,1)
      FLOX(JM1)=DLONG(J,1)
      HBX(JM1)=DH(J,1)
11    ABX(J)=DARC(J+1,1)
      KEQ=KEQ-1
      GO TO 14
12    JEND=JUP(1)-1
      DO 13 J=2,JEND
      JJ=JUP(1)-J+2
      JM1=J-1
      BX(JM1)=DB(JJ,1)
      FLAX(JM1)=DLAT(JJ,1)
      FLOX(JM1)=DLONG(JJ,1)
      HBX(JM1)=DH(JJ,1)
13    ABX(J)=DARC(JJ,1)
      KEQ=JUP(1)+1-KEQ
14    IF (JUP(3).EQ.0) GO TO 16
      JEND=JUP(3)-1
```

```
      DO 15 J=2,JEND
      JPJ=JUP(1)+JUP(2)-5+J
      IF (JUP(2).EQ.0) JPJ=JPJ+2
      BX(JPJ)=DB(J,3)
      FLOX(JPJ)=DLONG(J,3)
      FLAX(JPJ)=DLAT(J,3)
      HBX(JPJ)=DH(J,3)
15    ABX(JPJ+1)=DARC(J+1,3)
      JLAST=JUP(1)+JUP(2)+JUP(3)-5
      IF (JUP(2).EQ.0) JLAST=JLAST+2
      JU=JUP(3)
      BX(JLAST)=DB(JU,3)
      FLOX(JLAST)=DLONG(JU,3)
      FLAX(JLAST)=DLAT(JU,3)
      HBX(JLAST)=DH(JU,3)
      GO TO 17
16    JLAST=JUP(1)-1
      IF (JUP(2).NE.0) JLAST=JLAST+JUP(2)-2
      JU=2
      IF (KASE.EQ.1) JU=JUP(1)
      BX(JLAST)=DB(JU,1)
      FLOX(JLAST)=DLONG(JU,1)
      HBX(JLAST)=DH(JU,1)
      FLAX(JLAST)=DLAT(JU,1)
17    CONTINUE
      S(KEQ)=0.0
      IEND=KEQ-1
      DO 18 I=1,IEND
      II=KEQ-I
18    S(II)=S(II+1)-ABX(II+1)
      IEND=JLAST-1
      DO 19 I=KEQ,IEND
19    S(I+1)=S(I)+ABX(I+1)
C
C     DEFINE X-MESH IN CASE OF PREFIXED ALTITUDES (MESHB.EQ.1)
C
      IF (MESHB.EQ.0) GO TO 28
      ISTAR=3
      FSTAR=ISTAR
      ENC=.002*SL*SL*SL/(1.25*1.25*1.25)
20    IF (BX(KEQ).LT.(.311653/(SL*SL*SL)+FSTAR*ENC)) GO TO 21
      ISTAR=ISTAR+1
      FSTAR=FSTAR+1.
      GO TO 20
21    X(1)=BX(KEQ)
      DO 22 I=ISTAR,12
      EI=I
22    X(I)=.311653/(SL*SL*SL)+EI*ENC
      ISM1=ISTAR-1
      FIS=ISM1
      DEL=(X(ISTAR)-X(1))/FIS
      DO 23 I=2,ISM1
23    X(I)=X(I-1)+DEL
      CALL SPLINE (HBX,BX,Y2DOT,KEQ)
      DO 24 J=INT2,100
24    CALL YSPLN (HBX,BX,KEQ,Y2DOT,H(J),X(J))
      ISTAR=12
```

```
25    IF (X(ISTAR).LT.X(INT2)) GO TO 26
      ISTAR=ISTAR-1
      IF (ISTAR.GT.0) GO TO 25
      STOP
26    N=INT2-ISTAR
      EN=N
      DEL=(X(INT2)-X(ISTAR))/EN
      LIMIT=INT2-2
      DO 27 I=ISTAR,LIMIT
27    X(I+1)=X(I)+DEL
      GO TO 33
C
C     DETERMINE THE BOUNDS OF THE X-MESH COMPATIBLE WITH AVAILABLE
C     POINTS ON THE FIELD LINES,IN CASE OF PREFIXED X-MESH (MESHB.EQ.0)
C
28    DO 29 J=1,100
      IF (BX(KEQ).LT.X(J)) GO TO 30
29    CONTINUE
      WRITE (6,58)
      STOP
30    JMIN=J
      DO 31 J=JMIN,100
      IF (BX(JLAST).LT.X(J)) GO TO 32
31    CONTINUE
      JMAX=101
32    JMAX=J-1
33    CONTINUE
C
C     INTERPOLATE ALTITUDES AND ARC LENGTHS ONTO X MESH
C
      IF (((FLOX(1).LE.100.).AND.(FLOX(JLAST).GT.250.)).OR.((FLOX(1).GT.
     1250.).AND.(FLOX(JLAST).LE.100.))) GO TO 34
      GO TO 36
34    DO 35 I=1,JLAST
      IF (FLOX(I).GT.180.) FLOX(I)=FLOX(I)-360.
35    CONTINUE
36    CONTINUE
      KEQ1=KEQ-1
      DO 37 I=1,KEQ1
      II=KEQ-I
      SKN(I)=S(II)
37    BKN(I)=BX(II)
      CALL SPLINE (BKN,SKN,Y2DOT,KEQ1)
      JJ=JMIN
      IF (MESHB.GT.0) ) JJ=2
      DO 38 J=JJ,JMAX
      CALL YSPLN (BKN,SKN,KEQ1,Y2DOT,X(J),SXN(J))
38    CONTINUE
      NPTS=IEND+1-KEQ
      DO 39 J=1,NPTS
      JK1=J+KEQ
      SKN(J)=S(JK1)
39    BKN(J)=BX(JK1)
      CALL SPLINE (BKN,SKN,Y2DOT,NPTS)
      DO 40 J=JJ,JMAX
40    CALL YSPLN (BKN,SKN,NPTS,Y2DOT,X(J),SXP(J))
      NPTS=IEND+1
```

```
      CALL SPLINE (S,HBX,Y2DOT,NPTS)
      DO 41 J=JMIN,JMAX
      CALL YSPLN (S,HBX,NPTS,Y2DOT,SXN(J),HXN(J))
41    CALL YSPLN (S,HBX,NPTS,Y2DOT,SXP(J),HXP(J))
      CALL SPLINE (S,FLAX,Y2DOT,NPTS)
      DO 42 J=JMIN,JMAX
      CALL YSPLN (S,FLAX,NPTS,Y2DOT,SXN(J),LATN(J))
42    CALL YSPLN (S,FLAX,NPTS,Y2DOT,SXP(J),LATP(J))
      CALL SPLINE (S,FLOX,Y2DOT,NPTS)
      DO 43 J=JMIN,JMAX
      CALL YSPLN (S,FLOX,NPTS,Y2DOT,SXN(J),LONGN(J))
43    CALL YSPLN (S,FLOX,NPTS,Y2DOT,SXP(J),LONGP(J))
C
C
C     OUTPUT - - COORDINATES OF X-MESH POINTS ALONG THE NORTH AND THE
C     SOUTH PORTIONS OF THE FIELD LINE
C
      DO 44 J=JMIN,JMAX
      IF (LONGN(J).GT.180.) LONGN(J)=LONGN(J)-360.
      IF (LONGP(J).GT.180.) LONGP(J)=LONGP(J)-360.
      IF (LONGN(J).LT.-180.) LONGN(J)=LONGN(J)+360.
      IF (LONGP(J).LT.-180.) LONGP(J)=LONGP(J)+360.
      SXN(J)=ABS(SXN(J))*6371.2
44    SXP(J)=SXP(J)*6371.2
      IF (LATN(JMAX).LT.LATP(JMAX)) GO TO 45
      IF (MOD(LINE,NPR).EQ.0) WRITE (6,59) (I,X(I),HXN(I),HXP(I),SXN(I),
     1SXP(I),LATN(I),LATP(I),LONGN(I),LONGP(I),I=JMIN,JMAX)
      GO TO 46
45    IF (MOD(LINE,NPR).EQ.0) WRITE (6,59) (I,X(I),HXP(I),HXN(I),SXP(I),
     1SXN(I),LATP(I),LATN(I),LONGP(I),LONGN(I),I=JMIN,JMAX)
46    CONTINUE
      DRIFT=DRIFT*PATH
      PHI=EVN(3)
      BACKSPACE 10
      WRITE (10) SL,FLONG,LINE,SLOPE,DRIFT,CLONG,DPHIDX,PHI,(X(I),HXN(I)
     1,HXP(I),SXN(I),SXP(I),I=JMIN,JMAX)
      END FILE 10
C
C     INITIALIZE FOR THE NEXT LINE
C
      DO 47 I=1,3
47    EVN1(I)=EVN(I)
      IF ((FLONG.LT.SLONG2).OR.(FLONG.GT.SLONG3)) GO TO 48
      FLONG=FLONG+DLONG1
      IF (ABS(FLONG-SFLONG).LT.(DLONG1*.99)) GO TO 51
      GO TO 50
48    IF ((FLONG.LT.SLONG1).OR.(FLONG.GT.SLONG4)) GO TO 49
      FLONG=FLONG+DLONG2
      IF (ABS(FLONG-SFLONG).LT.(DLONG2*.99)) GO TO 51
      GO TO 50
49    FLONG=FLONG+DLONG3
      IF (ABS(FLONG-SFLONG).LT.(DLONG3*.99)) GO TO 51
50    LINE=LINE+1
      GO TO 4
51    CONTINUE
C
52    FORMAT (3F10.4,I10)
```

```
53    FORMAT (4F10.2)
54    FORMAT (I10,2F10.3)
55    FORMAT (2I10)
56    FORMAT (3F10.2,5F10.5)
57    FORMAT (//1X8HAT LONG=,F8.2,1X,34HTHE COORDINATES OF THE POINT AT
     1L=,F6.2,1X,7HAND B =,F7.4,1X,9HARE ALT =,F10.2,1X,9HAND LAT =,F6.2
     2/1X,51HTHE EQUATORIAL POINT OF THE FIELD LINE IS AT EALT =,F10.2,1
     3X,6HELAT =,F6.2,1X,7HELONG =,F8.2,1X,15HAND HAS B VALUE,F7.4/1X,32
     4HTHE DRIFT PARAMETERS ARE DRIFT =,F7.3,2X,7HSLOPE =,F7.3,2X,6HPATH
     5 =,F8.3,2X,8HPSLOPE =,F7.3,2X,8HDPHIDX =,E12.3,///)
58    FORMAT (20H CHOOSE ANOTHER MESH)
59    FORMAT (5X,11HMESH POINTS,11X,13HALTITUDES(KM),7X,27HARC LENGTH FR
     1OM EQUATOR(KM),7X,9HLATITUDES,15X,10HLONGITUDES//7X,7H(GAUSS),10X,
     25HNORTH,7X,5HSOUTH,12X,5HNORTH,7X,5HSOUTH,9X,5HNORTH,5X5HSOUTH,9X,
     35HNORTH,6X,5HSOUTH//(I4,F10.5,4X,2F12.1,4X,2F12.1,4X,2F10.2,4X,2F1
     41.2))
      END


      SUBROUTINE TRACE (DB,DARC,DH,KEQ,ALT,FLAT,FLONG,ERR,ERRB,ERRL,SB,S
     1L,KASE,JUP,DRIFT,CLONG,SLOPE,EVN,DLONG,DLAT,PATH,PSLOPE)
      COMMON B(200),VN1(200),VN2(200),ARC(200),VNEAR(3),VCONJ(3),VLAST(3
     1),VPLAST(3),VBO(3),VSAVE(3),BCONJ,VO,BO,ALTO,MMM,JNEAR,BNEAR,BSAVE
     2,BRSAVE,BPSAVE,BTSAVE,VN3(200),JSTOP
C
C     THIS SUBROUTINE COMPLETES TRACING OF THE FIELD LINE DEFINED
C     IN BLRING, TO THE ALTITUDE LEVEL GIVEN BY ALTO, AND TO ITS
C     CONJUGATE POINT (WHERE THE SAME B-VALUE IS ATTAINED IN THE
C     OPPOSITE HEMISPHERE). THE HEMISPHERE IN WHICH TRACING TO ALTO IS
C     PERFORMED, IS SELECTED SO THAT THE ALTITUDE OF THE CONJUGATE
C     POINT IS GREATER THAN ALTO
C
      DIMENSION V(3), DB(200,3), DARC(200,3), DH(200,3), JUP(3), EVN(3)
      DIMENSION DLONG(200,3), DLAT(200,3)
      ERR1=2.5E-04*ERR
      ERR2=2.5E-05*ERR
      KK=0
      CALL BLRING (ALT,FLAT,FLONG,ERR,ERRB,ERRL,SB,SL,JUP(1),DRIFT,SLOPE
     1,AEQ,EB,EALT,PATH,PSLOPE)
      JU=JUP(1)
      DO 1 J=1,JU
      DB(J,1)=B(J)
      DARC(J,1)=ABS(ARC(J))
      DLAT(J,1)=90.-VN2(J)*57.2957795
      DLONG(J,1)=VN3(J)*57.2957795
      SIT=ABS(SIN(VN2(J)))
      SSQ=SIT*SIT
      OER=(6356.912+SSQ*(21.3677+.108*SSQ))/6371.2
      AER=VN1(J)-OER
1     DH(J,1)=AER*6371.2
      VSAVE(1)=VN1(2)
      VSAVE(2)=VN2(2)
      VSAVE(3)=VN3(2)
      VLAST(1)=VN1(JU)
      VLAST(2)=VN2(JU)
      VLAST(3)=VN3(JU)
      DO 2 I=1,3
2     EVN(I)=VNEAR(I)
C
```

```
C       INSERT EQUATORIAL POINT
C
3       IF (ABS(EB-DB(JNEAR,1)).LE.2.E-6) GO TO 6
        IF (AEQ.GE.DARC(JNEAR,1)) GO TO 5
        JU=JUP(1)
        DO 4 J=JNEAR,JU
        JJ=JUP(1)-J+JNEAR+1
        DB(JJ,1)=DB(JJ-1,1)
        DLAT(JJ,1)=DLAT(JJ-1,1)
        DLONG(JJ,1)=DLONG(JJ-1,1)
        DH(JJ,1)=DH(JJ-1,1)
4       DARC(JJ,1)=DARC(JJ-1,1)
        JUP(1)=JUP(1)+1
        KEQ=JNEAR
        DB(JNEAR,1)=EB
        DLAT(JNEAR,1)=90.-VNEAR(2)*57.2957795
        DLONG(JNEAR,1)=VNEAR(3)*57.2957795
        DH(JNEAR,1)=EALT
        DARC(JNEAR+1,1)=AEQ
        DARC(JNEAR,1)=DARC(JNEAR,1)-AEQ
        GO TO 8
5       IF (AEQ.EQ.DARC(JNEAR,1)) GO TO 7
        AEQ=AEQ-DARC(JNEAR,1)
        JNEAR=JNEAR-1
        GO TO 3
6       KEQ=JNEAR
        GO TO 8
7       KEQ=JNEAR-1
8       CONTINUE
C
C       COMPLETE THE TRACING OF THE FIELD LINE TO ALTO (USUALLY 100 KM)
C       AND TO THE CONJUGATE OF THIS POINT (WHERE THE SAME B-VALUE IS
C       ATTAINED).
C
        IF (VLAST(1).LT.VSAVE(1)) GO TO 11
9       KASE=1
        RUM1=VLAST(1)
        RUM2=VLAST(2)
        RUM3=VLAST(3)
        DO 10 I=1,3
10      VCONJ(I)=VSAVE(I)
        GO TO 13
11      DO 12 I=1,3
12      VCONJ(I)=VLAST(I)
        KASE=2
        RUM1=VSAVE(1)
        RUM2=VSAVE(2)
        RUM3=VSAVE(3)
13      CONTINUE
        CALL INSECT (CLONG,ERR2,JUP(2))
        IF (JUP(2).EQ.0) GO TO 15
        JU=JUP(2)
        DO 14 J=3,JU
        DB(J,2)=B(J)
        DLAT(J,2)=90.-VN2(J)*57.2957795
        DLONG(J,2)=VN3(J)*57.2957795
        DARC(J,2)=ABS(ARC(J))
```

```
      SIT=ABS(SIN(VN2(J)))
      SSQ=SIT*SIT
      OER=(6356.912+SSQ*(21.3677+.108*SSQ))/6371.2
      AER=VN1(J)-OER
14    DH(J,2)=AER*6371.2
      BO=BCONJ
15    JU=JUP(1)
      IF (JUP(2).EQ.0) BO=DB(JU,1)
      IF ((JUP(2).EQ.0).AND.(KASE.EQ.1)) BO=DB(2,1)
      CALL BESECT (RUM1,PUM2,RUM3,BALT,ERR2,JUP(3))
      IF ((JUP(2).EQ.0).OR.(BALT.GT.100.).OR.(KK.EQ.1)) GO TO 16
      KK=1
      GO TO (11,9), KASE
16    IF (JUP(3).EQ.0) GO TO 18
      JU=JUP(3)
      DO 17 J=2,JU
      DB(J,3)=B(J)
      DLAT(J,3)=90.-VN2(J)*57.2957795
      DLONG(J,3)=VN3(J)*57.2957795
      DARC(J,3)=ABS(ARC(J))
      SIT=ABS(SIN(VN2(J)))
      SSQ=SIT*SIT
      OER=(6356.912+SSQ*(21.3677+.108*SSQ))/6371.2
      AER=VN1(J)-OER
17    DH(J,3)=AER*6371.2
18    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BLRING (ALT,FLAT,FLONG,ERR,ERRB,ERRL,SB,SL,JU,DRIFT,SLO
     1PE,AEQ,EB,EALT,PATH,PSLOPE)
      COMMON B(200),VN1(200),VN2(200),ARC(200),VNEAR(3),VCONJ(3),VLAST(3
     1),VPLAST(3),VBO(3),VSAVE(3),BCONJ,VO,BO,ALTO,MMM,JNEAR,BNEAR,BSAVE
     2,BRSAVE,BPSAVE,BTSAVE,VN3(200),JSTOP
      DIMENSION SVB(200), SV1(200), SV2(200), SV3(200)
      DIMENSION V(3), SARC(200)
C
C     THIS SUBROUTINE DEFINES THE FIELD LINE THROUGH AN SB-SL POINT,THE
C     GRADIENT OF THE SECOND INVARIANT, AND THE VARIATION OF THE DRIFT
C     FACTOR ALONG THE FIELD LINE
C
      ERR1=2.5E-04*ERR
      ITEST=1
      SSB=SB
      BZERO=.311653/(SL*SL*SL)
      SB=(SB+BZERO)*0.5
      SSL=SL
    1 SL=SL*1.03
      CALL SEARCH (ALT,FLAT,FLONG,SB,SL,V,BB,FL,FI,ERR,ERRB,ERRL,JU)
      FI1=FI
      VS12=VSAVE(2)
      CALL EQUAT (VNEAR(1),VNEAR(2),VNEAR(3),EB,SL,EALT,ERR1,AEQ)
      SL=SSL
      VN11=VNEAR(1)
      VN12=VNEAR(2)
      VN13=VNEAR(3)
      CALL SEARCH (ALT,FLAT,FLONG,SB,SL,V,BB,FL,FI,ERR,ERRB,ERRL,JU)
      C11=VSAVE(1)
      C12=VSAVE(2)
      C13=VSAVE(3)
      C21=VN1(4)
      C22=VN2(4)
      C23=VN3(4)
      VS22=VSAVE(2)
      SIT1=ABS(SIN(VSAVE(2)))
      FACT=BPSAVE/(BTSAVE*SIT1)
      IF (ITEST.EQ.1) GO TO 3
      IF (JSTOP.GT.0) JU=JSTOP
      JS=JNEAR
      DO 2 J=1,JU
      SARC(J)=ARC(J)
      SVB(J)=B(J)
      SV1(J)=VN1(J)
      SV2(J)=VN2(J)
    2 SV3(J)=VN3(J)
    3 CALL EQUAT (VNEAR(1),VNEAR(2),VNEAR(3),EB,SL,EALT,ERR1,AEQ)
      VN21=VNEAR(1)
      VN22=VNEAR(2)
      VN23=VNEAR(3)
      CALL INVAR (C21,C22,C23,ERR1,BIN2,DUM,FII2,NO)
      CALL INVAR (C11,C12,C13,ERR1,BIN1,DUM,FII1,NO)
      CORR=FACT*(VS12-VS22)
      DPHI=ABS(VN13-VN23+CORR)
      DTHE=ABS(VN22-VN12)
      DR=ABS(VN21-VN11)
      SSQ=SIT1*SIT1
```

```
      DIST=SQRT(DR*DR+VN21*VN21*(DTHE*DTHE+DPHI*DPHI*SSQ))
      QQQ=BZERO/SB
      QQ=BZERO/EB
      IF (ITEST.EQ.2) GO TO 4
      PATH1=FII1+2.*BIN1*(FII1-FII2)/(BIN1-BIN2)
      DRIFT1=(FI1-FI)*QQ/(DIST*PATH1)
      ROOT1=SQRT(QQQ)
      SB=SSB
      ITEST=2
      GO TO 1
    4 PATH=FII1+2.*BIN1*(FII1-FII2)/(BIN1-BIN2)
      DRIFT=(FI1-FI)*QQ/(DIST*PATH)
      ROOT=SQRT(QQQ)
      SLOPE=(DRIFT-DRIFT1)/(ROOT1-ROOT)
      DRIFT=DRIFT-SLOPE*(1.-ROOT)
      PSLOPE=(PATH-PATH1)/(ROOT1-ROOT)
      PATH=PATH-PSLOPE*(1.-ROOT)
C
C     DRIFT IS THE GEOMETRIC FACTOR IN THE EQUATORIAL DRIFT VELOCITY
C     OF PARTICLES MIRRORING AT THE EQUATOR.  DRIFT FOR  PARTICLES
C     MIRRORING AT ANY OTHER POINT OF FIELD STRENGTH X IS OBTAINED
C     BY THE EXPRESSION DRIFT+SLOPE*(1.-SQRT(BZERO/X)).
C     TO OBTAIN THE REAL EQUATORIAL DRIFT VELOCITY IN CM/SEC, MULTIPLY
C     DRIFT BY 2.5766 E 05 * (SL**3) * (MASS) * (BETA**2) * (GAMMA)
C     MASS IN ELECTRON MASSES, BETA AND GAMMA USUAL RELATIVISTIC FACTOR
C     THE RECTIFIED LENGTH OF THE PARTICLES TRAJECTORY BETWEEN TWO
C     CONJUGATE MIRROR POINTS AT X, IS GIVEN BY
C     PATH + PSLOPE*(1.-SQRT(BZERO/X)). TO OBTAIN REAL PATH IN CM,
C     MULTIPLY PATH BY 6.3712E 08
C
      JNEAR=JS
      VNEAR(1)=VN21
      VNEAR(2)=VN22
      VNEAR(3)=VN23
      DO 5 J=1,JU
      ARC(J)=SARC(J)
      B(J)=SVB(J)
      VN1(J)=SV1(J)
      VN2(J)=SV2(J)
    5 VN3(J)=SV3(J)
      RETURN
      END
```

```
      SUBROUTINE SEARCH (ALT,FLAT,FLONG,SB,SL,V,BB,FL,FI,ERR,ERRB,ERRL,J
     1UP)
      COMMON B(200),VN1(200),VN2(200),ARC(200),VNEAR(3),VCONJ(3),VLAST(3
     1),VPLAST(3),VBO(3),VSAVE(3),BCONJ,VO,BO,ALTO,MMM,JNEAR,BNEAR,BSAVE
     2,BRSAVE,BPSAVE,BTSAVE,VN3(200),JSTOP
C
C     THIS SUBROUTINE FINDS A FIELD LINE THROUGH A POINT WITH A PREFIXED
C     B AND L VALUE FOR A GIVEN LONGITUDE.
C     SUBROUTINE IS CALLED WITH ALT,FLAT,FLONG = APPROXIMATE STARTING
C     COORDINATES WHICH SHOULD NOT BE TOO FAR AWAY FROM REAL B-L POINT
C     (E.G. NOT MORE THAN 20 DEGREES LATITUDE)
C     FLAT DETERMINES THE HEMISPHERE OF THE B-L POINT
C     FLONG REMAINS CONSTANT
C
      DIMENSION V(3), V1(3), V2(3)
      SERR=ERR
      SERRB=ERRB
      SERRL=ERRL
      SSB=SB
1     CONTINUE
      VO=0.
      DV=0.02
      MCHECK=0
      ICHECK=0
      V(1)=ALT/6371.2
      V(2)=(90.-FLAT)/57.2957795
      V(3)=FLONG/57.2957795
      DCLT=1.5708-0.2007*COS(V(3)+1.239)
      SIT=ABS(SIN(V(2)))
      SSQ=SIT*SIT
      OER=(6356.912+SSQ*(21.3677+.108*SSQ))/6371.2
      V(1)=V(1)+OER
      VO=ALTO/6371.2+OER
      ICON=1
2     ILIT=1
      DELV2=DV
      ICHECK=ICHECK+1
      IF (ICHECK.GT.15) GO TO 12
3     SIT=ABS(SIN(V(2)))
      CALL NEWMAG (V(1),SIT,V(3),BR,BP,BT,BB,V(2))
4     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
      V1(1)=V(1)
      V1(2)=V(2)
      MCHECK=MCHECK+1
      IF (MCHECK.GT.15) GO TO 13
      CALL NEWMAG (V(1),SIT,V(3),BR,BP,BT,BB,V(2))
      IF (ABS((BB-SB)/SB).GT.ERRB) GO TO 4
      MCHECK=0
      IF (ILIT.NE.1) GO TO 8
      ILIT=2
      CALL INVAR (V(1),V(2),V(3),ERR,BB,FL,FI,JUP)
      V2(1)=V(1)
      V2(2)=V(2)
      B2=BB
```

```
      FL2=FL
      IF (ABS((FL-SL)/SL).LE.ERRL) GO TO 9
      IF (ABS(V(2)-DCLT).LT.0.1) GO TO 7
      SGN=SIGN(1.,(FL-SL))
      IF (V(2).LT.DCLT) GO TO 5
      DELV2=-SGN*DELV2
      GO TO 6
5     DELV2=SGN*DELV2
6     V(2)=V(2)+DELV2
      GO TO 3
7     V(2)=V(2)+DELV2
      CALL INVAR (V(1),V(2),V(3),ERR,BB,FL,FI,JUP)
      IF ((FL-SL)*(FL-FL2).LE.0.) GO TO 3
      V(2)=V(2)-2.*DELV2
      GO TO 3
8     B1=BB
      CALL INVAR (V(1),V(2),V(3),ERR,BB,FL,FI,JUP)
      IF (ABS((FL-SL)/SL).LE.ERRL) GO TO 9
      FACT=(SL-FL)/(FL2-FL)
      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 2
      DV=Y
      GO TO 2
9     CONTINUE
      IF (ICON.EQ.2) GO TO 10
      ICON=2
      DV=DV*0.1
      ERR=ERR*0.025
      ERRB=ERRB*0.05
      ERRL=ERRL*0.01
      GO TO 2
10    SIT=ABS(SIN(V(2)))
      SSQ=SIT*SIT
      OER=(6356.912+SSQ*(21.3677+.108*SSQ))/6371.2
      AER=V(1)-OER
C
C     WHEN THE CALCULATION IS COMPLETED, ALT AND FLAT BECOME FINAL
C     COORDINATES OF THE B-L POINT. VSAVE ARE COORDINATES OF SAME
C     POINT IN GEOCENTRIC SYSTEM
C
      ALT=AER*6371.2
      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 11 I=1,3
11    VSAVE(I)=V(I)
      GO TO 15
12    WRITE (6,16) ICHECK,MCHECK,ALT,FLAT,FLONG,SB,SL
      GO TO 14
13    WRITE (6,17) ICHECK,MCHECK,ALT,FLAT,FLONG,SB,SL
14    SB=SB+.005
      ERR=SERR
      ERRB=SERRB
```

```
      ERRL=SERRL
      GO TO 1
15    ERR=SERR
      ERRB=SERRB
      ERRL=SERRL
      SB=SSB
      RETURN
C
16    FORMAT (51H  SORRY,BUT I CANNOT FIND THAT DAMN POINT IN ICHECK/2I1
     10,5E15.5)
17    FORMAT (51H  SORRY,BUT I CANNOT FIND THAT DAMN POINT IN MCHECK/2I1
     10,5E15.5)
      END
```

```
      SUBROUTINE EQUAT (DUM1,DUM2,DUM3,EB,SL,EALT,ERR,AEQ)
      COMMON B(200),VN1(200),VN2(200),ARC(200),VNEAR(3),VCONJ(3),VLAST(3
     1),VPLAST(3),VBO(3),VSAVE(3),BCONJ,VO,BO,ALTO,MMM,JNEAR,BNEAR,BSAVE
     2,BRSAVE,BPSAVE,BTSAVE,VN3(200),JSTOP
C
C     THIS SUBROUTINE TRACES THE FIELD LINE FROM AN INITIAL POINT
C     TO THE POINT OF MINIMUM INTENSITY.
C
      DIMENSION V(3,3), VN(3), VP(3), R1(3), R2(3), R3(3)
      ERR1=ERR
      MMM=1
      V(1,2)=DUM1
      V(2,2)=DUM2
      V(3,2)=DUM3
      ARC(1)=0.
      DCLT=1.5708-0.2007*COS(V(3,2)+1.239)
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)
      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 (6,7) ERR
      GO TO 1
5     ERR=ERR1
      JUP=J
      AEQ=0.0
      DO 6 J=3,JUP
6     AEQ=AEQ+ABS(ARC(J))
      EB=BNEAR
      SIT=ABS(SIN(VNEAR(2)))
      SSQ=SIT*SIT
      OER=(6356.912+SSQ*(21.3677+.108*SSQ))/6371.2
      AER=VNEAR(1)-OER
      EALT=AER*6371.2
      RETURN
C
7     FORMAT (28H INCREASE OF ERR IN EQUAT TO,E16.4)
      END
```

```
      SUBROUTINE INSECT (CLONG,ERR,JUP)
      COMMON B(200),VN1(200),VN2(200),ARC(200),VNEAR(3),VCONJ(3),VLAST(3
     1),VPLAST(3),VBO(3),VSAVE(3),BCONJ,VO,BO,ALTO,MMM,JNEAR,BNEAR,BSAVE
     2,BRSAVE,BPSAVE,BTSAVE,VN3(200),JSTOP
C
C     THIS SUBROUTINE TRACES THE FIELD LINE FROM AN INITIAL POINT
C     DOWNWARDS TO A PREFIXED (LOWER) ALTITUDE
C
      DIMENSION V(3,3), VN(3), VP(3), R1(3), R2(3), R3(3)
      ERR1=ERR
      J=0
      SIT=ABS(SIN(VCONJ(2)))
      SSQ=SIT*SIT
      OER=(6356.912+SSQ*(21.3677+.108*SSQ))/6371.2
      AER=VCONJ(1)-OER
      HR=6371.2*AER
      IF (HR.LE.ALTO) GO TO 5
      MMM=2
      V(1,2)=VCONJ(1)
      V(2,2)=VCONJ(2)
      V(3,2)=VCONJ(3)
      ARC(1)=0.
      DCLT=1.5708-0.2007*COS(V(3,2)+1.239)
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
      CLONG=VCONJ(3)*57.2957795
      IF (CLONG.GT.180.) CLONG=CLONG-360.
      IF (CLONG.LT.(-180.)) CLONG=CLONG+360.
      RETURN
C
6     FORMAT (29H INCREASE OF ERR IN INSECT TO,E16.4)
      END
```

```
      SUBROUTINE BESECT (RUM1,RUM2,RUM3,BALT,ERR,JUP)
      COMMON B(200),VN1(200),VN2(200),ARC(200),VNEAR(3),VCONJ(3),VLAST(3
     1),VPLAST(3),VBO(3),VSAVE(3),BCONJ,VO,BO,ALTO,MMM,JNEAR,BNEAR,BSAVE
     2,BRSAVE,BPSAVE,BTSAVE,VN3(200),JSTOP
C
C     THIS SUBROUTINE TRACES THE FIELD LINE DOWNWARDS FROM AN INITIAL
C     POINT TO A PREFIXED (HIGHER) B-VALUE
C
      DIMENSION V(3,3), VN(3), VP(3), R1(3), R2(3), R3(3)
      ERR1=ERR
      J=0
      MMM=3
      V(1,2)=RUM1
      V(2,2)=RUM2
      V(3,2)=RUM3
      ARC(1)=0.
      DCLT=1.5708-0.2007*COS(V(3,2)+1.239)
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 (B(2).GT.BO) 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 6
      ERR=4.*ERR
      WRITE (6,7) ERR
      GO TO 1
5     VBO(2)=RUM2
      VBO(1)=RUM1
6     ERR=ERR1
      JUP=J
      SIT=ABS(SIN(VBO(2)))
      SSQ=SIT*SIT
      OER=(6356.912+SSQ*(21.3677+.108*SSQ))/6371.2
      AER=VBO(1)-OER
      BALT=AER*6371.2
      RETURN
C
7     FORMAT (29H INCREASE OF ERR IN BESECT TO,E16.4)
      END
```

```
      SUBROUTINE INVAR (DUM1,DUM2,DUM3,ERR,BB,FL,FI,JUP)
      COMMON B(200),VN1(200),VN2(200),ARC(200),VNEAR(3),VCONJ(3),VLAST(3
     1),VPLAST(3),VBO(3),VSAVE(3),BCONJ,VO,BO,ALTO,MMM,JNEAR,BNEAR,BSAVE
     2,BRSAVE,BPSAVE,BTSAVE,VN3(200),JSTOP
C
C     SUBROUTINES INVAR,START,INTEGER,CARMEL, AND NEWMAG ARE BASED ON
C     ON MCILWAINS  1965 CODE.
C     INVAR MUST BE CALLED WITH GEOCENTRIC COORDINATES.
C     MMM IS A TRACING CONTROL PARAMETER DETERMINED IN EQUAT,INSECT,
C     BESECT AND INVAR.
C
      DIMENSION V(3,3), VN(3), VP(3), BEG(200), BEND(200), BLOG(200), EC
     1O(200), R1(3), R2(3), R3(3)
      V(1,2)=DUM1
      V(2,2)=DUM2
      V(3,2)=DUM3
      ERR1=ERR
      MMM=0
      ARC(1)=0.
      DCLT=1.5708-0.2007*COS(V(3,2)+1.239)
1     ARC(2)=DUM1*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)
      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 (6,8) 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)
      CALL CARMEL (B(2),FLINT,FL)
      FI=FLINT
      BB=B(2)
      RETURN
C
8     FORMAT (28H INCREASE OF ERR IN INVAR TO,E16.4)
      END
```

```
      SUBROUTINE START (R1,R2,R3,B,ARC,ERR,V)
      COMMON B(200),VN1(200),VN2(200),ARC(200),VNEAR(3),VCONJ(3),VLAST(3
     1),VPLAST(3),VBO(3),VSAVE(3),BCONJ,VO,BO,ALTO,MMM,JNEAR,BNEAR,BSAVE
     2,BRSAVE,BPSAVE,BTSAVE,VN3(200),JSTOP
      DIMENSION V(3,3), R1(3), R2(3), R3(3)
      M=MMM
      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 NEWMAG (V(1,2),SIT,V(3,2),BR,BT,BP,B(2),V(2,2))
      BSAVE=B(2)
      BRSAVE=BR
      BPSAVE=BP
      BTSAVE=BT
      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 NEWMAG (V,SIT,V(3,1),BR,BT,BP,B(1),V(2,1))
      IF (M.LT.2) GO TO 7
      IF (B(1)-B(2)) 9,9,8
7     IF (B(1)-B(2)) 8,9,9
8     ARC(2)=-ARC(2)
      GO TO 4
9     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 10 I=1,3
10    V(I,1)=V(I,2)-ARC(2)*(R1(I)+R2(I))/2.
      SIT=ABS(SIN(V(2,1)))
      IS=IS+1
      GO TO (6,11), IS
11    DO 12 I=1,3
12    V(I,3)=V(I,2)+ARC(3)*((1.5)*R2(I)-.5*R1(I))
      RETURN
      END
```

```
      SUBROUTINE LINES (R1,R2,R3,B,ARC,ERR,J,VP,VN)
      COMMON B(200),VN1(200),VN2(200),ARC(200),VNEAR(3),VCONJ(3),VLAST(3
     1),VPLAST(3),VBO(3),VSAVE(3),BCONJ,VO,BO,ALTO,MMM,JNEAR,BNEAR,BSAVE
     2,BRSAVE,BPSAVE,BTSAVE,VN3(200),JSTOP
      INTEGER FLAG1,FLAG2
      DIMENSION R1(3), R2(3), R3(3), VN(3), VP(3), RA(3)
      JSTOP=0
      M=MMM
      FLAG1=0
      FLAG2=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 NEWMAG (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)))
      B(J)=B(J)*((PRE1/VN(1))**3)
C
C     WHEN CALLED FROM EQUAT,ARC LENGTH IS MAINTAINED CONSTANT
C
      IF (M.EQ.1) GO TO 22
      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 (22,19,22), ILP
19    IF (X-3.3) 22,20,20
20    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 21 I=1,3
      VN(I)=VP(I)
      R3(I)=R2(I)
      R2(I)=R1(I)
21    R1(I)=RA(I)
      GO TO 38
22    VN1(J)=VN(1)
      VN2(J)=VN(2)
      VN3(J)=VN(3)
      IF (M.EQ.2) GO TO 28
      IF (M.EQ.3) GO TO 30
      IF (B(J-1).GT.B(J)) GO TO 32
      IF (M.EQ.1) GO TO 24
      IF (FLAG1.EQ.1) GO TO 26
      FLAG1=1
      DO 23 I=1,3
23    VNEAR(I)=VN(I)
      BNEAR=B(J)
      JNEAR=J
      GO TO 26
24    BNEAR=B(J-1)
      DO 25 I=1,3
25    VNEAR(I)=VP(I)
      GO TO 42
```

```
26    IF (FLAG2.EQ.1) GO TO 32
      IF ((VO+DEL).LT.VN(1)) GO TO 32
      DO 27 I=1,3
27    VCONJ(I)=VP(I)
      JSTOP=J-1
      FLAG2=1
      GO TO 32
28    SIT=ABS(SIN(VN(2)))
      SSQ=SIT*SIT
      OER=(6356.912+SSQ*(21.3677+.108*SSQ))/6371.2
      AER=VN(1)-OER
      HR=6371.2*AER
      IF (HR.GT.ALTO) GO TO 32
      SIT=ABS(SIN(VP(2)))
      SSQ=SIT*SIT
      OER=(6356.912+SSQ*(21.3677+.108*SSQ))/6371.2
      AER=VP(1)-OER
      HRP=6371.2*AER
      FAC=(ALTO-HRP)/(HR-HRP)
      DO 29 I=1,3
29    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)
      SIT=ABS(SIN(VCONJ(2)))
      CALL NEWMAG (VCONJ,SIT,VCONJ(3),BR,BT,BP,BB,VCONJ(2))
      BCONJ=BB
      B(J)=BB
      GO TO 42
30    IF (B(J).LT.BO) GO TO 32
      FAC=(BO-B(J-1))/(B(J)-B(J-1))
      DO 31 I=1,3
31    VBO(I)=VP(I)+(VN(I)-VP(I))*FAC
      ARC(J)=ARC(J)*FAC
      B(J)=BO
      VN1(J)=VBO(1)
      VN2(J)=VBO(2)
      VN3(J)=VBO(3)
      GO TO 42
32    IF (J.GE.200) GO TO 42
      A1=A2
      IF (M.NE.0) GO TO 33
      IF (B(J)-B(2)) 33,33,39
33    ILP=2
      A2=A3
      IF (M.EQ.1) GO TO 38
      A3=A3*.2*(8.+X)/(.8+X)
      AM=(2.-R3(2)*VN(1))*VN(1)*CRE
      IF (ABS(A3)-AM) 35,35,34
34    A3=SNA*AM
35    IF (SNA*R3(1)+.5) 36,36,38
36    AM=-.5*SNA*VN(1)/R3(1)
      IF (ABS(A3)-AM) 38,38,37
37    A3=SNA*AM
38    ARC(J+1)=A3
      AAB=ABS(A3)
      GO TO 3
39    BLAST=B(J)
      DO 40 ILAST=1,3
      VPLAST(ILAST)=VP(ILAST)
40    VLAST(ILAST)=VN(ILAST)
      IF (FLAG2.NE.0) GO TO 42
      DO 41 I=1,3
41    VCONJ(I)=VLAST(I)
42    RETURN
      END
```

```
      SUBROUTINE INTEG (ARC,BEG,BEND,B,JEP,ECO,FI)
      COMMON B(200),VN1(200),VN2(200),ARC(200),VNEAR(3),VCONJ(3),VLAST(3
     1),VPLAST(3),VB0(3),VSAVE(3),BCONJ,VO,BO,ALTO,MMM,JNEAR,BNEAR,BSAVE
     2,BRSAVE,BPSAVE,BTSAVE,VN3(200),JSTOP
      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 CARMEL (B,XI,VL)
      COMMON B(200),VN1(200),VN2(200),ARC(200),VNEAR(3),VCONJ(3),VLAST(3
     1),VPLAST(3),VBO(3),VSAVE(3),BCONJ,VO,BO,ALTO,MMM,JNEAR,BNEAR,BSAVE
     2,BRSAVE,BPSAVE,BTSAVE,VN3(200),JSTOP
      IF (XI-1.0E-36) 1,1,2
1     VL=(0.311653/B)**(1./3.)
      RETURN
2     XX=3.0*ALOG(XI)
      XX=XX+ALOG(B/0.311653)
      IF (XX+22.) 7,7,3
3     IF (XX+3.) 8,8,4
4     IF (XX-3.) 9,9,5
5     IF (XX-11.7) 10,10,6
6     IF (XX-23.) 11,11,12
7     GG=.333338*XX+.30062102
      GO TO 13
8     GG=((((((((-8.1537735E-14*XX+8.2232531E-13)*XX+1.0066362E-9)*XX+8.
     11048663E-8)*XX+3.2916354E-6)*XX+8.2711096E-5)*XX+1.3714667E-3)*XX+
     22.015017245)*XX+.43432642)*XX+.62337691
      GO TO 13
9     GG=((((((((2.6047023E-10*XX+2.3028767E-9)*XX-2.1997983E-8)*XX-5.39
     177642E-7)*XX-3.3408822E-6)*XX+3.8379917E-5)*XX+1.1784234E-3)*XX+1.
     224492441E-2)*XX+.43352783)*XX+.6228644
      GO TO 13
10    GG=((((((((6.2271665E-10*XX-3.958306E-8)*XX+9.9766148E-07)*XX-1.25
     12937E-5)*XX+7.9451313E-5)*XX-3.2077032E-4)*XX+2.1680398E-3)*XX+1.
     222817956E-2)*XX+.43510529)*XX+.6222355
      GO TO 13
11    GG=(((((2.8212095E-8*XX-3.8049276E-6)*XX+2.170224E-4)*XX-6.7310339
     1E-3)*XX+.12038224)*XX-.18461796)*XX+2.0007187
      GO TO 13
12    GG=XX-3.0460681
13    VL=(((1.0+EXP(GG))*0.311653)/B)**(1./3.)
      RETURN
      END
```

```
      SUBROUTINE NEWMAG (R,ST,PHI,BR,BT,BP,B,THET)
      COMMON B(200),VN1(200),VN2(200),ARC(200),VNEAR(3),VCONJ(3),VLAST(3
     1),VPLAST(3),VBO(3),VSAVE(3),BCONJ,VO,BO,ALTO,MMM,JNEAR,BNEAR,BSAVE
     2,BRSAVE,BPSAVE,BTSAVE,VN3(200),JSTOP
      DIMENSION G(7,7)
C     CAIN COEF.(SEE HENDRICKS,CAIN,JGR,71,JAN 1966,P346, EPOCH 1960
C     GSFC(9-65)- TRUNCATED
      DATAG/0.0,
     C    3.042500E+04,   2.304000E+03,  -3.252500E+03,  -4.191250E+03,
     C    1.795500E+03,  -7.218750E+02,  -5.775000E+03,   2.162000E+03,
     C   -5.196152E+03,   6.083920E+03,  -4.443791E+03,  -3.659969E+03,
     C   -1.153091E+03,   3.377499E+03,  -1.766692E+02,  -1.372650E+03,
     C   -2.498074E+03,  -1.968299E+03,  -1.775284E+03,  -7.472116E+01,
     C    1.319663E+03,  -4.473296E+02,   1.027740E+02,  -6.885860E+02,
     C    8.241101E+02,   1.458926E+02,   2.411003E+03,  -8.411659E+02,
     C    1.048716E+03,  -6.274950E+00,   1.856170E+02,  -2.004072E+02,
     C    3.483092E+02,   5.456862E+00,  -9.149923E+01,  -9.299108E+02,
     C    5.459207E+02,   2.440383E+02,  -5.612486E+01,   4.560145E+01,
     C    0.          ,   2.268375E+02,  -1.539256E+03,  -6.077321E+02,
     C    1.473353E+02,   2.792177E+01,   8.060319E+00,   7.321457E+01
     C/,J/0/,NMAX/7/
C     GSFC(9-65)     FOR NOV.,1965
C    C    3.030346E+04,   2.561535E+03,  -3.292325E+03,  -4.168019E+03,
C    C    1.674698E+03,  -6.622481E+02,  -5.751990E+03,   2.126600E+03,
C    C   -5.208415E+03,   6.264570E+03,  -4.505827E+03,  -3.653971E+03,
C    C   -1.264619E+03,   3.517501E+03,  -9.593829E+01,  -1.369074E+03,
C    C   -2.516355E+03,  -1.940594E+03,  -1.843298E+03,  -1.452579E+02,
C    C    1.202240E+03,  -4.804629E+02,   1.456861E+02,  -6.825223E+02,
C    C    8.302805E+02,   1.569993E+02,   2.287563E+03,  -7.823949E+02,
C    C    1.083347E+03,  -4.576530E+01,   2.096141E+02,  -1.820821E+02,
C    C    3.574717E+02,  -4.605592E+01,  -2.714477E+02,  -1.061405E+03,
C    C    5.986773E+02,   2.283311E+02,  -5.943623E+01,   3.897872E+01,
C    C    1.372820E+00,   3.829773E+02,  -1.548073E+03,  -6.959031E+02,
C    C    1.698721E+02,   2.929459E+01,   6.475123E+00,   7.281827E+01
C    C/,J/0/,NMAX/7/
1     P22=ST
      P21=SQRT(1.-P22*P22)
      AR=R
      IF (THET-1.570796327) 3,3,2
2     P21=-P21
3     IF (J) 5,5,4
4     SSQ=P22*P22
      AR=AR+(14.288-SSQ*(21.3677+.108*SSQ))/6371.2
5     AR=1./AR
C     N= 2
      DP22=P21
      SP2=SIN(PHI)
      CP2=COS(PHI)
      DP21=-P22
      AOR=AR*AR*AR
      C2=G(2,2)*CP2+G(1,2)*SP2
      BR=-(AOR+AOR)*(G(2,1)*P21+C2*P22)
      BT=AOR*(G(2,1)*DP21+C2*DP22)
      BP=AOR*(G(1,2)*CP2-G(2,2)*SP2)*P22
      IF (NMAX-3) 11,6,6
6     SP3=(SP2+SP2)*CP2
      CP3=(CP2+SP2)*(CP2-SP2)
```

```
      P31=P21*P21-0.333333333
      P32=P21*P22
      P33=P22*P22
      DP31=-P32-P32
      DP32=P21*P21-P33
      DP33=-DP31
      AOR=AOR*AR
      C2=G(3,2)*CP2+G(1,3)*SP2
      C3=G(3,3)*CP3+G(2,3)*SP3
      BR=BR-3.0*AOR*(G(3,1)*P31+C2*P32+C3*P33)
      BT=BT+AOR*(G(3,1)*DP31+C2*DP32+C3*DP33)
      BP=BP-AOR*((G(3,2)*SP2-G(1,3)*CP2)*P32+2.0*(G(3,3)*SP3-G(2,3)*CP3)
     1*P33)
C     N= 4
      IF (NMAX-4) 11,7,7
7     SP4=SP2*CP3+CP2*SP3
      CP4=CP2*CP3-SP2*SP3
      P41=P21*P31-0.26666666*P21
      DP41=P21*DP31+DP21*P31-0.26666666*DP21
      P42=P21*P32-0.20000000*P22
      DP42=P21*DP32+DP21*P32-0.20000000*DP22
      P43=P21*P33
      DP43=P21*DP33+DP21*P33
      P44=P22*P33
      DP44=3.0*P43
      AOR=AOR*AR
      C2=G(4,2)*CP2+G(1,4)*SP2
      C3=G(4,3)*CP3+G(2,4)*SP3
      C4=G(4,4)*CP4+G(3,4)*SP4
      BR=BR-4.0*AOR*(G(4,1)*P41+C2*P42+C3*P43+C4*P44)
      BT=BT+AOR*(G(4,1)*DP41+C2*DP42+C3*DP43+C4*DP44)
      BP=BP-AOR*((G(4,2)*SP2-G(1,4)*CP2)*P42+2.0*(G(4,3)*SP3-G(2,4)*CP3)
     1*P43+3.0*(G(4,4)*SP4-G(3,4)*CP4)*P44)
      IF (NMAX-5) 11,8,8
8     SP5=(SP3+SP3)*CP3
      CP5=(CP3+SP3)*(CP3-SP3)
      P51=P21*P41-0.25714285*P31
      DP51=P21*DP41+DP21*P41-0.25714285*DP31
      P52=P21*P42-0.22857142*P32
      DP52=P21*DP42+DP21*P42-0.22857142*DP32
      P53=P21*P43-0.14285714*P33
      DP53=P21*DP43+DP21*P43-0.14285714*DP33
      P54=P21*P44
      DP54=P21*DP44+DP21*P44
      P55=P22*P44
      DP55=4.0*P54
      AOR=AOR*AR
      C2=G(5,2)*CP2+G(1,5)*SP2
      C3=G(5,3)*CP3+G(2,5)*SP3
      C4=G(5,4)*CP4+G(3,5)*SP4
      C5=G(5,5)*CP5+G(4,5)*SP5
      BR=BR-5.0*AOR*(G(5,1)*P51+C2*P52+C3*P53+C4*P54+C5*P55)
      BT=BT+AOR*(G(5,1)*DP51+C2*DP52+C3*DP53+C4*DP54+C5*DP55)
      BP=BP-AOR*((G(5,2)*SP2-G(1,5)*CP2)*P52+2.0*(G(5,3)*SP3-G(2,5)*CP3)
     1*P53+3.0*(G(5,4)*SP4-G(3,5)*CP4)*P54+4.0*(G(5,5)*SP5-G(4,5)*CP5)*P
     255)
C     N= 6
```

```
      IF (NMAX-6) 11,9,9
9     SP6=SP2*CP5+CP2*SP5
      CP6=CP2*CP5-SP2*SP5
      P61=P21*P51-0.25396825*P41
      DP61=P21*DP51+DP21*P51-0.25396825*DP41
      P62=P21*P52-0.23809523*P42
      DP62=P21*DP52+DP21*P52-0.23809523*DP42
      P63=P21*P53-0.19047619*P43
      DP63=P21*DP53+DP21*P53-0.19047619*DP43
      P64=P21*P54-0.11111111*P44
      DP64=P21*DP54+DP21*P54-0.11111111*DP44
      P65=P21*P55
      DP65=P21*DP55+DP21*P55
      P66=P22*P55
      DP66=5.0*P65
      AOR=AOR*AR
      C2=G(6,2)*CP2+G(1,6)*SP2
      C3=G(6,3)*CP3+G(2,6)*SP3
      C4=G(6,4)*CP4+G(3,6)*SP4
      C5=G(6,5)*CP5+G(4,6)*SP5
      C6=G(6,6)*CP6+G(5,6)*SP6
      BR=BR-6.0*AOR*(G(6,1)*P61+C2*P62+C3*P63+C4*P64+C5*P65+C6*P66)
      BT=BT+AOR*(G(6,1)*DP61+C2*DP62+C3*DP63+C4*DP64+C5*DP65+C6*DP66)
      BP=BP-AOR*((G(6,2)*SP2-G(1,6)*CP2)*P62+2.0*(G(6,3)*SP3-G(2,6)*CP3)
     1*P63+3.0*(G(6,4)*SP4-G(3,6)*CP4)*P64+4.0*(G(6,5)*SP5-G(4,6)*CP5)*P
     265+5.0*(G(6,6)*SP6-G(5,6)*CP6)*P66)
      IF (NMAX-7) 11,10,10
10    SP7=(SP4+SP4)*CP4
      CP7=(CP4+SP4)*(CP4-SP4)
      P71=P21*P61-0.25252525*P51
      DP71=P21*DP61+DP21*P61-0.25252525*DP51
      P72=P21*P62-0.24242424*P52
      DP72=P21*DP62+DP21*P62-0.24242424*DP52
      P73=P21*P63-0.21212121*P53
      DP73=P21*DP63+DP21*P63-0.21212121*DP53
      P74=P21*P64-0.16161616*P54
      DP74=P21*DP64+DP21*P64-0.16161616*DP54
      P75=P21*P65-0.09090909*P55
      DP75=P21*DP65+DP21*P65-0.09090909*DP55
      P76=P21*P66
      DP76=P21*DP66+DP21*P66
      P77=P22*P66
      DP77=6.0*P76
      AOR=AOR*AR
      C2=G(7,2)*CP2+G(1,7)*SP2
      C3=G(7,3)*CP3+G(2,7)*SP3
      C4=G(7,4)*CP4+G(3,7)*SP4
      C5=G(7,5)*CP5+G(4,7)*SP5
      C6=G(7,6)*CP6+G(5,7)*SP6
      C7=G(7,7)*CP7+G(6,7)*SP7
      BR=BR-7.0*AOR*(G(7,1)*P71+C2*P72+C3*P73+C4*P74+C5*P75+C6*P76+C7*P7
     17)
      BT=BT+AOR*(G(7,1)*DP71+C2*DP72+C3*DP73+C4*DP74+C5*DP75+C6*DP76+C7*
     1DP77)
      BP=BP-AOR*((G(7,2)*SP2-G(1,7)*CP2)*P72+2.0*(G(7,3)*SP3-G(2,7)*CP3)
     1*P73+3.0*(G(7,4)*SP4-G(3,7)*CP4)*P74+4.0*(G(7,5)*SP5-G(4,7)*CP5)*P
     275+5.0*(G(7,6)*SP6-G(5,7)*CP6)*P76+6.0*(G(7,7)*SP7-G(6,7)*CP7)*P77
     3)
11    BP=BP/P22*1.E-5
      BT=BT*1.E-5
      BR=BR*1.E-5
      B=SQRT(BR*BR+BT*BT+BP*BP)
      RETURN
      END
```

```
      SUBROUTINE SPLINE (X,Y,Y2DOT,M)
      DIMENSION X(400), Y(400), D(400), P(400), E(400), A(400,3), B(400)
     1, Y2DOT(400)
C
C     THIS SUBROUTINE GENERATES COEFFICIENTS NEEDED FOR A SPLINE
C     INTERPOLATION.  GIVEN M VALUES OF X (INDEPENDENT VARIABLE)
C     AND THE CORRESPONDING Y VALUES (DEPENDENT VARIABLE),  THE ROUTINE
C     CALCULATES THE VALUES OF THE 2ND DERIVATIVE (Y2DOT) AT EACH VALUE
C     X.  M IS THE NUMBER OF DATA POINTS.
C
      DO 1 I=2,M
      IF (X(I).LE.X(I-1)) GO TO 2
1     CONTINUE
      GO TO 3
2     WRITE (6,8)
      WRITE (6,9) (I,X(I),Y(I),I=1,M)
3     CONTINUE
      IF (M.GT.400) WRITE (6,10)
      IF (M.GT.400) STOP
      IF (M.LT.4) WRITE (6,11)
      IF (M.LT.4) STOP
      MM=M-1
      DO 4 K=1,MM
      D(K)=X(K+1)-X(K)
      P(K)=D(K)/6.
4     E(K)=(Y(K+1)-Y(K))/D(K)
      DO 5 K=2,MM
5     B(K)=E(K)-E(K-1)
      A(1,2)=-1.-D(1)/D(2)
      A(1,3)=D(1)/D(2)
      A(2,3)=P(2)-P(1)*A(1,3)
      A(2,2)=2.*(P(1)+P(2))-P(1)*A(1,2)
      A(2,3)=A(2,3)/A(2,2)
      B(2)=B(2)/A(2,2)
      DO 6 K=3,MM
      A(K,2)=2.*(P(K-1)+P(K))-P(K-1)*A(K-1,3)
      B(K)=B(K)-P(K-1)*B(K-1)
      A(K,3)=P(K)/A(K,2)
6     B(K)=B(K)/A(K,2)
      Q=D(M-2)/D(M-1)
      A(M,1)=1.+Q+A(M-2,3)
      A(M,2)=-Q-A(M,1)*A(M-1,3)
      B(M)=B(M-2)-A(M,1)*B(M-1)
      Y2DOT(M)=B(M)/A(M,2)
      MN=M-2
      DO 7 I=1,MN
      K=M-I
7     Y2DOT(K)=B(K)-A(K,3)*Y2DOT(K+1)
      Y2DOT(1)=-A(1,2)*Y2DOT(2)-A(1,3)*Y2DOT(3)
      RETURN
C
8     FORMAT (33H X SHOULD BE INCREASING IN SPLINE.)
9     FORMAT (I10,2E20.6)
10    FORMAT (73H INCREASE THE DIMENSION OF ALL DIMENSIONED VARIABLES IN
     1 SPLINE AND YSPLN.)
11    FORMAT (46H AT LEAST 4 POINTS ARE NEEDED IN THIS ROUTINE.)
      END
```

```
      SUBROUTINE YSPLN (X,Y,NPTS,Y2DOT,UNKX,UNKY)
C
C     SPLINE INTERPOLATION ROUTINE USING Y2DOT VALUES CALCULATED IN
C     SUBROUTINE SPLINE.
C
C     GIVEN NPTS VALUES OF X (INDEPENDENT VARIABLE) AND THEIR CORRESPON-
C     DING Y VALUES (DEPENDENT VARIABLE) AND VALUES OF THE 2ND DERIV-
C     ATIVE (Y2DOT) AT EACH POINT, ROUTINE INTERPOLATES FOR Y (UNKY)
C     AT X=UNKX.
C
      DIMENSION X(400), Y(400), Y2DOT(400)
C
C     INTERPOLATION INTEPVAL SEARCH
C
      NBG=1
      NND=NPTS
      MDL=NPTS/2+1
      IF (NPTS-2) 1,2,2
1     WRITE (6,18)
      STOP
2     IF (UNKX-X(MDL)) 3,6,8
3     IF (MDL-NBG-1) 4,4,7
4     IF (UNKX-X(NBG)) 5,5,6
5     NK=NBG-1
      GO TO 13
6     NK=MDL-1
      GO TO 13
7     NND=MDL
      MDL=NBG+(NND-NBG)/2
      GO TO 2
8     IF (NND-MDL-1) 9,9,12
9     IF (UNKX-X(NND)) 10,11,11
10    NK=MDL
      GO TO 13
11    NK=NND
      GO TO 13
12    NBG=MDL
      MDL=NBG+(NND-MDL)/2
      GO TO 2
13    IF (NK) 15,14,15
14    NK=1
      GO TO 17
15    IF (NK-NPTS) 17,16,17
16    NK=NPTS-1
C
C     INTERPOLATION FOR UNKY
C
17    VRX1=X(NK+1)-UNKX
      VRX2=UNKX-X(NK)
      VRY1=Y(NK)
      VRY2=Y(NK+1)
      DNK=X(NK+1)-X(NK)
      TRM1=Y2DOT(NK)*VRX1**3
      TRM2=Y2DOT(NK+1)*VRX2**3
      TRM3=(TRM1+TRM2)/(6.0*DNK)
      UNKY=TRM3
      TRM1=VRY1/DNK
```

```
      TRM2=Y2DOT(NK)*DNK/6.0
      TRM3=VRX1*(TRM1-TRM2)
      UNKY=UNKY+TRM3
      TRM1=VRY2/DNK
      TRM2=Y2DOT(NK+1)*DNK/6.0
      TRM3=VRX2*(TRM1-TRM2)
      UNKY=UNKY+TRM3
C
C     THE COMPLETE EXPRESSION FOR UNKY IS GIVEN BY
C     UNKY=(Y2DOT(NK)*VRX1**3+Y2DOT(NK+1)*VRX2**3)/(6.0*DNK)+VRX1*(VRY1/
C    1DNK-Y2DOT(NK)*DNK/6.0)+VRX2*(VRY2/DNK-Y2DOT(NK+1)*DNK/6.0)
C
      RETURN
C
18    FORMAT (79H1PROGRAM HALT IN SUBROUTINE YSPLN----SEARCH LIST CONTAI
     1NS LESS THAN TWO POINTS.)
      END
1511
```

REFERENCES

1. Northrop, T. G., The Adiabatic Motion of Charged Particles, Interscience Publishers, 1963.

2. McIlwain, C. E., "Coordinates for Mapping the Distribution of Magnetically Trapped Particles," Journal of Geophysical Research, 66, 1961, pp. 3681-3691.

3. Mead, G. D., "The Motion of Trapped Particles in a Distorted Field," Radiation Trapped in the Earth's Magnetic Field, edited by B. McCormac, D. Reidel Publishing Co., 1966.

4. Stone, E. C., "The Physical Significance and Application of L, Bo, and Ro to Geomagnetically Trapped Particles," Journal of Geophysical Research, 68, 1963, pp. 4157-4166.

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

6. Roederer, J. G., "Southern Hemisphere Anomalies," Space Research VI, to be published, 1966.

7. Northrop, T. G. and E. Teller, "Stability of the Adiabatic Motion of Charged Particles in the Earth's Field," Physical Review, 177, 1960, pp. 215-225.

8. Hamlin, D. A., et al., "Mirror Azimuthal Drift Frequencies for Geomagnetically Trapped Particles," Journal of Geophysical Research, 66, 1961, p. 104.

9. Dudziak, W. F., et al., Graphic Displays of Geomagnetic Geometry, RM 63TMP-2, DASA 1372, General Electric Company, Santa Barbara, California, 1963.

10. Roederer, J. G., et al., Conjugate Intersects to Selected Geophysical Stations, NASA X-642-65-182, 1965.

11. McIlwain, C. E., University of California, distributed communication.

12. Hendricks, S. J. and J. C. Cain, "Magnetic Field Data for Trapped-Particle Evaluations," Journal of Geophysical Research, 71, 1966, pp. 346-347.

13. Pennington, R. H., Introductory Computer Methods and Numerical Analysis, The Macmillan Company, New York, 1965.

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R&D (Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)		
1. ORIGINATING ACTIVITY (Corporate author) Air Force Weapons Laboratory (WLRTH) Kirtland Air Force Base, New Mexico		2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP
3. REPORT TITLE A COMPUTATIONAL MODEL FOR GEOMAGNETICALLY TRAPPED PARTICLE SHELLS AND KINEMATIC PARAMETERS		
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) October 1965 to June 1966		
5. AUTHOR(S) (Last name, first name, initial) Roederer, Juan G.; Herod, James V., Capt, USA		
6. REPORT DATE September 1966	7a. TOTAL NO. OF PAGES 80	7b. NO. OF REFS 13
8a. CONTRACT OR GRANT NO. b. PROJECT NO. 5710 c. Subtask No. 07.047 d.	9a. ORIGINATOR'S REPORT NUMBER(S) AFWL-TR-66-78 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
10. AVAILABILITY/LIMITATION NOTICES Distribution of this document is unlimited.		
11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY AFWL (WLRTH) Kirtland AFB, NM 87117	

13. ABSTRACT

A computer program has been constructed which calculates a model for geomagnetically trapped particle shells and the associated kinematic parameters. Included is the calculation of longitude-dependent drift velocities and bounce paths. This document explains the methods used in constructing the program as well as the input and output of the code.

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
geomagnetic field line						
geomagnetically trapped particle shells						
drift velocity						
bounce period						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2*a*. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8c, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through
________________________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through
________________________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through
________________________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.

DISTRIBUTION

No. cys	
	HEADQUARTERS USAF
	Hq USAF, Wash, DC 20330
1	(AFCOA)
1	(AFRNE-B)
1	(AFNIN)
1	(AFTAC)
1	USAF Dep, The Inspector General (AFIDI), Norton AFB, Calif 92409
1	USAF Directorate of Nuclear Safety (AFINS), Kirtland AFB, NM 87117
	MAJOR AIR COMMANDS
	AFSC, Andrews AFB, Wash, DC 20331
1	(SCTR)
1	(SCTN)
1	SAC (DICC), Offutt AFB, Nebr 68113
1	ADC (ADOOA), Ent AFB, Colo 80912
1	AUL, Maxwell AFB, Ala 36112
1	USAFA (DFLBA), Colo 80840
	AFSC ORGANIZATIONS
1	AFSC Scientific and Technical Liaison Office, Research and Technology Division, AFUPO, Los Angeles, Calif 90045
1	FTD (TDBTL), Wright-Patterson AFB, Ohio 45433
1	AF Materials Laboratory, Wright-Patterson AFB, Ohio 45433
1	AF Avionics Laboratory, Wright-Patterson AFB, Ohio 45433
1	AF Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio 45433
1	AF Aero-Propulsion Laboratory, Wright-Patterson AFB, Ohio 45433
	RTD, Bolling AFB, Wash, DC 20332
1	(RTTW)
1	(RTS)
1	AF Msl Dev Cen (RRRT), Holloman AFB, NM 88330
1	AFMTC (MU-135, Tech Library), Patrick AFB, Fla 32925
3	SSD (SCL-6), Los Angeles AFS, AFUPO, Los Angeles, Calif 90045
1	ESD (ESTI), L.G. Hanscom Fld, Bedford, Mass 01731
	BSD, Norton AFB, Calif 92409
1	(BSOT)
1	(BSY)

DISTRIBUTION (cont'd)

No. cys	
	KIRTLAND AFB ORGANIZATIONS
1	AFSWC (SWEH), Kirtland AFB, NM 87117
1	ADC (ADSWO), Special Weapons Office, Kirtland AFB, NM 87117
1	SAC Res Rep (SACLO), AFSWC, Kirtland AFB, NM 87117
	AFWL, Kirtland AFB, NM 87117
10	(WLIL)
1	(WLAA)
1	(WLRB)
1	(WLRP)
50	(WLRT)
1	(WLF)
	OTHER AIR FORCE AGENCIES
	Director, USAF Project RAND, via: Air Force Liaison Office, The RAND Corporation, 1700 Main Street, Santa Monica, Calif 90406
1	(RAND Physics Div)
1	(RAND Library)
1	School of Aerospace Medicine, Aerospace Med Cen, Chief, Bionucleonics Division, Brooks AFB, Tex 78235
1	OAR (RROS), 1400 Wilson Blvd, Arlington, Va 22209
1	AFOSR (SRGL), 1400 Wilson Blvd, Arlington, Va 22209
	ARMY ACTIVITIES
1	Chief of Research and Development, Department of the Army (CRD/P, Scientific and Technical Information Division), Wash, DC 20310
1	The Surgeon General, Department of the Army (Chief, Research and Development Division), Wash, DC 20012
1	US Army Materiel Command, Harry Diamond Laboratories (ORDTL 06.33, Technical Library), Wash, DC 20438
1	Commanding Officer, US Army Combat Developments Command, Institute of Nuclear Studies, Ft Bliss, Tex 79916
1	ARGMA Liaison Office, Bell Telephone Laboratories, Whippany, NJ 07981
1	Redstone Scientific Information Center, US Army Missile Command, Chief, Document Section, Redstone Arsenal, Ala 35809
1	Commanding Officer, Ballistic Research Laboratories (AMXBR-TB/ Mr. J. Meszaros), Aberdeen Proving Ground, Md 21005
1	Commanding Officer, Picatinny Arsenal (SMUPA-VC1, Samuel Feltman Research Laboratories), Dover, NJ 07801

DISTRIBUTION (cont'd)

No. cys	
1	Commanding General, US Continental Army Command, ATTN: ATUTR-TNG, Ft Monroe, Va 23351
1	US Army Research Office, ATTN: Lt Gregory D. Atmore, Box CM, Duke Station, Durham, NC 27706
1	Director, Army Research Office, 3045 Columbia Pike, Arlington, Va 22204
2	Director, US Army Engineer Research and Development Laboratories, ATTN: STINFO Branch, Ft Bhlvoir, Va 20260
1	Commanding General, White Sands Missile Range (Tech Library), White Sands, NM 88002
	NAVY ACTIVITIES
1	Chief of Naval Research, Department of the Navy, Wash, DC 20390
1	Commanding Officer, Naval Research Laboratory, Wash, DC 20390
1	Superintendent, US Naval Postgraduate School, ATTN: George R. Luckett, Monterey, Calif 93940
1	Commanding Officer and Director, Naval Applied Science Laboratory, Brooklyn, NY 11251
1	Commander, Naval Ordnance Laboratory, ATTN: Dr. Rudlin, White Oak, Silver Spring, Md 20910
1	Office of Naval Research, Wash, DC 20360
1	Director of Naval Warfare Analyses, Institute of Naval Studies, Office of the Chief of Navy Ops, 545 Technology Square, Cambridge, Mass 02139
1	Commanding Officer, NWEF (Code 404), Kirtland AFB, NM 87117
	OTHER DOD ACTIVITIES
2	Director, DASA (Document Library Branch), Wash, DC 20301
5	Commander, Field Command, DASA (FCAG3, Special Weapons Publication Distribution), Sandia Base, NM 87115
1	Director, ARPA, Department of Defense, The Pentagon, Wash, DC 20301
1	Office of Director of Defense Research and Engineering, ATTN: John E. Jackson, Office of Atomic Programs, Rm 3E1071, The Pentagon, Wash, DC 20330
20	DDC (TIAAS), Cameron Station, Alexandria, Va 22314
	AEC ACTIVITIES
1	US Atomic Energy Commission (Hq Library, Reports Section), Mail Station G-017, Wash, DC 20545
1	Sandia Corporation (Information Distribution Division), Box 5800, Sandia Base, NM 87115
1	Sandia Corporation (Tech Library), P.O. Box 969, Livermore, Calif 94551

DISTRIBUTION (cont'd)

No. cys	
1	Chief, Division of Technical Information Extension, US Atomic Energy Commission, Box 62, Oak Ridge, Tenn 37831
1	University of California Lawrence Radiation Laboratory, ATTN: Director's Office, P.O. Box 808, Livermore, Calif 94551
1	University of California Lawrence Radiation Laboratory (Tech Info Div, ATTN: Dr. R.K. Wakerling), Berkeley, Calif 94720
1	Director, Los Alamos Scientific Laboratory (Helen Redman, Report Library), P.O. Box 1663, Los Alamos, NM 87554
1	Oak Ridge National Laboratory (Tech Library), Oak Ridge, Tenn 37831
	OTHER
1	Clearinghouse, ATTN: 410.11, 5285 Port Royal Road, Springfield, Va 22151
1	Manned Spacecraft Center (NASA), ATTN: Chief, Tech Info Div, Houston, Tex 77001
1	Massachusetts Institute of Technology, Lincoln Laboratory (Document Library), P.O. Box 73, Lexington, Mass 02173
1	Auburn University, Auburn Research Foundation, Auburn, Ala 36830
1	General Electric Company, Spacecraft Dept, ATTN: Anton A. Sinisgalli, Development Engineer, King of Prussia Park, P.O. Box 8661, CC&F 7, Rm 7246, Philadelphia, Pa 19101
1	General Electric TEMPO, ATTN: DASA Data Center, 735 State Street, Santa Barbara, Calif 93101
20	Mr. Juan Roederer, Centro Nacional De Radiacion Cosmica, Rivadavia 1917, Buenos Aires, Argentina
1	Official Record Copy (WLRTH/Capt Rekenthaler)