ADA057206

(11)

LEVEL II

RADC-TR-77-183
IN-HOUSE REPORT
MAY 1977

(12) 114 p.

Coordinate Conversion Technique for OTH Backscatter Radar

TERENCE J. ELKINS
JOSEPH GIBBS

AD No. ______
DDC FILE COPY

(16) 2305 (17) J2

Approved for public release; distribution unlimited.

ROME AIR DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
GRIFFISS AIR FORCE BASE, NEW YORK 13441

309 050

78 07 24 044

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

APPROVED:

TERENCE J. ELKINS
Chief, Ionospheric Radio
Physics Branch
Electromagnetic Sciences Division

APPROVED:

ALLAN C. SCHELL
Acting Chief
Electromagnetic Sciences Division

FOR THE COMMANDER:

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-77-183	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COORDINATE CONVERSION TECHNIQUE FOR OTH BACKSCATTER RADAR		5. TYPE OF REPORT & PERIOD COVERED In-House
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Terence J. Elkins Joseph Gibbs*		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Deputy for Electronic Technology (RADC/ETEI) Hanscom AFB Massachusetts 01731		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 2305J201
11. CONTROLLING OFFICE NAME AND ADDRESS Deputy for Electronic Technology (RADC/ETEI) Hanscom AFB, Massachusetts 01731		12. REPORT DATE May 1977
		13. NUMBER OF PAGES 111
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distrib ion unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

*ARCON Corporation
18 Lakeside Office Park
Wakefield, MA 01880

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

OTH radar
Ionospheric propagation
Ionospheric modelling

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A radio propagation simulation computer program, known as WIMP (World-Wide Ionospheric Modelling Program), has been studied and documented. The purpose of the program is to permit the transformation from Over-The-Horizon (OTH) Backscatter radar coordinates to geographic coordinates, given a vertical incidence and a backscatter ionogram measured at the radar site. The program consists of a three-dimensional ionospheric modelling algorithm, which contains a provision for updating on the basis of the ionograms, followed by a ray-tracing procedure. These algorithms have

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (Cont)

been deduced from a program source deck and the program structure has been documented in terms of mathematical algorithms and flow charts, together with program listings.

The algorithms have been compared with others in use at RADC and quantitative results of this comparison are presented. These consist of ionospheric model comparisons for representative ranges of the model parameters. Finally, a measured backscatter ionogram is simulated, using the RADC program, and the ionospheric structure along the sounder boresight is deduced.

Contents

Illustrations

Tables

Coordinate Conversion Technique for OTH Backscatter Radar

1. INTRODUCTION

The WIMP Program is a complex algorithm which purports to permit accurate conversion from radar target range coordinates to geographic target coordinates in OTH Backscatter HF radar applications; the complexity is fundamentally due to a deeply developed iterative and reiterative algorithm that ultimately involves the user. The program consists of three main components:

(a) A set of subprograms which construct a three-dimensional three-layer ionospheric model.

(b) A set of subprograms which simulate HF radio wave propagation in the ionosphere constructed in (a).

(c) One of three driving programs controlling the calling sequence to the radio wave propagation simulator in (b); these driving programs perform the following tasks: (1) From a given ionosphere, generate the leading edge of a simulated oblique ionogram including scale factors to the F_2-layer parameters to force agreement with a given vertical ionogram; (2) from a comparison of the simulation in (1) to a given oblique ionogram generate range gradient factors to apply to f_oF_2 and $M(3000)F_2$ to force agreement; (3) from the final ionosphere generated in (1) and (2), simulate predicted radio frequency propagation paths, from which range vs group path functions may be tabulated.

(Received for publication 24 May 1977)

We first describe in some detail the WIMP three-dimensional three-layer ionospheric model, followed by a general description of the radio wave propagation simulation subprogram (ray-tracing algorithm). A functional description is given of the three driving programs; the operating instructions for their use are included next.

Program block diagrams, flow charts, and program listings are included in Appendixes A, B, and C.

Comparisons have been made of the results of the WIMP ray tracings with techniques developed at RADC. The description and results of these studies are reported in Sections 19 through 21.

2. THE WIMP 3-D IONOSPHERIC MODEL

A complete three-dimensional ionospheric model generally consists of three fundamental aspects: (1) How many distinct layers are to be included, (2) How shall the layer parameters be modelled, (3) How shall the electron density profile be erected from the modelled parameters? A collateral question, equally important is, what ray-tracing technique is to be used, that is, which aspect dominates the simulation of HF propagation? To put the following discussion in proper perspective the latter question will be considered here in general, reserving a more detailed treatment to a latter section.

The WIMP ray-tracing algorithm, SUBROUTINE TRISL, is a control point technique based on layer parameters as in the ITS-78 program[1] and the DEVAN program.[2,3] The treatment of gradients of electron density, however, differs: ITS-78 does not consider gradients explicitly, DEVAN constructs an equivalent tilted reflecting surface from variations in virtual height, while WIMP constructs the tilt of the reflecting layer from the gradients of electron density. The predictions of the WIMP model are primarily sensitive to the layer parameter predictions, and only weakly sensitive to the electron density profile model. This latter dependence only enters the calculation at the reflection point (which generally is within the F_2 layer), and involves the horizontal gradients of the electron density which are determined primarily by the gradients of the F_2-layer parameters.

The WIMP model ray tracing considers three distinct layers, E, F_1, and F_2; an optional D-layer tail may be appended to the E layer. The E-layer critical

1. Barghausen, A.F., et al (1969) Predicting Long-Term Operational Parameters of High-Frequency Sky-Wave Telecommunication Systems, ESSA Technical Report ERL 110-ITS-78, Institute for Telecommunication Sciences, Boulder, CO.

2. Beckwith, R.I., Bailey, A.D., and Rao, N.N. (1972) An Investigation of Directional Propagation Effects in High-Frequency Radio Source Location, RRL Publication No. 409.

3. Beckwith, R.I. (1973) A Computer Program for the Rapid Prediction of Angles of Arrival of HF Radio Waves, RRL Publication No. 442.

frequency is determined from the local solar zenith angle. The model employed for the latter is sufficiently unique to describe in some detail below. The F_1 critical frequency is 25 percent larger than f_oE with an additional 0.5 MHz.

The ITS model provides predictions for the F_2 critical frequency and M(3000) factor. These parameters may be modified by ad hoc scale factors to force agreement with a given vertical ionogram, and ad hoc gradient factors to force agreement with a given oblique ionogram. The F_2 semithickness is determined from a linear sunspot model. All parameters thus far mentioned are used to generate the F_2-layer height.

The F_1 layer is fitted between the E and F_2 layers such that the F_1 bottom coincides with the E-layer height and the F_1-layer height coincides with the F2-layer bottom.

A more detailed description of the layer parameters and electron density model follows.

3. MODEL FOR SOLAR ZENITH ANGLE

The model for f_oE requires the solar zenith angle. From the law of cosines of a spherical triangle, this quantity is given by:

$$\cos Z = \sin \delta \sin \theta + \cos \delta \cos \theta \cos (\phi - \phi_s) , \tag{1}$$

where

- Z is the solar zenith angle,
- δ is the solar declination (= latitude of the subsolar point),
- ϕ_s is the east longitude of the subpolar point,
- θ is the geographic latitude of the field point,
- ϕ is the east longitude of the subsolar point.

The field point (θ, ϕ) is given, and the subsolar point (δ, ϕ_s) is required.

The subsolar point may be determined from the solution of the Kepler problem using mean elements from the American Ephemeris and Nautical Almanac. Such a solution, to second order in the eccentricity of the earth's orbit and neglecting lunar perturbations, produces an ephemeris accurate to within a nautical mile.

Alternatively, a simplified model for the solar ephemeris may be constructed, accurate to one degree of arc, as follows:

$$\delta = E \sin \Lambda, \quad \Lambda = 2\pi(d-80)/365.25 , \qquad (2)$$
$$\phi_s = \pi - UT ,$$

where, E is the obliquity of the earth's equator (=23°.45), Λ is the mean longitude of the sun measured in the ecliptic counterclockwise from the first point of Aries, UT is the universal time, and d the day of year.

Clearly, models between these two may be constructed; in particular, the annual variation of the equation of time may be modelled as a correction to the subsolar longitude. (The equation of time is the correction to be added to "sundial time" to reduce it to local mean time.) Such an approach is used in the WIMP program, with the following algorithm implemented in subroutine PTHP each time it is called. The solar declination ("zenith correction angle for day of year") is approximated by:

$$\delta' = E \sin(2\pi(d-82.5)/364) . \tag{3}$$

The equation of time in hours is modelled by:

$$X_j = aY* [A_1/(B_1 + X_1)] - [A_2/(B_2 + X_2)] , \tag{4}$$

where

$a = 0.0235$ hours;
$Y = +1.$, $X_2 = d + 21.$, $0.0 < d < 162.5$;
$Y = -1.$, $X_2 = 344.5 - d$, $162.5 \le d < 344.5$;
$Y = +1.$, $X_2 = d - 344.5$, $344.5 \le d < 366.5$;
$X_1 = (X_2/45.)^3$;
$A_1 = 64.$, $A_2 = 240.$, $B_1 = 4.$, $B_2 = 15.$

The following angle is defined, X_j being converted to radians:

$$D = \phi' + X_j - \phi , \tag{5}$$

where

$\phi' = \pi - UT$, $UT \ge \pi$,

$\phi' = \pi + UT$, $UT < \pi$.

The following approximation to the solar zenith angle is constructed:

$$Z_a^2 = \left[(D*\cos(0.6(\phi + \delta'))\right]^2 + (\theta - \delta')^2 , \tag{6}$$

and the cosine of the solar zenith angle is approximated by:

$$\cos Z = \cos (0.932\ Z_a)\ . \tag{7}$$

In the above description, angles are considered in radians, including UT; furthermore, longitudes are conventionally taken to be positive east. In the coding of PTHP, angles are conventionally in degrees, UT in hours, and longitudes are positive west.

4. MODEL FOR THE E-LAYER PARAMETERS

The E-layer critical frequency f_E, layer height h_E, semithickness y_E, and bottom altitude h_b are required. If the simple parabolic E-layer model is selected,

$$\begin{aligned} h_b &= 100 \text{ km}, \\ h_E &= 115 \text{ km}, \\ y_E &= 15 \text{ km}. \end{aligned} \tag{8a}$$

If, on the other hand, the "parabolic E- with D-layer tail" electron density model is selected,

$$\begin{aligned} h_b &= 100 - 4.\,F_1F_2, \\ h_E &= 115 \text{ km}, \\ y_E &= h_E - h_b\,. \end{aligned} \tag{8b}$$

where

$$\begin{aligned} F_1 &= 1 - (1 - R/3000)^2 && R < 6000 \text{ km}; \\ F_1 &= 0 && R \geq 6000 \text{ km}; \\ F_2 &= \cos(A) && A < \pi/2; \\ F_2 &= 0 && A \geq \pi/2; \\ R &= \text{DIST}\,(\theta,\ \phi,\ 76.0°\ \text{N},\ 102.0°\ \text{W}); \\ A &= \phi + \tfrac{1}{2}\,|t - \pi| - \tfrac{1}{2}\pi; \end{aligned}$$

and $(\theta,\ \phi)$ is the geographic north latitude and east longitude of the field point; DIST is the great circle distance (in km) between the two points specified in the argument list.

The critical frequency f_E is modelled from the sunspot number N_s and the cosine of the solar zenith angle cos Z [as modelled by Eq. (7)]. Let

$$C_1 = \frac{1}{4}(1 + \cos Z)^2, \quad S_1 = (1 + 0.004\ N_s)^{\frac{1}{2}},$$

$$f_n = 0.75\ S_1\ C_1, \quad f_x = 3.17\ S_1\ (\cos Z)^a, \qquad (9a)$$

$$f_d = (f_x{}^2 + f_n{}^2)^{\frac{1}{2}}, \quad a = 1/3.775.$$

Then, at night (cos Z negative) f_E is given by f_n; during the day (cos Z positive) f_E is given by f_d, unless f_d exceeds f_2 (f_oF_2 as modelled by the ITS-78 model) in which case f_n is taken. Furthermore, f_E is constrained not to exceed

$$f_{E\ max} = (0.99\ f_2 - 0.5)/1.26. \qquad (9b)$$

5. MODEL FOR THE F_1-LAYER CRITICAL FREQUENCY

The F_1-layer critical frequency f_1 is given by

$$f_1 = 0.5 + 1.26\ f_E, \qquad (10)$$

(10)

and is constrained [Eq. (9b)] not to exceed 0.99 f_2.

6. MODEL FOR f_oF_2 AND M(3000)F_2

These two parameters, f_2 and M_3, are given by the ITS-78 parameter prediction model. This model is sufficiently well known that it will not be described in detail herein. Generally, the model consists in reducing a set of coefficients for each modelled parameter to a given phase of the solar cycle, season of the year, universal time, and geographic position. The solar cycle is represented by the sunspot number, and the parameters are fitted to a first or second order polynomial. The seasonal variation of the various parameters is represented by different sets of coefficients for each month, except for f_oF_2. The latter is represented as a Fourier sum (9 terms) in the day of the year. The diurnal variation is represented by a Fourier expansion (up to 13 terms) and the global variation is represented by a (modified) spherical harmonic expansion. Parameters modelled are f_oF_2, M(3000)F_2, h'_bF_2, f_oE, upper decile f_oE_s, median f_oE_s, and the lower decile f_oE_s. Only the first two parameters are used herein.

7. MODEL FOR THE F_2-LAYER SEMITHICKNESS

The F_2-layer semithickness y_2 is given by

$$y_2 = F_y (56 + 0.489 N_s) . \tag{11}$$

The model allows for a correction scale factor F_y, which is initialized to unity; no provision is made for its redefinition.

8. MODEL FOR THE F_2-LAYER HEIGHT

The model employed for the construction of the F_2-layer height uses all of the parameters thus far described: f_E, f_1, f_2, M_2, y_2, h_b, h_m. The following frequencies are defined:

$$\begin{aligned}
f'_q &= f_2 - \frac{1}{4}(f_2 - f_1), \\
f''_q &= f_2 - 0.01 f_2 y_2, \\
f_q &= \max(f'_q, f''_q), \\
f_3 &= f_2 M_3, \\
f_o &= 4 f_q - 3 f_2, \\
f_a &= f_o + 3.78 f_E + 1.5, \\
f_b &= 1.26 f_E + 0.5, \\
f_c &= f_o + 8.82 f_E + 3.5, \\
f_d &= f_o - 1.26 f_E - 0.5 = f_o = f_1, \\
f_e &= 8.00 f_E, \\
f_f &= f_o + 7.78 f_E + 1.5, \\
f_h &= f_o - 0.22 f_E + 1.5, \\
f_i &= f_2 + f_q, \\
f_j &= f_2 - f_q .
\end{aligned} \tag{12}$$

The following parameters are defined (these are stored for later use in the construction of the $M(3000)F_2$ correction factor:

$$\begin{aligned}
H_3 &= 70 + 3100 (f_q/f_3)^2, \\
F_q &= f_q/f_2,
\end{aligned} \tag{13}$$

$$P_3 = (f_a/f_b)\log(f_c/f_d),$$
$$Z_1 = y_E\,(f_a/f_e)\log(f_f/f_h),$$
$$Z_2 = \frac{1}{2}y_2\,(f_q/f_2)\log(f_i/f_j)\ .$$

The F_2-layer height is given by:

$$h_2 = 8.0\,(H_3 - Z_1 - Z_2 - h_b)/P_3 + y_2 + h_E\ . \tag{14}$$

9. MODEL FOR THE F_1-LAYER HEIGHT AND SEMITHICKNESS

The F_1-layer height h_1 and semithickness y_1 are given by:

$$h_1 = h_2 - y_2,$$
$$y_1 = h_1 - h_E\ . \tag{15}$$

10. MODEL FOR F_2-LAYER CORRECTION FACTORS — VERTICAL IONOGRAM

The following information had been extracted from a vertical incidence ionogram as input to the program from which correction factors to the F_2 layer parameters are to be derived:

t_r — universal time of ionogram,

L_r — geographic north latitude of sounding station,

W_r — geographic west longitude of sounding station,

F_1 — f_oF_1 from ionogram,

F_2 — f_oF_2 from ionogram,

$F_q = F_1 + \frac{1}{4}(F_2 - F_1)$,

$F'_q = F_2 - \frac{1}{4}(F_2 - F_1)$,

H_v — F_2 virtual height at frequency F_q,

H'_v — F_2 virtual height at frequency F'_q,

N_s — sunspot number,

D_y — day of year.

The ionospheric parameters at the given event are constructed from the parameter prediction model; let these parameters be represented by the following notation:

f_E, f_1, f_2 — critical frequencies,

h_E, h_1, h_2 — layer heights,

y_E, y_1, y_2 — semithcikness,

h_b — bottom of ionosphere,

M_3 — $M(3000)F_2$,

f_q — "quarter point frequency", Eq. (14),

Z_1, Z_2, P_3 — Eq. (15) parameters.

Define the following frequencies:

$$f'_q = f_1 + \frac{1}{4}(f_2 - f_1),$$

$$f_a = f_2 + 3.78\ f_E + 1.5,$$

$$f_c = f_2 + 8.82\ f_E + 3.5,$$

$$f_d = f_2 - f_1,$$

$$f_f = f_2 + 7.78\ f_E + 1.5$$

$$f_h = f_2 - 0.22\ f_E + 1.5,$$

$$f_i = f_2 + f'_q,$$

$$f_j = f_2 - f'_q.$$

Define the following quantities:

$$Z'_1 = (y_E/8)*(f_a/f_E)*\ \log\ (f_f/f_h),$$

$$Z'_2 = (y_2/2)*(f'_q/f_2)*\ \log\ (f_1/f_j),$$

$$P'_3 = (f_a/f_1)*\ \log\ (f_c/f_d),$$

$$H'_3 = (P_3/P'_3)*(H_v - Z'_1 - Z'_{2\ 1} - h_b) + Z_1 + Z_2 + h_b,$$

$$M'_3 = (f'_q/f_2)*(3100/H'_3 - 70))^{\overline{2}}.$$

Then the correction factors R_f and R_m purporting to correct predicted f_oF_2 and $M(3000)F_2$ parameters are given by:

$$R_f = F_2/f_2,$$

(16)

$$R_m = M'_3/M_3\ .$$

11. MODEL FOR F_2-LAYER CORRECTION FACTORS – OBLIQUE IONOGRAM

Factors FR and MR are calculated in one run of the OBLFACT program, and are used as input for the next run of OBLFACT to be stored in elements 3 and 5 of/PERTC/. The reference point is also input directly into elements 1 and 2 of /PERTC/; the remaining elements are established by DATA statements. In the following description, the elements of /PERTC/ are referenced by the elements of (C_i; i=1, 16). The field point north latitude and west longitude (N, W) are arguments of PERTC, the algorithm for the calculation of the oblique correction factors. The reference latitude and longitude, C_1 and C_2, are alternatively referenced as N_r and W_r, respectively.

The algorithm commences by computing the distance and azimuth to the field point (N, W) from the reference point (N_r, W_r); let these two quantities be D and Z_o, respectively.

Define the following quantities:

$Z = Z_o - C_{16}$,

$Z_1 = (\pi/180)*(C_{13} - C_{12} - 90)$,

$S = \csc(Z_1)$,

$Y = 3600\ C_9\ C_{11}\ S$,

$W = C_{10}\ S$

$Z_2 = (\pi/180)*((D + Y)/360\ W)$,

$X = \sin(Z_2)$.

The following correction factors are then calculated:

$$R_a = (1 + C_i X) * (1 + C_j D/C_k) * (1 + C_m Z) \qquad (17)$$

where the f_oF_2 and $M(3000)F_2$ corrections are specified by the index set (a; i, j, k, m) being assigned values (f; 7, 3, 4, 14) and (m; 8, 5 6, 15), respectively. These factors are then applied to the appropriate F_2-level parameter.

12. ELEMENTS OF COMMON BLOCK/PERTC/

This common block is linked to the INPUT file via NAMELIST inputs in the main programs MAIN and FTDBLB; OBLFACT links elements of NAMELIST input (FRO and MRO) as arguments of subroutines SKIP and/or PEAK; the latter two routines then define elements 3 and 5 of /PERTC/ equal to FRO and MRO, respectively. OBLFACT also links the first two elements NLREF and WLREF to the input NAMELIST. All elements are assigned default values in a BLOCK DATA program; presumably, these defaults may be assigned different values by replacing the BLOCK DATA program. The elements of /PERTC/ are used by SUBROUTINE PERTC which computes additional corrections to f_oF_2 and $M(3000)F_2$ based on information extracted from an oblique ionogram. The reference point (NLREF, WLREF) is the same reference point used in the construction of the updating correction factors to f_oF_2 and $M(3000)F_2$ effected in SUBROUTINE FACTO.

Table 1. The Elements of /PERTC/ with the Mnenomic Names Used in PERTC and the Default Values Assigned in the BLOCK DATA Program

Element	Mnenomic	Default	Element	Nmenomic	Default
1	NLREF		2	WLREF	
3	FR	0.0	4	FD	1000.0
5	MR	0.0	6	MD	1000.0
7	FZ	0.0	8	MZ	0.0
9	DUT	0.0792	10	DW	148.5
11	X	0.225	12	GAMMA	180.0
13	AZM	38.808	14	DFAC	0.0
15	DMAC	0.0	16	AZREF	0.0

13. ELECTRON DENSITY MODEL

A continuous electron density profile is constructed from the predicted ionospheric parameters. Slope discontinuities may occur in the model at the critical points of the profile. The model allows for either a parabolic model for the lower E-layer (below the critical altitude) or an E-layer with a D-layer tail; these two models will be described first.

In the parabolic E-layer model, the base altitude, layer height, and semi-thickness are modelled to be 100, 115, and 15 km, respectively. Below the E-layer height h_E, the electron density N_e is given by:

$$N_e = 12400\, f_E^2\, (1 - Y^2), \quad h_b < h < h_E;$$
$$N_e = 0, \qquad h < h_b, \tag{18}$$

where

$Y = (h_E - h)/y_E$; h is the altitude; h_E, y_E, and f_E are the E-layer height, semithickness, and critical frequency.

In the E-layer/D-layer tail model the semithickness may increase by up to 4 km, h_b will lower correspondingly, and h_E remains at 115 km. First, an electron density N_{eo} is calculated by Eq. (20) with Y, however, constructed as follows:

$$Y' = 2\,(h_E - h)/y_E\,,$$
$$x = 1.5\,[1.7/(1.7 + Y')]^{\frac{1}{2}},$$
$$z = [1 + 0.1\,(Y' - 1)\,|Y' - 1|\,]\,/\,[1 - 0.1/(Y' + 1)^2]\,,$$
$$Y'' = 2 - \log\,(1 + 1.72\,(Y')^x),$$
$$Y = 1,\ Y''\ \text{negative},$$
$$Y = 1 - \frac{1}{2}\,(Y'')^z,\ Y''\ \text{not negative.}$$

Then

$$N_{eo} = 12400\, f_E^{\,2}\,(1 - Y^2)\,. \tag{19}$$

Calculate the correction for the D-layer tail N_d as follows:

$$C = \cos\,(0.85\,X),\ \text{where X is the solar zenith angle;}$$
$$N_d = 0,\ C\ \text{negative;} \tag{20}$$
$$N_d = (1 + 0.004\,N_s)\,C^{\frac{1}{4}}/(1 + 81\,W^4);$$

where

$W = (65 - h)/y_E$, and N_s is the sunspot number.

The electron density is finally given by:

$$N_e = N_{eo} + N_d\,. \tag{21}$$

Between h_E and H_2, the plasma frequency F_p is taken as the positive solution of a quadratic equation:

$$aF_p^{\,2} + bF_p + c = 0 \tag{22}$$

and the electron density is taken as

$$N_e = 12400 \, (F_p)^2 \, . \qquad (23)$$

The coefficients a, b, and c are calculated as follows:

First, define three reference heights h_a, h_b, h_c:

$$h_a = h_1 - y_1 \, (1 - (f_E/f_1)^2)^{\frac{1}{2}} \, ,$$

$$h_b = h_2 - y_2 \, (1 - (f_1/f_2)^2)^{\frac{1}{2}} \, ,$$

$$h_c = \frac{1}{2} \, (h_1 + h_b) \, .$$

For h between h_E and h_c, define coefficients B_1, B_2, B_3, B_4 as follows:

$$B_1 = \frac{1}{2} \, (h_a - h_E + h_b - h_1)/(f_1 - f_E),$$

$$B_2 = (h_i) - \frac{1}{2} \, (f_1 (h_a - h_E) + f_E (h_b - h_1)/(f_1 - f_E),$$

$$B_3 = y_1/f_1,$$

$$B_4 = y_1 \, .$$

For h between h_c and h_2, the B_i's are given by:

$$B_1 = \frac{1}{2} (h_b - h_1)/(f_2 - f_1), \quad B_2 = h_2 - f_2 B_1;$$

$$B_3 = y_2/f_2 \quad , \quad B_4 = y_2 \, .$$

The coefficients a, b, c of the quadratic Eq. (22) are defined as:

$$a = B_1^{\,2} + B_3^{\,2},$$

$$b = 2 \, B_1 \, (B_2 - h),$$

$$c = (B_2 - h)^2 - B_4^{\,2} \, .$$

The topside model (h greater than h_2) is a modified Chapman profile. Let $Y = 2 \, (h - h_2)/y_2$; then,

$$N_e = 12400 \left\{ (f_2)^2 \exp \left[1 - Y - \exp(-Y) \right] + (0.5 + 0.1 \, f_2) \, h_2 \, (h - h_2)/h^2 \right\} . \qquad (24)$$

14. MODEL FOR THE GRADIENT OF THE ELECTRON DENSITY

Any ray-tracing technique will require the gradient of the electron density at a field point (r, θ, ϕ). Clearly, the complications of the model described above for the electron density mitigate against any attempt to derive analytic expressions for the gradient, even neglecting variations in the ionospheric parameters. Furthermore, should one desire to replace the electron density model with some other, either more or less complicated, a general numerical algorithm for the gradient is desired.

The gradient model employed in the WIMP program places the field point at the center of a cubical volume, and computes the electron density at each of the nine points so defined. The coordinates of these nine points are listed in the following table.

Table 2. Coordinates of Points Used for Gradient Calculation

Point	Radial Coordinate	Theta	Phi
1	r	θ	ϕ
2	$r + D_r$	$\theta + D_\theta$	$\phi + D_\phi$
3	$r + D_r$	$\theta - D_\theta$	$\phi + D_\phi$
4	$r + D_r$	$\theta + D_\theta$	$\phi - D_\phi$
5	$r + D_r$	$\theta - D_\theta$	$\phi - D_\phi$
6	$r - D_r$	$\theta + D_\theta$	$\phi + D_\phi$
7	$r - D_r$	$\theta - D_\theta$	$\phi + D_\phi$
8	$r - D_r$	$\theta + D_\theta$	$\phi - D_\phi$
9	$r - D_r$	$\theta - D_\theta$	$\phi - D_\phi$

(r, θ, ϕ) are the coordinates of the field point,

(D_r, D_θ, D_ϕ) are the grid spacings,

$D_\theta = D_r/r$,

$D_\phi = D_r/r \sin \theta$.

Given a function $f(x)$ evaluated at three points, x_o, $x_+ = x_o + d_x$, and $x_- = x_o - dx$, then the best estimate of the derivative of f at x_o is given numerically by:

$$f'_o = \frac{1}{2}(f_+ - f_-)/dx^2 \tag{25}$$

where f_-, f_o, and f_+ represent $f(x_-)$, $f(x_o)$, and $f(x_+)$. The best estimate of the second derivative is given by:

$$f''_o = \frac{1}{2}(f_+ - 2f_o + f_-)/dx^2 . \tag{26}$$

Unfortunately, the defined set of points does not contain quite the correct subset to implement the numerical model for the derivative specified by Eq. (25). However, appropriate partial derivatives may be evaluated at the center of each side of the defined cube. Each coordinate then has four approximations to the appropriate partial derivative of the electron density; for instance, the r partial derivative will use pairs (2, 6), (3, 7), (4, 8), and (5, 9).

The algorithm adopted to represent the partial derivatives is the average of the four approximate values.

The value chosen for the grid space is 1 / 3-1/2 km.

15. THE WIMP 3-D RAY-TRACING TECHNIQUE

The ray-tracing model employed in the WIMP program depends upon the application of Bouger's rule and parabolic electron density models in multiple layers; the gradient of the electron density at the reflection height is employed to determine the tilt of the reflecting layer.

The program allows for refraction and/or reflecting in the E, F_1, and F_2 layers. The F_1 layer is inserted between the E and F_2 such that the bottom of the F_1 layer is at the E-layer height and the F_1-layer height is at the bottom of the F_2 layer.

The ray tracing is accomplished in SUBROUTINE TRISL.

15.1 Bouger's Rule

Bouger's rule states that in a refracting medium with spherical symmetry electromagnetic radiation propagates in such a way that the quantity $\mu r \cos\beta$ remains constant where μ is the local index of refraction, r is the distance to the field point from the center of symmetry, and β is the local elevation angle.

15.2 Refraction Through a Layer

Refraction through a layer is depicted in Figure 1. The group path correction $\Delta P'$ phase path correction ΔP, and range correction Δ are given by

$$\begin{aligned}
\Delta P' &= \left(\frac{y}{\sin\beta_c}\right)\left\{\frac{1}{2} u \log\left(\frac{u+1}{u-1}\right) - 1\right\}, \\
\Delta P &= \left\{1 - \frac{1}{2}\sin^2\beta_c\,(1 + 1/u^2)\right\}\Delta P' - \frac{1}{2} y \sin\beta_c/u^2, \\
\Delta &= \Delta P' \cos\beta_c/(R_o + h_m),
\end{aligned} \tag{27}$$

where $u = f \sin \beta_c / f_c$ and f is the operating frequency, f_c the critical frequency, R_o the radius of the earth, h_m the layer height, y the layer semithickness, and β_c the local elevation angle of the unrefracted ray at the layer height.

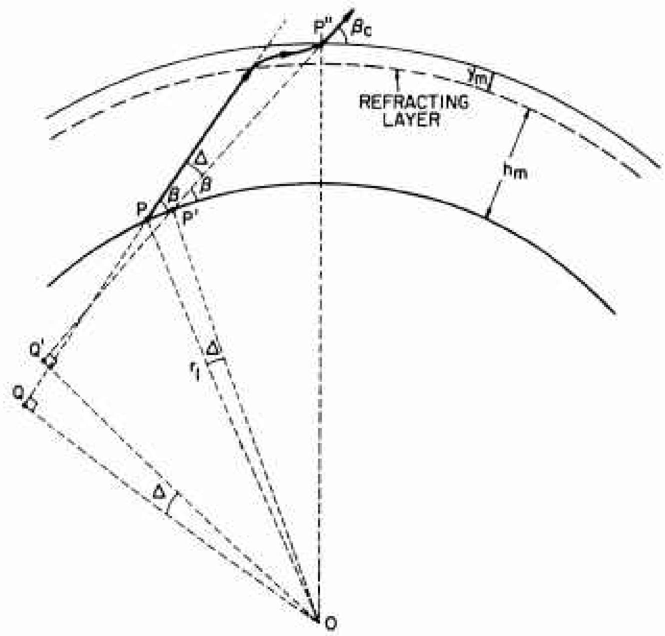


Figure 1. Refraction Through Layer

The group path length P' and phase path length P upon exiting the layer are given by

$$
\begin{aligned}
P' &= D_o + \Delta P', \\
P &= D_o + \Delta P,
\end{aligned}
\qquad (28)
$$

where D_o is the distance P' P'' in Figure 1.

15.3 Reflection from a Layer

Reflection from a layer is depicted in Figure 2. The group path corrections ΔP are given by

$$\Delta P' = \frac{y}{\sin \beta_r} \frac{1}{2} u \log \frac{u+1}{u-1},$$

$$\Delta P = \frac{1}{2} y \sin \beta_r + \left\{ 1 - \frac{1}{2} \sin^2 \beta_r (1 + 1/u^2) \right\} \Delta P', \quad (29)$$

where $u = f \sin \beta_r / f_c$, and β_r is the local elevation angle at the virtual height P''. These corrections apply for the upward trace to the reflection point; equal corrections obtain for the downward trace.

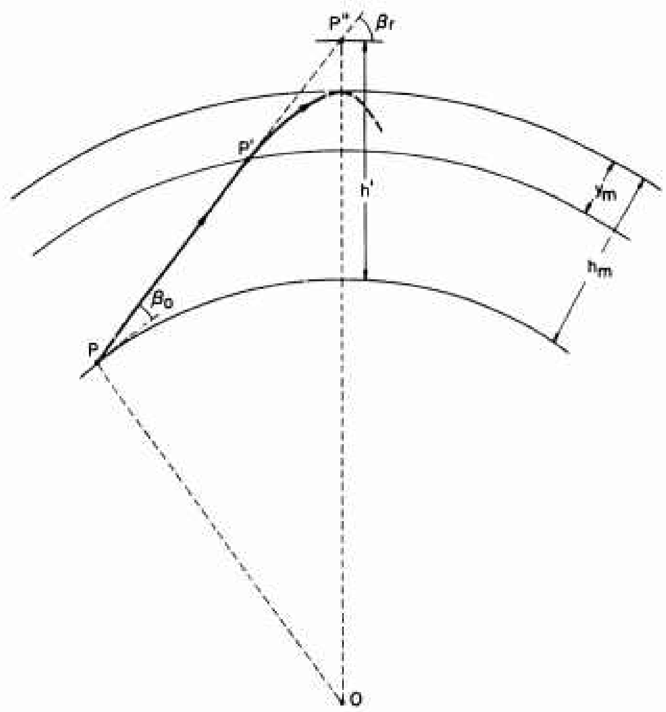


Figure 2. Reflection From Layer

15.4 Tilted Reflection Layer

The gradient of the electron density at the ray reflection point is evaluated and employed to construct an effective tilted reflecting layer. Underlying refracting layers are considered concentric with the earth; however, gradients in these layers are taken into account in that the upward and downward legs of the ray trace through the underlying layers are spatially separated.

The introduction of a tilted reflecting layer, of course, generates considerable complication in the ray trace in that the ray direction, layer normal, and radius vector are no longer coplanar. The program accounts for these bookkeeping complications by setting up orthonormal triads as the ray attacks successive layers.

The y axis is taken parallel to the ray direction. The yz plane contains the center of curvature of the tilted reflecting layer (or the center of the earth in a refracting layer).

15.5 Iteration Scheme

In each hop of the ray trace, the various calculations are iterated to refine the location of the profiles at which refraction through a layer and reflection from a layer occur. In the reflecting layer, the layer tilt is included within the iteration scheme.

At each stage of the ray-trace tests are made for pathological cases such as convergence failures in the allotted number of iterations, inconsistencies between stages, extremely tilted reflecting layers, ducting of the ray between layers, and so forth. Error diagnostic messages are printed prior to returning the main program should TRISL be required to abort the task.

16. FUNCTIONAL DESCRIPTION OF PROGRAM OBLFACT

16.1 Initialization (Block I)

This part of the program clears the INPUT file (NAMELIST INPUT). The elements of the NAMELIST are listed in Table 3 including the mnemonic name used, definition, and default value. A call to RIIP clears the INPUT file of the ITS parameter prediction coefficients. A call to FACTO calculates the F_2 layer correction factors from f_oF_2 and $h'_{min}F_2$ scaled from the vertical ionogram and appearing in the NAMELIST input. Successive calls to PEAK and SKIP with the remote correction factors FRϕ and MRϕ, calculated from a previous run of OBLFACT, check that the ray tracing proceeds as expected (ground-to-ground mode). If not, the mission is aborted and the next task is initiated.

Table 3. Elements of NAMELIST/INPUT/

Mnemonic	Definition	Unit	Notes
SN	Sunspot Number		UNDEF
YRMD	Year & Month, concatenated		UNDEF
DOY	Day of Year	day	UNDEF
ZT	Universal Time, Vertical Ionogram (VI)	hours	UNDEF
RI	Geocentric Distance, VI	km	6370.
NL1	North Latitude, VI	degrees	
WL1	West Longitude, VI	degrees	

Table 3. Elements of NAMELIST/INPUT/ (Cont)

Mnemonic	Definition	Unit	Notes
ZIREF	Universal Time, Oblique Ionogram (OI)	hours	UNDEF
RS	Geocentric Distance to ray end point	km	6370.
NLREF	North Latitude, OI Sounder	degrees	
WLREF	West Longitude, OI Sounder	degrees	
AZ	Bore sight azimuth	degrees	
ELSTEP	Elevation step for SKIP search	degrees	1.0
FCF2R	f_oF_2 from VI	MHz	UNDEF
HMINR	$h'_{min}F_2$ from VI	km	UNDEF
FP	Peak Frequency	MHz	*
DELP	Peak Group Path Error	km	*
FS	Skip Frequency	MHz	*
DELS	Skip Group Path Error	km	*
DCHK	Tolerance in Group Path Deviations		10.0
MRϕ	Gradient Correction Factor, M(3000)F2		≠
FRϕ	Gradient Correction Factor, f_oF_2	MHz	≠
MODE	$\lvert MODE \rvert$ = number of hops		?
IRSTRT	Termination Flag (STOP if ϕ)		1

Explanation of Terms Used in Table 3

NOTES - A numerical value of this column is the default value set in OBLFACT.

UNDEF - Implies that OBLFACT does not establish a default value.

* - These values are determined from a comparison between the observed oblique ionogram (OOI) and the predicted oblique ionogram (POI) generated from a previous run of the BLOBZ driving program. Refer to "Step 5" of the Operating Instructions. Default values of ϕ are assigned to F and DELP.

≠ - For the first run of OBLFACT, these parameters are set to ϕ. Corrected values will be part of the output of OBLFACT to be used as inputs of a new run of OBLFACT at the user level of iteration. Refer to Steps 4 to 7 of the Operating Instructions.

? - This parameter is an element of the argument list of the ray tracing program TRISL. Neither the operating instruction or the program flow chart illuminate the meaning of its sign. OBLFACT assigns a default value of -1.

16.2 Refinement of MR (Block II)

This part of the program iterates calls to PEAK in order to refine the M(3000) gradient correction factor MR. Since the peak frequency depends on antenna pattern and, thus upon elevation, it therefore depends primarily on the height of the layer. The Block II iteration maintains the f_oF_2 gradient correction factor fixed. The value of MR is adjusted initially by 0.001 and subsequently by a Newton iterative technique until the predicted peak group path agrees with that from the oblique ionogram to within the specified tolerance (DCHK, nominally 10 km), at which point transfer is made to Block III.

Block II is ultimately reentered from Block IV after refinement of the f_oF_2 gradient correction factor. A call to PEAK establishes whether further refinement is required. If the resultant group path is within tolerance, the task is terminated and the results printed.

Should the required information to effect this refinement be not available, the default values of ϕ assigned to FP and DELP automatically cause this part of the program to be bypassed. Thus, only the f_oF_2 gradient correction is determined; the M(3000)F_2 gradient remains by default the same as that of the parameter model (ITS-78), which, presumably, is the best available in the absence of evidence to the contrary.

16.3 Refinement of FR (Block III and IV)

This part of the program iterates calls to SKIP in order to refine the f_oF_2 gradient correction factor FR. In SKIP, rays are traced at frequency FS from 0° to 20° in steps given by ELSTEP (nominally 1° and must not be less than 1/2°). The array of group paths is searched to locate the minimum. This is compared with the minimum group path extracted from the leading edge of the oblique ionogram at frequency FS. If the difference is within tolerance, the MR iteration is re-entered. If not, FR is corrected for the next iteration.

Since OBLFACT is in a late user stage of simulation and a previous stage BLOB 2 produces the general ray-trace results, the user should have limiting information on the elevations between which the minimum group path should be located. The built-in 0° to 20° may not be sufficient to properly locate the minimum group path. The using facility might consider modifying the program to allow the user to specify the elevation domain to be searched. This may be accomplished as follows:

(a) Link the initial elevation, ELBEG and number of elevations to be traced, NELM to NAMELIST/INPUT/ in OBLFACT.

(b) Add ELBEG and NELM to COMMON/FIDDLE/ in OBLFACT, SKIP, and PEAK.

(c) Modify the elevation DO loop in SKIP as follows:

PRESENT VERSION	RECOMMENDED VERSION
NELM=20, DØ/ELSTEP + 1.DØ	(delete)
.	
.	EL=ELBEG
DO 10 IEL=1, NELM	DO 10 IEL=1, NELM
EL=(IEL-1)* ELSTEP	.
.	.
.	.
.	.
.	.
.	.
10 CONTINUE	10 EL=EL + ELSTEP

17. SUBROUTINE PEAK (FRR, MRR, GPP, IGOBAK)

17.1 Argument List

FRR - f_oF_2 correction quantity (INPUT)

MRR - $M(3000)F_2$ correction quantity (INPUT)

GPP - Predicted peak group path (OUTPUT)

IGOBAK - Error condition returned (OUTPUT)

17.2 Common Blocks

/PERT C/ linked to PERT F, the routine that calculates F_2-layer correction factors from parameters scaled from the oblique ionogram.

/REMWEN/ linked from TRISL. Contains reflection layer information.

/FIDDLE/ linked from OBLFACT. Contains the "peak frequency" FP scaled from the oblique ionogram, transmitter location, and take-off azimuth.

17.3 Function

Computes the "predicted peak group path" corresponding to oblique ionogram correction factors FRR and MRR, and "peak frequency" FP.

17.4 Algorithm

The input parameters FRR and MRR are stored in elements 3 and 5 of /PERT C/ for use by PERT F for calculating correction factors to f_oF_2 and $M(3000)F_2$.

An elevation is selected from the following critical table according to the peak frequency, element 1 of /FIDDLE/:

Table 4. Elevation Determination

FP less than	Elevation	FP less than	Elevation
5.74	Undefined	16.74	5.2
6.74	11.9	17.74	4.8
7.74	10.3	18.74	4.5
8.74	8.9	19.74	4.2
9.74	7.7	20.74	4.0
10.74	6.7	21.74	3.8
11.74	5.9	23.74	3.7
12.74	5.2	25.74	3.6
13.74	4.6	28.74	3.5
14.74	4.1	30.74	3.4
15.74	3.9	≥ 30.74	Undefined

The calling program is required to insure that 5.74 < FP < 30.74.

The ray-tracing program TRISL is called to compute the predicted peak group path GPP.

Elements of /REMWEN/ are checked to insure reflection consistency occurs in the F2 layer. Furthermore, the geocentric distance of the ray endpoint is checked that it is less than 6378.85 km. If these tests are passed, the error condition flag IGOBAK is set to zero; otherwise, an error condition is noted by setting IGOBAK to unity.

18. SUBROUTINE SKIP (FRR, MRR, GPS, IOPT, IGOBAK)

18.1 Argument List

FRR - f_oF_2 correction quantity (INPUT)
MRR - $M(3000)F_2$ correction quantity (INPUT)
GPS - Predicted minimum group path (OUTPUT)
IOPT - Correction control parameter (INPUT)
IGOBAK - Error condition returned (OUTPUT)

18.2 Common Blocks

/PERT C/ linked to PERT F
/REMWEN/ linked from TRISL
/FIDDLE/ linked from calling program (OBLFACT)

18.3 Function

Computes the predicted minimum group path corresponding to the oblique ionogram correction terms FRR and MRR, and "skip frequency" FS (element 2 of /FIDDLE/).

18.4 Algorithm

SKIP transfers FRR and MRR of the argument list to elements 3 and 5 of /PERT C/ respectively, to be used in PERT F to calculate F_2 correction factors.

A sequence of elevations between 0° and 20° is selected for ray tracing via calls to TRISL. The elevation step is contained in element 8 of /FIDDLE/ and a sufficient number of elevations are selected to include the entire domain (0°, 20°+). Provision is made for up to 41 ray traces; thus, the user must insure that the elevation step size is no greater than 0.5°. The elevation scan is terminated at the first occurrence of a penetration through the ionosphere if such occurs before the normal loop termination.

The rays traced in this elevation scan are partitioned into three sets:

(a) Rays not reflecting from the F_2 layer. These are excluded from further consideration in (b) and (c).

(b) Rays with end point at geocentric distance no greater than 6378.85 km.

(c) Rays with end point at geocentric distance greater than 6378.85 km.

If set (b) is empty, the error flag IGOBAK is set to unity and control is returned to the calling program. Otherwise, the element of (b) with the smallest group path is determined. This least group path is established as the predicted minimum group path GPS unless the following conditions obtain:

(a) SKIP had been called with IOPT not zero, and

(b) The ray with next higher elevation belongs to set (c) and has a smaller group path length.

In the latter case, the minimum group path returned as GPS is given by:

$$GPS = G_b + \left\{(G_c - G_b)/(R_c - R_b)\right\} (6378.85 - R_b) ,$$

where the G's and R's refer to group paths and end point geocentric distance, and the subscripts b and c refer to the above selected elements of set (b) and (c).

Should calls be issued to SKIP more than 100 times in any run of OBLFACT, the error flag is set to two and the task immediately aborted.

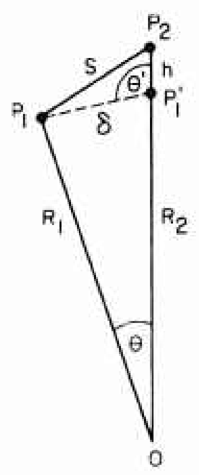
19. REMARKS ON THE USE OF DOUBLE PRECISION ARITHMETIC

The version of the program that the present author has studied has been modified from CDC 6600 machines for use on UNIVAC machines. The essence of this modification is to assign the double precision attribute to every real variable. In the opinion of the present author, wholesale use of double precision arithmetic is not justified, reduction notwithstanding.

In the first place, double precision arithmetic is expensive. Analysis of double precision arithmetic reveals that (1) twice the core storage is required for the variables involved in the calculation, (2) the coding requires, in general, three times the number of instructions, and (3) the execution time increases by approximately a factor of four.

In the second place, increasing the precision of a calculation does not increase the accuracy of the computed result. For example, the geomagnetic field employed in conjunction with the ITS parameter prediction model is a sixth order spherical harmonic expansion. The expansion coefficients are given to one part per million or less. An eight significant digit machine in single precision can calculate the geomagnetic field at a point to the accuracy of the model. The ionospheric parameter model is essentially a one percent model; double precision arithmetic will not improve this situation.

On the other hand, there are situations where single precision arithmetic may produce inaccurate results; for example, in the solution of plane triangles in certain situations.



Consider the situation depicted here. The distances R_1 and R_2 and the angle θ are known and the distance S between P_1 and P_2 is required. The law of cosines applies and

$$S_2 = R_1^2 + R_2^2 - 2R_1 R_2 \cos \theta \ . \qquad (30)$$

Typically, the distances R_1 and R_2 are of the order of 6000 km, the arc $P_1P'_1$ might be 60 km (θ = 0.01 rad), and the altitude difference P'_1P_2 might also be of the order of 60 km. Thus, take R_2 and $\cos\theta$ to be given exactly by

$$R_2 = R_1(1 + 0.01),$$

$$\cos \theta = 1 - \frac{1}{2} \times 10^{-4} \ .$$

With single precision of 8 significant digits,

$$S^2 = R_1^2 \quad 1+(1+2\times 10^{-2} + 10^{-4} \pm 10^{-9}) - (2+2\times 10^{-2} - 10^{-4} - 10^{-6} \pm 10^{-8})$$

$$S^2 + R_1^2 \quad 2\times 10^{-4} + 10^{-6} \pm 10^{-8} \ .$$

In this case, the truncation error has resulted in reduction of precision to one part in 10^4.

This loss of precision may or may not be tolerable; in either case it is not necessary, for the law of cosines may be reformulated to solve the triangle $P_1 P_2 P_1'$ for S, retaining full single precision results. The length of the cord of the angle θ is given by

$$\delta = 2R_1 \sin \frac{1}{2} \theta \ .$$

Furthermore,

$$\cos \theta' = - \sin \frac{1}{2} \theta = -\delta / 2R_1 \ ,$$

$$R_2 = R_1 + h \ .$$

Then,

$$S^2 = h^2 + \delta^2 + h\delta^2 / R_1 \ . \tag{31}$$

This form of the law of cosines will retain single precision results, and should be used wherever possible. To implement this recommendation in SUBROUTINE TRISL, however, would require considerable effort and analysis; a totally new program may be required.

20. MODEL COMPARISON

For purposes of comparison, the WIMP program modelling subroutines were used to generate ionospheric models for various combinations of parameters (Sunspot Number, Season, Universal Time) for a midlatitude location. These were compared with similar models derived from a program in use at RADC.[4] These comparisons are shown in Figures 3 through 13.

4. Rush, C.M., Miller, D., and Gibbs, J. (1974) The relative daily variability of F_oF_2 and h_mF_2 and their implications for HF radio propagation, Radio Science, 9:749-756.

Both models use the same formulation for f_oF_2 the maximum plasma frequency of the F-layer (the ITS-78 model). Figures 3 through 5 show the Universal Time variation of the height of maximum electron density (h_mF_2) for three seasons and two sunspot numbers [W(XX) and R(XX) refer to the WIMP and RADC models for SSN = XX]. Similarly, Figures 6 through 8 show the comparison of parabolic semithickness (y_mF_2); WIMP uses a semithickness value which is time independent. The WIMP model contains a discontinuity at the lower side of the F layer, which results in the most significant departures from the RADC model in the vertical profile. In order to illustrate these departures, the difference between the plasma frequency was calculated by the two models at the altitude of the F_1-layer maximum used in WIMP. These results are shown for four seasons in Figures 9 through 12, where the times of sunrise and sunset are also indicated. Figure 13 shows a comparison of the two models for given time (χ is the solar zenith angle). The relevant ionospheric model parameters are indicated in the figure.

The results of ray tracing obliquely through the 3-D ionosphere of which Figure 13 represents a single vertical profile are shown in Figure 14. The ray-trace program used was the standard RADC program, known as ARCON II, ITS. As expected, the greatest discrepancies between the oblique group path for the WIMP and RADC models are to be found corresponding to the profile discontinuity at the F_1 layer (compare Figures 9 through 12). In order to examine the differences in computed leading edge of a backscatter ionogram, due to the different models, a one-dimensional ionospheric model was generated, having the vertical profile given by the RADC and WIMP models in Figure 13, (that is, having no horizontal gradients). Leading edges were then computed for simulated backscatter ionograms, using the ARCON II ray-trace program, and the results shown in Figure 15.

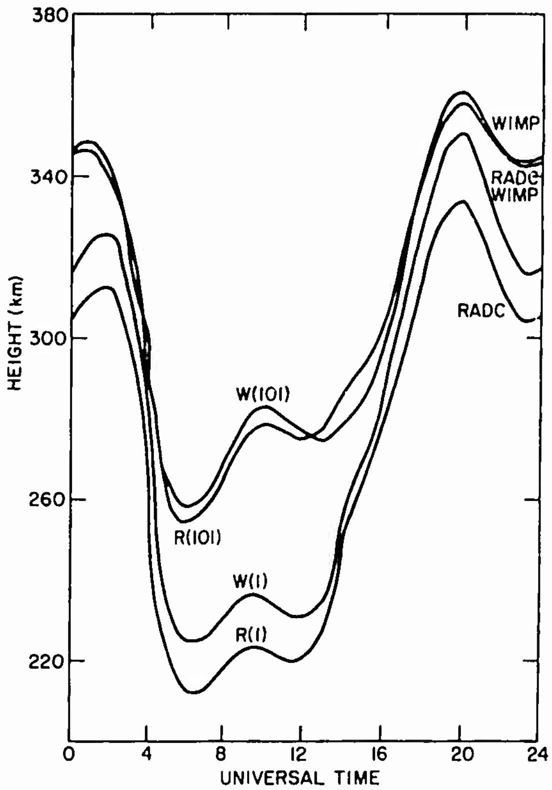


Figure 3. RADC and WIMP Model h_mF_2, Winter Solstice

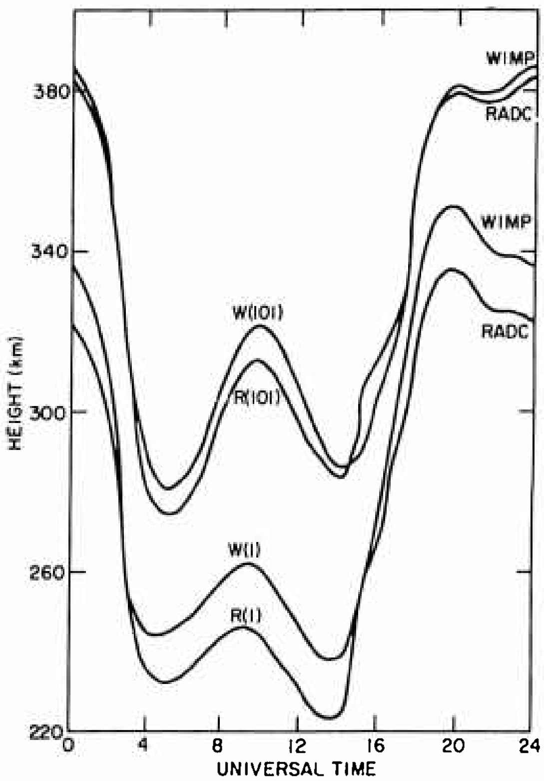


Figure 4. RADC and WIMP Model h_mF_2, Vernal Equinox

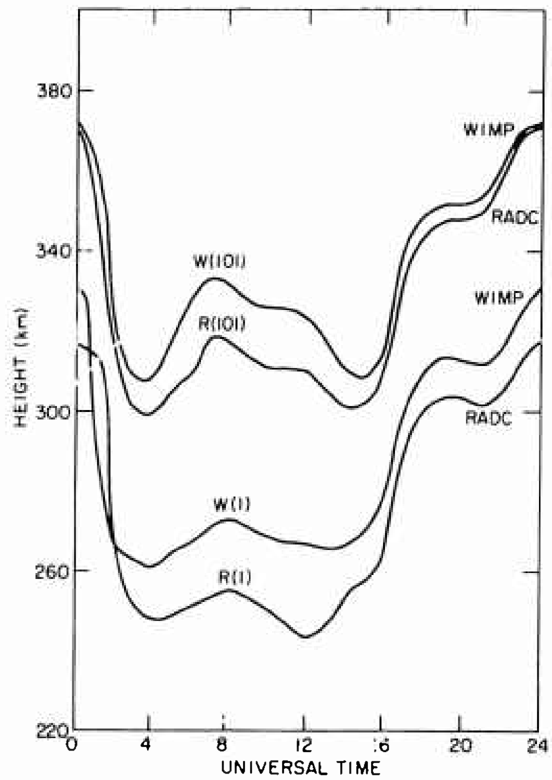


Figure 5. RADC and WIMP Model h_mF_2, Summer Solstice

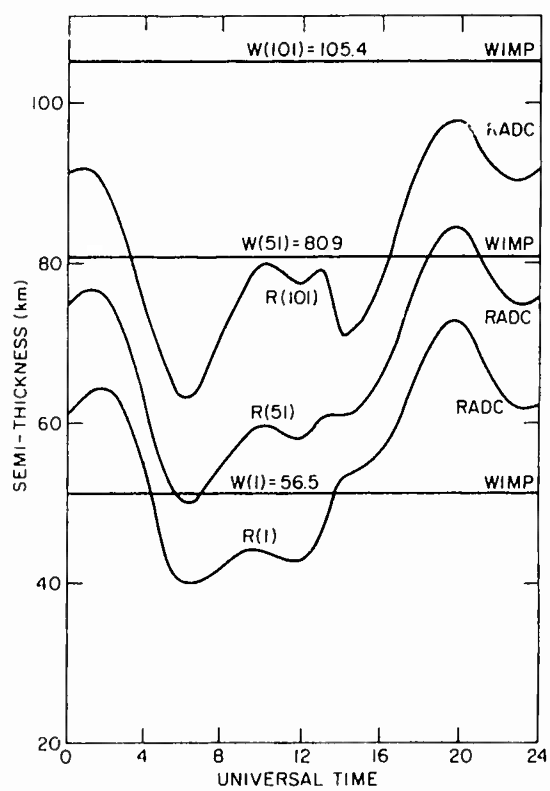


Figure 6. RADC and WIMP Model y_mF_2, Winter Solstice

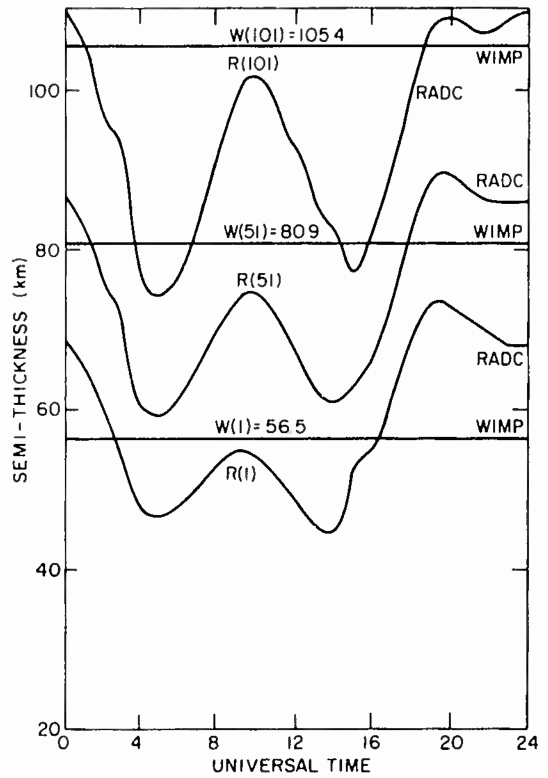


Figure 7. RADC and WIMP Model $y_m F_2$, Vernal Equation

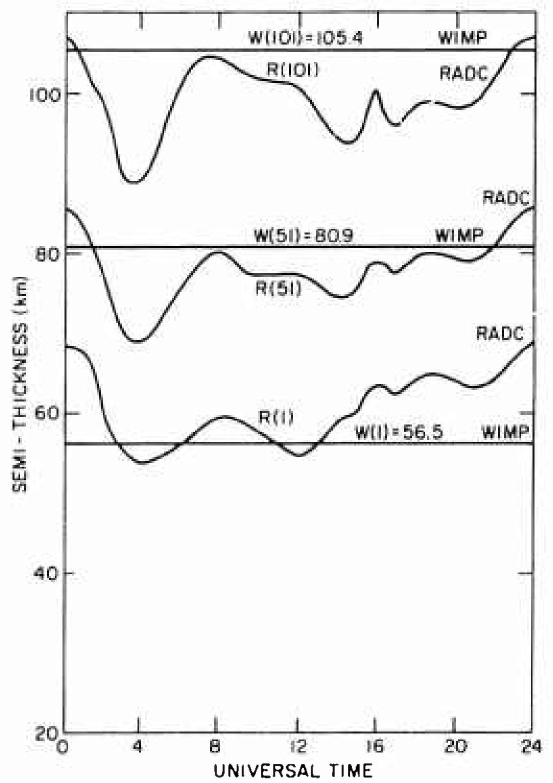


Figure 8. RADC and WIMP Model $y_m F_2$, Summer Solstice

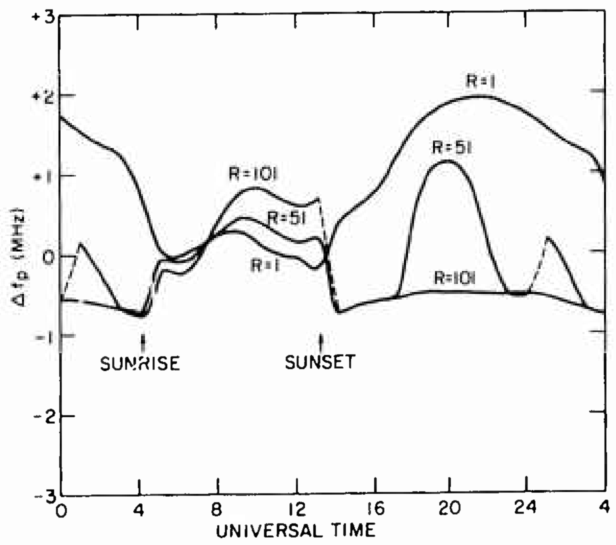


Figure 9. RADC Plasma Frequency at WIMP h_mF_1 Minus WIMP f_oF_1, Winter Solstice

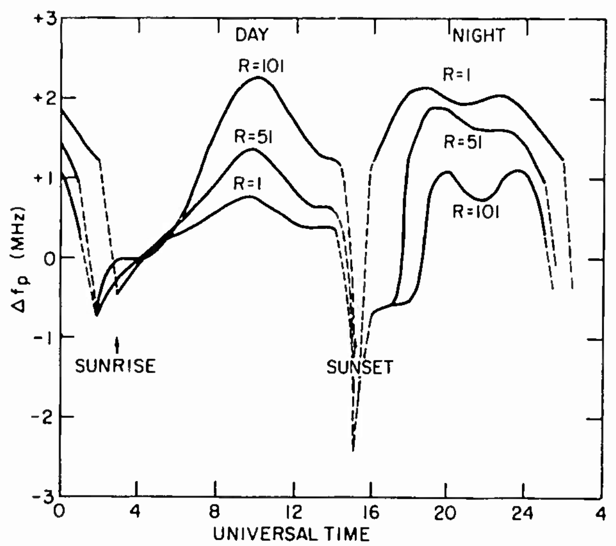


Figure 10. RADC Plasma Frequency at WIMP h_mF_1 Minus WIMP f_oF_1, Vernal Equinox

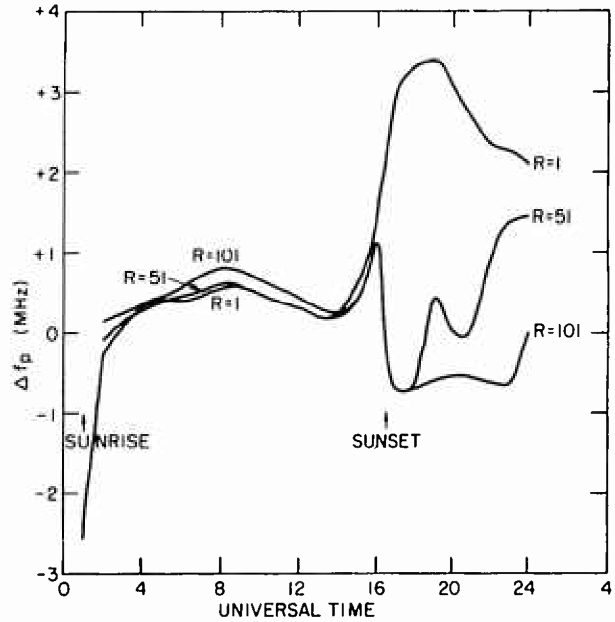


Figure 11. RADC Plasma Frequency at WIMP h_mF_1 Minus WIMP f_oF_1, Summer Solstice

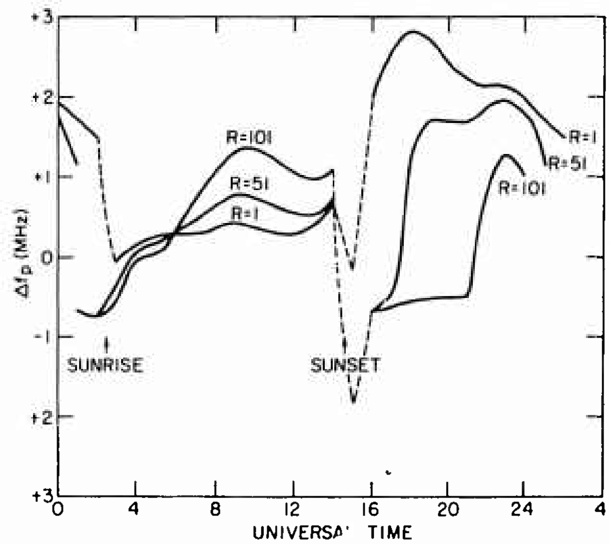


Figure 12. RADC Plasma Frequency at WIMP h_mF_1 Minus WIMP f_oF_1 Autumnal Equinox

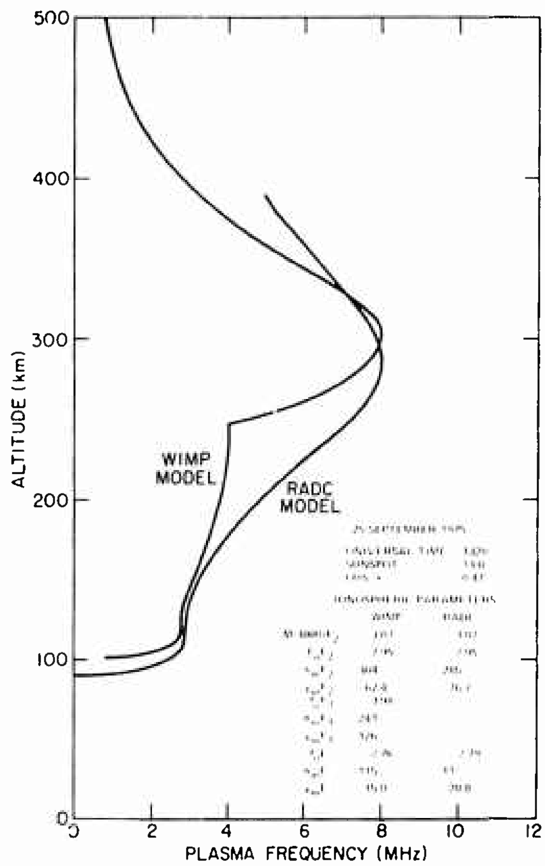


Figure 13. RADC and WIMP Plasma Frequency Profiles

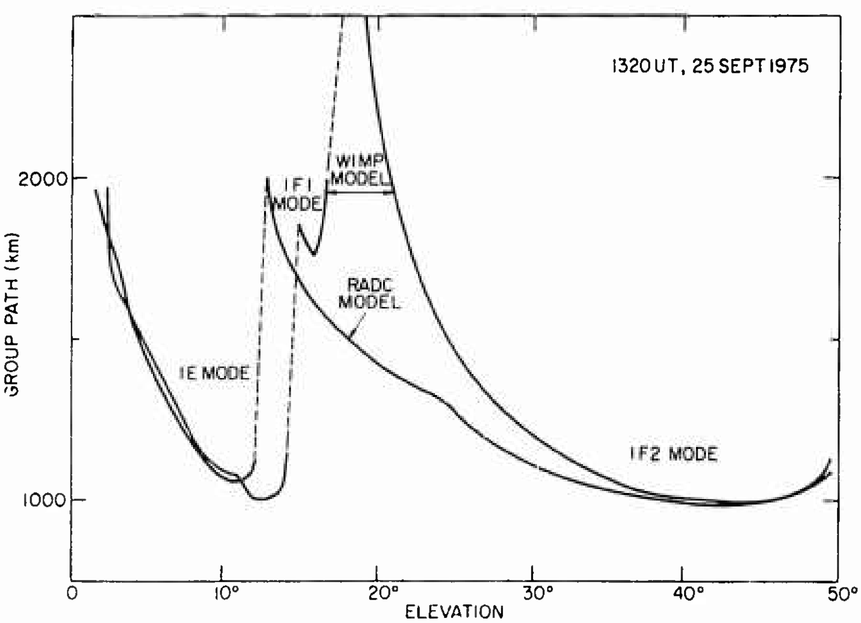


Figure 14. ARCON II 10-MHz Ray Traces, RADC and WIMP Profiles

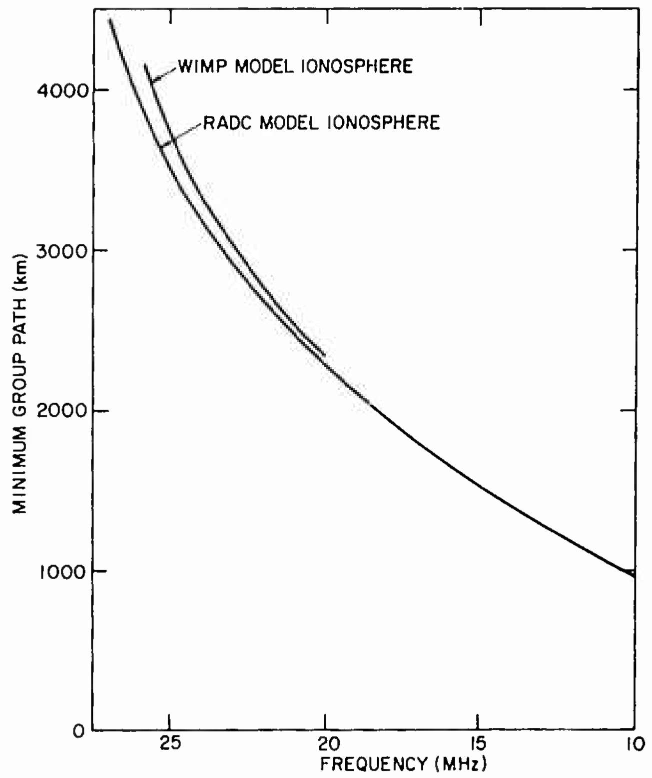


Figure 15. ARCON II Predicted Leading Edges, RADC and WIMP Profiles

21. COORDINATE CONVERSION

The conversion from apparent radar range (that is, group path (P) to true range (R) can be accomplished if the ionospheric structure is known, by using the procedures described earlier. The RADC model for 1320 UT on 25 September 1975 was used, together with the ARCON II ray-trace program, to compute (P-R) as a function of P for frequencies in the range 18 to 26 MHz, as shown in Figure 16. The locus of the backscatter leading edge is indicated by the curve labelled P_{min}, corresponding to the minimum group path at each frequency.

Figure 17 shows the comparison of the actual leading edge of a backscatter ionogram measured at 1320 UT (Curve 2) with the simulated leading edge (Curve 6). Curve 1 shows a leading edge measured at 1340 UT, indicating that the ionosphere is moderately unstable, since Curves 1 and 2 are noticeably different. Curve 3 indicates a secondary leading edge appearing in the ionograms, perhaps a result of azimuthal gradients in the ionosphere. The ionospheric model was then updated by modifying it on the basis of the measured vertical ionogram and allowing for a linear gradient, such that

$$N_e = BN_{eo} (1 + A R_+) ,$$

where N_e is the modelled electron density at any location.

N_{eo} is the modelled electron density (from the RADC program)

$$B = \frac{(f_oF_2) \text{ measured}}{(f_oF_2) \text{ model}} ,$$

A = constant ,

R_+ = boresight range (in radians) .

It should be noted that the range is here measured in terms of the angle subtended at the earth's center by the arc on the earth's surface which is the ground range. Curve 5 in Figure 17 shows the computed leading edge for A = 2.0 and Curve 4 shows the best fit (A = 1.61) with the measured data. By repeating the procedure illustrated in Figure 16, the best estimated conversion from radar range to true range can be determined for any operating frequency. Although azimuthal radar information is not available, it is possible to ray trace at various azimuths to establish a locus on the ground of possible ranges for a target having any given radar range.

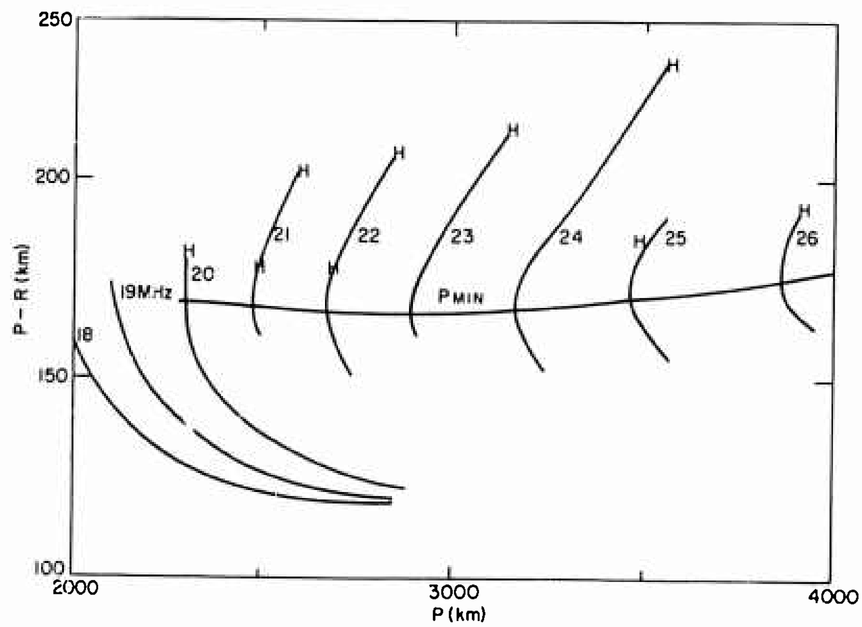


Figure 16. Illustrating Range Conversion

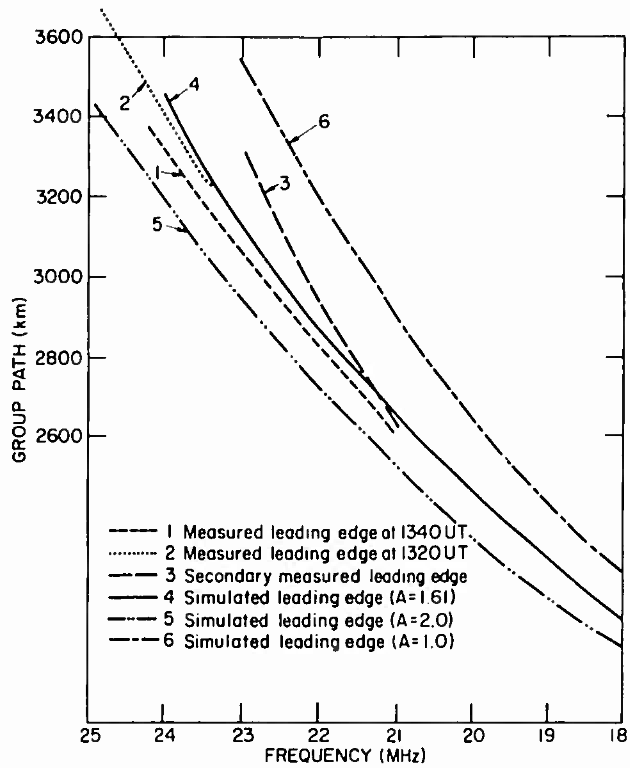


Figure 17. Backscatter Leading Edge Comparison

Appendix A

Program Listings for the WIMP Model Ionosphere

Program listings are included for the WIMP model ionosphere; each subroutine is briefly described as follows:

1. RIIP - Calling control program for layer parameters and electron density.
2. CRPL2 - Generates f_oF_2 and $M(3000)F_2$.
3. MMDIP - Geomagnetic Field Model.
4. PERT F - f_oF_2 and $M(3000)F_2$ gradient factors.
5. DFP3 - Great Circle Range/Azimuth Calculations.
6. DFP - Great Circle Range Calculation.
7. PTHP - E, F_1, Y_mF_2, and h_mF_2 calculations.
8. NFROMR - Electron Density Profile.

```
      SUBROUTINE RIIP(RFEC,BNLATA,BWLONA,PLASD)                         RII
C                                                                       RII
C     PROGRAMMED FOR O.ODOM BY COMPUTATIONS,INC. -OCTOBER,1967          RII
C                                                                       RII
C                                                                       RII
C                                                                       RII
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               RII
      DOUBLE PRECISION M3000,NLATA                                      RII
      DOUBLE PRECISION MFAC                                             RII
      LOGICAL SMOOTH,TILT,NIP,FAST                                      RII
      COMMON/RIIPAR/M3000,FCF2,FCF1,FCE,HBE,HAE,HME,HAF1,HMF1,HAF2,HMF2, RII
     1SN,ZT,YR,D,HAESET,MFAC,F2FAC,F1FAC,EFAC,H2FAC,SNSQRT,QRAD,CQ,IDL,I RII
     2A,SMOOTH                                                          RII
      COMMON/RPERT/RRFEC,ITIP/NIPRIP/NIP/TILTC/TILT,FAST                RII
      COMMON/RIIFWA/IX(4)                                               RI2
      REAL IX                                                           RI2
                                                                        RI2
      DATA NLATA,WLONGA                                                 RI2
      DATA SN,ZT,YR,D/13.D0,13.D0,7509.D0,269.D0/                       RI2
      DATA M3000,FCF2,FCF1,FCE/3.05D0,8.D0,4.08D0,2.79D0/               RI2
      DATA HBE,HME,HMF1,HMF2/90.D0,110.D0,180.D0,288.D0/                RI2
      DATA HAE,HAF1,HAF2/20.D0,70.D0,108.D0/                            RI2
      DATA MFAC,F2FAC,F1FAC,EFAC,H2FAC/5*1.D0/                          RI2
      DATA HAESET/15.D0/                                                RI2
                                                                        RI2
      NLATA=BNLATA*57.2957795130823 2D0                                 RII
      WLONGA=BWLONA *57.29577951308232D0                                RII
      RRFEC=RFEC                                                        RII
      NLATA=DMOD(NLATA,90.D0)                                           RII
      WLONGA=DMOD(WLONGA,360.D0)+180.D0-DSIGN(180.D0,WLONGA)            RII
 1004 CALL  CRPL2(NLATA,WLONGA)                                         RII
      M3000=M3000*MFAC                                                  RI2
      FCF2 = FCF2*F2FAC                                                 RI2
C   PREDICTED BOTTOM OF E LAYER                                         RII
 1005 BA=100.D0                                                         RII
      AA=BA+HAESET                                                      RII
      IF(IDL*IA.EQ.0) GO TO 1006                                        RII
      CALL  DFP(NLATA,WLONGA,76.D0,102.D0,DIST)                         RII
      BA=100.D0-4.D0*DMAX1((1.D0-DABS(DIST-3.D3)/3.D3)**2,0.0D0)*DMAX1( RII
     1 DCOS((DABS(15.D0*DABS(ZT-12.D0)-WLONGA)-90.D0)/(2.D0*            RII
     2 57.29577951308232D0)),0.0D0)                                     RII
C     PREDICTED TRUE HEIGHT PROFILE                                     RII
 1006 CALL  PTHP(NLATA,WLONGA,AA,BA)                                    RII
C   GET ELECTRON DENSITY AT REF. POINT                                  RII
C     GET N FROM R                                                      RII
      IF(RFEC.LE.6370.D0) RETURN                                        RII
      CALL  NFROMR(RFEC,PLASD)                                          RII
      RETURN                                                            RII
      END                                                               RII
```

```
      SUBROUTINE  CRPL2(NLAT,WLONG)                                        CRP
C     ...TO FIND CRITICAL FREQUENCY AS A FUNCTION OF THETA AND PHI         CRP
C        FROM NEW NBS IONISPHERIC PREDICTIONS FOR A PARTICULAR TIME...     CRP
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                  CRP
      DOUBLE PRECISION NLAT,M3000,MFAC                                     CRP
      DIMENSION   SINTO(11),COSTO(11),G(76),GAMMA(2),DKTO(76,2)            CRP
      DIMENSION   CT(8),ST(8)                                              CRP
      COMMON/RIIPAR/M3000,FCF20,DUM(9),SN,ZT,YR,DAY,HAESET,MFAC, FFAC,     CRP
     1DUM1(6),ID(3)                                                        CRP
     */LABCRP/NHARM(2),KI(2),KII(2),KIII(2),KIV(2),KV(2),KVI(2),KVII(2),   CRP
     *KVIII(2),KIX(2),D(13,76,2)                                           CRP
      DATA RAD/.0174532925199433D0/                                        CRP
      G(1)=1.0D0                                                           CRP
      CALL  MMDIP(NLAT,WLONG,SSMX)                                         CRP
      THERAD=(360.D0-WLONG)*RAD                                            CRP
      COSLAM=DCOS(NLAT*RAD)                                                CRP
      CT(1)=COSLAM*DCOS(THERAD)                                            CRP
      ST(1)=COSLAM*DSIN(THERAD)                                            CRP
      DO 63047 IJK=2,8                                                     CRP
      CT(IJK)=CT(1)*CT(IJK-1)-ST(1)*ST(IJK-1)                              CRP
63047 ST(IJK)=CT(1)*ST(IJK-1)+ST(1)*CT(IJK-1)                              CRP
      IF(DABS(ZT-ZTP).LT.1.D-3.AND.DABS(YR-YRP).LT.1.D-1)GO TO 20          CRP
      ZTP=ZT                                                               CRP
      YRP=YR                                                               CRP
      TO=15.0D0*ZT-180.0D0                                                 CRP
      TORAD=TO*RAD                                                         CRP
      ARG=0.0D0                                                            CRP
      NHARML=MAX0(NHARM(1),NHARM(2))                                       CRP
      DO 1  J=1,NHARML                                                     CRP
      ARG=TORAD+ARG                                                        CRP
      SINTO(J)=DSIN(ARG)                                                   CRP
1     COSTO(J)=DCOS(ARG)                                                   CRP
      DO 2 L=1,2                                                           CRP
      NHARML=NHARM(L)                                                      CRP
      K9=KIX(L)+1                                                          CRP
      DO 2 K=1,K9                                                          CRP
      DKTO(K,L)=D(1,K,L)                                                   CRP
      DO 3 J=1,NHARML                                                      CRP
3     DKTO(K,L)=DKTO(K,L)+D(2*J+1,K,L)*COSTO(J)+D(2*J,K,L)*SINTO(J)        CRP
2     CONTINUE                                                             CRP
20    DO 100 L=1,2                                                         CRP
      NHARML=NHARM(L)                                                      CRP
      K1=KI(L)+1                                                           CRP
      K2=KII(L)+1                                                          CRP
      K3=KIII(L)+1                                                         CRP
      K4=KIV(L)+1                                                          CRP
      K5=KV(L)+1                                                           CRP
      K6=KVI(L)+1                                                          CRP
      K7=KVII(L)+1                                                         CRP
      K8=KVIII(L)+1                                                        CRP
      K9=KIX(L)+1                                                          CRP
      K1P1=K1+1                                                            CRP
      K1P2=K1+2                                                            CRP
      K1P3=K1+3                                                            CRP
      K2P1=K2+1                                                            CRP
      K2P2=K2+2                                                            CRP
      K2P3=K2+3                                                            CRP
```

```
      K3P1=K3+1                                                    CRP
      K3P2=K3+2                                                    CRP
      K3P3=K3+3                                                    CRP
      K4P1=K4+1                                                    CRP
      K4P2=K4+2                                                    CRP
      K4P3=K4+3                                                    CRP
      K5P1=K5+1                                                    CRP
      K6P1=K6+1                                                    CRP
      K7P1=K7+1                                                    CRP
      K8P1=K8+1                                                    CRP
      DO 4 K=2,K1                                                  CRP
4     G(K)=G(K-1)*SSMX                                             CRP
      IF(K1.EQ.K2) GO TO 9                                         CRP
      G(K1P1)=CT(1)                                                CRP
      G(K1P2)=ST(1)                                                CRP
      DO 5 K=K1P3,K2,2                                             CRP
      G(K)=G(K-2)*SSMX                                             CRP
5     G(K+1)=G(K-1)*SSMX                                           CRP
      IF(K2.EQ.K3) GO TO 9                                         CRP
      G(K2P1)=CT(2)                                                CRP
      G(K2P2)=ST(2)                                                CRP
      DO 6 K=K2P3,K3,2                                             CRP
      G(K)=G(K-2)*SSMX                                             CRP
6     G(K+1)=G(K-1)*SSMX                                           CRP
      IF(K3.EQ.K4) GO TO 9                                         CRP
      G(K3P1)=CT(3)                                                CRP
      G(K3P2)=ST(3)                                                CRP
      DO 7 K=K3P3,K4,2                                             CRP
      G(K)=G(K-2)*SSMX                                             CRP
7     G(K+1)=G(K-1)*SSMX                                           CRP
      IF(K4.EQ.K5) GO TO 9                                         CRP
      G(K4P1)=CT(4)                                                CRP
      G(K4P2)=ST(4)                                                CRP
      DO 8 K=K4P3,K5,2                                             CRP
      G(K)=G(K-2)*SSMX                                             CRP
8     G(K+1)=G(K-1)*SSMX                                           CRP
      IF(K5.EQ.K6) GO TO 9                                         CRP
      G(K5P1)=CT(5)                                                CRP
      G(K6)=ST(5)                                                  CRP
      IF(K6.EQ.K7) GO TO 9                                         CRP
      G(K6P1)=CT(6)                                                CRP
      G(K7)=ST(6)                                                  CRP
      IF(K7.EQ.K8) GO TO 9                                         CRP
      G(K7P1)=CT(7)                                                CRP
      G(K8)=ST(7)                                                  CRP
      IF(K8.EQ.K9) GO TO 9                                         CRP
      G(K8P1)=CT(8)                                                CRP
      G(K9)=ST(8)                                                  CRP
9     GAMMA(L)=0.0D0                                               CRP
      DO 10 K=1,K9                                                 CRP
10    GAMMA(L)=GAMMA(L)+DKTO(K,L)*G(K)                             CRP
100   CONTINUE                                                     CRP
      FCF20=GAMMA(1)*DABS(FFAC)                                    CRP
      M3000=GAMMA(2)*DABS(MFAC)                                    CRP
      IF(FFAC.LT.0.0D0)CALL PERTF(ZT,NLAT,WLONG,FCF20)             CRP
      IF(MFAC.LT.0.0D0)CALL PERTH(ZT,NLAT,WLONG,M3000)             CRP
      RETURN                                                       CRP
```

```
      SUBROUTINE  PTHP(NLAT,WLONG,A,B)                                  PTH
C     PREDICTED TRUE HEIGHT PROFILE                                     PTH
C     PROGRAMMED FOR D.ODOM BY I.STUHLER                                PTH
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               PTH
      DOUBLE PRECISION NLAT,M3000,M,J,KK                                PTH
      LOGICAL     SMOOTH                                                PTH
      COMMON/FACCOM/FM,ZZ,KK,P3000                                      PTH
      COMMON/RIIPAR/M3000,FCF20,FCF10,FCEO,HBEP,HPE,HMEP,HPF1,HMF1P,HPF2 PTH
     1,HMF2P,SN,ZT,YR,D,DUM(3),F1FAC,EFAC,H2FAC,                        PTH
     2 SNSQRT,QRAD,CQ,ID(2),SMOOTH                                      PTH
      DATA  RAD/57.2957795130823D0/                                     PTH
C   SOLVE FOR ZENITH CORR. ANGLE FOR DAY OF YEAR                        PTH
      M=23.45D0*DSIN(180.D0*(D-82.5D0)/(182.D0*RAD))                    PTH
C   GET F3000 PROPAGATION FREQ FOR F2 LAYER                             PTH
      F3000=FCF20*M3000                                                 PTH
C   CHECK FOR INCORRECT DAY(NEG)                                        PTH
      IF(D.LT.0.0D0) GO TO 2000                                         PTH
C   IS DAY PAST SPRING                                                  PTH
      IF(D.GE.162.5D0) GO TO 1000                                       PTH
      X=D+365.5D0                                                       PTH
      Y=1.0D0                                                           PTH
      GO TO 1001                                                        PTH
C   IS DAY PAST FALL                                                    PTH
1000  IF(D.GE.344.5D0) GO TO 1002                                       PTH
      X=D                                                               PTH
      Y=-1.0D0                                                          PTH
      GO TO 1001                                                        PTH
C   CHECK FOR INCORRECT DAY (TOO LARGE)                                 PTH
1002  IF(D.GE.366.5D0) GO TO 2001                                       PTH
      X=D                                                               PTH
      Y=1.D0                                                            PTH
C   SOLVE FOR SOLAR ZENITH ANGLE IN RAD                                 PTH
1001  J=Y*.0235D0*(64.D0/(4.D0+DABS((X-344.5D0)/45.D0)**3)-240.D0/(15.D0 PTH
     1 +DABS(X-344.5D0)))                                               PTH
      IF(ZT.LT.12.D0)ZT=-ZT                                             PTH
      Z=DABS(15.D0*DABS(ZT-12.D0)-J*15.D0-WLONG)                        PTH
      IF(Z.GE.180.D0) Z=360.D0-Z                                        PTH
      ZSQ=Z*Z                                                           PTH
      QSQ=ZSQ*(DCOS(0.6D0*(NLAT+M)/RAD)**2)+(NLAT-M)**2                 PTH
1005  Q=DSQRT(QSQ)                                                      PTH
      QRAD=Q/RAD                                                        PTH
      CQ=DCOS(.932D0*QRAD)                                              PTH
C    FUNCTION FOR E-LAYER NIGHT TIME FORMATION                          PTH
      CCQ=((CQ+1.D0)*.5D0)**2                                           PTH
      SNSQRT=DSQRT(1.D0+.004D0*SN)                                      PTH
      IF(CQ.LE.0.D0) GO TO 1101                                         PTH
C   SOLVE FOR PRED E-LAYER CRITICAL FREQ                                PTH
C  10.0489=3.17*3.17,    .5625=.75*.75                                  PTH
      FCEO=SNSQRT*DSQRT(10.0489D0*CQ**(2.D0/3.775D0)+.5625D0*CCQ**2)    PTH
      IF(FCEO.GE.FCF20)FCEO=.75D0*SNSQRT*CCQ                            PTH
      GO TO 1102                                                        PTH
1101  FCEO=.75D0*SNSQRT*CCQ                                             PTH
C   SOLVE FOR PRED F1-LAYER CIRTICAL FREQ                               PTH
1102  FCF10=1.26D0*FCEO+0.5D0                                           PTH
      IF(FCF10.LT.FCF20*.99D0) GO TO 9982                               PTH
      FCF10=.99D0*FCF20                                                 PTH
      FCEO=DMAX1((FCF10-.5D0)/1.26D0,1.D-5)                             PTH
```

```
C   SOLVE FOR REF. FREQ. NO. 1                                          PTH
9982  FK1=FCF20-(FCF20-FCF10)/4.D0                                      PTH
C   CALCULATE PRED. SEMITHICKNESS OF F2 LAYER                           PTH
      HPF2=(0.489D0*SH+56.0D0)*H2FAC                                    PTH
C   SOVE FOR REF FREQ. NO. 2                                            PTH
      FK2=FCF20*(1.D0-(DSQRT(HPF2)/100.0D0))                            PTH
C   SET FM TO GREATER OF TWO REF. FREQ.                                 PTH
      IF(FK1.GT.FK2) GO TO 1003                                         PTH
      FM=FK2                                                            PTH
      GO TO 1004                                                        PTH
1003  FM=FK1                                                            PTH
C   SOLVE FOR VIRTUAL HT. OF F3000                                      PTH
1004  XX=FM/F3000                                                       PTH
      H3000=3100.D0*XX**2+70.0D0                                        PTH
      FC=4.D0*FM-3.D0*FCF10                                             PTH
      P3000=((FC+3.78D0*FCEO+1.5D0)/(1.26D0*FCEO+.5D0))*DLOG((FC+8.82D0 PTH
     1 *FCEO+3.5D0)/(FC-1.26D0*FCEO-0.5D0))                             PTH
      ZZ=(A-B)*(FC+3.78D0*FCEO+1.5D0)/(8.D0*FCEO)*DLOG((FC+7.78D0       PTH
     *  *FCEO+1.5D0)/(FC-.22D0*FCEO+1.5D0))                             PTH
      FF=FM/FCF20                                                       PTH
      KK=HPF2*FF*DLOG((1.D0+FF)/(1.D0-FF))/2.0D0                        PTH
C   CALCULATE PRED. HT. OF F2 LAYER MAX.                                PTH
      HMF2P=8.D0*(H3000-ZZ-B-KK)/P3000+HPF2+A                           PTH
C   CALC. HTS. OF OTHER LAYERS                                          PTH
      HMF1P=HMF2P-HPF2                                                  PTH
      HMEP=A                                                            PTH
      HBEP=B                                                            PTH
      HPF1=HMF1P-HMEP                                                   PTH
      HPE=A-B                                                           PTH
      ZT=DABS(ZT)                                                       PTH
      FCEO=DMIN1(FCEO*EFAC,(.99D0*FCF20-.5D0)/1.26D0)                   PTH
      IF(F1FAC)1014,1014,1015                                           PTH
1014  FCF10=1.26D0*FCEO+.5D0                                            PTH
      GO TO 1016                                                        PTH
1015  FCF10=FCF10*F1FAC                                                 PTH
1016  IF(SMOOTH)FCF10=FCF10/.8660254037844390D0                         PTH
      FCF10=DMIN1(FCF10,.99D0*FCF20)                                    PTH
      RETURN                                                            PTH
2000  WRITE(6,4)D                                                       PTH
4     FORMAT(4H D =D13.5,12H IS NEGATIVE)                               PTH
      STOP                                                              PTH
2001  WRITE(6,5)D                                                       PTH
5     FORMAT(4H D =D13.5,22H IS GREATER THAN 366.5)                     PTH
      STOP                                                              PTH
      END                                                               PTH
```

```
      SUBROUTINE  NFROMR(RFEC,PLASD)                                    NFR
C                                                                       NFR
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               NFR
      DOUBLE PRECISION K,J,M3000                                        NFR
      LOGICAL SMOOTH                                                    NFR
      COMMON/RIIPAR/M3000,FCF2,FCF1,FCE,HBE,HAE,HME,HAF1,HMF1,HAF2,HMF2, NFR
     1DUM(10),SNSQRT,QRAD,CQ,ID,IA ,SMOOTH                              NFR
C   HEIGHT ABOVE GROUND OF REF. PT.                                     NFR
      OH=RFEC-6370.D0                                                   NFR
C   NO - IS PCINT OF INTEREST ABOVE E LAYER MAX                         NFR
      IF(OH.GT.HME) GO TO 201                                           NFR
      IF(ID.EQ.1) GO TO 2001                                            NFR
C   IS PT. OF INTEREST BELOW E LAYER                                    NFR
      IF((HME-OH).GE.HAE) GO TO 300                                     NFR
C   NO - SOLVE FOR E LAYER PLASMA DENSITY                               NFR
      FO2=FCE*FCE*(1.D0-((HME-OH)/HAE)**2)                              NFR
      GO TO 202                                                         NFR
C   E LAYER WITH D LAYER TAIL                                           NFR
2001  ZP=2.D0*(HME-OH)/HAE                                              NFR
      K=1.5D0*DSQRT(1.7D0/(1.7D0+ZP))                                   NFR
      J=1.D0+(ZP-1.D0)*DABS(ZP-1.D0)/10.D0                              NFR
      J=J/(1.D0-.1D0/(1.D0+ZP)**2)                                      NFR
      ZPP=2.D0                                                          NFR
      IF(ZP.GT.0.D0)ZPP=2.D0-DLOG(1.D0+1.72D0*ZP**K)                    NFR
      IF(ZPP.LE.0.D0) GO TO 300                                         NFR
      ZPP=2.D0-ZPP**J                                                   NFR
      FO2=FCE*FCE*(1.D0-ZPP*ZPP/4.D0)                                   NFR
      GO TO 202                                                         NFR
C   SOLVE FOR REF. HTS. 1 THRU 3                                        NFR
201   HS2=HMF2-HAF2*DSQRT(1.D0-(FCF1/FCF2)**2)                          NFR
      HS1=HMF1-HAF1*DSQRT(1.D0-(FCE/FCF1)**2)                           NFR
      HT2=(HS2+HMF1)/2.D0                                               NFR
C   IS PT. OF INTEREST ABOVE REF PT 3                                   NFR
      IF(OH.GT.HT2) GO TO 203                                           NFR
C   NO- SOLVE FOR F1 LAYER PLASMA DENSITY                               NFR
      BIGK1=(HS1-HME+HS2-HMF1)/(2.D0*(FCF1-FCE))                        NFR
      BIGK3=HAF1/FCF1                                                   NFR
      BIGK2=HMF1-(FCF1*(HS1-HME)+FCE*(HS2-HMF1))/(2.D0*(FCF1-FCE))      NFR
      BIGA=BIGK1**2+BIGK3**2                                            NFR
      BIGB=2.D0*BIGK1*(BIGK2-OH)                                        NFR
      BIGC=OH**2-2.D0*OH*BIGK2+BIGK2**2-HAF1**2                         NFR
      FO=(-BIGB+DSQRT(BIGB**2-4.D0*BIGA*BIGC))/(2.D0*BIGA)              NFR
      IF(FO.LT.FCE)FO=FCE                                               NFR
      FO2=FO*FO                                                         NFR
      GO TO 202                                                         NFR
C   SOLVE FOR DENSITY ABOVE F2 LAYER MAX                                NFR
211   OHH=(OH-HMF2)/OH                                                  NFR
      Z=(OH-HMF2)/HAF2*2.D0                                             NFR
      FO2=FCF2**2*DEXP(1.D0-Z-DEXP(-Z))+(0.5D0+FCF2/10.0D0)*OHH*HMF2/OH NFR
      FO2=DMIN1(FO2,FCF2**2)                                            NFR
      GO TC 202                                                         NFR
C   IS POINT OF INTEREST ABOVE F2-LAYER MAX                             NFR
203   IF(OH.GT.HMF2) GO TO 211                                          NFR
C   NO-SOLVE FOR F2-LAYER PLASMA DENSITY                                NFR
      BIGK1=(HS2-HMF1)/(2.D0*(FCF2-FCF1))                               NFR
      BIGK3=HAF2/FCF2                                                   NFR
      BIGK2=HMF2-FCF2*BIGK1                                             NFR
```

```
      BIGA=BIGK1**2+BIGK3**2                                              NFR
      BIGB=2.D0*BIGK1*(BIGK2-OH)                                          NFR
      BIGC=OH**2-2.D0*OH*BIGK2+BIGK2**2-HAF2**2                           NFR
      FO=(-BIGB+DSQRT(BIGB**2-4.D0*BIGA*BIGC))/(2.D0*BIGA)                NFR
      FO2=FO*FO                                                           NFR
202   PLASD=FO2*12400.D0                                                  NFR
      IF(ID.EQ.1) GO TO 301                                               NFR
      RETURN                                                              NFR
300   PLASD=0.D0                                                          NFR
      IF(ID.EQ.0)RETURN                                                   NFR
301   CQ=DCOS(QRAD*.85D0)                                                 NFR
      IF(CQ.GT.0.0D0) GO TO 302                                           NFR
      CQ=0.D0                                                             NFR
      RETURN                                                              NFR
302   CQ=CQ**(0.25D0)                                                     NFR
303   PLASD=PLASD+SNSQRT*SNSQRT*CQ*1.D2/(1.D0+81.D0*((65.D0-OH)/HAE)**4)  NFR
      RETURN                                                              NFR
      END                                                                 NFR
```

```
      SUBROUTINE  DFP(NLATA,WLONGA,NLATB,WLONGB,DIST)                    DFP
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                DFP
      DOUBLE PRECISION NLATA,NLATB                                       DFP
      DATA  RAD/57.2957795130823200/                                     DFP
      DACOS(DUMMY)=DATAN2(DSQRT(1.0D0-DUMMY*DUMMY),DUMMY)                DFP
C   SHIFT INPUT LAT. BY ONE QUADRANT                                     DFP
      PPA=90.0D0-NLATA                                                   DFP
      PPB=90.0D0-NLATB                                                   DFP
      GGH=WLONGA                                                         DFP
      GGK=WLONGB                                                         DFP
C   FIND SMALLEST ANGULAR DIFF. BETWEEN A AND B                          DFP
      HHGK=GGH-GGK                                                       DFP
      ABHHGK=DABS(HHGK)                                                  DFP
      HHK=ABHHGK                                                         DFP
      IF(HHK.GT.180.D0) HHK=360.D0-HHK                                   DFP
C   SOLVE FOR BEARING FROM A TO B                                        DFP
      RAAPB=HHK/RAD                                                      DFP
      CSAAPB=DCOS(RAAPB)                                                 DFP
      RPPA=PPA/RAD                                                       DFP
      SINPPA=DSIN(RPPA)                                                  DFP
      COSPPA=DCOS(RPPA)                                                  DFP
      RPPB=PPB/RAD                                                       DFP
      SINPPB=DSIN(RPPB)                                                  DFP
      COSPPB=DCOS(RPPB)                                                  DFP
      COSAAB=COSPPA*COSPPB+SINPPA*SINPPB*CSAAPB                          DFP
      RAAB=DACOS(COSAAB)                                                 DFP
C   SOLVE FOR GREAT CIRCLE DISTANCE BETWEEN A AND B                      DFP
      DIST=6370.0D0*RAAB                                                 DFP
      RETURN                                                             DFP
      END                                                                DFP
```

```
      SUBROUTINE  MMDIP(NLAT,WLONG,SSMX)                                DMM
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               DMM
C COMPUTE MAGNETIC FIELD COMPONENTS MAGNETIC DIP AND MODIFIED MAGNETIC   DMM
C     DIP                                                               DMM
      DOUBLE PRECISION NLAT                                             DMM
      DIMENSION   P(7,7),DP(7,7),CP(7),AOR(7),SP(7)                     DMM
      DIMENSION   CT(7,7),H(7,7),G(7,7)                                 DMM
      DATA P,DP,SP/1.0D0,104*0.0D0/,CP/1.0D0,6*0.0D0/                   DMM
      DATA  RD/57.2957795130823200        /,HC/6371.2D3/                DMM
      DATA CT/2*0.D0,.3333333300,.2666666700,.2571428600,               DMM
     1 .2539682500,.2525252500,3*0.D0,.2D0,.2285714200,.2380952300,     DMM
     2 .2424242400,4*0.D0,.1428571400,.1904761900,.2121212100,          DMM
     3 5*0.D0,.1111111100,.1616161600,6*0.D0,.0909090900,14*0.D0/       DMM
      DATA G/0.D0,.3041120D0,.02403500,-.03151800,-.04179400,           DMM
     1 .01625600,-.01952300,0.D0,.02147400,-.05125300,                  DMM
     2 .06213000,-.04529800,-.03440700,-.00485300,2*0.D0,               DMM
     3 -.01338100,-.02489800,-.02179500,-.01944700,.00321200,           DMM
     4 3*0.D0,-.00649600,.00700800,-.00060800,.02141300,                DMM
     5 4*0.D0,-.00204400,.00277500,.00105100,5*0.D0,.00069700,          DMM
     6 .00022700,6*0.D0,.00111500/                                      DMM
      DATA H/8*0.D0,-.05798900,.03312400,.01487000,-.01182500,          DMM
     1 -.00079600,-.00575800,2*0.D0,-.00157900,-.00407500,              DMM
     2 .01000600,-.002D0,-.00873500,3*0.D0,.0002100,.0004300,           DMM
     3 .00459700,-.00340600,4*0.D0,.00138500,.00242100,                 DMM
     4 -.00011800,5*0.D0,-.00121800,-.00111600,6*0.D0,                  DMM
     5 -.00032500/                                                      DMM
      P1=NLAT                                                           DMM
      P2=360.0D0-(WLONG)                                                DMM
      IF(P1-89.9D0)2,4,1                                                DMM
 1    P1=89.9D0                                                         DMM
      P2=0.D0                                                           DMM
      GO TO 4                                                           DMM
 2    IF(P1+89.9D0)3,4,4                                                DMM
 3    P1=-89.9D0                                                        DMM
      P2=0.D0                                                           DMM
 4    PHI=P2/RD                                                         DMM
      AR=HC/(HC+3.D5)                                                   DMM
      C=DSIN(P1/RD)                                                     DMM
      S=DSQRT(1.-C**2)                                                  DMM
      SP(2)=DSIN(PHI)                                                   DMM
      CP(2)=DCOS(PHI)                                                   DMM
      AOR(1)=AR**2                                                      DMM
      AOR(2)=AR**3                                                      DMM
      DO 5 M=3,7                                                        DMM
      SP(M)=SP(2)*CP(M-1)+CP(2)*SP(M-1)                                 DMM
      CP(M)=CP(2)*CP(M-1)-SP(2)*SP(M-1)                                 DMM
 5    AOR(M)=AR*AOR(M-1)                                                DMM
      BV=0.D0                                                           DMM
      BN=0.D0                                                           DMM
      BPHI=0.D0                                                         DMM
      DO 6 N=2,7                                                        DMM
      FN=DBLE(FLOAT(N))                                                 DMM
      SUMR=0.D0                                                         DMM
      SUMT=0.D0                                                         DMM
      SUMP=0.D0                                                         DMM
      DO 7 M=1,N                                                        DMM
      IF(N-M)8,9,8                                                      DMM
```

```
9       P(N,N)=S*P(N-1,N-1)                                          DMM
        DP(N,N)=S*DP(N-1,N-1)+C*P(N-1,N-1)                           DMM
        GO TO 10                                                     DMM
8       P(N,M)=C*P(N-1,M)-CT(N,M)*P(N-2,M)                           DMM
        DP(N,M)=C*DP(N-1,M)-S*P(N-1,M)-CT(N,M)*DP(N-2,M)             DMM
10      FM=DBLE(FLOAT(M-1))                                          DMM
        TS=G(N,M)*CP(M)+H(N,M)*SP(M)                                 DMM
        SUMR=SUMR+P(N,M)*TS                                          DMM
        SUMT=SUMT+DP(N,M)*TS                                         DMM
7       SUMP=SUMP+FM*P(N,M)*(-G(N,M)*SP(M)+H(N,M)*CP(M))             DMM
        BV=BV+AOR(N)*FN*SUMR                                         DMM
        BN=BN-AOR(N)*SUMT                                            DMM
6       BPHI=BPHI-AOR(N)*SUMP                                        DMM
         COSLAM=DCOS((NLAT)/RD)                                      DMM
        COM1=-BV                                                     DMM
        COM2=BN                                                      DMM
        COM3=-BPHI/S                                                 DMM
        TMP=COM2**2+COM3**2                                          DMM
        BIGI=DATAN(-COM1/DSQRT(TMP))                                 DMM
        SSMX=(BIGI/DSQRT(BIGI**2+COSLAM))                            DMM
C     NEXT CARD DELETED DSINCE SMX NOT USED BY CRPL2                 DMM
C       SMX=DASIN(SSMX)                                              DMM
        RETURN                                                       DMM
        END                                                          DMM
```

```
      SUBROUTINE PERTF(T,NL,WL,FCF20)                                    PER
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                PER
      DOUBLE PRECISION NLREF,MR,MZ,MD,NL                                 PER
      COMMON/PERTC/NLREF,WLREF,FR,FD,MR,MD,FZ,MZ,DUT,DW,X,GAMMA,AZM,     PER
     1 DFAC,DMAC,AZREF                                                   PER
      RTD=57.2957795130823200                                            PER
      CALL DFP3(NLREF,WLREF,NL,WL,DIST,AZ)                               PER
      AZ=AZ-AZREF                                                        PER
      STHINV=1.0D0/DABS(DSIN((AZM-(90.D0+GAMMA))/RTD))                   PER
      Y=DUT*3600.D0*X*STHINV                                             PER
      W=DW*STHINV                                                        PER
      SD=DSIN((DIST+Y)/W*360.D0/RTD)                                     PER
      FCF20=FCF20*(1.D0+FZ*SD)*(1.D0+FR*DIST/FD)*(1.D0+AZ*DFAC)          PER
      FCF20=AMAX1(.51D0,FCF20)                                           PER
      RETURN                                                             PER
      ENTRY PERTM(T,NL,WL,FCF20)                                         PER
      FCF20=FCF20*(1.D0+MZ*SD)*(1.D0+MR*DIST/MD)*(1.D0+AZ*DMAC)          PER
      FCF20=AMAX1(1.D0,FCF20)                                            PER
      RETURN                                                             PER
      END                                                                PER
```

```
      SUBROUTINE FACTO(NLREF,WLREF,FCF2R,HMINR,ZTREF)                   FAC
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                FAC
      DOUBLE PRECISION KK,M3000P,M3000R,MFAC,MM,NLREF,K1                FAC
      COMMON/FACCOM/FM,ZZ,KK,P3000                                      FAC
      COMMON/RIIPAR/M3000P,FCF2,FCF1,FCE,HBE,HAE,HME,HAF1,HMF1,HPF2,    FAC
     1 HMF2,SNS(5),MFAC,F2FAC,DUM(6),IDUM(3)                            FAC
      DATA RTD/57.2957795130823200/                                     FAC
      F2FAC=1.D0                                                        FAC
      MFAC=1.D0                                                         FAC
      IF(FCF2R.EQ.0.D0) RETURN                                          FAC
      ZTS=SNS(2)                                                        FAC
      SNS(2)=ZTREF                                                      FAC
      CALL CRPL2(NLREF,WLREF)                                           FAC
      F2FAC=DABS(FCF2R/FCF2)                                            FAC
      IF(HMINR.EQ.0.D0) GO TO 10                                        FAC
      CALL RIIP(6370.D0,NLREF/RTD,WLREF/RTD,PLASD)                      FAC
      P1=(FCF2+3.78D0*FCE+1.5D0)/FCF1*DLOG((FCF2+8.82D0*FCE+3.5D0)/     FAC
     1  (FCF2-FCF1))                                                    FAC
      MM=HAE/8.D0                                                       FAC
      Z1=MM*(FCF2+3.78D0*FCE+1.5D0)/FCE*DLOG((FCF2+7.78D0*FCE+1.5D0)/   FAC
     1 (FCF2-.22D0*FCE+1.5D0))                                          FAC
      FFM=FCF1+(FCF2-FCF1)/4.D0                                         FAC
      FF1=FFM/FCF2                                                      FAC
      K1=50.D0*FF1*DLOG((1.D0+FF1)/(1.D0-FF1))                          FAC
      H3000=P3000/P1*(DABS(HMINR)-Z1-HBE-HPF2*K1/100.D0)+ZZ+HBE+KK      FAC
      M3000R=(FM/FCF2)*DSQRT(3100.D0/(H3000-70.D0))                     FAC
      MFAC=DSIGN(M3000R/M3000P,HMINR)                                   FAC
   10 F2FAC=DSIGN(F2FAC,FCF2R)                                          FAC
      SNS(2)=ZTS                                                        FAC
      RETURN                                                            FAC
      END                                                               FAC
```

Appendix B

Block Diagram and Flow Chart for WIMP Ray-Tracing Program TRISL

Explanation:

BI. TRISL BLOCK DIAGRAM

This presentation shows the general communication network among the various functional processes of TRISL, and illustrates the iterative complications inherent in the program. The function of the various blocks is generally described as follows:

BLOCK I:	Initialization Procedures.
BLOCK II:	E-layer refraction upwards.
BLOCK III:	F_1-layer refraction upwards.
BLOCK IV:	Reflection layer initialization.
BLOCK V:	Tilt calculation and consistency checks.
BLOCK VI:	F_1-layer refraction downwards.
BLOCK VII:	E-layer refraction downwards.
BLOCK VIII:	Error checks.
BLOCK IX:	Iterate reflection and tilt.
BLOCK X:	Traffic control.
BLOCK XI:	Normal Terminal calculations.
BLOCK XII:	ERROR exits.

B2. DETAILED FLOW CHARTS

Detailed flow charts are presented for each block. Most line members of TRISL are accounted for except when so doing unduly complicates comprehension of the process.

B3. FLOW CHART SYMBOLS AND CONVENTIONS

(a) Process entry/exit point.

(b) Processing statements.

(c) Decision trees, two or three way branches.

(d) External procedures.

(e) Numbers in the form Snnn refer to statement numbers in the program.

(f) Numbers in the form nnn refer to line numbers of the program listing.

(g) Entry points to a block are labelled generally with block designation statement number and line number specifying the source.

(h) Exit points from a block are generally labelled by block designation, statement number, and line number specifying the destination.

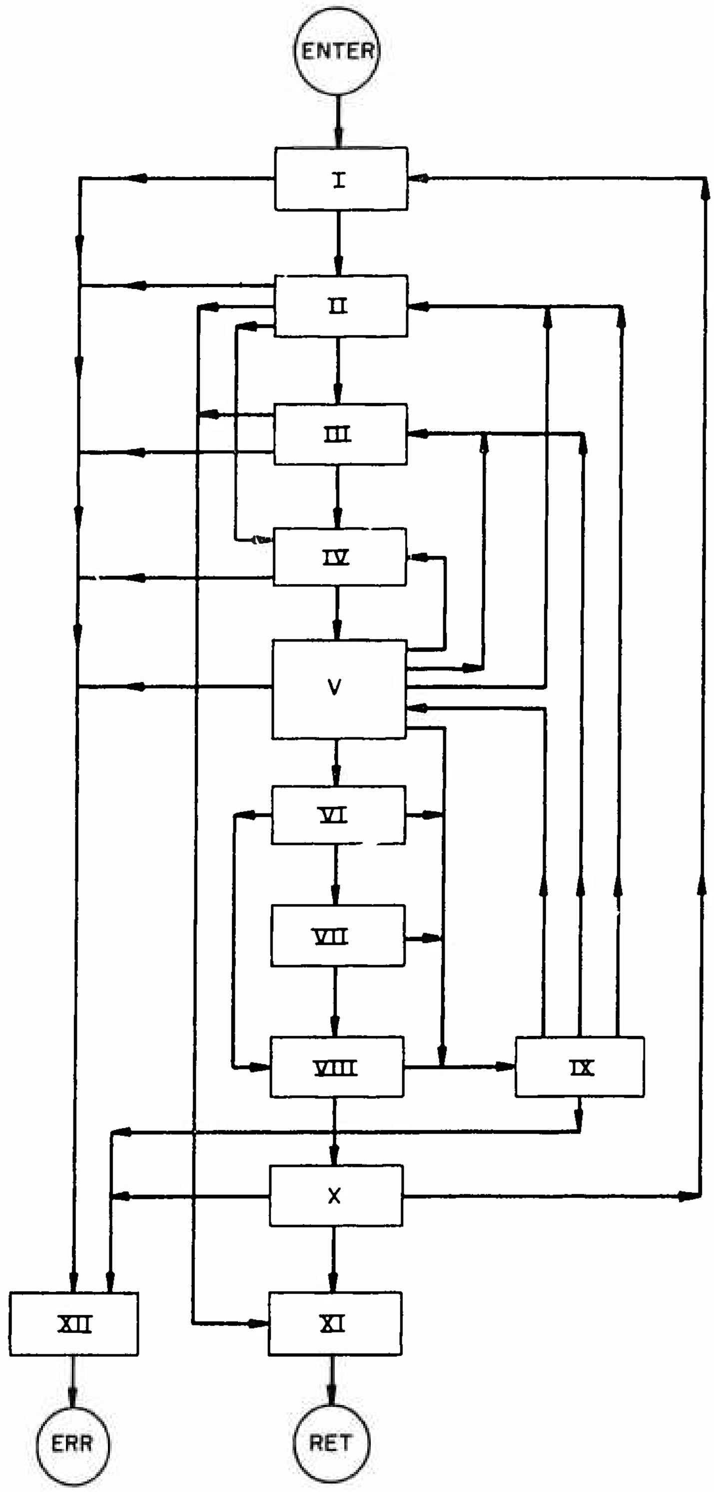
(i) Error exits are labelled by an error number and the error condition detected.

(j) Exits from the bottom generally enter the next block at the top.

(k) External call names and calling line numbers are contained within the external procedure symbols.

B4. PROGRAM LISTING

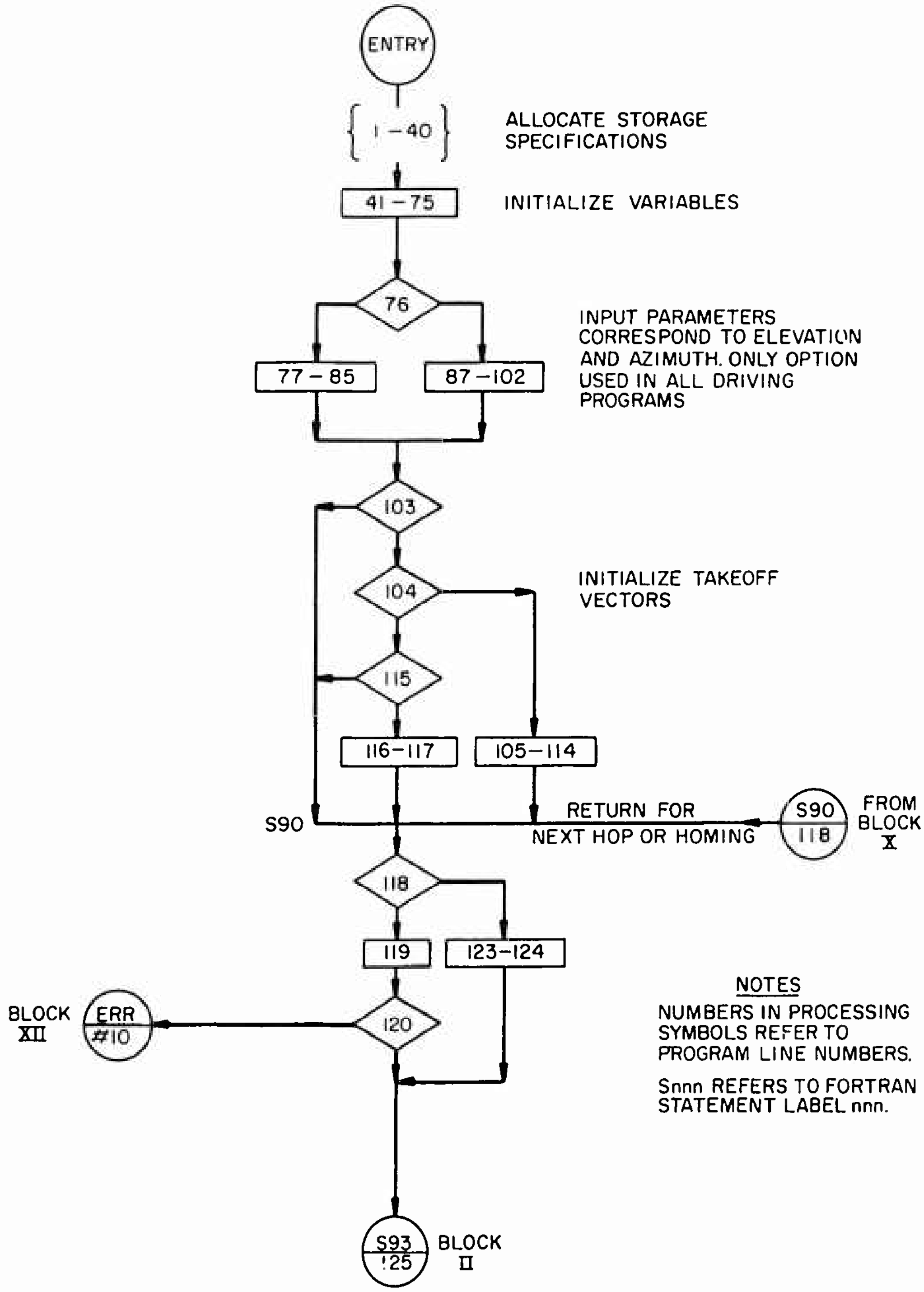
Listings are included for TRISL together with its externals.



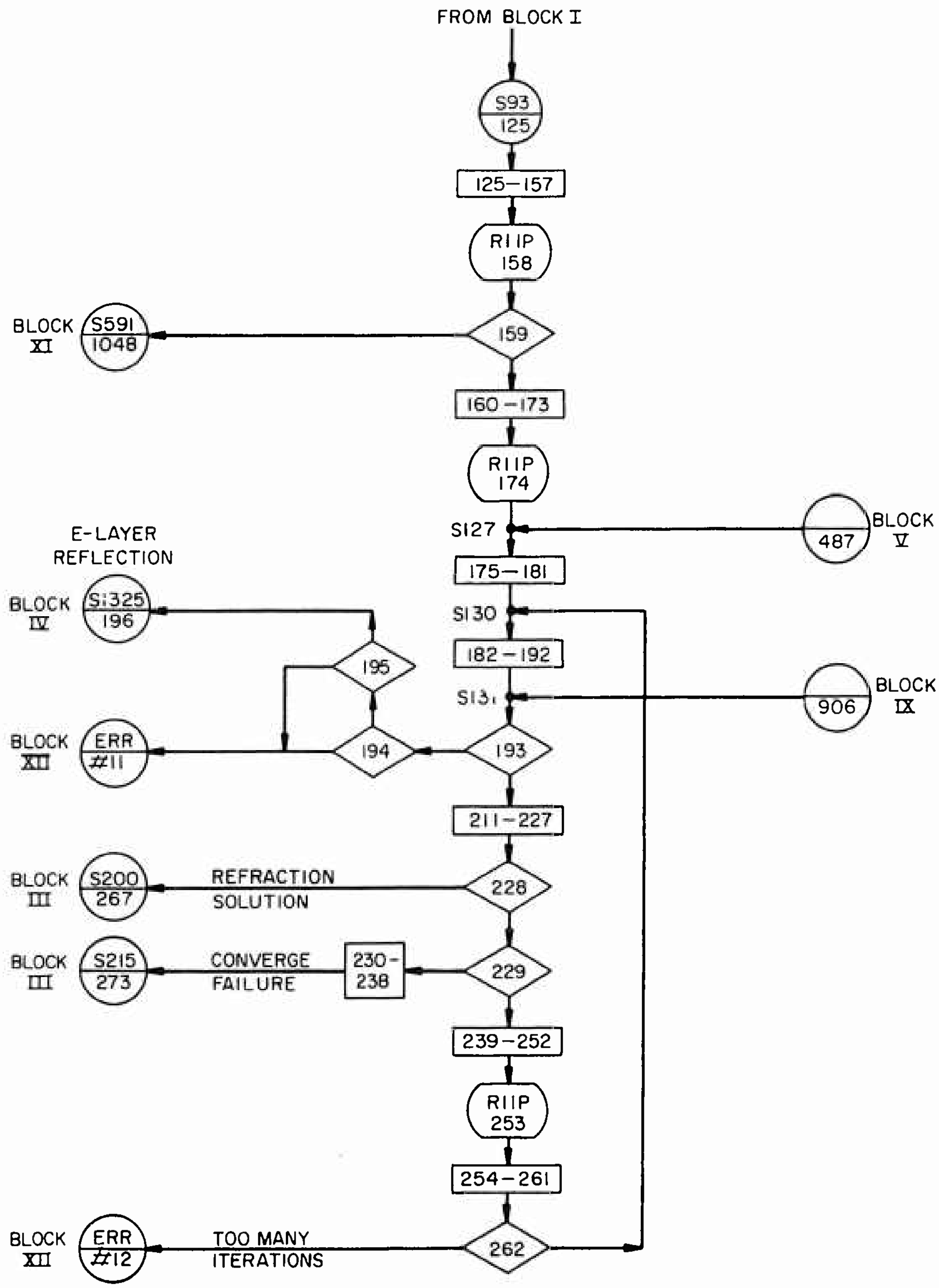
TRISL Block Diagram

TRISL: Block I, Argument List

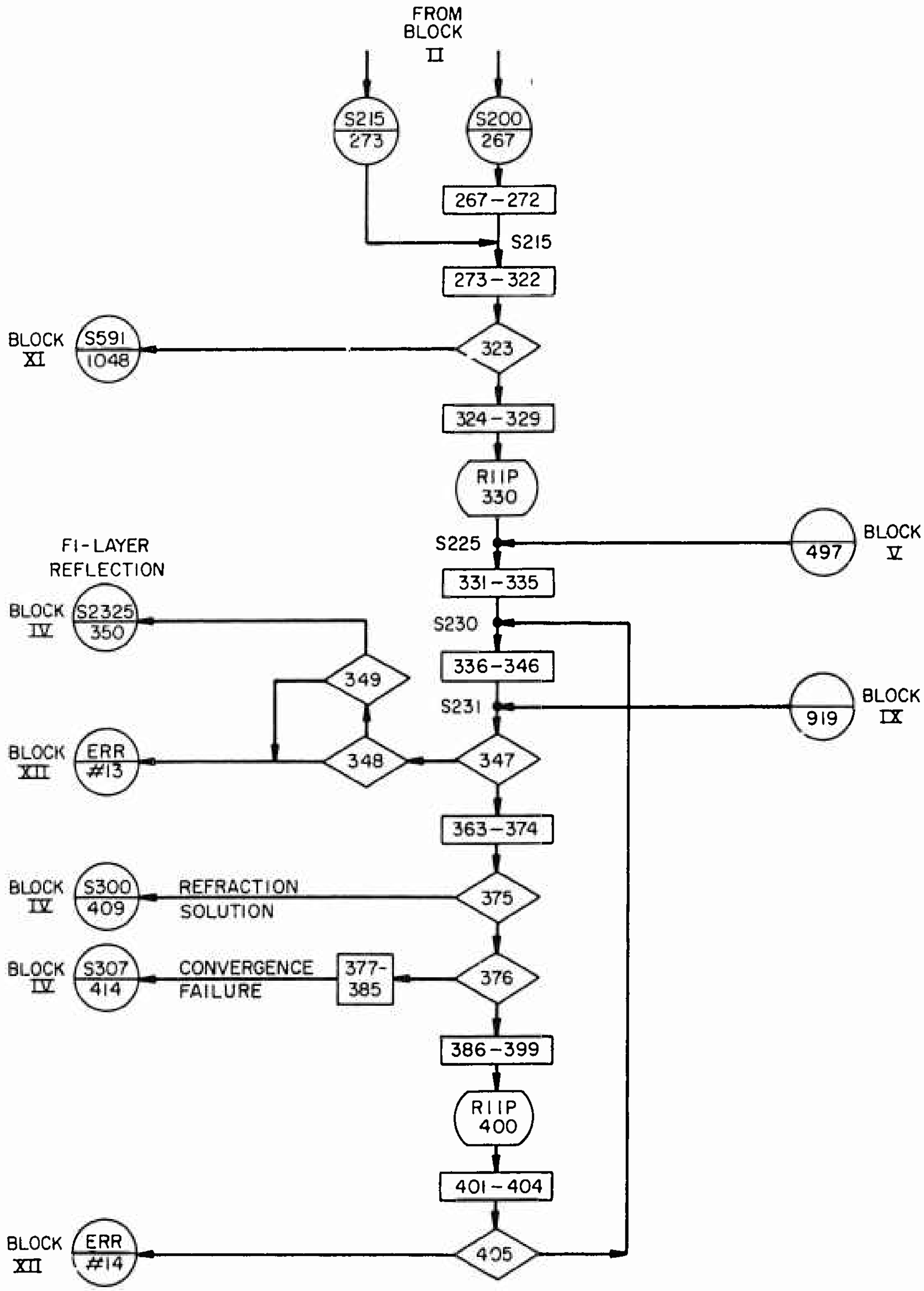
Element	I/D	Mnemonic	Type	Definition
1	I	MAI	Double	Geocentric Distance, Initial Point
2	I	NLAI	Double	North Latitude, Initial Point
3	I	WLAI	Double	West Longitude, Initial Point
4	I	NLAZ	Double	Takeoff Bearing (all calls) or Target North Latitude
5	I	WLAZ	Double	Takeoff Elevation (all calls) or Target West Longitude
6	I	MAZ	Double	Less than 6000 km (all calls) or Target Geocentric Distance
7	I	FREQ	Double	Frequency
8	I	MODE	Integer	|MODE| = Number of hops
9	I	RTARG	Double	= 6370 km, (all calls)
10	O	RS 2	Double	Geocentric Distance, End Point
11	O	NLTARG	Double	North Latitude, End Point
12	O	WLTARG	Double	West Longitude, End Point
13	O	PPTOT	Double	Phase Path Length
14	O	GPTOT	Double	Group Path Length
15	O	AZF	Double	Final to Initial Azimuth
16	O	ELF	Double	End Point Elevation
17	O	SRANGE	Double	Final-Initial Straight Line Distance
18	O	ELSR	Double	Takeoff Elevation
19	O	DIST	Double	Final-Initial Great Circle Ground Range
20	O	BEAR	Double	Initial to Final Azimuth
21	O	IGOBAK	Integer	Error Flag 8 = 0 No Error = 1 Fatal Error = 2 Penetrate



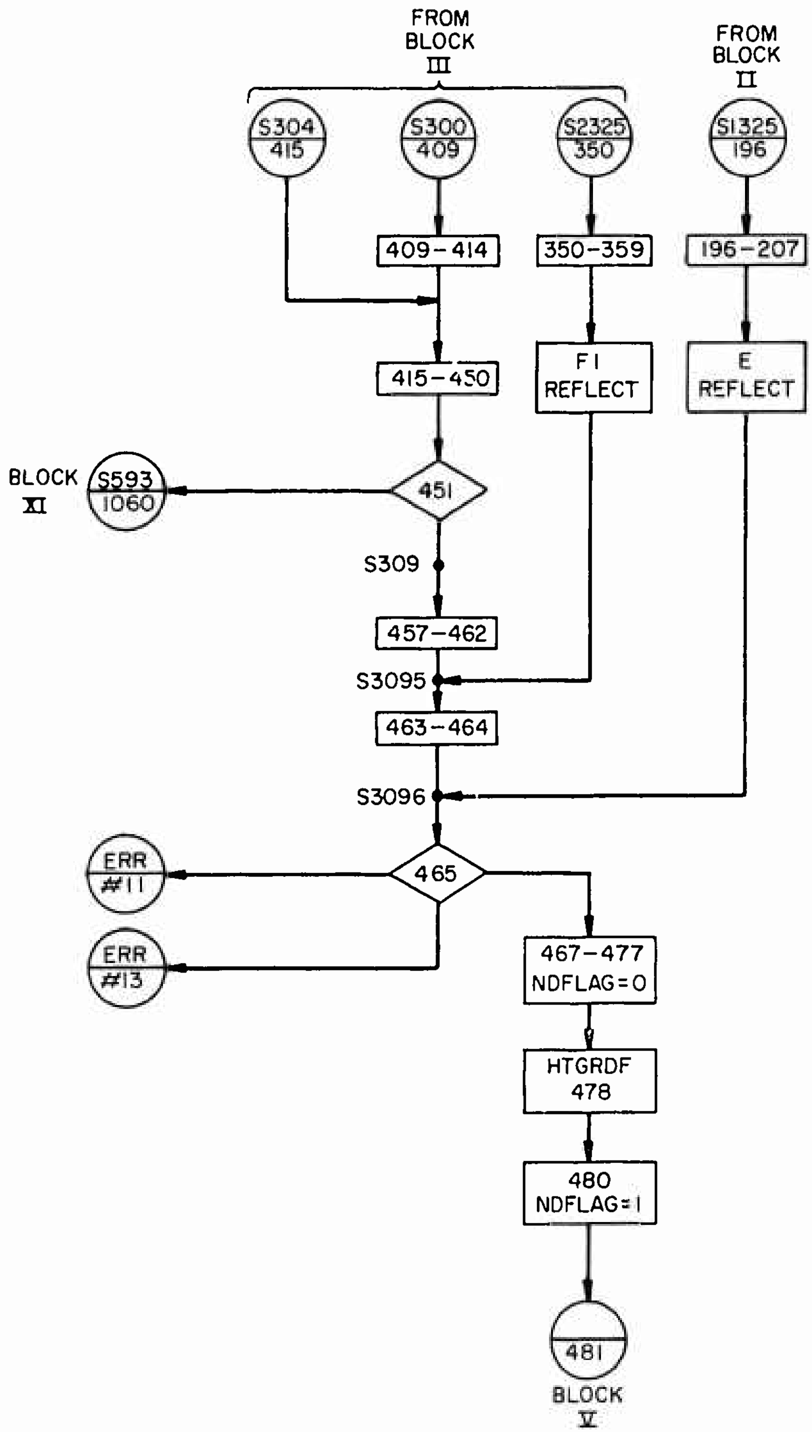
Block I, Initialization Procedures



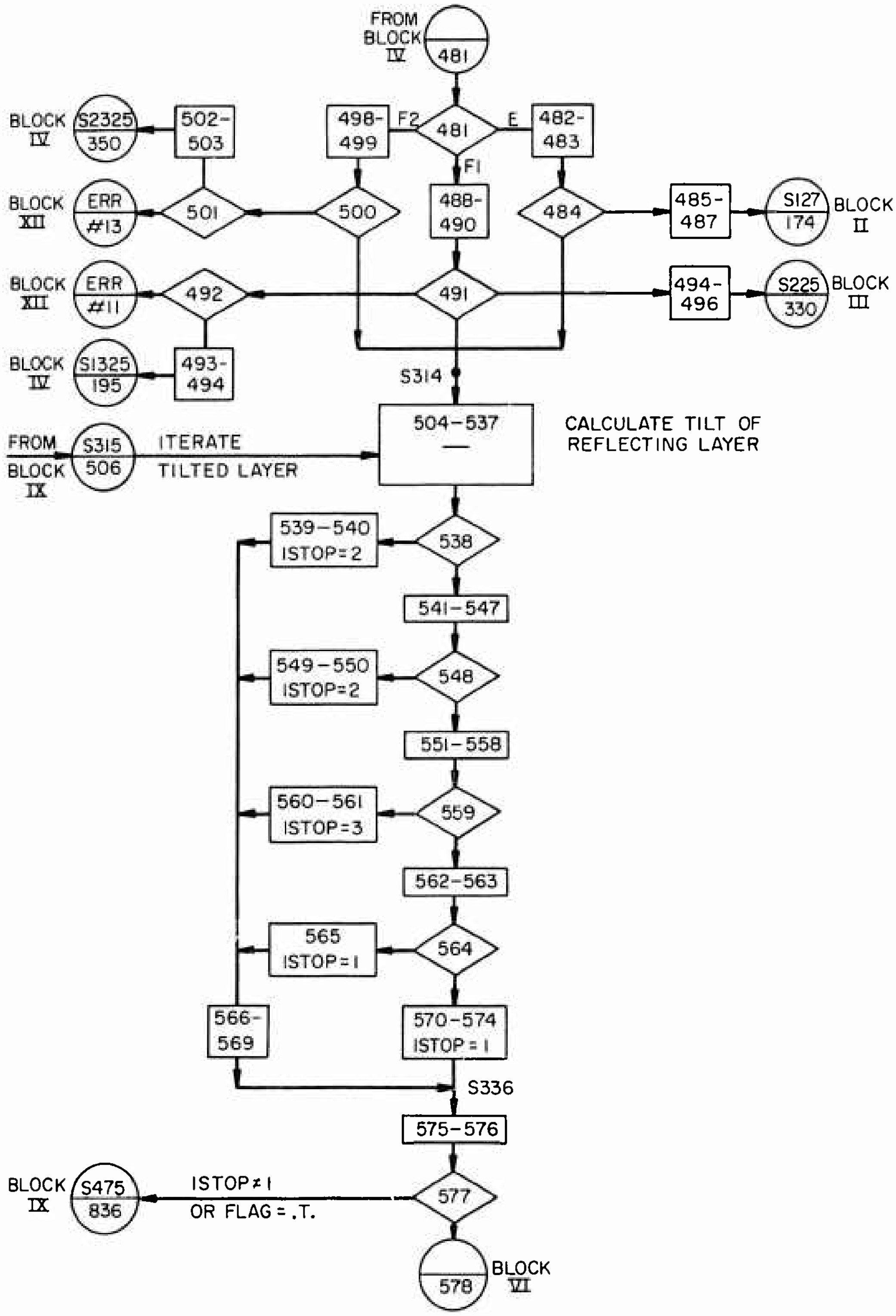
Block II, E-Layer Refraction Upwards



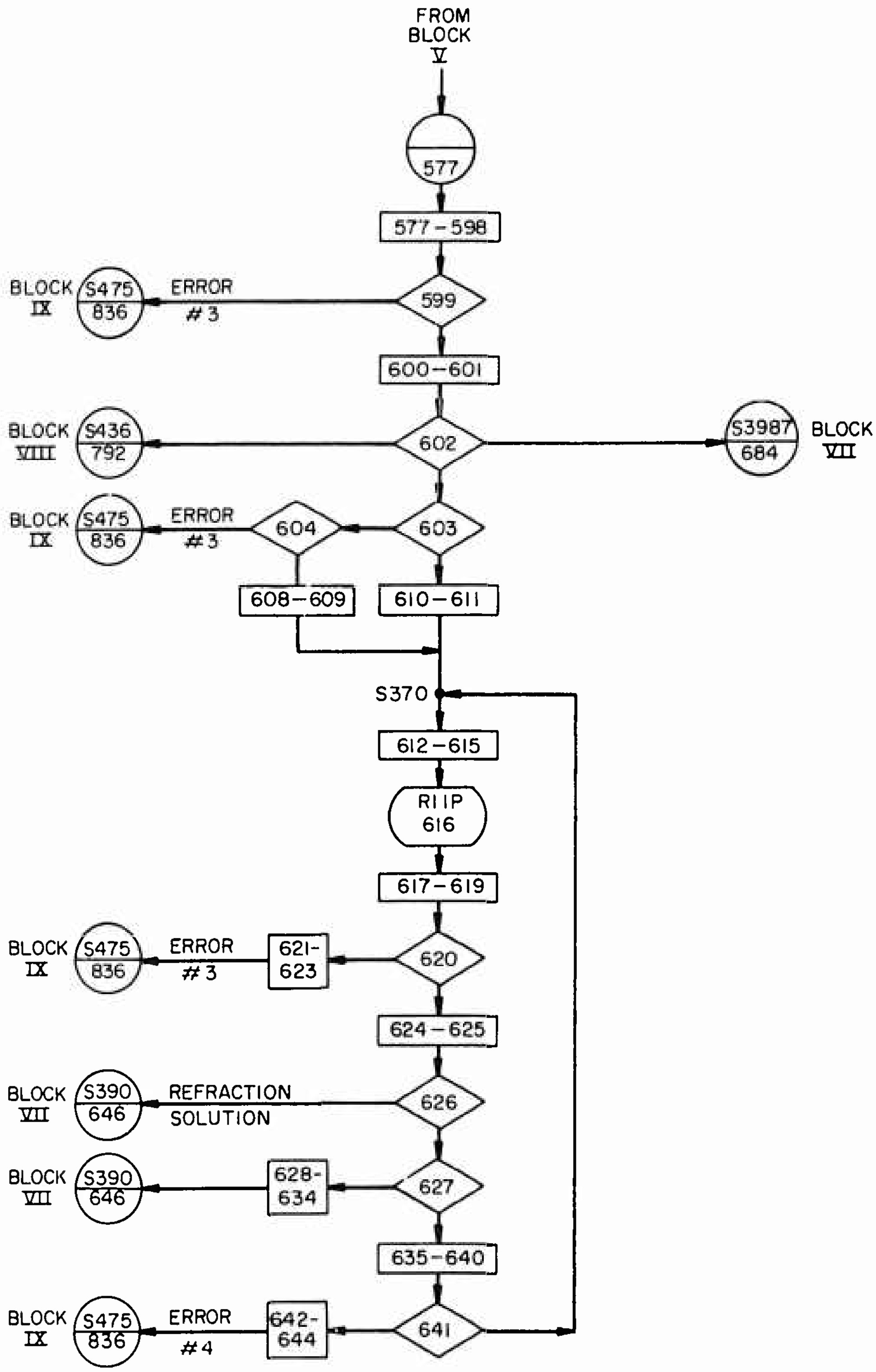
Block III, F_1-Layer Refraction Upwards



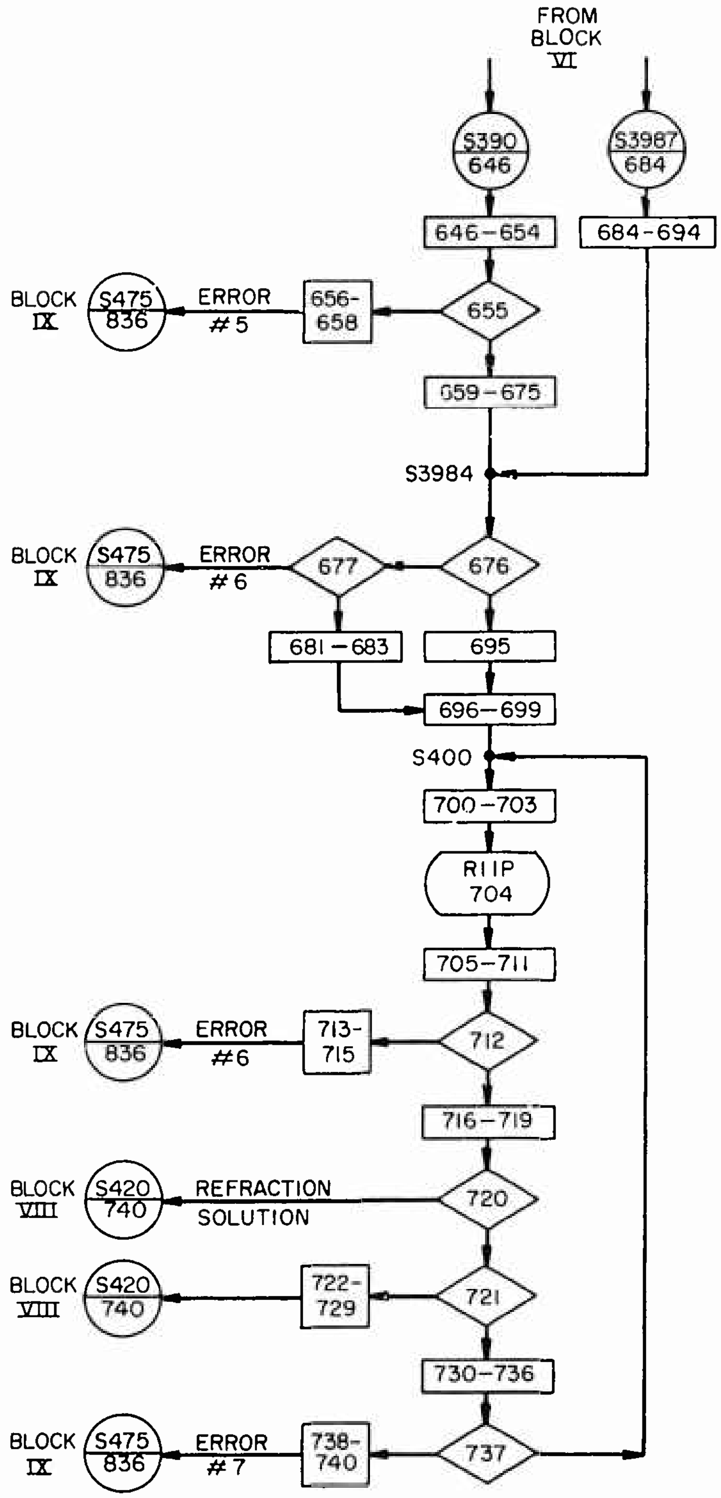
Block IV, Reflection Layer Initialization



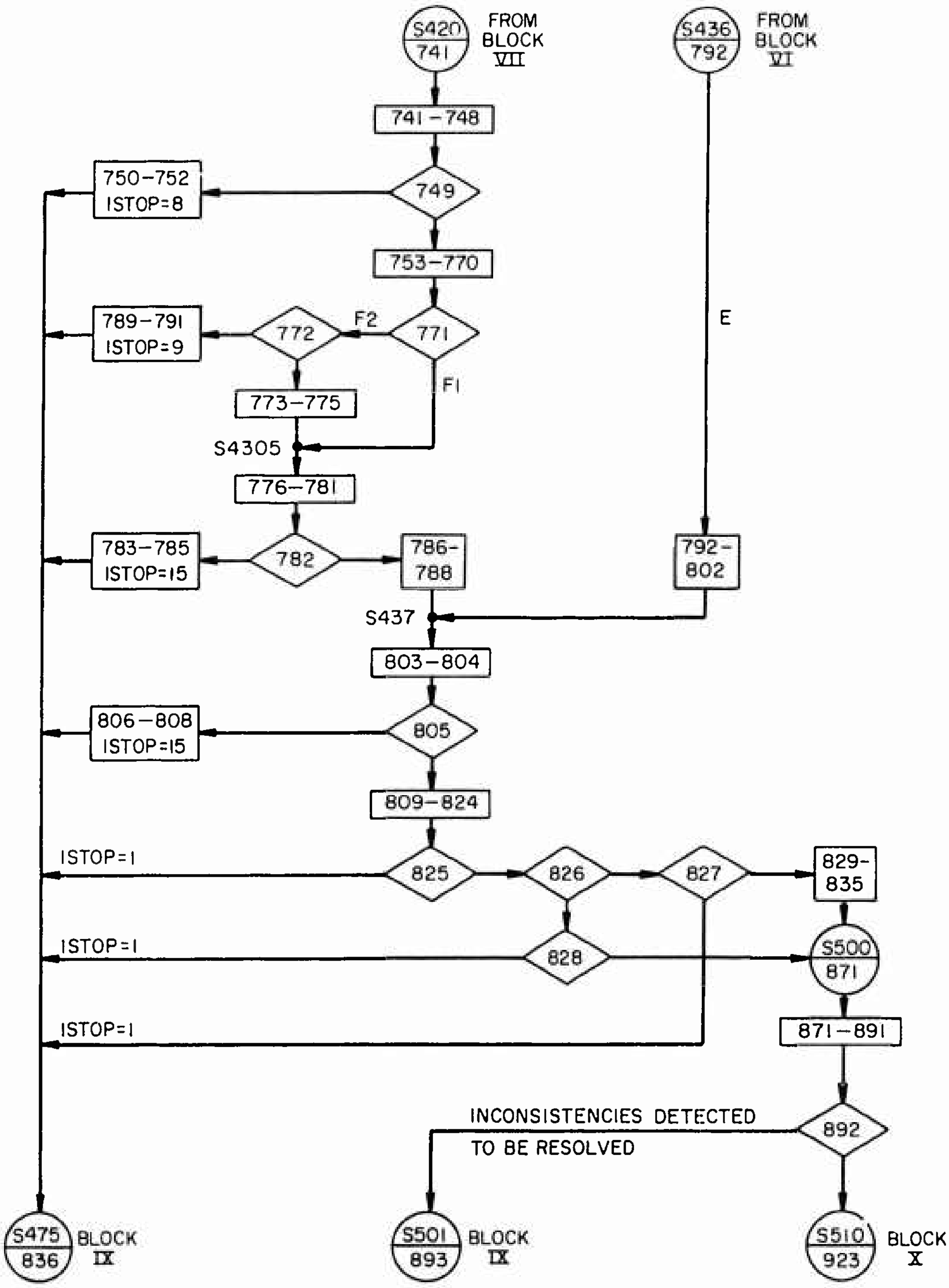
Block V, Tilt Calculation and Consistency Checks



Block VI, F_1-Layer Refraction Downwards



Block VII, E-Layer Refraction Downwards

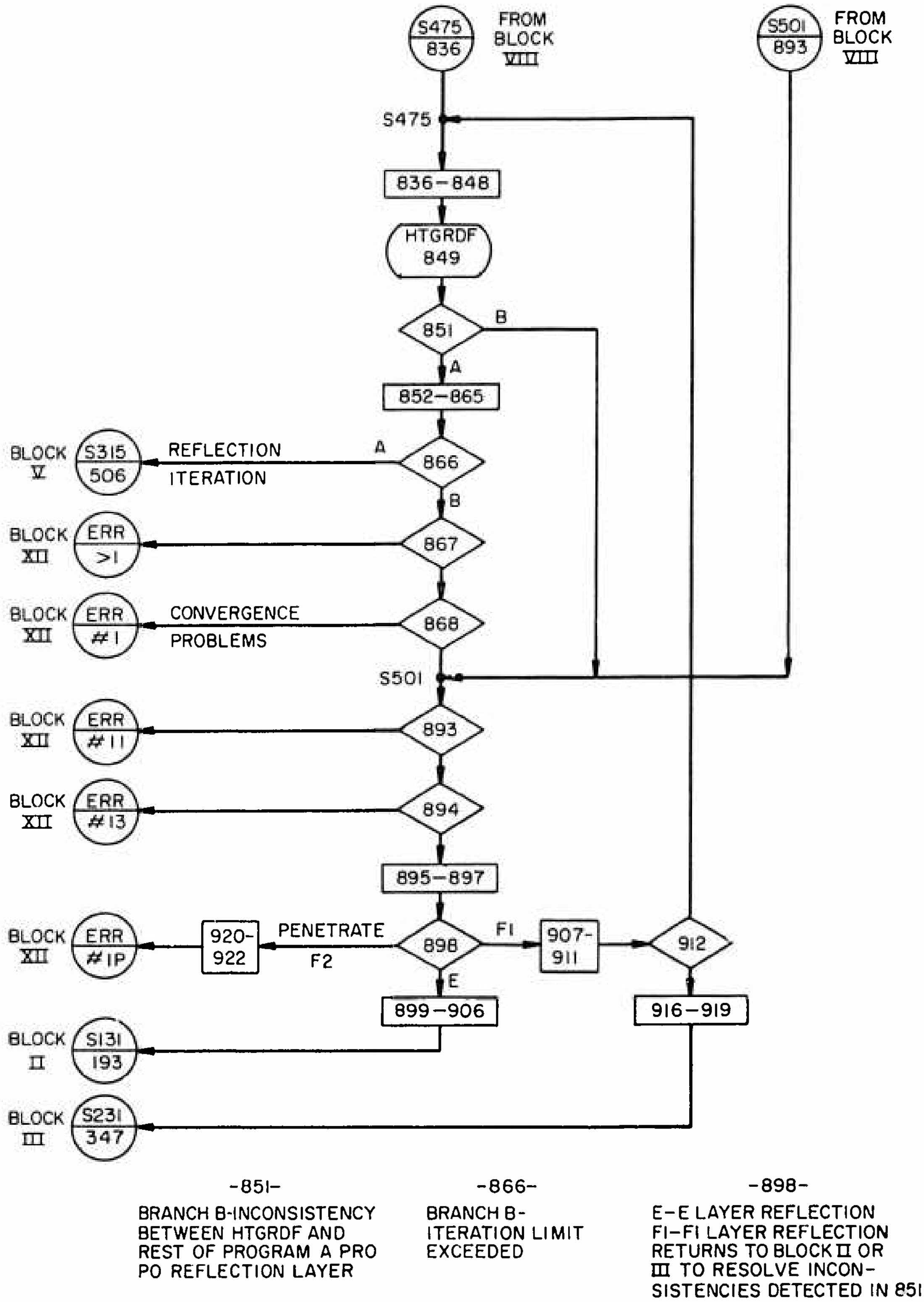


NOTES

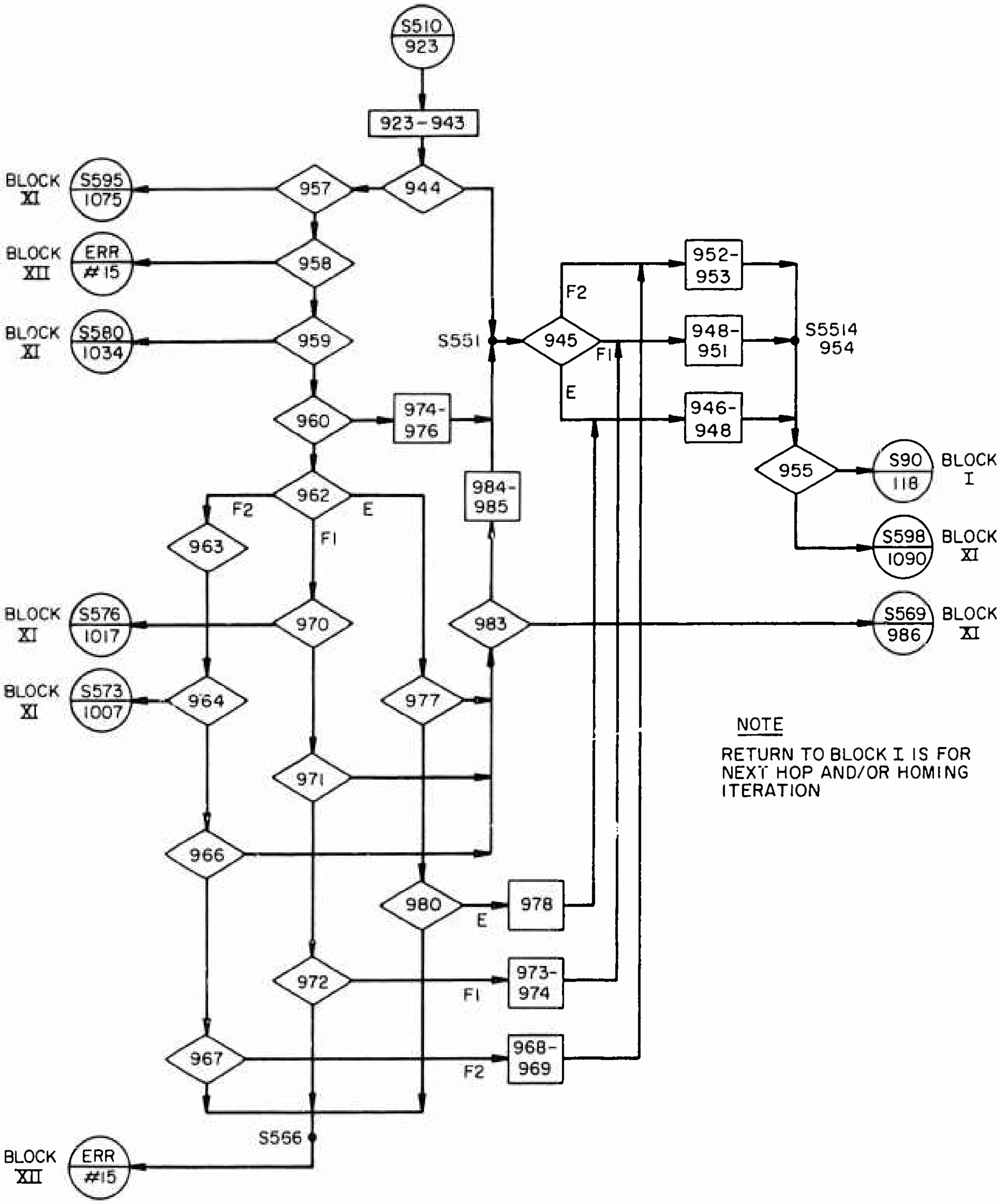
BLOCK X EXIT IS DESIRED CONVERGENT SOLUTION TO TILTED LAYER ITERATION

THE ISTOP=1 EXITS TO S475 ARE NORMAL. THE OTHER ERROR CONDITIONS MAY BE CORRECTED IN SUCCEEDING ITERATIONS

Block VIII, Error Checks



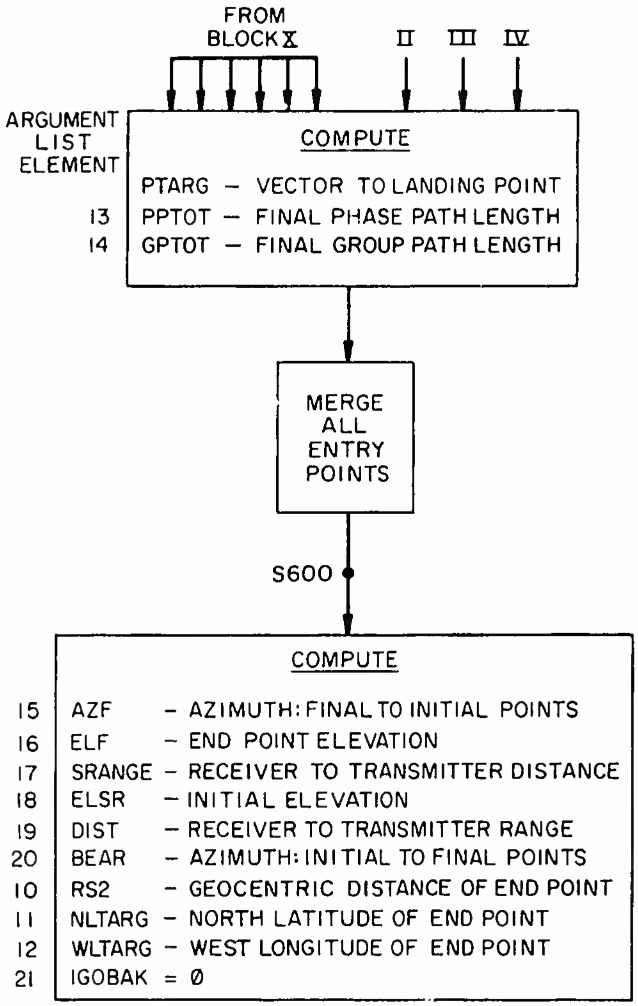
Block IX, Iterate Reflection and Tilt



NOTE

RETURN TO BLOCK I IS FOR NEXT HOP AND/OR HOMING ITERATION

Block X, Traffic Control



Block XI, Normal Terminal Calculations

```
      SUBROUTINE TRISL(MA1,NLA1,WLA1,NLA2,WLA2,MA2,FREQ,MODE,RTARG,RS2,  TR2
     *NLTARG,WLTARG,PPTOT,GPTOT,AZF,ELF,SRANGE,ELSR,DIST,BEAR,IGOBAK)     TR2
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                TR2
      LOGICAL HITS,FLAG,SEARCH,ASC,FLAG2,FLAG3,NOELAY,INCHOP,LP(6),NIP   TR2
      LOGICAL GPONLY                                                     TR2
      DOUBLE PRECISION  MESSPR                                           TR2
      DOUBLE PRECISION NLA1,MA1,NLA2,MA2,NLTARG,MAG,MA1B1,MB1,NLC1,NL4,  TR2
     1MRX1SQ,NLC2,MR3,NL8,NLC3,MN1,MA2C2,MC2,MA3C3,MC3,NLD,MRM1,MRM2,    TR2
     2MRY1SQ,MC4SQ,NLE,MRM3,MRM4,MC5SQ,NLW2,KP,L,N1,M,KPPP,LOC,          TR2
     3MAGVEC,NORTH(3),ME,MN,NORTH1(3),EAST1(3),MR2,MC1PSQ,M3000          TR2
      REAL OUT2                                                          TR2
      DIMENSION S1(3),A1(3),B1(3),A11(3),EAST(3),MESSPR(15),             TR2
     *       XI(3),KP(3),L(3,3),P1P(3),PC2(3),P8(3),R3(3),PC3(3),P10(3),C TR2
     *2(3),PR1(3),N1(3),P0(3),A2(3),M(3,3),C3(3),A3(3),PHI(3),KPPP(3),P( TR2
     *3,3),PD(3),PE(3),PQ(3),PW2(3),W2(3),PTARG(3),          RN1(3),R2(3 TR2
     *),PC1(3),P4(3)                                                     TR2
      COMMON/FLIMSY/DIST1,ICNPRB  /GPFLAG/GPONLY,IWRITE                  TR2
      COMMON/CPREV/PQ,P,RT1,C52PPP,C53PPP,MRM1,CBETM1,MRM4,RY,           TR2
     2  DPP1P,MC5SQ,RM1,YY2PPP,ZY2PPP,H1,MC4SQ,MRY1SQ,YY1PPP,            TR2
     2 ZY1PPP,C42PPP,C43PPP,YS3PPP,ZS3PPP,C32PPP,RTM1,DNDR,DNDNL,        TR2
     3 DNDWL,D,GNU,PP2,PPEF1D,CBETM2,HE,PPF2,RV,HF1,PP2A,DPP2P,          TR2
     4 MB1,SBET1,GPF2,GPEF1D,DGP2P,DGP1P,                                TR2
     5 F1,PREV(56),HITS,INCHOP,FLAG,FLAG2,FLAG3,NOELAY,LP                TR2
      COMMON/VECTRS/LOC(3,8),VEC(3,6),IREFL                              TR2
      COMMON/NIPRIP/NIP                                                  TR2
     */RIIPAR/M3000,FCF2,FCF1,FCE,HBE,HAE,HME,HAF1,HMF1,HAF2,HMF2,DUM(13 TR2
     *),ID(3)/RTCOM/COSBO,NUSE                                           TR2
      COMMON/REMWEN/INUSE,NNUSE(20),FCREF(20)                            TR2
      DATA R,EPS1,EPS2,EPS3,EPS4,EPS5/6670.D0,5*1.D0/                    TR2
      DATA  PI,RTD,DTR,R0/3.14159265358979300,57.29577951308232D0,       TR2
     *.01745329251994330D0,6370.D0/                                      TR2
      DATA  LIMEA,LIMF1A,LIMF1D,LIMED,LIMF2/5*15/                        TR2
      DATA MESSPR/7HF2 ITER,8HBIG TILT,8HBIG TILT,9HF1-D ITER,           TR2
     1 9HF1-D REFL,7HF1 MISS,8HE-D ITER,8HE-D REFL,6HE MISS,             TR2
     2 8HI-I REFL,9HE-F1 DUCT,8HE-A ITER,10HF1-F2 DUCT,                  TR2
     3 9HF1-A ITER,9HMISS TARG/                                          TR2
      WLON(ARG1,ARG2)=DSIGN(PI,ARG1)+PI-DATAN2(ARG1,ARG2)                TR2
      DACOS(ARG1)=DATAN2(DSQRT(1.D0-ARG1**2),ARG1)                       TR2
      DASIN(ARG1)=DATAN2(ARG1,DSQRT(1.D0-ARG1**2))                       TR2
      INCHOP=.TRUE.                                                      TR2
      HITS=.TRUE.                                                        TR2
      ASC=MODE.EQ.0                                                      TR2
      SEARCH=ASC                                                         TR2
      PP0=0.D0                                                           TR2
      PPEF1A=0.D0                                                        TR2
      DPP1=0.D0                                                          TR2
      PP2A=0.D0                                                          TR2
      PPF2=0.D0                                                          TR2
      PP2=0.D0                                                           TR2
      PPEF1D=0.D0                                                        TR2
      DPP2P=0.D0                                                         TR2
      PP0P=0.D0                                                          TR2
      PPTOT=0.D0                                                         TR2
      GPTOT=0.D0                                                         TR2
      GPEF1A=0.D0                                                        TR2
      DGP1=0.D0                                                          TR2
      DGP2=0.D0                                                          TR2
```

```
      GPEF1A=0.D0                                                       TR2
      GPF2=0.D0                                                         TR2
      GPEF1D=0.D0                                                       TR2
      DGP2P=0.D0                                                        TR2
      DGP1F=0.D0                                                        TR2
      ICNPRB=0                                                          TR2
      INUSE=0                                                           TR2
      IGOBAK=0                                                          TR2
      IHOPS=0                                                           TR2
      RS2=RTARG                                                         TR2
      CN=DCOS(NLA1*DTR)                                                 TR2
      SN=DSIN(NLA1*DTR)                                                 TR2
      CW=DCOS(WLA1*DTR)                                                 TR2
      SW=DSIN(WLA1*DTR)                                                 TR2
      A1(1)=CN*CW                                                       TR2
      A1(2)=-CN*SW                                                      TR2
      A1(3)=SN                                                          TR2
      IF(MA2.LT.6000.D0) GO TO 50                                       TR2
      CN=DCOS(NLA2*DTR)                                                 TR2
      DO 45 I=1,3                                                       TR2
      A11(I)=MA1*A1(I)                                                  TR2
45    A1(I)=A11(I)                                                      TR2
      B1(1)=MA2*DCOS(WLA2*DTR)*CN-A1(1)                                 TR2
      B1(2)=-MA2*DSIN(WLA2*DTR)*CN-A1(2)                                TR2
      B1(3)=MA2*DSIN(NLA2*DTR)-A1(3)                                    TR2
      MB1=MAG(B1)                                                       TR2
      DTA1B1=DOT(A1,B1)                                                 TR2
      GO TO 60                                                          TR2
C  WHEN MA2 IS INPUTTED AS A NUMBER LESS THAN 6000, NLA2 IS ASSUMED     TR2
C     TO BE THE BEARING,AND WLA2 IS ASSUMED TO BE THE TAKE-OFF ANGLE    TR2
50    EAST1(1)=SW                                                       TR2
      EAST1(2)=CW                                                       TR2
      EAST1(3)=0.D0                                                     TR2
      CALL CROSS(A1,EAST1,NORTH1)                                       TR2
      CAZ=DCOS(NLA2*DTR)                                                TR2
      SAZ=DSIN(NLA2*DTR)                                                TR2
      CEL=DCOS(WLA2*DTR)                                                TR2
      SEL=DSIN(WLA2*DTR)                                                TR2
      DO 55 I=1,3                                                       TR2
      B1(I)=CEL*(CAZ*NORTH1(I)+SAZ*EAST1(I))+SEL*A1(I)                  TR2
      A11(I)=MA1*A1(I)                                                  TR2
55    A1(I)=A11(I)                                                      TR2
      MB1=1.D0                                                          TR2
      DTA1B1=SEL*MA1                                                    TR2
60    IF(DABS(MA1-R0).LE.1.D-5) GO TO 90                                TR2
      IF(DASIN(R0/MA1)-DACOS(-DTA1B1/(MA1*MB1)))88,88,80                TR2
80    T1=(-DTA1B1-DSQRT(DTA1B1**2-DOT(B1,B1)*(DOT(A1,A1)-R0**2)))       TR2
     */DOT(B1,B1)                                                       TR2
      DO 85 J=1,3                                                       TR2
85    A1(J)=B1(J)*T1+A1(J)                                              TR2
      A=-2.D0*DOT(A1,B1)/DOT(A1,A1)                                     TR2
      DO 87 J=1,3                                                       TR2
87    B1(J)=B1(J)+A*A1(J)                                               TR2
      PPTOT=T1*MB1                                                      TR2
      GPTOT=PPTOT                                                       TR2
      GO TO 90                                                          TR2
88    IF(DTA1B1.GE.0.D0)GO TO 90                                        TR2
```

```
      PPTOT=-2.D0*DTA1B1/MB1                                      TR2
      GPTOT=PPTOT                                                 TR2
 90   IF(INCHOP) GO TO 92                                         TR2
      IONREF=IONREF+1                                             TR2
      IF(IONREF.LT.40) GO TO 93                                   TR2
      ISTOP=10                                                    TR2
      GO TO 7310                                                  TR2
 92   IONREF=0                                                    TR2
      IHOPS=IHOPS+1                                               TR2
 93   PREV(13)=0.D0                                               TR2
C   START OF E LAYER REFRACTION                                   TR2
      NIP=.FALSE.                                                 TR2
      INT=0                                                       TR2
      INUSE=INUSE+1                                               TR2
      FLAG=.FALSE.                                                TR2
      FLAG2=.FALSE.                                               TR2
      CALL  CROSS(A1,B1,XI)                                       TR2
      CALL  CROSS(XI,B1,KP)                                       TR2
      MA1B1=MAG(XI)                                               TR2
      MB1=MAG(B1)                                                 TR2
      A1MAG=MAG(A1)                                               TR2
C   EQN(2)                                                        TR2
      DO 100 J=1,3                                                TR2
      L(1,J)=XI(J)/MA1B1                                          TR2
      L(2,J)=B1(J)/MB1                                            TR2
 100  L(3,J)=KP(J)/(MA1B1*MB1)                                    TR2
      Y5P=0.D0                                                    TR2
      Z5P=0.D0                                                    TR2
C   EQN(9)                                                        TR2
      P1P(1)=0.D0                                                 TR2
      DO 110 I=2,3                                                TR2
 110  P1P(I)=L(I,1)*A1(1)+L(I,2)*A1(2)+L(I,3)*A1(3)               TR2
      YC1P=DSQRT(B**2-P1P(3)**2)                                  TR2
C   EQN(12)                                                       TR2
      DO 120 J=1,3                                                TR2
 120  PC1(J)=L(2,J)*YC1P+L(3,J)*P1P(3)                            TR2
C   EQN(14,15)                                                    TR2
      NLC1=DASIN(PC1(3)/B)                                        TR2
      WLC1=WLON(PC1(2),PC1(1))                                    TR2
C   EQN(17)                                                       TR2
      BETAC1=DASIN(DOT(PC1,B1)/(MB1*B))                           TR2
      BCBC1=B*DCOS(BETAC1)                                        TR2
 125  CALL  RIIP(R0,NLC1,WLC1,PLASD)                              TR2
      IF(SEARCH.AND.ASC.AND.RTARG.GT.BCBC1.AND.RTARG.LE.R0+HBE)GOTO 591  TR2
      NIP=.TRUE.                                                  TR2
      NOELAY=.FALSE.                                              TR2
      IF(FCE.GT.0.D0)GO TO 126                                    TR2
      FCE=1.D-2                                                   TR2
      NOELAY=.TRUE.                                               TR2
 126  RM=R0+HBE+HAE                                               TR2
      H=HAE                                                       TR2
      F=FREQ/FCE                                                  TR2
      Y3P=DSQRT(RM**2-P1P(3)**2)                                  TR2
      YBEP=DSQRT((R0+HBE)**2-P1P(3)**2)                           TR2
      DO 1265 J=1,3                                               TR2
 1265 PC1(J)=L(2,J)*Y3P+L(3,J)*P1P(3)                             TR2
      NLC1=  DASIN(PC1(3)/RM)                                     TR2
```

```
      WLC1=WLON(PC1(2),PC1(1))                                            TR2
      CALL  RIIP(R0,NLC1,WLC1,PLASD)                                      TR2
127   RM=HBE+HAE+R0                                                       TR2
      H=HAE                                                               TR2
      NOELAY=.FALSE.                                                      TR2
      IF(FCE.GT.0.D0)GO TO 129                                            TR2
      FCE=1.D-2                                                           TR2
      NOELAY=.TRUE.                                                       TR2
129   F=FREQ/FCE                                                          TR2
130   IF(.NOT.FLAG)GO TO 1305                                             TR2
      CBET1= DABS(Z10PP)/RM                                               TR2
      SBET1=DSQRT(1.D0-CBET1**2)                                          TR2
      Y3P=RM*SBET1                                                        TR2
      YBEP=(R0+HBE)*SBET1                                                 TR2
      GO TO 131                                                           TR2
1305  Y3P=DSQRT(RM**2-P1P(3)**2)                                          TR2
      YBEP=DSQRT((R0+HBE)**2-P1P(3)**2)                                   TR2
C    EQN(18)                                                              TR2
      CBET1=BCBC1/RM                                                      TR2
      SBET1=DSQRT(1.D0-CBET1*CBET1)                                       TR2
131   IF(F*SBET1.GT.1.D0) GO TO 135                                       TR2
      IF(FLAG) GO TO 134                                                  TR2
132   IF(FLAG2) GO TO 134                                                 TR2
1325  IREFL=1                                                             TR2
      RM=R0+HBE+HAE                                                       TR2
      MR3=A1MAG                                                           TR2
      SINO2=DOT(A1,B1)/(MB1*A1MAG)                                        TR2
      COSO2=DSQRT(1.D0-SINO2**2)                                          TR2
      Y9P=P1P(2)                                                          TR2
      Z9P=P1P(3)                                                          TR2
      C22P=L(2,1)*B1(1)+L(2,2)*B1(2)+L(2,3)*B1(3)                         TR2
      C23P=L(3,1)*B1(1)+L(3,2)*B1(2)+L(3,3)*B1(3)                         TR2
      Y10P=COSO2*(Y9P*COSO2+Z9P*SINO2)                                    TR2
      Z10P=COSO2*(Z9P*COSO2-Y9P*SINO2)                                    TR2
      GO TO 3096                                                          TR2
  134 FLAG=.FALSE.                                                        TR2
      ISTOP=11                                                            TR2
      GO TO 7310                                                          TR2
135   IF(.NOT.NOELAY) GO TO 137                                           TR2
      Y4P=Y3P                                                             TR2
      DGP1=0.D0                                                           TR2
      Z4P=P1P(3)                                                          TR2
      GO TO 139                                                           TR2
137   DGP1=-H/SBET1+H*F/2.D0*DLOG((F*SBET1+1.D0)/(F*SBET1-1.D0))          TR2
      GAM1=CBET1*DGP1/RM                                                  TR2
      Y4P=Y3P*DCOS(GAM1)-P1P(3)*DSIN(GAM1)                                TR2
      Z4P=P1P(3)*DCOS(GAM1)+Y3P*DSIN(GAM1)                                TR2
C    DIST BETWEEN SUCCESSIVE R<S                                          TR2
139   DISTB=DSQRT((Y4P-Y5P)**2+(Z4P-Z5P)**2)                              TR2
      IF(FLAG) GO TO 141                                                  TR2
      DO 140 J=1,3                                                        TR2
140   P4(J)=L(2,J)*Y4P+L(3,J)*Z4P                                         TR2
      GO TO 1418                                                          TR2
141   DO 1413 J=1,3                                                       TR2
1413  P4(J)=M(2,J)*Y4P+M(3,J)*Z4P+P0(J)                                   TR2
1418  IF(DISTB.LE.EPS1) GO TO 200                                         TR2
      IF(INT.LE.1.OR.DISTB.LT.DISTA) GO TO 145                            TR2
```

```
      OUT2=DISTA                                                        TR2
      IF(IWRITE.EQ.1) WRITE(6,142) OUT2,IHOPS                           TR2
142   FORMAT(24H E-LAYER CONV. PROBLEMS,,F6.1,17H KM USED, HOP NO.,I3)  TR2
      H=HP                                                              TR2
      Y3P=Y3PP                                                          TR2
      RM=RMP                                                            TR2
      DGP1=GPP                                                          TR2
      F=FP                                                              TR2
      GO TO 215                                                         TR2
145   NL4=DASIN(P4(3)/MAG(P4))                                          TR2
      WL4=WLON(P4(2),P4(1))                                             TR2
      DISTA=DISTB                                                       TR2
      GPP=DGP1                                                          TR2
      Y5P=Y4P                                                           TR2
      Z5P=Z4P                                                           TR2
      HP=H                                                              TR2
      FP=F                                                              TR2
      Y3PP=Y3P                                                          TR2
      RMP=RM                                                            TR2
      SINO1=SBET1                                                       TR2
      COSO1=CBET1                                                       TR2
      DO 148 J=1,3                                                      TR2
148   R2(J)=P4(J)                                                       TR2
      CALL  RIIP(R0,NL4,WL4,PLASD)                                      TR2
      NOELAY=.FALSE.                                                    TR2
      IF(FCE.GT.0.D0) GO TO 149                                         TR2
      FCE=1.D-2                                                         TR2
      NOELAY=.TRUE.                                                     TR2
149   RM=HBE+HAE+R0                                                     TR2
      H=HAE                                                             TR2
      F=FREQ/FCE                                                        TR2
      INT=INT+1                                                         TR2
      IF(INT.LE.LIMEA) GO TO 130                                        TR2
      ISTOP=12                                                          TR2
      GO TO 7310                                                        TR2
C    END OF E LAYER REFRACTION                                          TR2
C    START OF F1 LAYER REFRACTION                                       TR2
200   Y5P=Y4P                                                           TR2
      Z5P=Z4P                                                           TR2
      SINO1=SBET1                                                       TR2
      COSO1=CBET1                                                       TR2
      DO 210 J=1,3                                                      TR2
210   R2(J)=P4(J)                                                       TR2
215   MR2=RM                                                            TR2
      RR1=MR2-H                                                         TR2
      SINO1P=SINO1                                                      TR2
      COSO1P=COSO1                                                      TR2
      Y6P=COSO1*(Y5P*COSO1+Z5P*SINO1)                                   TR2
      Z6P=COSO1*(Z5P*COSO1-Y5P*SINO1)                                   TR2
      IF(.NOT.FLAG) GO TO 2155                                          TR2
      DO 2151 J=1,3                                                     TR2
2151  P4(J)=M(2,J)*Y3P+M(3,J)*Z10PP                                     TR2
      RR1=-H/MAG(P4)                                                    TR2
      DO 2152 J=1,3                                                     TR2
2152  P4(J)=R2(J)+RR1*P4(J)                                             TR2
      RR1=MAG(P4)                                                       TR2
      MR2=MAG(R2)                                                       TR2
```

```
      COSO1P=BCBC1/MR2                                                  TR2
      SINO1P=DSQRT(1.D0-COSO1P**2)                                      TR2
      DO 2153 J=1,3                                                     TR2
      P4(J)=M(2,J)*Y6P+M(3,J)*Z6P+P0(J)                                 TR2
 2153 P8(J)=R2(J)-P4(J)                                                 TR2
      CALL  CROSS(R2,P8,XI)                                             TR2
      CALL  CROSS(XI,P8,KP)                                             TR2
      CNST=MAG(XI)                                                      TR2
      FLOG=MAG(P8)                                                      TR2
      DO 2154 J=1,3                                                     TR2
      L(1,J)=XI(J)/CNST                                                 TR2
      L(2,J)=P8(J)/FLOG                                                 TR2
 2154 L(3,J)=KP(J)/(CNST*FLOG)                                          TR2
      Y5P=L(2,1)*R2(1)+L(2,2)*R2(2)+L(2,3)*R2(3)                        TR2
      Z5P=L(3,1)*R2(1)+L(3,2)*R2(2)+L(3,3)*R2(3)                        TR2
      Y6P=L(2,1)*P4(1)+L(2,2)*P4(2)+L(2,3)*P4(3)                        TR2
      Z6P=L(3,1)*P4(1)+L(3,2)*P4(2)+L(3,3)*P4(3)                        TR2
      FLAG=.FALSE.                                                      TR2
 2155 INT=0                                                             TR2
      FLAG3=.FALSE.                                                     TR2
      HEA=H                                                             TR2
      HE=H                                                              TR2
      PPEF1A=0.D0                                                       TR2
      IF(.NOT.NOELAY)PPEF1A=H*(.5D0*SINO1-1.D0/SINO1+F/4.D0*(1.D0-F**(-2 TR2
     *)+COSO1**2)*DLOG((F*SINO1+1.D0)/(F*SINO1-1.D0)))                  TR2
      DPP1=PPEF1A                                                       TR2
      PP0=RR1*DSQRT(1.D0-(COSO1P*MR2/RR1)**2)                           TR2
      TAU15A=MR2*SINO1P-PP0                                             TR2
      IF(.NOT.HITS.OR.COSO1P*MR2/A1MAG.GE.1.D0) GO TO 217               TR2
      PP0=PP0-A1MAG*DSQRT(1.D0-(COSO1P*MR2/A1MAG)**2)                   TR2
 217  C12P=Y5P-Y6P                                                      TR2
      C13P=Z5P-Z6P                                                      TR2
      MC1PSQ=C12P**2+C13P**2                                            TR2
      MRX1SQ=BCBC1**2                                                   TR2
 218  RM=R0+HBE+HAE+HAF1                                                TR2
      TC2=DSQRT((RM**2-MRX1SQ)/MC1PSQ)                                  TR2
      IF(SEARCH.AND.RTARG.GT.DSQRT(MRX1SQ).AND.RTARG.LE.RR1) GO TO 591  TR2
      YC2P=C12P*TC2+Y6P                                                 TR2
      ZC2P=C13P*TC2+Z6P                                                 TR2
      DO 220 J=1,3                                                      TR2
 220  PC2(J)=L(2,J)*YC2P+L(3,J)*ZC2P                                    TR2
      NLC2=  DASIN(PC2(3)/RM)                                           TR2
      WLC2=WLON(PC2(2),PC2(1))                                          TR2
      CALL  PIIF(R0,NLC2,WLC2,PLASD)                                    TR2
 225  RM=R0+HBE+HAE+HAF1                                                TR2
      H=HAF1                                                            TR2
      F=FREQ/FCF1                                                       TR2
      Y9P=0.D0                                                          TR2
      Z9P=0.D0                                                          TR2
 230  IF(.NOT.FLAG) GO TO 2305                                          TR2
      CBET4= DABS(Z10PP)/RM                                             TR2
      SBET4=DSQRT(1.D0-CBET4**2)                                        TR2
      Y7P=RM*SBET4                                                      TR2
      GO TO 231                                                         TR2
 2305 TDEL3=DSQRT((RM**2-MRX1SQ)/MC1PSQ)                                TR2
      Y7P=C12P*TDEL3+Y6P                                                TR2
      Z7P=C13P*TDEL3+Z6P                                                TR2
```

```
      MR3=RM                                                          TR2
      CBET4=BCBC1/MR3                                                 TR2
      SBET4=DSQRT(1.D0-CBET4*CBET4)                                   TR2
231   IF(F*SBET4.GT.1.D0) GO TO 235                                   TR2
      IF(FLAG) GO TO 234                                              TR2
232   IF(FLAG3) GO TO 234                                             TR2
2325  IREFL=2                                                         TR2
      RM=R0+HBE+HAE+HAF1                                              TR2
      Y9P=Y5P                                                         TR2
      Z9P=Z5P                                                         TR2
      MR3=MR2                                                         TR2
      COSO2=COSO1                                                     TR2
      SINO2=SINO1                                                     TR2
      Y10P=COSO2*(Y9P*COSO2+Z9P*SINO2)                                TR2
      Z10P=COSO2*(Z9P*COSO2-Y9P*SINO2)                                TR2
      GO TO 3095                                                      TR2
  234 FLAG=.FALSE.                                                    TR2
      ISTOP=13                                                        TR2
      GO TO 7310                                                      TR2
235   DGP2=H*(F/2.D0*DLOG((F*SBET4+1.D0)/(F*SBET4-1.D0))-1.D0/SBET4)  TR2
      GAM2=CBET4*DGP2/RM                                              TR2
      Y8P=Y7P*DCOS(GAM2)-Z7P*DSIN(GAM2)                               TR2
      Z8P=Z7P*DCOS(GAM2)+Y7P*DSIN(GAM2)                               TR2
C    DIST BETWEEN SUCCESSIVE R<S                                      TR2
      DISTB=DSQRT((Y8P-Y9P)**2+(Z8P-Z9P)**2)                          TR2
      IF(FLAG) GO TO 241                                              TR2
      DO 240 J=1,3                                                    TR2
240   P8(J)=L(2,J)*Y8P+L(3,J)*Z8P                                     TR2
      GO TO 2425                                                      TR2
241   DO 242 J=1,3                                                    TR2
242   P8(J)=M(2,J)*Y8P+M(3,J)*Z8P+P0(J)                               TR2
2425  IF(DISTB.LE.EPS2) GO TO 300                                     TR2
      IF(INT.LE.1.OR.DISTB.LT.DISTA)GO TO 245                         TR2
      OUT2=DISTA                                                      TR2
      IF(IWRITE.EQ.1) WRITE(6,243)OUT2,IHOPS                          TR2
243   FORMAT(25H F1-LAYER CONV. PROBLEMS,,F6.1,17H KM USED, HOP NO.,I3) TR2
      H=HP                                                            TR2
      F=FP                                                            TR2
      Y7P=Y7PP                                                        TR2
      RM=RMP                                                          TR2
      DGP2=GPP                                                        TR2
      GO TO 307                                                       TR2
245   NL8=  DASIN(P8(3)/MAG(P8))                                      TR2
      WL8=WLON(P8(2),P8(1))                                           TR2
      Y9P=Y8P                                                         TR2
      Z9P=Z8P                                                         TR2
      HP=H                                                            TR2
      FP=F                                                            TR2
      Y7PP=Y7P                                                        TR2
      RMP=RM                                                          TR2
      GPP=DGP2                                                        TR2
      COSO2=CBET4                                                     TR2
      SINO2=SBET4                                                     TR2
      DO 247 J=1,3                                                    TR2
247   R3(J)=P8(J)                                                     TR2
      DISTA=DISTB                                                     TR2
      CALL  RIIP(R0,NL8,WL8,PLASD)                                    TR2
```

```
      RM=R0+HBE+HAE+HAF1                                              TR2
      H=HAF1                                                          TR2
      F=FREQ/FCF1                                                     TR2
      INT=INT+1                                                       TR2
      IF(INT.LE.LIMF1A) GO TO 230                                     TR2
      ISTOP=14                                                        TR2
      GO TO 7310                                                      TR2
C   END OF F1 LAYER REFRACTION                                        TR2
300   Y9P=Y8P                                                         TR2
      Z9P=Z8P                                                         TR2
      SINO2=SBET4                                                     TR2
      COSO2=CBET4                                                     TR2
      DO 305 J=1,3                                                    TR2
305   R3(J)=P8(J)                                                     TR2
307   MR3=RM                                                          TR2
      IREFL=3                                                         TR2
      RM=R0+HMF2                                                      TR2
      RR2=MR3-H                                                       TR2
      Y10P=COSO2*(Y9P*COSO2+Z9P*SINO2)                                TR2
      Z10P=COSO2*(Z9P*COSO2-Y9P*SINO2)                                TR2
      COSO2P=COSO2                                                    TR2
      SINO2P=SINO2                                                    TR2
      IF(.NOT.FLAG) GO TO 3075                                        TR2
      DO 3071 J=1,3                                                   TR2
3071  P8(J)=M(2,J)*Y7P+M(3,J)*Z10P+P0(J)                              TR2
      RR2=-H/MAG(P8)                                                  TR2
      DO 3072 J=1,3                                                   TR2
3072  P8(J)=R3(J)+RR2*P8(J)                                           TR2
      RR2=MAG(P8)                                                     TR2
      MR3=MAG(R3)                                                     TR2
      COSO2P=BCRC1/MR3                                                TR2
      SINO2P=DSQRT(1.D0-COSO2P**2)                                    TR2
      DO 3073 J=1,3                                                   TR2
      P8(J)=M(2,J)*Y10P+M(3,J)*Z10P+P0(J)                             TR2
3073  P4(J)=R3(J)-P8(J)                                               TR2
      CALL  CROSS(R3,P4,XI)                                           TR2
      CALL  CROSS(XI,P4,KP)                                           TR2
      CNST=MAG(XI)                                                    TR2
      FLOG=MAG(P4)                                                    TR2
      DO 3074 J=1,3                                                   TR2
      L(1,J)=XI(J)/CNST                                               TR2
      L(2,J)=P4(J)/FLOG                                               TR2
3074  L(3,J)=KP(J)/(CNST*FLOG)                                        TR2
      Y9P=L(2,1)*R3(1)+L(2,2)*R3(2)+L(2,3)*R3(3)                      TR2
      Z9P=L(3,1)*R3(1)+L(3,2)*R3(2)+L(3,3)*R3(3)                      TR2
      Y10P=L(2,1)*P8(1)+L(2,2)*P8(2)+L(2,3)*P8(3)                     TR2
      Z10P=L(3,1)*P8(1)+L(3,2)*P8(2)+L(3,3)*P8(3)                     TR2
      FLAG=.FALSE.                                                    TR2
3075  DO 308 J=1,3                                                    TR2
308   PREV(J)=0.D0                                                    TR2
      IF(.NOT.SEARCH) GO TO 309                                       TR2
      IF(RTARG.GT.RR1.AND.RTARG.LE.MR2) GO TO 593                     TR2
      IF(IREFL.EQ.1) GO TO 309                                        TR2
      IF(MR2.LE.RR2.AND.RTARG.LE.RR2.AND.RTARG.GT.MR2) GO TO 593      TR2
      IF(RR1.GT.MR3.AND.RTARG.LE.RR1.AND.RTARG.GT.MR3) GO TO 593      TR2
C   TILTED F2 LAYER                                                   TR2
309   HF1A=H                                                          TR2
```

```
      HF1=H                                                             TR2
      TAU=MR3*SINO2P-RR2*DSQRT(1.D0-(MR3/RR2*COSO2P)**2)+TAU1EA         TR2
      PPEF1A=PPEF1A+H*(.5D0*SINO2+F/4.D0*(1.D0-F**(-2)+COSO2**2)*DLOG((F TR2
     **SINO2+1.D0)/(F*SINO2-1.D0))-1.D0/SINO2)+TAU                      TR2
      GPEF1A=DGP1+DGP2+TAU                                              TR2
 3095 C22P=Y9P-Y10P                                                     TR2
      C23P=Z9P-Z10P                                                     TR2
 3096 TC3=(RM**2-Y10P**2-Z10P**2)/(C22P**2+C23P**2)                     TR2
      IF(TC3.LT.0.D0)GO TO (134,134,234),IREFL                          TR2
      TC3=DSQRT(TC3)                                                    TR2
      YC3P=C22P*TC3+Y10P                                                TR2
      ZC3P=C23P*TC3+Z10P                                                TR2
      DO 310 J=1,3                                                      TR2
      PC3(J)=L(2,J)*YC3P+L(3,J)*ZC3P                                    TR2
      P10(J)=L(2,J)*Y10P+L(3,J)*Z10P                                    TR2
 310  C2(J)=L(2,J)*C22P+L(3,J)*C23P                                     TR2
      BETA22=0.D0                                                       TR2
      NLC3=DASIN(PC3(3)/RM)                                             TR2
      WLC3=WLON(PC3(2),PC3(1))                                          TR2
      NDFLAG=0                                                          TR2
      CALL HTGRDF(NDFLAG,FREQ,RTD*NLC3,RTD*WLC3,RTD*DACOS(BCBC1/RM),0.  TR2
     *D0,RM,RM1,RT1,RTM1,DNDR,DNDNL,DNDWL,PNTFLG)                       TR2
      NDFLAG=1                                                          TR2
      GO TO (311,312,313),IREFL                                         TR2
 311  F1=FREQ/FCE                                                       TR2
      H1=HAE                                                            TR2
      IF(NUSE.EQ.1) GO TO 314                                           TR2
      FLAG2=.TRUE.                                                      TR2
      INT=0                                                             TR2
      GO TO 127                                                         TR2
 312  F1=FREQ/FCF1*.86602540378443900                                   TR2
      RM1=RM1+HAF1                                                      TR2
      H1=HAF1*2.D0                                                      TR2
      GO TO (3123,314,3125),NUSE                                        TR2
 3123 IF(FLAG2) GO TO 134                                               TR2
      FLAG2=.TRUE.                                                      TR2
      GO TO 1325                                                        TR2
 3125 FLAG3=.TRUE.                                                      TR2
      INT=0                                                             TR2
      GO TO 225                                                         TR2
 313  F1=FREQ/FCF2                                                      TR2
      H1=HAF2                                                           TR2
      IF(NUSE.EQ.3) GO TO 314                                           TR2
      IF(FLAG3) GO TO 234                                               TR2
      FLAG3=.TRUE.                                                      TR2
      GO TO 2325                                                        TR2
 314  INTF2=0                                                           TR2
      IRCYCL=0                                                          TR2
 315  COPHC3=PC3(3)/RM                                                  TR2
      SIPHC3=DSQRT(1.D0-COPHC3**2)                                      TR2
      NNUSE(INUSE)=NUSE                                                 TR2
      FCREF(INUSE)=FREQ/F1                                              TR2
C     CORRECTION IN F1 CRITICAL PRINT OUT                               TR2
      IF(NUSE.EQ.2) FCREF(INUSE)=FCF1                                   TR2
      COTHC3=PC3(1)/DSQRT(PC3(1)**2+PC3(2)**2)                          TR2
      SITHC3=PC3(2)/DSQRT(PC3(1)**2+PC3(2)**2)                          TR2
      PR1(1)=RTM1*COTHC3*SIPHC3                                         TR2
```

```
      PR1(2)=RTM1*SITHC3*SIPHC3                                         TR2
      PR1(3)=RTM1*COPHC3                                                TR2
      N1(1)=COTHC3*SIPHC3*DNDR-COTHC3*COPHC3*DNDNL/RTM1+SITHC3*DNDHL/(RT TR2
     *M1*SIPHC3)                                                        TR2
      N1(2)=SITHC3*SIPHC3*DNDR-SITHC3*COPHC3*DNDNL/RTM1-COTHC3*DNDHL/(RT TR2
     *M1*SIPHC3)                                                        TR2
      N1(3)=COPHC3*DNDR+SIPHC3*DNDNL/RTM1                               TR2
      MN1=MAG(N1)                                                       TR2
      T0=-RTM1/MN1                                                      TR2
      ISTOP=1                                                           TR2
      DO 320 J=1,3                                                      TR2
      P0(J)=N1(J)*T0+PR1(J)                                             TR2
  320 A2(J)=P10(J)-P0(J)                                                TR2
      CALL  CROSS(A2,C2,XI)                                             TR2
      CALL  CROSS(XI,C2,KP)                                             TR2
      MA2C2=MAG(XI)                                                     TR2
      MC2=MAG(C2)                                                       TR2
      DO 330 J=1,3                                                      TR2
      M(1,J)=XI(J)/MA2C2                                                TR2
      M(2,J)=C2(J)/MC2                                                  TR2
  330 M(3,J)=KP(J)/(MA2C2*MC2)                                          TR2
      Z10PP=M(3,1)*A2(1)+M(3,2)*A2(2)+M(3,3)*A2(3)                      TR2
      RB=RM1-H1                                                         TR2
      IF(RB.GE. DABS(Z10PP))GO TO 332                                   TR2
      ISTOP=2                                                           TR2
      GO TO 3345                                                        TR2
  332 COBETB= DABS(Z10PP)/RB                                            TR2
      SIBETB=DSQRT(1.D0-COBETB**2)                                      TR2
      YS1PP=RB*SIBETB                                                   TR2
      DO 333 J=1,3                                                      TR2
  333 S1(J)=M(2,J)*YS1PP+M(3,J)*Z10PP+P0(J)                             TR2
      R4=MAG(S1)                                                        TR2
      PP2=1.D0-(MR3/R4*COSO2)**2                                        TR2
      IF(PP2.GE.0.D0) GO TO 3331                                        TR2
      ISTOP=2                                                           TR2
      GO TO 3345                                                        TR2
 3331 PP2=R4*DSQRT(PP2)-MR3*SINO2                                       TR2
      PP2A=PP2                                                          TR2
      IF(IREFL.GT.1) GO TO 3336                                         TR2
      IF(HITS) GO TO 3334                                               TR2
      PP0=R4*DSQRT(1.D0-(MR3/R4*COSO2)**2)                              TR2
      GO TO 3336                                                        TR2
 3334 PP0=PP2                                                           TR2
 3336 CBTTM1=RB*COBETB/RTM1                                             TR2
      IF(CBTTM1.LE.1.D0)GO TO 334                                       TR2
      ISTOP=3                                                           TR2
      GO TO 3345                                                        TR2
  334 SBTTM1=DSQRT(1.D0-CBTTM1**2)                                      TR2
      FLAG=.FALSE.                                                      TR2
      IF(1.D0-F1*SBTTM1.GT.0.D0) GO TO 335                              TR2
      FLAG=.TRUE.                                                       TR2
 3345 THETA=1.D0/RTM1                                                   TR2
      DIST1=1.D10                                                       TR2
      IRCYCL=0                                                          TR2
      GO TO 336                                                         TR2
  335 FLOG=H1*F1*DLOG((1.D0+F1*SBTTM1)/(1.D0-F1*SBTTM1))/2.D0           TR2
      GPF2=FLOG*2.D0                                                    TR2
```

```
      THETA=CBTTM1*FLOG/RTM1                                        TR2
      PPF2=H1*SBTTM1+(1.D0-F1**(-2)+CBTTM1**2)*FLOG                 TR2
      IRCYCL=IRCYCL+1                                               TR2
 336  YS2PP=YS1PP*DCOS(THETA)-Z10PP*DSIN(THETA)                     TR2
      ZS2PP=Z10PP*DCOS(THETA)+YS1PP*DSIN(THETA)                     TR2
      IF(FLAG.OR.ISTOP.GT.1) GO TO 475                              TR2
      YS3PP=YS1PP*DCOS(2.D0*THETA)-Z10PP*DSIN(2.D0*THETA)           TR2
      ZS3PP=Z10PP*DCOS(2.D0*THETA)+YS1PP*DSIN(2.D0*THETA)           TR2
      ALPHAV=PI/2.D0-THETA-DASIN(SIBETB)                            TR2
      RV=RB*COBETB/DSIN(ALPHAV)                                     TR2
      YVPP=YS2PP*RV/RB                                              TR2
      C32PP=YS3PP-YVPP                                              TR2
      C33PP=ZS3PP-Z10PP                                             TR2
      DO 340 J=1,3                                                  TR2
      C3(J)=M(2,J)*C32PP+M(3,J)*C33PP                               TR2
 340  A3(J)=M(2,J)*YS3PP+M(3,J)*ZS3PP+P0(J)                         TR2
      CALL  CROSS(A3,C3,PHI)                                        TR2
      CALL  CROSS(PHI,C3,KPPP)                                      TR2
      MA3C3=MAG(PHI)                                                TR2
      MC3=MAG(C3)                                                   TR2
      C32PPP=MC3                                                    TR2
      DO 350 J=1,3                                                  TR2
      P(1,J)=PHI(J)/MA3C3                                           TR2
      P(2,J)=C3(J)/MC3                                              TR2
 350  P(3,J)=KPPP(J)/(MA3C3*MC3)                                    TR2
C    F1 LAYER DOWNWARD                                              TR2
      YS3PPP=P(2,1)*A3(1)+P(2,2)*A3(2)+P(2,3)*A3(3)                 TR2
      IF(YS3PPP.GE.0.D0) GO TO 3555                                 TR2
      ZS3PPP=P(3,1)*A3(1)+P(3,2)*A3(2)+P(3,3)*A3(3)                 TR2
      INT=0                                                         TR2
      GO TO(436,3987,353),IREFL                                     TR2
 353  IF(MR3.GE.DABS(ZS3PPP)) GO TO 360                             TR2
      IF(B.GE. DABS(ZS3PPP)) GO TO 356                              TR2
 3555 ISTOP=3                                                       TR2
      IRCYCL=0                                                      TR2
      GO TO 475                                                     TR2
 356  YDPPP=-DSQRT(B**2-ZS3PPP**2)                                  TR2
      GO TO 370                                                     TR2
 360  YDPPP=-DSQRT(MR3**2-ZS3PPP**2)                                TR2
C    F1 LAYER ITERATION 370 TO 390                                  TR2
 370  DO 380 J=1,3                                                  TR2
 380  PD(J)=P(2,J)*YDPPP+P(3,J)*ZS3PPP                              TR2
      NLD=DASIN(PD(3)/MAG(PD))                                      TR2
      WLD=WLON(PD(2),PD(1))                                         TR2
      CALL  RIIP(R0,NLD,WLD,PLASD)                                  TR2
      RM=HBE+HAE+HAF1+R0                                            TR2
      H=HAF1                                                        TR2
      F=FREQ/FCF1                                                   TR2
      IF(RM.GE.DABS(ZS3PPP)) GO TO 382                              TR2
      ISTOP=3                                                       TR2
      IRCYCL=0                                                      TR2
      GO TO 475                                                     TR2
 382  YDPPP2=-DSQRT(RM**2-ZS3PPP**2)                                TR2
      DISTB=DABS(YDPPP2-YDPPP)                                      TR2
      IF(DISTB.LE.EPS3) GO TO 390                                   TR2
      IF(INT.LE.1.OR.DISTB.LT.DISTA) GO TO 384                      TR2
      H=HP                                                          TR2
```

```
      F=FP                                                              TR2
      RM=RMP                                                            TR2
      YDPPP2=YDPPP                                                      TR2
      OUT2=DISTA                                                        TR2
      IF(IWRITE.EQ.1)WRITE(6,243)OUT2,IHOPS                             TR2
      GO TO 390                                                         TR2
 384  YDPPP=YDPPP2                                                      TR2
      DISTA=DISTB                                                       TR2
      RMP=RM                                                            TR2
      FP=F                                                              TR2
      HP=H                                                              TR2
      INT=INT+1                                                         TR2
      IF(INT.LE.LIMF1D) GO TO 370                                       TR2
      ISTOP=4                                                           TR2
      IRCYCL=0                                                          TR2
      GO TO 475                                                         TR2
C    E LAYER DOWNWARD                                                   TR2
 390  YM1PPP=YDPPP2                                                     TR2
      HF1=H                                                             TR2
      MRM1=DSQRT(YM1PPP**2+ZS3PPP**2)                                   TR2
      DO 395 J=1,3                                                      TR2
 395  RN1(J)=P(2,J)*YM1PPP+P(3,J)*ZS3PPP                                TR2
      SBETM1=DABS(DOT(RN1,C3)/(MRM1*MC3))                               TR2
      CBETM1=DSQRT(1.D0-SBETM1*SBETM1)                                  TR2
      R4=MAG(A3)                                                        TR2
      PP2=PP2+R4*DSQRT(1.D0-(MRM1/R4*CBETM1)**2)-MRM1*SBETM1            TR2
      IF(F*SBETM1.GT.1.D0)GO TO 398                                     TR2
      ISTOP=5                                                           TR2
      IRCYCL=0                                                          TR2
      GO TO 475                                                         TR2
 398  FLOG=DLOG((F*SBETM1+1.D0)/(F*SBETM1-1.D0))                        TR2
      DGP2P=H*(F/2.D0*FLOG-1.D0/SBETM1)                                 TR2
      TH12=CBETM1*DGP2P/RM                                              TR2
      PPEF1D=H*(.5D0*SBETM1+F/4.D0*(1.D0-F**(-2)+CBETM1**2)*FLOG-1.D0/SB TR2
     *ETM1)                                                             TR2
      DPP2P=PPEF1D                                                      TR2
      RR2=MRM1*SBETM1                                                   TR2
      RR2C=MRM1*CBETM1                                                  TR2
      YM2PPP=YM1PPP*DCOS(TH12)-ZS3PPP*DSIN(TH12)                        TR2
      ZM2PPP=ZS3PPP*DCOS(TH12)+YM1PPP*DSIN(TH12)                        TR2
      MRM2=DSQRT(YM2PPP**2+ZM2PPP**2)                                   TR2
      YY1PPP=CBETM1*(YM2PPP*CBETM1-ZM2PPP*SBETM1)                       TR2
      ZY1PPP=CBETM1*(ZM2PPP*CBETM1+YM2PPP*SBETM1)                       TR2
      C42PPP=YY1PPP-YM2PPP                                              TR2
      C43PPP=ZY1PPP-ZM2PPP                                              TR2
      MRY1SQ=YY1PPP**2+ZY1PPP**2                                        TR2
      MC4SQ=C42PPP**2+C43PPP**2                                         TR2
 3984 IF(MR2**2.GE.MRY1SQ)GO TO 399                                     TR2
      IF(B**2.GE.MRY1SQ) GO TO 3986                                     TR2
      ISTOP=6                                                           TR2
      IRCYCL=0                                                          TR2
      GO TO 475                                                         TR2
 3986 TE=-DSQRT((B**2-MRY1SQ)/MC4SQ)                                    TR2
      RM=B                                                              TR2
      GO TO 3995                                                        TR2
 3987 CNST=-DOT(A3,C3)/MC3**2                                           TR2
      C42PPP=P(2,1)*C3(1)+P(2,2)*C3(2)+P(2,3)*C3(3)                     TR2
```

```
      C43PPP=P(3,1)*C3(1)+P(3,2)*C3(2)+P(3,3)*C3(3)
      YY1PPP=YS3PPP+CNST*C42PPP
      ZY1PPP=ZS3PPP+CNST*C43PPP
      MRY1SQ=YY1PPP**2+ZY1PPP**2
      MC4SQ=MC3**2
      MRM1=DSQRT(YS3PPP**2+ZS3PPP**2)
      CBETM1=DSQRT(MRY1SQ)/MRM1
      SBETM1=DSQRT(1.D0-CBETM1**2)
      GO TO 3984
399   TE=-DSQRT((MR2**2-MRY1SQ)/MC4SQ)
3995  YEPPP=C42PPP*TE+YY1PPP
      ZEPPP=C43PPP*TE+ZY1PPP
C     E LAYER ITERATION 400 TO 420
      INT=0
400   DO 410 J=1,3
410   PE(J)=P(2,J)*YEPPP+P(3,J)*ZEPPP
      NLE=DASIN(PE(3)/MAG(PE))
      WLE=WLON(PE(2),PE(1))
      CALL  RIIP(RO,NLE,WLE,PLASD)
      NOELAY=.FALSE.
      IF(FCE.GT.0.D0) GO TO 4105
      FCE=1.D-2
      NOELAY=.TRUE.
4105  RM=HBE+HAE+RO
      H=HAE
      F=FREQ/FCE
      IF(RM**2.GE.MRY1SQ)  GO TO 411
      ISTOP=6
      IRCYCL=0
      GO TO 475
411   TE=-DSQRT((RM**2-MRY1SQ)/MC4SQ)
      YEPPP2=C42PPP*TE+YY1PPP
      ZEPPP2=C43PPP*TE+ZY1PPP
      DISTB=DSQRT((YEPPP-YEPPP2)**2+(ZEPPP-ZEPPP2)**2)
      IF(DISTB.LE.EPS4)  GO TO 420
      IF(INT.LE.1.OR.DISTB.LT.DISTA)  GO TO 414
      H=HP
      F=FP
      RM=RMP
      YEPPP2=YEPPP
      ZEPPP2=ZEPPP
      OUT2=DISTA
      IF(IWRITE.EQ.1)WRITE(6,142)OUT2,IHOPS
      GO TO 420
414   YEPPP=YEPPP2
      ZEPPP=ZEPPP2
      RMP=RM
      FP=F
      HP=H
      DISTA=DISTB
      INT=INT+1
      IF(INT.LE.LIMED)  GO TO 400
      ISTOP=7
      IRCYCL=0
      GO TO 475
420   YM3PPP=YEPPP2
      ZM3PPP=ZEPPP2
```

```
      HE=H                                                          TR2
C    END OF E LAYER                                                 TR2
      MRM3=DSQRT(YM3PPP**2+ZM3PPP**2)                               TR2
      CBETM2=MRM1*CBETM1/MRM3                                       TR2
      SBETM2=DSQRT(1.D0-CBETM2*CBETM2)                              TR2
      IF(IREFL.EQ.2) PP2=PP2+MRM1*SBETM1-MRM3*SBETM2                TR2
      IF(F*SBETM2.GT.1.D0) GO TO 430                                TR2
      ISTOP=8                                                       TR2
      IRCYCL=0                                                      TR2
      GO TO 475                                                     TR2
430   IF(.NOT.NOELAY) GO TO 4302                                    TR2
      FLOG=0.D0                                                     TR2
      DPP1P=0.D0                                                    TR2
      TH34=0.D0                                                     TR2
      DGP1P=0.D0                                                    TR2
      YM4PPP=YM3PPP                                                 TR2
      ZM4PPP=ZM3PPP                                                 TR2
      MRM4=MRM3                                                     TR2
      GO TO 4304                                                    TR2
4302  FLOG=DLOG((F*SBETM2+1.D0)/(F*SBETM2-1.D0))                    TR2
      DGP1P=H*(F/2.D0*FLOG-1.D0/SBETM2)                             TR2
      TH34=CBETM2*DGP1P/RM                                          TR2
      YM4PPP=YM3PPP*DCOS(TH34)-ZM3PPP*DSIN(TH34)                    TR2
      ZM4PPP=ZM3PPP*DCOS(TH34)+YM3PPP*DSIN(TH34)                    TR2
      MRM4=DSQRT(YM4PPP**2+ZM4PPP**2)                               TR2
      DPP1P=H*(.5D0*SBETM2+F/4.D0*(1.D0-F**(-2)+CBETM2**2)*FLOG-1.D0/SBE TR2
     *TM2)                                                          TR2
4304  RR1=MRM4-H                                                    TR2
      IF(IREFL.LT.3) GO TO 4305                                     TR2
      IF( DABS(RR2C).GT.RR1) GO TO 431                              TR2
      TAU=RR2-RR1*DSQRT(1.D0-(RR2C/RR1)**2)                         TR2
      PPEF1D=PPEF1D+DPP1P+TAU                                       TR2
      GPEF1D=DGP1P+DGP2P+TAU                                        TR2
4305  YY2PPP=CBETM2*(YM4PPP*CBETM2-ZM4PPP*SBETM2)                   TR2
      ZY2PPP=CBETM2*(ZM4PPP*CBETM2+YM4PPP*SBETM2)                   TR2
      C52PPP=YY2PPP-YM4PPP                                          TR2
      C53PPP=ZY2PPP-ZM4PPP                                          TR2
      MC5SQ=C52PPP**2+C53PPP**2                                     TR2
      SBM4H2=1.D0-(CBETM2/(1.D0-HE/MRM4))**2                        TR2
      IF(SBM4H2.GT.0.D0) GO TO 4307                                 TR2
      ISTOP=15                                                      TR2
      IRCYCL=0                                                      TR2
      GO TO 475                                                     TR2
4307  TAU1ED=MRM4*SBETM2-(MRM4-HE)*DSQRT(SBM4H2)                    TR2
      RY=DSQRT(MRY1SQ)                                              TR2
      GO TO 437                                                     TR2
431   ISTOP=9                                                       TR2
      IRCYCL=0                                                      TR2
      GO TO 475                                                     TR2
436   MB1=MAG(A3)                                                   TR2
      SBET1=-DOT(A3,C3)/(MB1*MC3)                                   TR2
      CBET1=DSQRT(1.D0-SBET1**2)                                    TR2
      RY=MB1*CBET1                                                  TR
      MRY1SQ=RY**2                                                  TR
      CNST=MB1/MC3*SBET1                                            TR
      C52PPP=P(2,1)*C3(1)+P(2,2)*C3(2)+P(2,3)*C3(3)                 TR
      C53PPP=P(3,1)*C3(1)+P(3,2)*C3(2)+P(3,3)*C3(3)                 TR
```

```
      YY2PPP=YS3PPP+CNST*C52PPP                                          TR
      ZY2PPP=ZS3PPP+CNST*C53PPP                                          TR
      MC5SQ=MC3**2                                                       TR
437   HITS=R0.GT.RY                                                      TR
      INCHOP=(R0+50.D0).GE.RY                                            TR
      IF(R0+HBE.GT.RY) GO TO 438                                         TR
      ISTOP=15                                                           TR
      IRCYCL=0                                                           TR
      GO TO 475                                                          TR
438   TQ=-DSQRT(((R0+HBE)**2-MRY1SQ)/MC5SQ)                              TR
      YBEPPP=C52PPP*TQ+YY2PPP                                            TR
      ZBEPPP=C53PPP*TQ+ZY2PPP                                            TR
      IF(.NOT.HITS) GO TO 450                                            TR
      TQ=-DSQPT((R0**2-MRY1SQ)/MC5SQ)                                    TR
      YQPPP=C52PPP*TQ+YY2PPP                                             TR
      ZQPPP=C53PPP*TQ+ZY2PPP                                             TR
      DO 440 J=1,3                                                       TR
440   PQ(J)=P(2,J)*YQPPP+P(3,J)*ZQPPP                                    TR
      GO TO 461                                                          TR
450   DO 460 J=1,3                                                       TR
460   PQ(J)=P(2,J)*YY2PPP+P(3,J)*ZY2PPP                                  TR2
C   STARTS TO ITERATE TILTED F2                                          TR2
461   DIST2=DSQRT((PQ(1)-PREV(1))**2+(PQ(2)-PREV(2))**2+(PQ(3)-PREV      TR2
     *(3))**2)                                                           TR2
      RTDIF=DABS(RT1-PREV(13))                                           TR2
      IF(IRCYCL.LE.2) GO TO 475                                          TR2
      IF(DIST2-EPS5)463,463,462                                          TR2
462   IF(DIST1-DIST2)470,475,475                                         TR2
463   IF(RTDIF-EPS5)500,500,475                                          TR2
470   CALL PRVGET                                                        TR2
      ICNPRB=1                                                           TR2
      IF(IWRITE.EQ.1)WRITE(6,473) IHOPS,EPS5,DIST1,RTDIF                 TR2
473   FORMAT(48H TILTED LAYER CONVERGENCE PROBLEMS ON HOP NUMBER,I3,     TR2
     1 1H,,F6.1,29H KM. CRITERIA NOT MET, DIST1=,F8.2,8H, RTDIF=,        TR2
     2 F8.2)                                                             TR2
      GO TO 500                                                          TR2
475   YW2PP=RTM1*YS2PP/RB                                                TR2
      ZW2PP=RTM1*ZS2PP/RB                                                TR2
      DO 480 J=1,3                                                       TR2
480   PW2(J)=M(2,J)*YW2PP+M(3,J)*ZW2PP+P0(J)                             TR2
      NLW2=DASIN(PW2(3)/MAG(PW2))                                        TR2
      WLW2=WLON(PW2(2),PW2(1))                                           TR2
      DO 490 J=1,3                                                       TR2
490   W2(J)=PW2(J)-P0(J)                                                 TR2
      BETA22=DACOS(DOT(PW2,W2)/(MAG(W2)*MAG(PW2)))                       TR2
      BETTM1=DASIN(SBTTM1)                                               TR2
      CALL PRVSTO                                                        TR2
      DIST1=DIST2                                                        TR2
      BP=RTM1                                                            TR2
      CALL HTGROF(NDFLAG,FREQ,RTD*NLW2,RTD*WLW2,RTD*BETTM1,RTD*BETA22,B  TR2
     *P,RM1,RT1,RTM1,DNDR,DNDNL,DNDWL,CNTFLG)                            TR2
      IF(IREFL.NE.NUSE) GO TO 501                                        TR2
      GO TO(4916,4917,4918),IREFL                                        TR2
4916  F1=FREQ/FCE                                                        TR2
      H1=HAE                                                             TR2
      GO TO 4919                                                         TR2
4917  F1=FREQ/FCF1*.86602540378443900                                    TR2
```

```
      RM1=RM1+HAF1                                                   TR2
      H1=HAF1*2.D0                                                   TR2
      GO TO 4919                                                     TR2
4918  F1=FREQ/FCF2                                                   TR2
      H1=HAF2                                                        TR2
4919  INTF2=INTF2+1                                                  TR2
      RM=MAG(PW2)                                                    TR2
      DO 492 J=1,3                                                   TR2
492   PC3(J)=PW2(J)                                                  TR2
      IF(INTF2.LE.LIMF2) GO TO 315                                   TR2
      IF(ISTOP.GT.1) GO TO 7310                                      TR2
494   IF(FLAG.OR.PNTFLG.GT.500) GO TO 501                            TR2
      GO TO 7310                                                     TR2
C    ITERATION FINISHED                                              TR2
500   DO 5003 J=1,3                                                  TR2
      VEC(J,1)=L(2,J)                                                TR2
      LOC(J,1)=L(2,J)*YBEP+L(3,J)*P1P(3)                             TR2
      VEC(J,2)=L(2,J)*C12P+L(3,J)*C13P                               TR2
      LOC(J,2)=R2(J)                                                 TR2
      VEC(J,3)=M(2,J)                                                TR2
      LOC(J,3)=L(2,J)*Y9P+L(3,J)*Z9P                                 TR2
      VEC(J,4)=P(2,J)                                                TR2
      LOC(J,4)=M(2,J)*YS1PP+M(3,J)*Z10PP+P0(J)                       TR2
      LOC(J,5)=A3(J)                                                 TR2
      LOC(J,6)=P(2,J)*YM1PPP+P(3,J)*ZS3PPP                           TR2
      VEC(J,5)=P(2,J)*C42PPP+P(3,J)*C43PPP                           TR2
      LOC(J,7)=P(2,J)*YM3PPP+P(3,J)*ZM3PPP                           TR2
      VEC(J,6)=P(2,J)*C52PPP+P(3,J)*C53PPP                           TR2
5003  LOC(J,8)=P(2,J)*YBEPPP+P(3,J)*ZBEPPP                           TR2
      DO 5008 K=1,6                                                  TR2
      MAGVEC=MAG(VEC(1,K))                                           TR2
      IF(MAGVEC.EQ.0.D0) GO TO 5008                                  TR2
      DO 5007 J=1,3                                                  TR2
5007  VEC(J,K)=VEC(J,K)/MAGVEC                                       TR2
5008  CONTINUE                                                       TR2
      IF(PNTFLG.LT..5) GO TO 510                                     TR2
501   IF(FLAG2) GO TO 134                                            TR2
      IF(FLAG3) GO TO 234                                            TR2
      RM=RM1                                                         TR2
      INT=0                                                          TR2
      FLAG=.TRUE.                                                    TR2
      GO TO(502,503,504),IREFL                                       TR2
502   CBET1=COBETB*RB/RM                                             TR2
      FLAG2=.TRUE.                                                   TR2
      SBET1=DSQRT(1.D0-CBET1**2)                                     TR
      Y3P=DSQRT(RM**2-Z10PP**2)                                      TR
      P1P(3)=Z10PP                                                   TR
      F=FREQ/FCE                                                     TR
      H=HAE                                                          TR
      GO TO 131                                                      TR
503   F=FREQ/FCF1                                                    TR
      FLAG3=.TRUE.                                                   TR
      H=HAF1                                                         TR
      MR3=RM                                                         TR
      CBET4=COBETB*RB/RM                                             TR
      IF(DABS(CBET4).LT.1.D0.AND.DABS(RM).GT.DABS(Z10PP)) GOTO 5035  TR
      ISTOP=6                                                        TR
```

```
        IRCYCL=0
        GO TO 475
 5035   SBET4=DSQRT(1.D0-CBET4**2)
        Y7P=DSQRT(RM**2-Z10PP**2)
        Z7P=Z10PP
        GO TO 231
 504    IF(IWRITE.EQ.1)WRITE(6,505)IHOPS
 505    FORMAT(23H PENETRATION ON HOP NO.,I3,1H.)
        GO TO 7311
 510    IF(HITS) GO TO 530
        GO TO (512,514,514),IREFL
 512    PP0P=MB1*SBET1
        GO TO 516
 514    PP0P=(MRM4-HE)*DSQRT(1.D0-(CBETM2/(1.D0-HE/MRM4))**2)
 516    DO 520 J=1,3
        B1(J)=P(2,J)*C52PPP+P(3,J)*C53PPP
 520    A1(J)=PQ(J)
        GO TO 550
 530    GO TO (531,532,532),IREFL
 531    ALPHAQ=DASIN(MB1/R0*CBET1)
        PP0P=MB1*SBET1-R0*DSQRT(1.D0-(RY/R0)**2)
        GO TO 533
 532    ALPHAQ=DASIN(MRM1*CBETM1/R0)
        PP0P=(MRM4-HE)*DSQRT(1.D0-(CBETM2/(1.D0-HE/MRM4))**2)-R0*DSQRT(1.D
       *0-(MRM4/R0*CBETM2)**2)
 533    B22PPP=-C52PPP*DCOS(2.D0*ALPHAQ)+C53PPP*DSIN(2.D0*ALPHAQ)
        B23PPP=-C53PPP*DCOS(2.D0*ALPHAQ)-C52PPP*DSIN(2.D0*ALPHAQ)
        DO 540 J=1,3
        A1(J)=PQ(J)
 540    B1(J)=P(2,J)*B22PPP+P(3,J)*B23PPP
 550    IF(IHOPS.GE.IABS(MODE))GO TO 552
 551    GO TO (5511,5512,5513),IREFL
 5511   PPTOT=PPTOT+PP0+PPF2+PP0P
        GPTOT=GPTOT+PP0+GPF2+PP0P
        GO TO 5514
 5512   PPTOT=PPTOT+PP0+DPP1+PPF2+PP2+DPP1P+PP0P+TAU1EA+TAU1ED
        GPTOT=GPTOT+PP0+DGP1+GPF2+PP2+DGP1P+PP0P+TAU1EA+TAU1ED
        GO TO 5514
 5513   PPTOT=PPTOT+PP0+PPEF1A+PPF2+PP2+PPEF1D+PP0P
        GPTOT=GPTOT+PP0+GPEF1A+GPF2+PP2+GPEF1D+PP0P
 5514   MB1=MAG(A1)
        IF(.NOT.HITS.AND.IHOPS.GE.IABS(MODE).AND.MB1.GE.RTARG) GO TO 598
        GO TO 90
 552    IF(SEARCH.AND.RTARG.GT.MR3-HF1A.AND.RTARG.LE.DMAX1(RV,MR3))GOTO595
        IF(ASC) GO TO 566
        IF(DABS(RTARG-R0).LE.1.D-5.AND.HITS) GO TO 580
        IF(MODE.GT.0) GO TO 563
C     DESCENDING MODE
        GO TO(564,562,560),IREFL
 560    IF(RTARG.LE.DMAX1(RV,MRM1).AND.RTARG.GT.MRM1-HF1) GO TO 576
        IF(DMIN1(MRM4-HE,MRM1).LT.RTARG.AND.RTARG.LE.DMAX1(MRM4,MRM1-HF1))
       * GO TO 573
        IF(RTARG.LE.MRM4-HE) GO TO 568
        IF(INCHOP) GO TO 566
        SEARCH=.TRUE.
        GO TO 5513
 562    IF(RTARG.LE.RV.AND.RTARG.GT.MRM4) GO TO 576
```

```
          IF(RTARG.LE.MRM4) GO TO 568                                     TR2
          IF(INCHOP) GO TO 566                                            TR2
          SEARCH=.TRUE.                                                   TR2
          GO TO 5512                                                      TR2
  563     SEARCH=.TRUE.                                                   TR2
          ASC=.TRUE.                                                      TR2
          GO TO 551                                                       TR2
  564     IF(RTARG.LE.RV) GO TO 568                                       TR2
          SEARCH=.TRUE.                                                   TR2
          IF(.NOT.INCHOP) GO TO 5511                                      TR2
  566     ISTOP=15                                                        TR2
          GO TO 7310                                                      TR2
    568 IF(RTARG.GE.RY.OR.INCHOP) GO TO 569                               TR2
          SEARCH=.TRUE.                                                   TR2
          GO TO 551                                                       TR2
    569 TTARG=0.D0                                                        TR2
          IF(RTARG.GT.RY)TTARG=-DSQRT((RTARG**2-MRY1SQ)/MC5SQ)            TR2
          YTARGP=C52PPP*TTARG+YY2PPP                                      TR2
          ZTARGP=C53PPP*TTARG+ZY2PPP                                      TR2
          DO 570 J=1,3                                                    TR2
          B1(J)=C52PPP*P(2,J)+C53PPP*P(3,J)                               TR2
  570     PTARG(J)=P(2,J)*YTARGP+P(3,J)*ZTARGP                            TR2
          PP0P=0.D0                                                       TR2
          IF(RTARG.GT.RY)PP0P=-RTARG*DSQRT(1.D0-(RY/RTARG)**2)            TR2
          GO TO (571,5715,5715),IREFL                                     TR2
  571     PP0P=MB1*SBET1+PP0P                                             TR2
          PPTOT=PPTOT+PP0+PPF2+PP0P                                       TR2
          GPTOT=GPTOT+PP0+GPF2+PP0P                                       TR2
          GO TO 600                                                       TR2
  5715    PP0P=(MRM4-HE)*DSQRT(1.D0-(CBETM2/(1.D0-HE/MRM4))**2)+PP0P      TR2
          GO TO(5720,5720,5725),IREFL                                     TR2
  5720    PPTOT=PPTOT+PP0+DPP1+PPF2+PP2+DPP1P+PP0P+TAU1EA+TAU1ED          TR2
          GPTOT=GPTOT+PP0+DGP1+GPF2+PP2+DGP1P+PP0P+TAU1EA+TAU1ED          TR2
          GO TO 600                                                       TR2
  5725    PPTOT=PPTOT+PP0+PPEF1A+PPF2+PP2+PPEF1D+PP0P                     TR2
          GPTOT=GPTOT+PP0+GPEF1A+GPF2+PP2+GPEF1D+PP0P                     TR2
          GO TO 600                                                       TR2
  573     TTARG=-DSQRT((RTARG**2-MRY1SQ)/MC4SQ)                           TR2
          YTARGP=C42PPP*TTARG+YY1PPP                                      TR
          ZTARGP=C43PPP*TTARG+ZY1PPP                                      TR
          DO 575 J=1,3                                                    TR
          B1(J)=C42PPP*P(2,J)+C43PPP*P(3,J)                               TR
  575     PTARG(J)=P(2,J)*YTARGP+P(3,J)*ZTARGP                            TR
          PPTOT=PPTOT+PPEF1A+PPF2+PP2+PP0+PPEF1D                          TR
          GPTOT=GPTOT+GPEF1A+GPF2+PP2+PP0+GPEF1D                          TR
          GO TO 600                                                       TR
  576     YTARGP=-DSQRT(RTARG**2-ZS3PPP**2)                               TR
          ZTARGP=ZS3PPP                                                   TR
          DO 579 J=1,3                                                    TR
          B1(J)=C32PPP*P(2,J)                                             TR
  579     PTARG(J)=P(2,J)*YTARGP+P(3,J)*ZTARGP                            TR
          GO TO(5793,5793,5797),IREFL                                     TR
  5793    TAU=MRM4*SBETM2-(MRM4-HE)*DSQRT(1.D0-(CBETM2/(1.D0-HE/MRM4))**2)+P TR
         * P0+PP2                                                         TR
          PPTOT=PPTOT+TAU+DPP1+PPF2+TAU1EA                                TR
          GPTOT=GPTOT+TAU+DGP1+GPF2+TAU1EA                                TR
          GO TO 600                                                       TR
```

```
 5797  TAU=MRM1*SBETM1-(MRM1-HF1)*DSQRT(1.D0-(CBETM1/(1.D0-HF1/MRM4))**2)  TF
      *+PP0+PP2                                                             TF
       PPTOT=PPTOT+TAU+PPEF1A+PPF2+DPP2P                                    TF
       GPTOT=GPTOT+TAU+GPEF1A+GPF2+DGP2P                                    TF
       GO TO 600                                                            TR
 C     TARGET ON GROUND                                                     TR
 580   DO 590 J=1,3                                                         TR
       B1(J)=C52PPP*P(2,J)+C53PPP*P(3,J)                                    TR
 590   PTARG(J)=A1(J)                                                       TR
       GO TO(5902,5904,5906),IREFL                                          TR
 5902  PPTOT=PPTOT+PP0+PPF2+PP0P                                            TR
       GPTOT=GPTOT+PP0+GPF2+PP0P                                            TR
       GO TO 600                                                            TR
 5904  PPTOT=PPTOT+PP0+DPP1+PPF2+PP2+DPP1P+PP0P+TAU1EA+TAU1ED               TR.
       GPTOT=GPTOT+PP0+DGP1+GPF2+PP2+DGP1P+PP0P+TAU1EA+TAU1ED               TR:
       GO TO 600                                                            TR
 5906  PPTOT=PPTOT+PP0+PPEF1A+PP2+PPF2+PPEF1D+PP0P                          TR:
       GPTOT=GPTOT+PP0+GPEF1A+PP2+GPF2+GPEF1D+PP0P                          TR:
       GO TO 600                                                            TR2
 C     ASCENDING MODE                                                       TR2
 591   YTARGP=DSQRT(RTARG**2-P1P(3)**2)                                     TR2
       DO 592 J=1,3                                                         TR2
 592   PTARG(J)=L(2,J)*YTARGP+L(3,J)*P1P(3)                                 TR2
       TAU=RTARG*DSQRT(1.D0-(RY/RTARG)**2)                                  TR2
       PPTOT=PPTOT+TAU                                                      TR2
       GPTOT=GPTOT+TAU                                                      TR2
       INUSE=INUSE-1                                                        TR2
       IF(.NOT.HITS) GO TO 600                                              TR2
       TAU=-R0*DSQRT(1.D0-(RY/R0)**2)                                       TR2
       PPTOT=PPTOT+TAU                                                      TR2
       GPTOT=GPTOT+TAU                                                      TR2
       GO TO 600                                                            TR2
 593   TTARG=DSQRT((RTARG**2-MRX1SQ)/MC1PSQ)                                TR2
       YTARGP=C12P*TTARG+Y6P                                                TR2
       ZTARGP=C13P*TTARG+Z6P                                                TR2
       DO 594 J=1,3                                                         TR2
       PTARG(J)=L(2,J)*YTARGP+L(3,J)*ZTARGP                                 TR2
 594   B1(J)=L(2,J)*C12P+L(3,J)*C13P                                        TR2
       TAU=(MR3-HF1A)*DSQRT(1.D0-(COSO2/(1.D0-HF1A/MR3))**2)                TR2
       PPTOT=PPTOT+DPP1+TAU                                                 TR2
       GPTOT=GPTOT+DGP1+TAU                                                 TR2
       INUSE=INUSE-1                                                        TR2
       IF(.NOT.HITS) GO TO 600                                              TR2
       TAU=-R0*DSQRT(1.D0-((MR2-HEA)/R0*COSO1)**2)                          TR2
       PPTOT=PPTOT+TAU                                                      TR2
       GPTOT=GPTOT+TAU                                                      TR2
       GO TO 600                                                            TR2
 595   TTARG=DSQRT((RTARG**2-Y10P**2-Z10P**2)/(C22P**2+C23P**2))            TR2
       YTARGP=C22P*TTARG+Y10P                                               TR2
       ZTARGP=C23P*TTARG+Z10P                                               TR2
       DO 596 J=1,3                                                         TR2
       PTARG(J)=L(2,J)*YTARGP+L(3,J)*ZTARGP                                 TR2
 596   B1(J)=L(2,J)*C12P+L(3,J)*C13P                                        TR2
       TAU=(MR2-HEA)*DSQRT(1.D0-(COSO1/(1.D0-HEA/MR2))**2)+PP2A             TR2
       PPTOT=PPTOT+PPEF1A+PPF2/2.D0+TAU                                     TR2
       GPTOT=GPTOT+GPEF1A+GPF2/2.D0+TAU                                     TR2
       IF(.NOT.ASC) GO TO 600                                               TR2
```

```
      INUSE=INUSE-1                                                     TR2
      TAU=-R0*DSQRT(1.D0-((MR2-HEA)/R0*COSO1)**2)                       TR2
      PPTOT=PPTOT+TAU                                                   TR2
      GPTOT=GPTOT+TAU                                                   TR2
      GO TO 600                                                         TR2
598   RS2=MAG(A1)                                                       TR2
      DO 599 I=1,3                                                      TR2
599   PTARG(I)=A1(I)                                                    TR2
      GO TO 601                                                         TR2
600   IF(.NOT.HITS) RS2=MAG(PTARG)                                      TR2
601   IF(GPONLY) RETURN                                                 TR2
      NLTARG=RTD*DASIN(PTARG(3)/RS2)                                    TR2
      WLTARG=RTD*WLON(PTARG(2),PTARG(1))                                TR2
      EAST(1)=-PTARG(2)                                                 TR2
      EAST(2)=PTARG(1)                                                  TR2
      EAST(3)=0.D0                                                      TR2
      CALL CROSS(PTARG,EAST,NORTH)                                      TR2
      CNST1=DOT(PTARG,B1)/RS2                                           TR2
      ELF=RTD*DASIN(CNST1/MAG(B1))                                      TR2
      ME=MAG(EAST)                                                      TR2
      MN=MAG(NORTH)                                                     TR2
      CNST=DOT(EAST,B1)/ME                                              TR2
      AZF=RTD*DATAN2(CNST,DOT(NORTH,B1)/MN)+180.D0-DSIGN(180.D0,CNST)   TR2
      B1(1)=PTARG(1)-A11(1)                                             TR2
      B1(2)=PTARG(2)-A11(2)                                             TR2
      B1(3)=PTARG(3)-A11(3)                                             TR2
      SRANGE=MAG(B1)                                                    TR2
      ELSR=RTD*DASIN(DOT(A11,B1)/(SRANGE*MA1))                          TR2
      DIST=R0*DACOS(DOT(A11,PTARG)/(RS2*MA1))                           TR2
      CNST=DOT(EAST1,B1)                                                TR2
      BEAR=RTD*DATAN2(CNST,DOT(NORTH1,B1))+180.D0-DSIGN(180.D0,CNST)    TR2
      RETURN                                                            TR2
7310  IGOBAK=1                                                          TR2
      IF(IWRITE.EQ.1)WRITE(6,7309) MESSPR(ISTOP)                        TR2
7309  FORMAT(11H  MESSUP---,A10)                                        TR2
      RETURN                                                            TR2
7311  IGOBAK=2                                                          TR2
      RETURN                                                            TR2
      END                                                               TR2
```

```
UNCTION MAG          74/74    OPT=1                               FTN 4.5+414        04/27/

             DOUBLE PRECISION FUNCTION MAG(A)                                          MAG
             IMPLICIT DOUBLE PRECISION(A-Z)                                            MAG
             DIMENSION A(3)                                                            MAG
             MAG=DSORT(A(1)*A(1)+A(2)*A(2)+A(3)*A(3))                                  MAG
             RETURN                                                                    MAG
             END                                                                       MAG
```

```
JNCTION DOT          74/74    OPT=1                               FTN 4.5+414        04/27/

             DOUBLE PRECISION FUNCTION DOT(A,B)                                        DOT
             IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                       DOT
             DIMENSION A(3),B(3)                                                       DOT
             DOT=A(1)*B(1)+A(2)*B(2)+A(3)*B(3)                                         DOT
             RETURN                                                                    DOT
             END                                                                       DOT
```

```
ROUTINE CROSS        74/74    OPT=1                               FTN 4.5+414        04/27/

             SUBROUTINE CROSS(A,B,C)                                                   VEC
             IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                       VEC
             DIMENSION A(3),B(3),C(3)                                                  VEC
             C(1)=A(2)*B(3)-A(3)*B(2)                                                  VEC
             C(2)=A(3)*B(1)-A(1)*B(3)                                                  VEC
             C(3)=A(1)*B(2)-A(2)*B(1)                                                  VEC
             RETURN                                                                    VEC
             END                                                                       VEC
```

```
ROUTINE PRVSTO      74/74    OPT=1                              FTN 4.5+414          04/27/

          SUBROUTINE PRVSTO                                                  PRV
          DOUBLE PRECISION P1,P2                                             PRV
          LOGICAL LP1,LP2                                                    PRV
          COMMON/CPREV/P1(56),P2(56),LP1(6),LP2(6)                           PRV
          DO 1 I=1,56                                                        PRV
    1     P2(I)=P1(I)                                                        PRV
          DO 2 I=1,6                                                         PRV
    2     LP2(I)=LP1(I)                                                      PRV
          RETURN                                                             PRV
          ENTRY PRVGET                                                       PRV
          DO 3 I=1,56                                                        PRV
    3     P1(I)=P2(I)                                                        PRV
          DO 4 I=1,6                                                         PRV
    4     LP1(I)=LP2(I)                                                      PRV
          RETURN                                                             PRV
          END                                                                PRV
```

```
ROUTINE HTGROF      74/74    OPT=1                              FTN 4.5+414          04/27/

          SUBROUTINE  HTGROF(NOFLAG,FREQ,NLAT,WLONG,ANGLE1,ANGLE2,RANG12,RMU HTG
         *,RTU,RTMU,DNDR,DNDT,DNDP,PNTFLG)                                   HTG
          IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                HTG
          DOUBLE PRECISION NLAT,NLATR                                        HTG
          LOGICAL TILT,FAST                                                  HTG
          COMMON/TILTC/TILT,FAST                                             HTG
          NLATR=NLAT/57.2957795130823200                                     HTG
          WLONGR=WLONG/57.2957795130823200                                   HTG
          CALL RIIP(6370.D0,NLATR,WLONGR,EN)                                 HTG
          CALL  RTFIND(NOFLAG,FREQ,ANGLE1,ANGLE2,RANG12,RTU,RMU,RTMU,PNTFLG) HTG
          IF(TILT) GO TO 1                                                   HTG
          DNDR=1.0D0                                                         HTG
          DNDP=0.0D0                                                         HTG
          DNDT=0.0D0                                                         HTG
          RETURN                                                             HTG
    1     CALL  DENSE(RTU,NLAT,WLONG,EN,DNDR,DNDT,DNDP)                      HTG
          RETURN                                                             HTG
          END                                                                HTG
```

```
      SUBROUTINE  RTFIND(NDFLAG,FREQ,ANGLE1,ANGLE2,RANG12,RTU,RMU,RTMU,P RTF
     *NTFLG)                                                            RTF
C  NDFLAG FLAG TO INDICATE WHICH PASS                                   RTF
C    =0 FIRST PASS                                                      RTF
C    =1 SUBSEQUENT PASSES                                               RTF
C  NOTE - THIS ROUTINE REQUIRES INITIALIZATION .                        RTF
C           ON FIRST PASS, ANGLE1 MUST = THE INITIAL TAKE OFF ANGLE ON  RTF
C           GROUND, ANGLE2 MUST =0, RANG12 MUST = RADIUS OF EARTH       RTF
C  FREQ-OPERATING FREQUENCY IN MHZ.                                     RTF
C  ANGLE1 - ANGLE (DEG) BETWEEN TILTED LAYER AND RAY INCIDENT ON THE LAY RTF
C  ANGLE2 - TILT ANGLE (DEG) - THAT IS, ANGLE BETWEEN TILTED LAYER AND A RTF
C          EXACTLY HORIZONTAL LAYER AT THE REFLECTION POINT             RTF
C  RANG12 - DISTANCE FROM CENTER OF EARTH (KM) OF REFLECTION POINT USED RTF
C          COMPUTE ANGLE1+2                                             RTF
C  RTU-COMPUTED DISTANCE FROM CENTER OF EARTH IN KM. OF POINT OF INTERES RTF
C  RMU-DISTANCE FROM CENTER OF EARTH TO LAYER IN KM.                    RTF
C  PNTFLG-PENETRATION FLAG                                              RTF
C        =0.  REFLECTION                                                RTF
C        =1.  PENETRATION                                               RTF
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               RTF
      DOUBLE PRECISION M3000                                            RTF
      DIMENSION   RM(3),H(3),FC(3),RTM(3)                               RTF
      COMMON/RIIPAR/M3000,FCF2,FCF1,FCE,HBE,HAE,HME,HAF1,HMF1,HAF2,HMF2, RTF
     1DUM(13),ID(3)                                                     RTF
      COMMON/RTCOM/COSBO,NUSE                                           RTF
      DATA RAD/0.17453292519943300-01/                                  RTF
      DACOS(DUMMY)=DATAN2(DSQRT(1.0D0-DUMMY*DUMMY),DUMMY)               RTF
      PNTFLG=0.0D0                                                      RTF
      IF(NDFLAG)4,4,44                                                  RTF
4     IF(ANGLE2.EQ.0.0D0)GO TO 44                                       RTF
      WRITE(6,100) ANGLE2                                               RTF
100   FORMAT(22H THE VALUE OF ANGLE2 =,D15.8,13H IS IN ERROR.)          RTF
      PNTFLG=1.0D0                                                      RTF
44    FC(1)=FCE                                                         RTF
      FC(2)=FCF1                                                        RTF
      FC(3)=FCF2                                                        RTF
      H(1)=HAE                                                          RTF
      H(2)=HAF1                                                         RTF
      H(3)=HAF2                                                         RTF
      RM(3)=6370.D0+HMF2                                                RTF
      RM(2)=RM(3)-H(3)                                                  RTF
      RM(1)=RM(2)-H(2)                                                  RTF
      IF(NDFLAG.GT.0) GO TO 45                                          RTF
      COSBO=(DCOS(ANGLE1*RAD)*RANG12/6370.D0)                           RTF
45    DO 10 N=1,3                                                       RTF
C   COMPUTE RTM FOR EACH LAYER                                          RTF
      IF(FC(N).EQ.0.0D0)GO TO 6                                         RTF
      FAC=(RM(N)**2)/16.D0-(H(N)**2/2.D0)*((FREQ/FC(N))**2-1.D0)        RTF
      IF(FAC)6,5,5                                                      RTF
5     RTM(N)=0.75D0*RM(N)+DSQRT(FAC)                                    RTF
      GO TO 10                                                          RTF
6     RTM(N)=RM(3)                                                      RTF
10    CONTINUE                                                          RTF
C   FIND MAX RTM                                                        RTF
      RMM=RTM(3)                                                        RTF
      IF(RTM(3).EQ.RM(3))RMM=RM(3)/2.CD0                                RTF
3000  IF(NDFLAG.GT.0) GO TO 14                                          RTF
```

```
C    IF ALL RTM ARE EQUAL, USE F2 LAYER                                RTF
       IF(RTM(2).NE.RTM(3).OR.RTM(1).NE.RTM(3))GO TO 12                RTF
       NUSE=3                                                          RTF
       RTMUSE=RTM(3)                                                   RTF
       GO TO 14                                                        RTF
C    FIND MINIMUM RTM                                                  RTF
12     RTMUSE=RTM(1)                                                   RTF
       NUSE=1                                                          RTF
       DO 13 I=2,3                                                     RTF
       IF(RTMUSE.LT.RTM(I))GO TO 13                                    RTF
       RTMUSE=RTM(I)                                                   RTF
       NUSE=I                                                          RTF
13     CONTINUE                                                        RTF
       IF(NDFLAG.GT.0) GO TO 14                                        RTF
123    ANGLE1=(DACOS(COSB0*6370.D0/RANG12))/RAD                        RTF
C    USE PARAMETERS ASSOC. WITH MIN RTM                                RTF
14     RTMUSE=RTM(NUSE)                                                RTF
       RMUSE=RM(NUSE)                                                  RTF
       HUSE=H(NUSE)                                                    RTF
       FCUSE=FC(NUSE)                                                  RTF
144    CBTM=DCOS(ANGLE1*RAD)*RANG12/RTMUSE                             RTF
       IF(CBTM-1.0D0)15,15,16                                          RTF
15     BTM=DACOS(CBTM)                                                 RTF
17     X=1.0D0-((FREQ/FCUSE)**2)*((DSIN(BTM))**2)                      RTF
       IF(X.GE.0.0D0)GO TO 20                                          RTF
16     IF(RTMUSE.NE.RM(3))GO TO 18                                     RTF
C    PENETRATION                                                       RTF
185    RTU=RMM                                                         RTF
       PNTFLG=1.0D0                                                    RTF
       RMU=RM(3)                                                       RTF
       RTMU=RTMUSE                                                     RTF
2000   RETURN                                                          RTF
18     IF(NUSE-3)19,185,185                                            RTF
C    ELIMINATE RTM USED AND FIND NEXT SMALLEST RTM                     RTF
19     NUSE=NUSE+1                                                     RTF
       GO TO 123                                                       RTF
C    REFLECTION                                                        RTF
20     RT=RTMUSE-HUSE*DSQRT(X)                                         RTF
       RMU=RMUSE                                                       RTF
       RTMU=RTMUSE                                                     RTF
       A=RMU-RT                                                        RTF
       B=A/DCOS(ANGLE2*RAD)                                            RTF
       IF(B.GE.HUSE)B=HUSE                                             RTF
       RTU=RMU-B                                                       RTF
       END                                                             RTF
```

```
      SUBROUTINE  DENSE(R,THETA,PHI,EN,DNDR,DNDT,DNDP)                  DEN
C  R-DISTANCE FROM CENTER OF EARTH IN KM.                               DEN
C  THETA-NORTH LATITUDE IN DEGREES                                      DEN
C  PHI-WEST LONGITUDE IN DEGREES                                        DEN
C  EN-VALUE OF ELECTRON DENSITY AT GIVEN POINT                          DEN
C  DNDR-PARTIAL DERIVATIVE OF ELEC. DEN. WITH RESPECT TO R              DEN
C  DNDT-PARTIAL DERIVATIVE OF ELEC. DEN. WITH RESPECT TO THETA          DEN
C  DNDP-PARTIAL DERIVATIVE OF ELEC. DEN. WITH RESPECT TO PHI            DEN
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               DEN
      DIMENSION   RX(9),RY(9),ALPHA(9),FN(9)                            DEN
      COMMON/RPERT/RRIIP,ITIP                                           DEN
      DATA RAD/.017453292519943300/                                     DEN
10    RY(1)=R                                                           DEN
      RX(1)=THETA*RAD                                                   DEN
      ALPHA(1)=PHI*RAD                                                  DEN
      NSIG=1                                                            DEN
20    CALL  DERV(NSIG,FN,RY,RX,ALPHA,DNDR,DNDT,DNDP)                    DEN
      IF(NSIG.EQ.3) GO TO 30                                            DEN
      DO 25 I=2,5                                                       DEN
      CALL  RIIP(RY(I),RX(I),ALPHA(I),FN(I))                            DEN
      GO TO (22,23),ITIP                                                DEN
22    CALL  NFROMR(RY(I+4),FN(I+4))                                     DEN
      GO TO 25                                                          DEN
23    CALL  RIIP(RY(I+4),RX(I+4),ALPHA(I+4),FN(I+4))                    DEN
25    CONTINUE                                                          DEN
      CALL  RIIP(RY(1),RX(1),ALPHA(1),FN(1))                            DEN
      NSIG=2                                                            DEN
      GO TO 20                                                          DEN
30    EN=FN(1)                                                          DEN
      RETURN                                                            DEN
      END                                                               DEN
```

```
      SUBROUTINE  DERV(NSIG,FN,RY,RX,ALPHA,DNDR,DNDT,DNDP)                 DER
C  NSIG-SWITCH TO INDICATE WHETHER POINTS OR DERIVATIVES ARE REQUIRED      DER
C  FN-ARRAY OF 9 VALUES OF ELEC. DEN.                                      DER
C  RY-ARRAY OF 9 VALUES OF R IN KM.                                        DER
C  RX-ARRAY OF 9 VALUES OF THETA IN RAD.                                   DER
C  ALPHA-ARRAY OF 9 VALUES OF PHI IN RAD.                                  DER
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                  DER
      DIMENSION   FN(9),RY(9),RX(9),ALPHA(9),DF(3)                         DER
      DATA S/1.0D0/                                                        DER
      IF(NSIG-2)100,200,200                                                DER
C FIND 8 POINTS ON CUBIC SURROUNDING RY(1),RX(1),ALPHA(1)                  DER
100   X1=S/(0.17320508D+01)                                                DER
      X2=X1/RY(1)                                                          DER
      C=RY(1)*DCOS(RX(1))                                                  DER
      IF(C.EQ.0.0D0)C=100.0D0                                              DER
      X3=X1/C                                                              DER
      DO 10 I=2,5                                                          DER
      RY(I)=RY(1)+X1                                                       DER
      RY(I+4)=RY(1)-X1                                                     
10    CONTINUE                                                             DER
      DO 20 I=2,8,2                                                        DER
      RX(I)=RX(1)+X2                                                       DER
      RX(I+1)=RX(1)-X2                                                     DER
20    CONTINUE                                                             DER
      DO 30 I=2,6,4                                                        DER
      ALPHA(I)=ALPHA(1)+X3                                                 DER
      ALPHA(I+1)=ALPHA(I)                                                  DER
      ALPHA(I+2)=ALPHA(1)-X3                                               DER
      ALPHA(I+3)=ALPHA(I+2)                                                DER
30    CONTINUE                                                             DER
      RETURN                                                               DER
C  FIND PARTIALS                                                           DER
200   AM=0.125D0                                                           DER
      DF(1)=AM*(FN(2)-FN(6)+FN(3)-FN(7)+FN(4)-FN(8)+FN(5)-FN(9))           DER
      DF(2)=AM*(FN(2)-FN(3)+FN(4)-FN(5)+FN(6)-FN(7)+FN(8)-FN(9))           DER
      DF(3)=AM*(FN(2)-FN(4)+FN(3)-FN(5)+FN(6)-FN(8)+FN(7)-FN(9))           DER
240   DNDR=DF(1)/X1                                                        DER
      DNDT=DF(2)/X2                                                        DER
      DNDP=DF(3)/X3                                                        DER
      NSIG=3                                                               DER
      RETURN                                                               DER
      END                                                                  DER
```

Appendix C

Block Diagram, Flow Chart, and Program Listing for WIMP Driving Program OBLFACT

Explanation:

C1. OBLFACT BLOCK DIAGRAM

This presentation illustrates the communication and iteration network employed to generate the $M(3000)F_2$ and f_oF_2 gradient correction factors required to match the predicted leading edge to the given oblique ionogram. The function of the various blocks is generally described as follows:

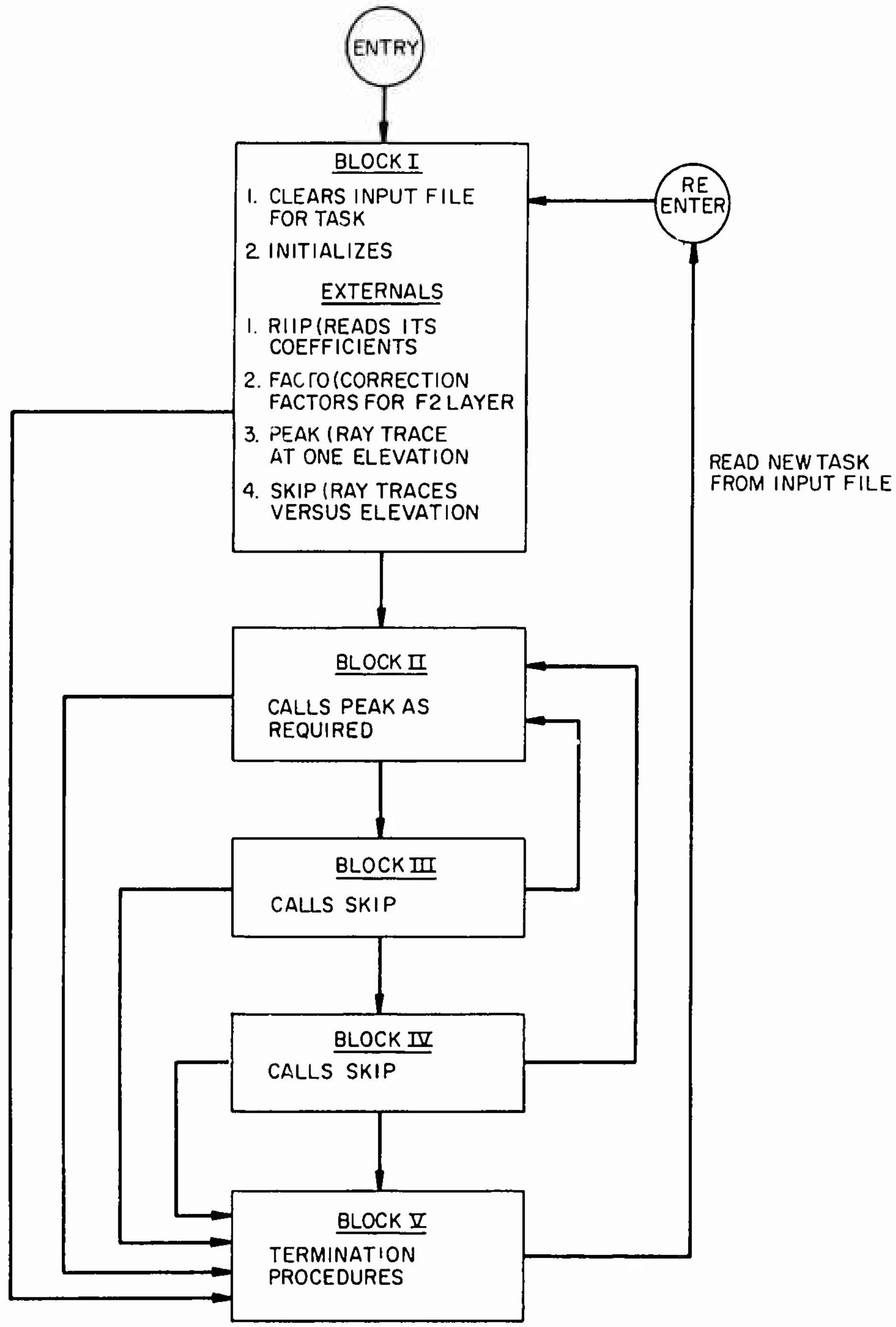
BLOCK I:	Initialization and Input Procedures.
BLOCK II:	Refinement of $M(3000)F_2$ Gradient Correction Factor.
BLOCKS III and IV:	Refinement of f_oF_2 Gradient Correction Factor.
BLOCK V:	Termination and Output Procedures.

C2. DETAILED FLOW CHARTS

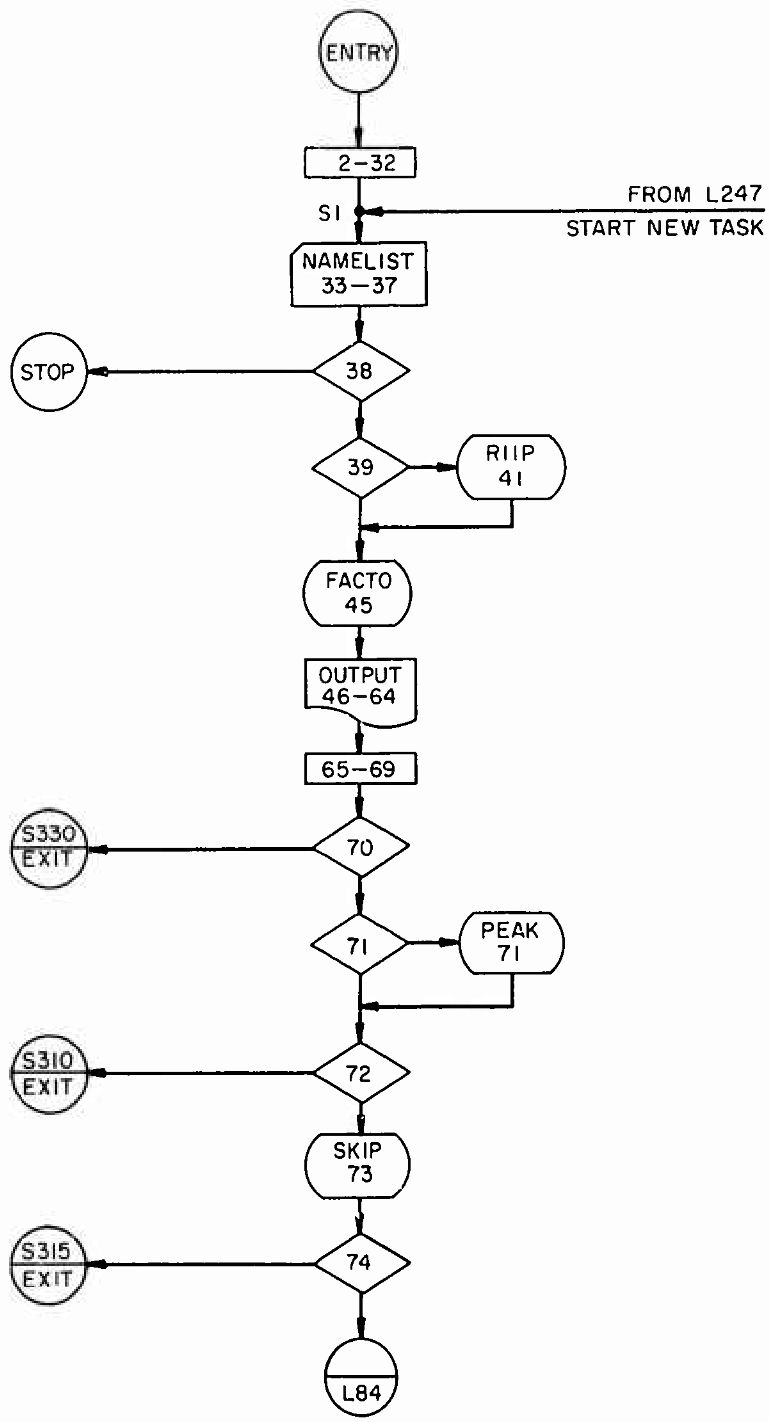
Detailed flow charts are presented for each block. Symbols and conventions are the same as described in Appendix A.

C3. PROGRAM LISTING

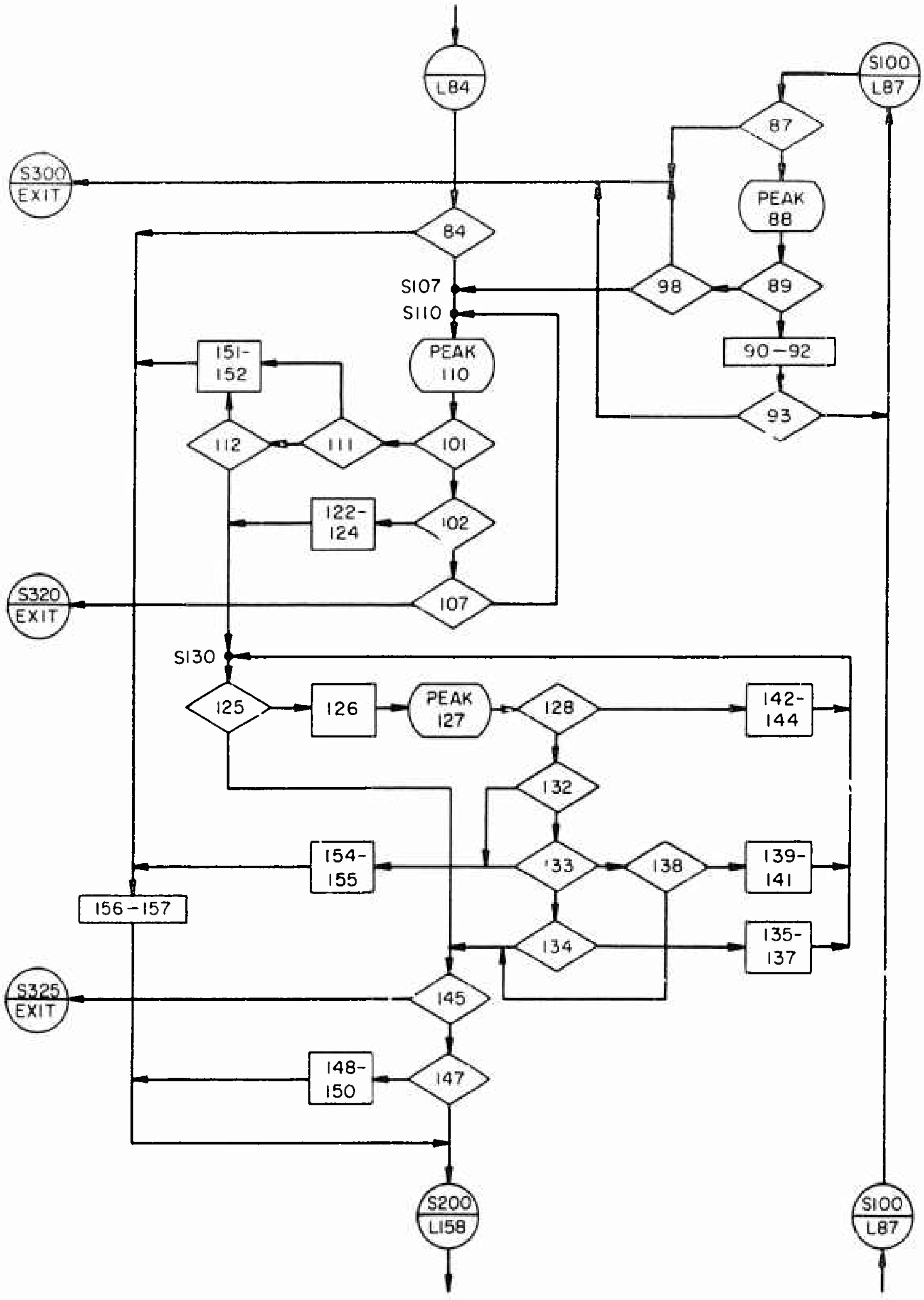
Listings are included for OBLFACT, PEAK, and SKIP.



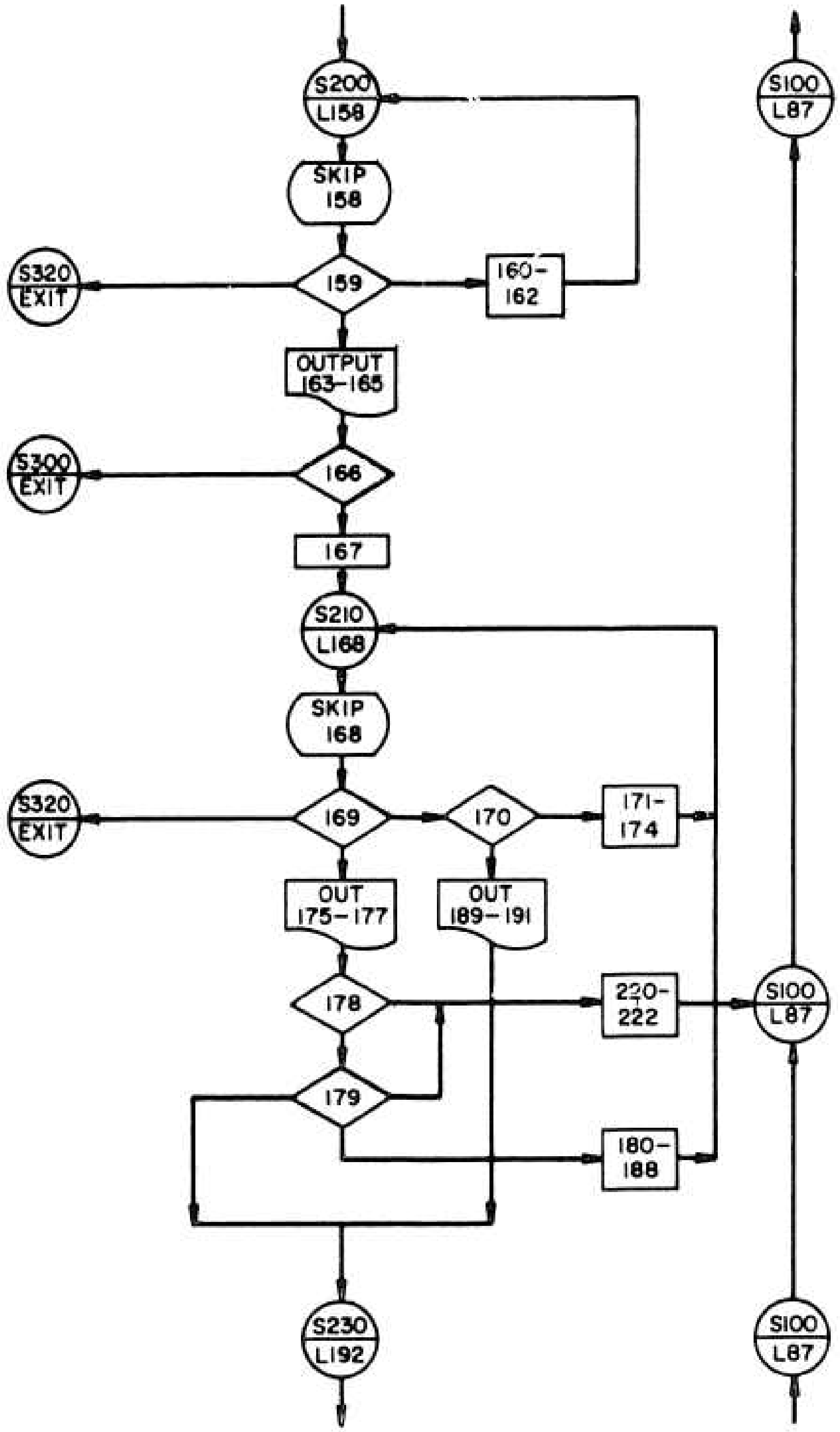
OBLFACT Block Diagram



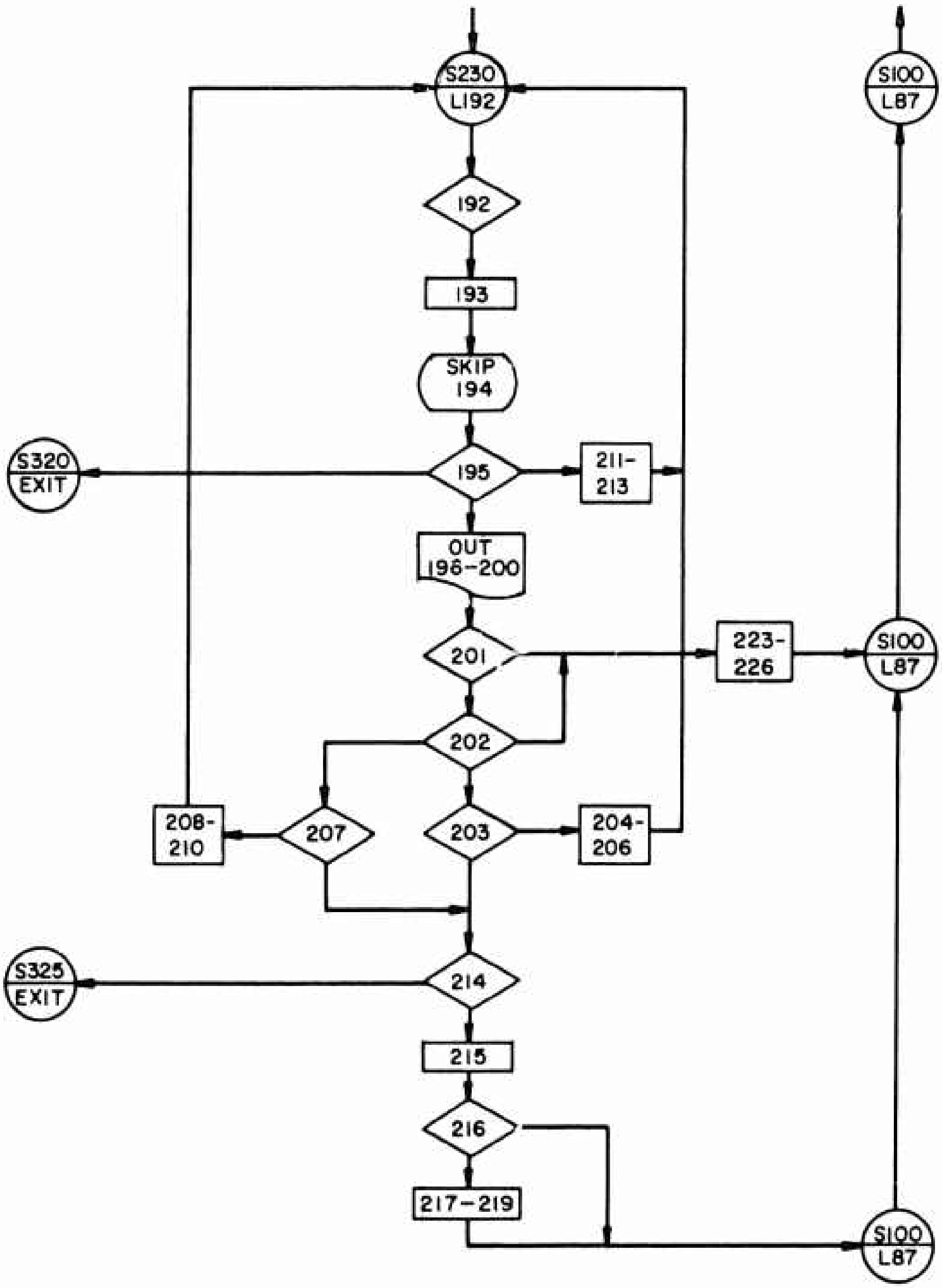
Block I, Detailed Flowchart



Block II, Detailed Flowchart



Block III, Detailed Flowchart



Block IV, Detailed Flowchart

```
      SUBROUTINE OBL                                                    OBL
C     PROGRAM OBLFACT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)            OBL
      IMPLICIT DOUBLE PRECISION(A-H,M,O-Z)                              OBL
      INTEGER MODE                                                      OBL
      DOUBLE PRECISION NL1,NLREF                                        OBL
      LOGICAL SMOOTH,GPONLY                                             OBL
      COMMON/GPFLAG/GPONLY,IWRITE                                       OBL
      COMMON/FIDDLE/FP,FS,R1,NL1,WL1,AZ,RS,ELSTEP,MODE,NSKPCL           OBL
      COMMON/PERTC/NLREF,WLREF,PERTP(14)                                OBL
      COMMON/RIIPAR/DUM1(11),SN,ZT,YRMO,DOY,HAESET,MFAC,F2FAC,F1FAC,    OBL
     1 EFAC,H2FAC,DUM2(3),ID,IA,SMOOTH                                  OBL
      NAMELIST/INPUT/SN,ZT,YRMO,DOY,MFAC,F2FAC,                         OBL
     1NL1,WL1,R1,NLREF,WLREF,RS,AZ,ELSTEP,MODE,                         OBL
     1FCF2R,HMINR,OCHK,FP,DELP,FS,DELS,MR0,FR0                          OBL
     3,IRSTRT,ZTREF                                                     OBL
      IRSTRT=1                                                          OBL
      GPONLY=.TRUE.                                                     OBL
      IWRITE=0                                                          OBL
      ID=1                                                              OBL
      IA=0                                                              OBL
      R1=6370.D0                                                        OBL
      RS=6370.D0                                                        OBL
      MODE=-1                                                           OBL
      NL1=                                                              OBL
      WL1=                                                              OBL
      OCHK=10.D0                                                        OBL
      SN30=0.D0                                                         OBL
      NLREF=NL1                                                         OBL
      WLREF=WL1                                                         OBL
      AZ=                                                               OBL
      ELSTEP=1.D0                                                       OBL
1     FR0=0.D0                                                          OBL
      MR0=0.D0                                                          OBL
      FP=  0.D0                                                         OBL
      DELP=0.D0                                                         OBL
      READ(5,INPUT)                                                     OBL
      IF(IRSTRT.EQ.0) STOP                                              OBL
      IF(SN30.EQ.YRMO)  GO TO 4                                         OBL
      SN30=YRMO                                                         OBL
      CALL RIIP(0.D0,0.D0,0.D0,0.D0)                                    OBL
    4 IPKCNT=0                                                          OBL
      FCF2R=- DABS(FCF2R)                                               OBL
      HMINR=- DABS(HMINR)                                               OBL
      CALL FACTO(NLREF,WLREF,FCF2R,HMINR,ZTREF)                         OBL
CCCC  WRITE(6,INPUT)                                                    OBL
      WRITE(6,900)SN,ZT,YRMO,DOY,MFAC,F2FAC                             OBL
  900 FORMAT(1X,[SN      =[,11X,D24.18/1X,[ZT      =[,11X,D24.18/       OBL
     *       1X,[YRMO    =[,11X,D24.18/1X,[DOY     =[,11X,D24.18/       OBL
     *       1X,[MFAC    =[,11X,D24.18/1X,[F2FAC   =[,11X,D24.18)       OBL
      WRITE(6,901)NL1,WL1,R1,NLREF,WLREF,RS                             OBL
  901 FORMAT(1X,[NL1     =[,11X,D24.18/1X,[WL1     =[,11X,D24.18/       OBL
     *       1X,[R1      =[,11X,D24.18/1X,[NLREF   =[,11X,D24.18/       OBL
     *       1X,[WLREF   =[,11X,D24.18/1X,[RS      =[,11X,D24.18)       OBL
      WRITE(6,902)AZ,ELSTEP,MODE,FCF2R,HMINR,OCHK                       OBL
  902 FORMAT(1X,[AZ      =[,11X,D24.18/1X,[ELSTEP  =[,11X,D24.18/       OBL
     *       1X,[MODE    =[,16X,I3    /1X,[FCF2R   =[,11X,D24.18/       OBL
     *       1X,[HMINR   =[,11X,D24.18/1X,[OCHK    =[,11X,D24.18)       OBL
```

```
      WRITE(6,903)FP,DELP, FS,DELS, MR0,FR0                          OBL
  903 FORMAT(1X,[FP        =[,11X,D24.18/1X,[DELP      =[,11X,D24.18/ OBL
     *        1X,[FS        =[,11X,D24.18/1X,[DELS      =[,11X,D24.18/ OBL
     *        1X,[MR0       =[,11X,D24.18/1X,[FR0       =[,11X,D24.18) OBL
      WRITE(6,904) IRSTRT,ZTREF                                      OBL
  904 FORMAT(1X,[IRSTRT  =[,16X,I3     /1X,[ZTREF    =[,11X,D24.18)  OBL
      IM01=0                                                         OBL
      IF01=0                                                         OBL
      NSKPCL=0                                                       OBL
      IGOBAK=0                                                       OBL
      GPP0=0.D0                                                      OBL
      IF(DABS(DELP).LT.DCHK.AND.DABS(DELS).LT.DCHK) GO TO 330         OBL
      IF(FP.GE.5.74D0) CALL PEAK(FR0,MR0,GPP0,IGOBAK)                 OBL
      IF(IGOBAK.GT.0) GO TO 310                                      OBL
      CALL SKIP(FR0,MR0,GPS0,0,IGOBAK)                               OBL
      IF(IGOBAK.GT.0) GO TO 315                                      OBL
      GPPDES=GPP0+DELP                                               OBL
      GPSDES=GPS0+DELS                                               OBL
      WRITE(6,111) GPPDES,GPSDES                                     OBL
  111 FORMAT([  DESIRED GPP=[,4PD15.2,[, GPS=[,D15.2)                OBL
      DELP1=DELP                                                     OBL
      DELS1=DELS                                                     OBL
      DELP0=DELP                                                     OBL
      DELS0=DELS                                                     OBL
      MR1=MR0                                                        OBL
      IF(DABS(DELP).LT.DCHK) GO TO 197                               OBL
      GO TO 107                                                      OBL
C MR FIDDLING                                                        OBL
  100 IF(FP.LT.5.74D0) GO TO 300                                     OBL
      CALL PEAK(FR0,MR0,GPP0,IGOBAK)                                 OBL
      IF(IGOBAK.EQ.0) GO TO 105                                      OBL
      MR1=MR0                                                        OBL
      MR0=MR1+.001D0                                                 OBL
      IPKCNT=IPKCNT+1                                                OBL
      IF(IPKCNT-100) 100,320,320                                     OBL
  105 DELP0=GPPDES-GPP0                                              OBL
      NSTMT=105                                                      OBL
      WRITE(6,1111) NSTMT,FR0,MR0,DELP0                              OBL
 1111 FORMAT(I10,1P3D15.7)                                           OBL
      IF(DABS(DELP0).LT.DCHK) GO TO 300                              OBL
  107 MR1=MR0-DSIGN(.001D0,DELP0)                                    OBL
  110 CALL PEAK(FR0,MR1,GPP1,IGOBAK)                                 OBL
      IF(IGOBAK.EQ.0) GO TO 115                                      OBL
      IF(DELP0.GT.0.D0) GO TO 128                                    OBL
      MR0=MR1                                                        OBL
      MR1=MR0-DSIGN(.001D0,DELP0)                                    OBL
      DELP0=-10000.D0                                                OBL
      IPKCNT=IPKCNT+1                                                OBL
      IF(IPKCNT-100)110,320,320                                      OBL
  115 DELP1=GPPDES-GPP1                                              OBL
      NSTMT=115                                                      OBL
      WRITE(6,1111) NSTMT,FR0,MR1,DELP1                              OBL
      IF(DABS(DELP1).LT.DCHK) GO TO 190                              OBL
      IF(DELP1*DELP0) 130,190,120                                    OBL
  120 MRR=DELP1*(MR1-MR0)/(DELP0-DELP1)                              OBL
      IF(DABS(MRR).GT.0.1D0) MRR=DSIGN(0.1D0,MRR)                    OBL
      MRR=MR1+MRR                                                    OBL
```

```
      IF(-DELP0.EQ.10000.D00.OR.(MR1-MRR)*DELP0.LE.0.D0)           OBL
     1 MRR=MR1-DSIGN(.001D0,DELP1)                                  OBL
      MR0=MR1                                                       OBL
      MR1=MRR                                                       OBL
      DELP0=DELP1                                                   OBL
      GO TO 110                                                     OBL
 128  DELP1=-10000.D0                                               OBL
      NSTMT=128                                                     OBL
      WRITE(6,1111) NSTMT,FR0,MR1,DELP1                             OBL
  130 IF(DABS(MR1-MR0).LT..0001D0) GO TO 180                        OBL
      MRR=(MR1+MR0)*.5D0                                            OBL
      CALL PEAK(FR0,MRR,GPPR,IGOBAK)                                OBL
      IF(IGOBAK.GT.0) GO TO 160                                     OBL
      DELPR=GPPDES-GPPR                                             OBL
      NSTMT=130                                                     OBL
      WRITE(6,1111) NSTMT,FR0,MRR,DELPR                             OBL
      IF(DABS(DELPR).LT.DCHK) GO TO 195                             OBL
      IF(DELPR*DELP0) 150,195,140                                   OBL
  140 IF(DABS(DELPR).GT.DABS(DELP0)) GO TO 180                      OBL
      DELP0=DELPR                                                   OBL
      MR0=MRR                                                       OBL
      GOTO 130                                                      OBL
  150 IF(DABS(DELPR).GT.DABS(DELP1)) GO TO 180                      OBL
      DELP1=DELPR                                                   OBL
      MR1=MRR                                                       OBL
      GOTO 130                                                      OBL
  160 DELP1=-10000.D0                                               OBL
      MR1=MRR                                                       OBL
      GO TO 130                                                     OBL
  180 IF(IM01*IF01.EQ.1) GOTO 325                                   OBL
      IM01=1                                                        OBL
      IF(DABS(DELP1).GE.DABS(DELP0)) GO TO 200                      OBL
      DELP0=DELP1                                                   OBL
      MR0=MR1                                                       OBL
      GOTO 200                                                      OBL
  190 MR0=MR1                                                       OBL
      DELP0=DELP1                                                   OBL
      GOTO 197                                                      OBL
  195 MR0=MRR                                                       OBL
      DELP0=DELPR                                                   OBL
  197 IM01=0                                                        OBL
C   FR FIDDLE                                                       OBL
  200 CALL SKIP(FR0,MR0,GPS0,1,IGOBAK)                              OBL
      IF(IGOBAK-1) 205,201,320                                      OBL
  201 FR1=FR0                                                       OBL
      FR0=FR1+.001D0                                                OBL
      GO TO 200                                                     OBL
  205 DELS0=GPSDES-GPS0                                             OBL
      NSTMT=205                                                     OBL
      WRITE(6,1111) NSTMT,FR0,MR0,DELS0                             OBL
      IF(DABS(DELS0).LT.DCHK) GO TO 300                             OBL
      FR1=FR0-DSIGN(.001D0,DELS0)                                   OBL
  210 CALL SKIP(FR1,MR0,GPS1,1,IGOBAK)                              OBL
      IF(IGOBAK-1) 215,211,320                                      OBL
  211 IF(DELS0.GT.0.D0) GO TO 228                                   OBL
      FR0=FR1                                                       OBL
      FR1=FR0-DSIGN(.001D0,DELS0)                                   OBL
```

```
      DELS0=-10000.D0                                          OBL
      GO TO 210                                                OBL
  215 DELS1=GPSDES-GPS1                                        OBL
      NSTMT=215                                                OBL
      WRITE(6,1111) NSTMT,FR1,MR0,DELS1                        OBL
      IF(DABS(DELS1).LT.DCHK) GO TO 290                        OBL
      IF(DELS1*DELS0) 230,290,220                              OBL
  220 FRR=DELS1*(FR1-FR0)/(DELS0-DELS1)                        OBL
      IF(DABS(FRR).GT.0.1D0) FRR=DSIGN(0.1D0,FRR)              OBL
      FRR=FR1+FRR                                              OBL
      IF(-DELS0.EQ.10000.D00.OR.(FR1-FRR)*DELS0.LE.0.D0)       OBL
     1  FRR=FR1-DSIGN(.001D0,DELS1)                            OBL
      FR0=FR1                                                  OBL
      FR1=FRR                                                  OBL
      DELS0=DELS1                                              OBL
      GOTO 210                                                 OBL
228   DELS1=-10000.D0                                          OBL
      NSTMT=228                                                OBL
      WRITE(6,1111) NSTMT,FR1,MR0,DELS1                        OBL
  230 IF(DABS(FR1-FR0).LT..0001D0) GO TO 280                   OBL
      FRR=(FR1+FR0)*.5D0                                       OBL
      CALL SKIP(FRR,MR0,GPSR,1,IGOBAK)                         OBL
      IF(IGOBAK-1) 235,260,320                                 OBL
  235 DELSR=GPSDES-GPSR                                        OBL
      NSTMT=235                                                OBL
      WRITE(6,1111) NSTMT,FRR,MR0,DELSR                        OBL
                                                               OBL
                                                               OBL
      IF(DABS(DELSR).LT.DCHK) GO TO 295                        OBL
      IF(DELSR*DELS0) 250,295,240                              OBL
  240 IF(DABS(DELSR).GT.DABS(DELS0)) GO TO 280                 OBL
      DELS0=DELSR                                              OBL
      FR0=FRR                                                  OBL
      GOTO 230                                                 OBL
  250 IF(DABS(DELSR).GT.DABS(DELS1)) GO TO 280                 OBL
      DELS1=DELSR                                              OBL
      FR1=FRR                                                  OBL
      GOTO 230                                                 OBL
  260 DELS1=-10000.D0                                          OBL
      FR1=FRR                                                  OBL
      GO TO 230                                                OBL
  280 IF(IM01*IF01.EQ.1) GO TO 325                             OBL
      IF01=1                                                   OBL
      IF(DABS(DELS1).GE.DABS(DELS0)) GO TO 100                 OBL
      DELS0=DELS1                                              OBL
      FR0=FR1                                                  OBL
      GOTO 100                                                 OBL
  290 FR0=FR1                                                  OBL
      DELS0=DELS1                                              OBL
      GOTO 297                                                 OBL
  295 FR0=FRR                                                  OBL
      DELS0=DELSR                                              OBL
  297 IF01=0                                                   OBL
      GOTO 100                                                 OBL
  300 WRITE(6,305)  MR0,FR0,DELP0,DELS0                        OBL
  305 FORMAT([0 CONVERGENT VALUES\ MR=[,1PD15.7,[, FR=[,D15.7/ OBL
     1 10X,[ MISS DIST.  PEAK=[,D10.3,[, SKIP=[,D10.3)         OBL
```

```
      GOTO 1                                                          OBL
  310 WRITE(6,312)                                                    OBL
  312 FORMAT([0 PEAK RAYTRACE FAILS ON INPUT CASE, TRY DIFFERENT INPUT[) OBL
      GOTO 1                                                          OBL
  315 WRITE(6,317)                                                    OBL
  317 FORMAT([0 SKIP RAYTRACES FAIL ON INPUT CASE, TRY DIFFERENT INPUT[) OBL
      GOTO 1                                                          OBL
  320 WRITE(6,322)                                                    OBL
  322 FORMAT([0  100 ITERATIONS DONE BUT DOESN[,1H[,[T CONVERGE,[,    OBL
     1[ TRY DIFFERENT INPUT[)                                         OBL
      GOTO 300                                                        OBL
  325 WRITE(6,327)                                                    OBL
  327 FORMAT([0 CONVERGENCE PROBLEMS, TRY DIFFERENT INPUT[)           OBL
      GOTO 300                                                        OBL
  330 WRITE(6,332)                                                    OBL
  332 FORMAT([ YOUR ORIGINAL INPUT CONVERGES[)                        OBL
      GOTO 1                                                          OBL
      END                                                             OBL
```

```
      SUBROUTINE SKIP(FRR,MRR,GPS,IOPT,IGOBAK)                          SKI
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               SKI
      REAL OUT(3)                                                       SKI
      DOUBLE PRECISION MRR,MR,NL1,NLT,NLREF,GP(42),RS2(42)              SKI
      COMMON/PERTC/NLREF,WLREF,FR,FD,MR,DUM(11)                         SKI
      COMMON/FIDDLE/FP,FS,R1,NL1,WL1,AZ,RS,ELSTEP,MODE,NSKPCL           SKI
      COMMON/REMWEN/INUSE,NNUSE(20),FCREF(20)                           SKI
      NSKPCL=NSKPCL+1                                                   SKI
      IF(NSKPCL.GT.100) GO TO 120                                       SKI
      IGOBAK=0                                                          SKI
      NELM=20.D0/ELSTEP+1.D0                                            SKI
      FR=FRR                                                            SKI
      MR=MRR                                                            SKI
      DO 10 IEL=1,NELM                                                  SKI
      EL=(IEL-1)*ELSTEP                                                 SKI
      CALL TRISL(R1,NL1,WL1,AZ,EL,0.D0,FS,MODE,RS,RS2(IEL),NLT,WLT,     SKI
     1PP,GP(IEL),AZF,ELF,SRANGE,ELSR,DIST,BEAR,IRETRN)                  SKI
      IF(IRETRN-1) 6,8,12                                               SKI
    6 DO  7  I=1,INUSE                                                  SKI
      IF(NNUSE(I).NE.3)  GO TO 8                                        SKI
    7 CONTINUE                                                          SKI
      GO TO 10                                                          SKI
    8 RS2(IEL)=1.D20                                                    SKI
      GP(IEL)=1.D20                                                     SKI
   10 CONTINUE                                                          SKI
      IEL=NELM+1                                                        SKI
   12 IF(IEL.EQ.1) GO TO 110                                            SKI
      RS2(IEL)=1.D20                                                    SKI
      NELM=IEL-1                                                        SKI
      IMIN=1                                                            SKI
      IF=0                                                              SKI
      DO 20 IEL=1,NELM                                                  SKI
      IF(RS2(IEL) .GT.6378.85D0) GO TO 20                               SKI
      IF=1                                                              SKI
      IF(GP(IEL).LT.GP(IMIN)) IMIN=IEL                                  SKI
   20 CONTINUE                                                          SKI
      IF(IF.EQ.0) GO TO 110                                             SKI
      GPS=GP(IMIN)                                                      SKI
      IF(IOPT.EQ.0) GO TO 130                                           SKI
      IF(RS2(IMIN+1).LT.6378.85D0) GO TO 130                            SKI
      IF(RS2(IMIN+1).GE.0.9D20) GO TO 130                               SKI
      IF(GP(IMIN+1).GT.GP(IMIN)) GO TO 130                              SKI
      GPS=GP(IMIN)+(GP(IMIN+1)-GP(IMIN))*(6378.85D0-RS2(IMIN))/         SKI
     1(RS2(IMIN+1)-RS2(IMIN))                                           SKI
      GO TO 130                                                         SKI
  110 IGOBAK=1                                                          SKI
      GO TO 130                                                         SKI
  120 IGOBAK=2                                                          SKI
130   OUT(1)=FR                                                         SKI
      OUT(2)=MR                                                         SKI
      OUT(3)=GPS                                                        SKI
      WRITE(6,140) OUT,IGOBAK                                           SKI
140   FORMAT(L  SKIP(,2F10.6,F10.2,I5)                                  SKI
      RETURN                                                            SKI
      END                                                               SKI
```

```
      SUBROUTINE PEAK(FRR,MRR,GPP,IGOBAK)                                PEA
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 PEA
      REAL OUT(3)                                                        PEA
      DOUBLE PRECISION MRR,NL1,NLT,ELP(25),NLREF,MR                      PEA
      COMMON/PERTC/NLREF,WLREF,FR,FD,MR,DUM(11)                          PEA
      COMMON/FIDDLE/FP,FS,R1,NL1,WL1,AZ,RS,ELSTEP,MODE,NSKPCL            PEA
      COMMON/REMWEN/INUSE,NNUSE(20),FCREF(20)                            PEA
      DATA ELP/11.9D0,10.3D0,8.9D0,7.7D0,6.7D0,5.9D0,5.2D0,4.6D0,       PEA
     1 4.1D0,3.9D0,5.2D0,4.8D0,4.5D0,4.2D0,4.D0,3.8D0,2*3.7D0,          PEA
     2 2*3.6D0,3*3.5D0,2*3.4D0/                                          PEA
      FR=FRR                                                             PEA
      MR=MRR                                                             PEA
      IF=FP-4.74D0                                                       PEA
      CALL TRISL(R1,NL1,WL1,AZ,ELP(IF),0.D0,FP,MODE,RS,                  PEA
     1RS2,NLT,WLT,PP,GPP,AZF,ELF,SRANGE,ELSR,DIST,BEAR,IGOBAK)          PEA
      IF(RS2.GT.6378.85D0) IGOBAK=1                                      PEA
      DO  1  I=1,INUSE                                                   PEA
      IF(NNUSE(I).NE.3) IGOBAK=1                                         PEA
    1 CONTINUE                                                           PEA
      OUT(1)=FR                                                          PEA
      OUT(2)=MR                                                          PEA
      OUT(3)=GPP                                                         PEA
      WRITE(6,11) OUT,IGOBAK                                             PEA
   11 FORMAT([   PEAK[,2F10.6,F10.2,I5)                                  PEA
      RETURN                                                             PEA
      END                                                                PEA
```