Technical Report 816

A WKB PROGRAM FOR ELF/VLF
EARTH-IONOSPHERE EXCITATION BY SOURCES AT SATELLITE HEIGHTS

R. A. Pappert
L. R. Hitney

September 1982

Interim Report for Period May 1981—September 1982

Prepared for Defense Nuclear Agency

Approved for public release; distribution unlimited

NAVAL OCEAN SYSTEMS CENTER
San Diego, California 92152
ADMINISTRATIVE INFORMATION

This work, sponsored by the Defense Nuclear Agency under Program Element 62715H, Subtask S99QAXHB, and NOSC work unit 532-MP20A/VLF, was performed during the period May 1981 through September 1982.

Released by
JH Richter, Head
E M Propagation Division

Under authority of
JD Hightower, Head
Environmental Sciences Department

ACKNOWLEDGEMENTS

The authors are grateful to CH Shellman who developed the basic Runge-Kutta field integration program and to W DeHope for programming assistance.
**A WKB Program for ELF/VLF Earth-Ionosphere Excitation by Sources at Satellite Heights**

This report presents a variation of an earlier program for calculating ELF/VLF excitation of the earth-ionosphere waveguide by point dipoles at satellite heights. The variation consists of implementing WKB formalism developed for the purpose of speeding up the requisite field integrations. Allowance is made for the following:

1. Calculating excitation factors for all electric field components $E_x$, $E_y$, and $E_z$.
2. Calculating excitation by both electric and magnetic point dipoles.
3. Calculating excitation by vertical, horizontal end-on and horizontal broadside configurations.

A mode sum program is also included as an illustration of how the excitation factors are used in field calculations.
ABSTRACT

This report presents a variation of an earlier program [3] for calculating ELF/VLF excitation of the earth-ionosphere waveguide by point dipoles at satellite heights. The variation consists of implementing WKB formalism developed by Budden [2] for the purpose of speeding up the requisite field integrations. Allowance is made for the following: (1) calculating excitation factors for all electric field components $E_x$, $E_y$ and $E_z$; (2) calculating excitation by both electric and magnetic point dipoles; (3) calculating excitation by vertical, horizontal end-on and horizontal broadside configurations. A mode sum program is also included as an illustration of how the excitation factors are used in field calculations.
# CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>1</td>
</tr>
<tr>
<td>I INTRODUCTION</td>
<td>3</td>
</tr>
<tr>
<td>II SUMMARY OF WKB FORMULAS</td>
<td>6</td>
</tr>
<tr>
<td>III FIELD MATCHING</td>
<td>9</td>
</tr>
<tr>
<td>IV EXCITATION FACTORS</td>
<td>12</td>
</tr>
<tr>
<td>V PROGRAM DESCRIPTION</td>
<td>14</td>
</tr>
<tr>
<td>VI SAMPLE INPUT AND OUTPUT</td>
<td>26</td>
</tr>
<tr>
<td>VII MODE SUMS</td>
<td>42</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>47</td>
</tr>
<tr>
<td>APPENDIX A: SATELLITE EXCITATION FACTOR PROGRAM LISTING</td>
<td>49</td>
</tr>
<tr>
<td>APPENDIX B: MODE SUM AND PLOTTING PROGRAM LISTING</td>
<td>109</td>
</tr>
</tbody>
</table>
I. INTRODUCTION

In a recent report [1] WKB formalism developed by Budden [2] for an anisotropic medium was used to speed up calculation of ELF/VLF modal height gains in the ionosphere to heights well in excess of the peak of the F layer. Justification for use of the WKB formalism lies in the fact that for altitudes sufficiently high in the ionosphere (i.e., \( \geq 110 \) km for VLF, \( \geq 150 \) km for ELF under daytime ionospheres and \( \geq 250 \) km for ELF under nighttime ionospheres) the electromagnetic field is in the outgoing whistler mode and the ionospheric gradients are sufficiently weak. In this report the WKB method is appended to an earlier program [3] for calculating excitation of the earth-ionosphere waveguide by point dipoles at satellite altitudes. As with the latter program, provision is made for calculating excitation factors for all electric field components generated by both electric and magnetic point dipoles of arbitrary orientation. It can be expected that the present program will reduce the computer cost (Univac 1100) by about a factor of 1/3 in the lower ELF band (\( \sim 75 \) Hz) and by more than an order of magnitude in the VLF band when the source altitudes are \( \geq 500 \) km. We summarize in the following paragraphs the features necessary for an understanding of the present work.

All height dependent input quantities are referenced to a Cartesian coordinate system \((x, y, z)\) such that the \(z\) axis is taken along the vertical and positive into the ionosphere. The \((x, y, z)\) coordinate system is referred to in this report as the input coordinate system. The transmitter and receiver are located in the \(x, z\) plane with \(z = 0\) being the ground. The source is taken to be imbedded in a semi-infinite, homogeneous, but anisotropic slab with lower boundary termed WKBHT in the input coordinate system. Thus, reflections from above the source are ignored. The primary plane wave field radiated by the elevated antenna at an earth ionosphere eigenangle, \(\theta_n\), is decomposed into downcoming magneto-ionic components and evaluated at \(z = \text{WKBHT}\) in the input coordinate system. Next, two independent full wave starting solutions consistent with the condition of downcoming waves are assumed at an altitude termed LWSTHT in the input coordinate system. LWSTHT must be in the isotropic space between the ground and the lowest level of the ionosphere. The two independent solutions are numerically integrated upward to a level
termed TOPHT in the input coordinate system. At TOPHT the two solutions are combined to yield the proper modal polarization and decomposed into the downcoming whistler component. The full wave whistler component is then matched to the corresponding WKB component and the solution propagated to WKBHT via the WKB formulas. Finally, the strength of the full wave solution is determined by matching to the source generated whistler component previously determined at WKBHT.

The present program is a variant of a fields program [4] which necessitated starting the full wave integrations with outgoing wave and subsequent downward integration. That is in contrast with the present requirement of starting the full wave integrations with down-going wave and subsequent upward integration. To achieve this modification the specie profile and collision frequency read in (as was done in reference [3]) is left unaltered and coordinate transformations effected internally in the program. That is, read in of the latter quantities is referenced to what has been referred to previously as the input coordinate system (z axis positive into ionosphere, z = 0 at the ground). The profiles are read in from WKBHT down to level LWSTHT (or equivalently D) which is often taken to be ground (LWSTHT = 0). The coordinate system and profiles are then transformed by rotating 180° about the x-axis and translating in the z direction by the amount WKBHT. This is referred to as the prime coordinate system and represented by (x', y', z'). Mathematically the transformation is \( z' = -z + WKBHT \). This has the effect that \( TOPHT' = -LWSTHT + WKBHT \), \( LWSTHT' = -TOPHT + WKBHT \) and \( WKBHT' = 0 \), where the left hand side of the equations are (apart from the ') the program names after the coordinate transformation. The transformation from the \((x, y, z)\) system to the \((x', y', z')\) system is effected internally in the program and achieves the goal that the starting full-wave solutions in the transformed system correspond to upgoing wave and that the full wave integrations proceed as in the original full-wave program [4]. Transformation to the primed system is achieved in the subroutine XINPUT.

Basic inputs to the present program must be supplied from an ELF/VLF earth-ionosphere waveguide program such as that of reference [5]. The inputs include mode eigenangles, derivatives of the modal function evaluated at the eigenangles (or related quantities) and the elements of the plane wave ionospheric reflection matrix as defined in reference [2].
For completeness, Budden's WKB formulas [2] are summarized in section II. Formulas used in matching the full wave and WKB field are given in section III. Principal outputs of the program are the excitation factors defined in section IV. Flow of the program and subroutine descriptions are given in section V. Section VI contains a discussion of the card deck arrangement for input along with discussion of a sample output. Application of the excitation factors for calculating mode sums is discussed in section VII where sample mode sum plots are included. Appendix A contains a listing of the excitation factor program. Appendix B contains a listing of the mode sum program used to generate the results shown in section VII.
II. SUMMARY OF WKB FORMULAS

For plane wave incidence of an rf wave on the ionosphere, Maxwell's equations can be written as [6]

\[ e' = -iTe \]  

where the prime denotes \((k^{-1} \partial / \partial z)\), with \(k\) the free space wave number, \(T\) a 4 x 4 matrix given by Budden [2], and \(e\) a column matrix composed of components of the electric \((\mathbf{E})\) and magnetic \((\mathbf{H})\) fields of the rf wave. The transpose of \(e\) is

\[ e^T = (E_x', -E_y', Z_0 H_x', Z_0 H_y') \]  

where \(Z_0\) is the free space impedance. Henceforth the notation \(H_x = Z_0 H_x', H_y = Z_0 H_y\) and \(H_z = Z_0 H_z\) will be used.

The matrix \(T\) has four characteristic roots or eigenvalues, \(q_i\) \((i = 1, 2, 3, 4)\), which satisfy the characteristic equation

\[ \det (T - qI) = 0 \]  

where \(I\) is the unit 4 x 4 matrix. This equation is the Booker quartic. Corresponding to any root \(q_i\) there is an eigencolumn \(P_s(i)\) which satisfies

\[ T P_s(i) = q_i P_s(i) \]  

Let \(S\) be the 4 x 4 matrix whose columns are \(s(i)\). For points where \(S\) is nonsingular, the column matrix \(f\) can be defined as

\[ e = Sf \text{ or } f = S^{-1}e \]  

and it can be shown [2] that the elements of \(f\) satisfy

\[ f'_k = -i q_k f_k + \sum_{j=1}^{4} k_j f_j \]  

\(j, k = 1, 4\)
where

$$\Gamma = -S^{-1}S'$$  \hspace{1cm} (7)$$

The preceding transformation can be carried out only when $S$ is nonsingular. When two roots of the Booker quartic are equal, two of the columns $S^{(i)}$ are usually multiples of each other, and then $S$ is singular. Near such points some of the coupling coefficients $\Gamma_{kj}$ are very large and the points may be points of reflection or points where coupling between two up-going (or downgoing) waves is very strong. The present program cannot be used in such circumstances.

When the species densities and collision frequencies vary slowly enough with height and where no two of the $q_i$ become nearly equal, the terms of $\Gamma$ are small quantities and there is an approximate solution for which the non-diagonal elements of $\Gamma$ are ignored. This solution is associated with one particular root $q_j$ of the Booker quartic. It is [2]

$$f_j = \exp (-ikfZq_jdz + kf^2\Gamma_{jj}dz)$$  \hspace{1cm} (8)$$

and the corresponding field components [in Budden's notation] are

$$\left(E_x, E_y, H_x, H_y\right) = (A_j, F_j)^{-1/2} \left(a_3q_j + a_4, -A_j, q_jA_j, a_5q_j + a_6\right) \times \exp (-ikfZq_jdz + kf^2\Gamma_{jj}dz)$$  \hspace{1cm} (9)$$

where

$$a_1 = -(T_{11} + T_{44}) \hspace{2cm} a_4 = T_{14}T_{42} - T_{12}T_{44}$$

$$a_2 = T_{11}T_{44} - T_{14}T_{41} \hspace{2cm} a_5 = T_{42}$$

$$a_3 = T_{12} \hspace{2cm} a_6 = T_{41}T_{12} - T_{11}T_{42}$$

$$A_j = q_j^2 + a_1q_j + a_2$$

$$F_j = 2q_jA_j + (q_j^2 - T_{32})(2q_j + a_1) - (a_3b_5 + b_3a_5)$$  \hspace{1cm} (13)$$
\[ 2\Gamma_{jj} = (A_j F_j)^{-1} q_j^2 \{ a_3 b_5' - a_3 b_5 + a_5 b_3' - a_5 b_3 \} \\
+ a_{4,6} b_{6}' - a_{4,6} b_6' + a_{6,4} b_{4}' - a_{6,4} b_4' + q_j (a_3 b_6' - a_3 b_6 + a_4 b_5' - a_4 b_5 + a_5 b_4' \\
- a_5 b_4' + a_6 b_3' - a_6 b_3') \} \]

(14)

In these equations the \( T_{ij} \)'s are the elements of the \( T \) matrix given by Budden p 389.

The essence of the program documented in this report is the implementation of equation (9) for altitudes exceeding a height termed TOPHT in the input coordinate system. The mode extracted from the magneto-ionic set is the least attenuated downcoming wave. Runge-Kutta integrations are used to calculate two independent full wave solutions at TOPHT from starting conditions at LWSHT in the input coordinate system. The full-wave solutions at TOPHT are combined to satisfy the mode polarization condition and decomposed into magneto ionic components. The downcoming whistler component is then matched to the corresponding WKB component and the fields carried to WKBHT via equation (9).
III. FIELD MATCHING

As described in the introduction, it is necessary in the primed coordinate system to integrate two independent field solutions from the level \( z' = \text{TOPHT}' = -\text{LWSTHT} + \text{WKBHT} \) to the level \( z' = \text{LWSTHT}' = -\text{TOPHT} + \text{WKBHT} \). The initial solution vectors correspond in the prime system (at level \( \text{TOPHT}' \)) to outgoing waves in vacuum and are taken to correspond to vertical and horizontal polarizations. The initial solution vectors are calculated in the subroutine WF INIT and are denoted by the matrix \( P \) as shown below

\[
\begin{align*}
P(1,1) &= E_x = \cos(\theta_n) = C. \\
P(1,2) &= E_y = 0. \\
P(2,1) &= -E_y = 0. \\
P(2,2) &= -E_x = 1. \\
P(3,1) &= H_x = 0. \\
P(3,2) &= H_x = \cos(\theta_n) = C. \\
P(4,1) &= H_y = 1. \\
P(4,2) &= H_y = 0.
\end{align*}
\] (15)

where \( E_j \) and \( H_j \) represent the \( j^{th} \) component of the electric and magnetic fields of the rf wave in the prime system and \( \theta_n \) represents the eigenangle for the \( n^{th} \) earth ionosphere waveguide mode. The initial solutions are integrated to the level \( z' = \text{LWSTHT}' \) via the Runge-Kutta integration scheme implemented in the subroutine WF INTG. At \( z' = \text{LWSTHT}' \) each independent full-wave solution, \( P_i \) is decomposed into four magneto-ionic components as follows

\[
P_i = \sum_{j=1}^{4} \beta(i,j) S(j); i = 1,2
\] (16)

where \( S(j) \) are the magneto-ionic solution vectors. Thus,

\[
\beta(i,j) = S^{-1}(j)P_i
\] (17)

where

\[
S^{-1}(j)S(k) = \begin{cases} 0 & j \neq k \\ 1 & j \neq k \end{cases}
\] (18)
Because of the sorting achieved in the subroutine WF SORT, the upgoing (in the prime system) whistler mode is \( S(2) \). The total whistler component is structured from a linear combination of the components associated with the two independent full wave solutions. The two independent solutions are combined in such a way that they satisfy the modal polarization condition at LWSTHT in the input coordinate system

\[
f = \frac{E_y}{H_y} = \frac{1 - \nu R_{nn} \bar{R}_{nn}}{1 - \nu R_{nn} \bar{R}_{nn}} = \frac{\nu R_{nn} \bar{R}_{nn}}{1 - \nu R_{nn} \bar{R}_{nn}}
\]

In this equation \( R \) is the plane wave reflection coefficient associated with everything above LWSTHT and \( \bar{R} \) is the plane wave reflection coefficient from everything below LWSTHT in the input coordinate system. Consistent with convention, the first subscript on \( R \) or \( \bar{R} \) applies to the polarization of the incident wave whereas the second subscript applies to the polarization of the reflected wave, with \( \nu \) denoting vertical polarization and \( \perp \) denoting horizontal polarization.

The total upgoing (in the prime system) whistler component, \( S_T(2) \), obtained from decomposition of the two full wave solutions at \( z' = LWSTHT' \), is then given by

\[
S_T(2) = (\beta(1,2) - \beta(2,2)f)S(2)
\]

Note that the minus sign before \( \beta(2,2) \) follows from the fact that \( P(2,2) = -E_y \) (see equation (15)). Equation (9) with \( j = 2 \) is normalized such that it equals equation (20) at \( z' = LWSTHT' \) and the solution propagated to \( z' = WKBHT' = 0 \) via the WKB formula. If the solution at \( z' = 0 \) is denoted by \( [\beta(1,2) - \beta(2,2)fS(2)]_{z'=0} \), then the strength and final unknown, \( A_m \), of the full wave solution is determined by [7, 8]

\[
A_m = \frac{BP_{m}(1,2)e^{ikQ(2)Z_0'}}{(Q(2) - Q(1))(Q(2) - Q(3))(Q(2) - Q(4))[\beta(1,2) - \beta(2,2)fS(1,2)]_{z'=0}}
\]

where \( S(I,2)(I = 1,2,3,4) \) represents the \( I \)th field component of the magneto-ionic mode (i.e., \( I = 1 \cdot E_x, I = 2 \cdot -E_y, I = 3 \cdot H_x, I = 4 \cdot H_y \)). Any of
the field components can be used to determine $A_m$. The program uses $H_y$ or $I = 4$. The free space wave number is denoted by $k$, the Q's are Booker quartic solutions, and $z'_0$ is the transmitter location in the prime coordinate system. $BP_m(I,J)$ represents an amplitude factor associated with the decomposition of the primary source into an upgoing (in the prime system) magneto-ionic component ($J = 2$) associated with the eigenvalue $\theta_n$. It is determined from equations (17), (22), (24), (25), and (56) of reference [8] and will not be reproduced here. The subscript $m$ on $A$ and $BP$ denotes the source type and configuration in a way which will be described in the following section.

The calculations that have been described in this section are effected in the subroutine WF BNDY (or in the entry DECOMP in WF BNDY).
IV. EXCITATION FACTORS

In this section the excitation factors, \( \lambda \), which have been programmed are summarized and in section VII their usage in mode sum calculations is discussed. The formulas have been derived in reference [8] and include "height gain" effects associated with transmitter and receiver. In that sense they depart from the conventional definition. The formulas programmed differ from those of reference [8] only in choice of sign and by virtue of using here the complement of the eigenangle used in reference [8]. The formulas are:

\[
\lambda_1^m = \frac{2CS}{\partial F/\partial \phi} e^{-i\pi/4} (1+\hat{R}_\|)H_y(z_R)((1-\hat{R}_\perp \hat{R}_\perp)h_\| e_y + \hat{R}_\perp \hat{R}_\perp e_y^m)
\]

\[
\lambda_2^m = \frac{2C}{\partial F/\partial \phi} e^{i3\pi/4} (1+\hat{R}_\perp)E_y(z_R)((\hat{R}_\perp \hat{R}_\perp)h_\| e_y + \hat{R}_\perp \hat{R}_\perp e_y^m)
\]

\[
\lambda_3^m = \frac{2C}{\partial F/\partial \phi} e^{-i\pi/4} (1+\hat{R}_\|)E_x(z_R)((1-\hat{R}_\perp \hat{R}_\perp)h_\| e_y + \hat{R}_\perp \hat{R}_\perp e_y^m)
\]

(22)

In the above, \( F \) is the modal function (i.e., the determinant of \( 1 - RR \)). The superscript \( m \) denotes the source while the subscripts on \( \lambda \) denote the component of the electric field at the receiver for which the excitation factor applies. The convention governing \( m \) and \( j \) is:

\[
\lambda_j^m =
\]

\( m = \) 1 vertical electric dipole
\( m = \) 2 horizontal electric dipole broadside
\( m = \) 3 horizontal electric dipole end fire
\( m = \) 4 vertical magnetic dipole
\( m = \) 5 horizontal magnetic dipole broadside
\( m = \) 6 horizontal magnetic dipole end fire

The \( R \) and \( \tilde{R} \) in equation (22) have been defined in the previous section. The quantities \( h_y^m \) and \( e_y^m \) are the full wave \( y \) components of the magnetic and electric fields of the rf wave and are dependent upon the type and location of
the source as indicated by the superscript \( m \). The dependence on altitude of the receiver, \( z_R \), in the input coordinate system is given by the functions \( H_y(z_R), E_y(z_R), \) and \( E_x(z_R) \). A complete description of these functions is given in reference [3]. Finally, \( S \) and \( C \) are the sine and cosine, respectively, of an eigenangle, \( \theta_n \), and \( \partial F/\partial \theta \) is the derivative of the mode function evaluated at \( \theta_n \).
V. PROGRAM DESCRIPTION

This section describes the subroutines in the SATELLITE program listed in appendix A. Many of the subroutines are only slight modifications of those given in reference [3]. The subroutines WKB, WKBVAR, QGAMMA and DDKXMT have been added for the purpose of implementing the WKB formalism. Principal output of the program is the excitation factors defined in section IV. A chart showing the essential structure of the SATELLITE program follows on pages 22 through 25.

SUBROUTINE MAIN

MAIN first calls for input of ionic species data in XINPUT and for computation of height gain components and excitation factors via WAVFLD. After executing WAVFLD, excitation factors are available in the array EFIELD (M,J) where M is the source designation and J the field component designation consistent with the convention given in section IV.

SUBROUTINE XINPUT (ISTART, ISTOP)

XINPUT controls read-in of input parameters via NAMELIST statements and ionic species densities and collision frequencies as a function of altitude. Common areas are set up as required. ISTART is set to 1 before the first call to XINPUT and to 0 upon subsequent calls. ISTART = 1 implies all necessary data are to be read in and ISTART = 0 signals that previously read data are to be updated, with all unspecified parameters remaining unchanged. If a value ISTOP = 0 is returned by XINPUT, then more input data are specified in the data deck for subsequent calls to WAVFLD, whereas if a value ISTOP = 1 is returned, the data read were the last data in the data deck, so that XINPUT should not be called again. XINPUT effects the transformation from the input coordinate system (x, y, z) to the prime coordinate system (x', y', z').

The data deck is divided into several parts, each of which is marked by an identifier card with the identifier DATUMFOL, PROFILE, COLLFREQ, QUIT or STOP. Each of these identifiers is described in the following section.
SUBROUTINE WAVFLD (EX, EY, EZ, HX, HY, HZ)

WAVFLD calls for Runge-Kutta integration of two independent (corresponding to outgoing wave in the prime system at TOPHT') field solutions from TOPHT' to LWSTHT' at DELHT increments. The two solutions are combined to satisfy the modal polarization condition and decomposed into magneto-ionic components at LWSTHT'. At LWSTHT' the full wave whistler component is matched to the corresponding WKB component and the solution carried to WKBHT' via WKB formulas. At WKBHT' the strength of the solution is determined for each source and orientation. WAVFLD then calls for the back substitution of normalizing values which have been saved in WF STOR. The excitation factors are then calculated in WAVFLD and placed in the array EFIELD(M,J). The field components placed at DELHT intervals in the arrays EX, EY, EZ, HX, HY, HZ are extraneous output in the present application.

SUBROUTINE WFINTG (TOPHT', LWSTHT', DELHT, IFLAG)

WFINTG performs Runge-Kutta integration of the two solution vectors in P from TOPHT' to LWSTHT'. Numerical solutions are saved at DELHT increments. Accuracy is maintained by continually adjusting the step size used in the numerical integration. The current step size (call it h) is used to obtain an estimate of P, and then a better estimate is obtained by using two integrations with step size h/2. If the two solutions agree within an error of PRECSN (an input parameter normally set to 3.D-5), the better estimate is accepted. The step size is automatically decreased to h/2 if the two estimates differ by more than PRECSN, and the integration is repeated. If the error is significantly greater than PRECSN, a step size h/2 is used. If the error is significantly less than PRECSN, the step size 2h is used if it also yields an error less than PRECSN. These tests thus form an automatic step size correction. In this program the call to WFINTG sets IFLAG equal to zero.

ENTRY INIT T

INIT T is an entry in subroutine TMTRX. INIT T sets up height independent values to be used in T matrix calculation. These include the
internally set flag ISO (ISO = 1 for isotropic calculation, 0 otherwise). Also set are the angular radio frequency, the wave number, direction cosines of the geomagnetic field, the complex sine and cosine of THETA.

SUBROUTINE TMTRX(HT')

TMTRX computes the value of the T matrix and a specific height HT' (as usual the prime simply implies that HT is referred to the prime coordinate system). The T matrix depends upon input ionospheric parameters (species density, collision frequency, angle of propagation, magnetic field, etc.). The susceptibility matrix, M, for each species in the ionosphere is computed, the effect of earth curvature is included and the T matrix is computed from the susceptibility matrix elements. The equations used to evaluate M and T are given in Clemmow and Heading [6].

SUBROUTINE WFDENS (HT', EN, COLL)

WFDENS computes the species density (returned in EN) and collision frequency (returned in COLL) at height HT' for up to five species in the ionosphere. EN and COLL are determined from corresponding profiles in the common field WFPROF. LHT and MHT are integer values which indicate which profile values are to be interpolated to find values at the height HT'. The EN (or COLL) values are given by logarithmic interpolation of the profile values of heights ENHT(LHT + 1) and ENHT(LHT) or (COLLHT(MHT + 1) and COLLHT(MHT)).

SUBROUTINE WF INIT(P)

WF INIT sets the two starting field vectors P, at TOPHT' subject to the condition of outgoing wave in the prime system. The two starting solutions correspond to TE and TM polarizations.

SUBROUTINE PDERIV(P,DPDH)

PDERIV computes the height derivatives of the two field vectors in P according to Clemmow and Heading [6]. The derivative is returned in DPDH.
SUBROUTINE XFER (A, B, M)

Transfers the N element array A into B.

ENTRY WF STOR

This is an entry in WFSCAL where certain values obtained during integration through the ionosphere are saved for later use. The solution matrix P, the height for which P is a solution and a height integer index are saved along with orthogonalization and normalization values OSUM, APROD and BPROD. In addition, values of the susceptibility matrix elements M31, M32 and M33, which are needed to compute the EZ component of the electric field, are saved at each height.

SUBROUTINE WF STEP (P, DPH, HT', DELHT, IFLAG)

WF STEP numerically advances the solution matrix P, using the derivative DPH, from HT' to HT'-DELHT by the Runge-Kutta method. IFLAG, set internally, controls the calculations at intermediate points between HT' and HT'-DELHT at which evaluations are required for comparison of the second and fourth order Runge-Kutta integrations.

SUBROUTINE WF SCAL (PP, IFLAG)

WF SCAL normalizes and orthogonalizes the solution vectors PP according to the formulas of reference [4]. This scaling must later be removed to yield correct unscaled solutions. Control for calculating the quantities needed for removal of the scaling is the internally set IFLAG. Calling WF SCAL sets IFLAG = 0 in this program.

ENTRY DECOMP(HT')

This is an entry in WF BNDY. The two full wave solution vectors, P, are decomposed at HT' = LWSTHT' into upgoing (in the prime system) magneto-ionic components and combined to satisfy the modal polarization condition given in equation (19).
SUBROUTINE EIGVAL (A, SIG)

EIGVAL computes the eigenvalues (returned in SIG) or the arbitrary 4\times4 complex matrix A. The characteristic polynomial \( P(\lambda) \) of A is determined by explicitly computing \( \det(A-\lambda I) = P(\lambda) \). The roots of this quartic polynomial (as computed in closed form, see subroutine QUARTC) are the desired eigenvalues of A. If the ionosphere is isotropic, the eigenvalues of A are computed as the roots of two quadratics.

SUBROUTINE QUARTC (FOUR B3, SIX B2, FOUR B1, B0, Q)

QUARTC computes the four roots of the polynomial \( Q^4 + \text{FOUR B3} \times Q^3 + \text{SIX B2} \times Q^2 + \text{FOUR B1} \times Q + B0 \) in closed form [9]. Up to ten applications of Newton's iteration are then performed to improve the accuracy of the roots, if necessary.

SUBROUTINE EIGVEC (A, SIG, VEC)

EIGVEC computes the four eigenvectors of A (returned in the four columns of the matrix VEC). The eigenvalues of A are assumed given in SIG. Each eigenvector \( X_i \) is computed as the solution of \( (A - \sigma_i I)X_i = 0 \). Since \( \det(A - \sigma_i I) = 0 \), this system has a solution if one element of \( X_i \) is chosen arbitrarily. In order to reduce the special isotropic case, the first element of \( X_1 \) and \( X_4 \) are arbitrarily set to \( A_{23} \). The other three elements of each vector are then determined as the solution of a linear system in three unknowns. If the ionosphere is isotropic, the eigenvectors (corresponding to the quadratic roots described previously) are computed in simplified form.

SUBROUTINE WF SORT (Q, A IFAIL)

WF SORT arranges the eigenvalues Q and eigenvectors A of the Booker quartic so that they occur in the order of upgoing fast (evanescent), upgoing slow (travelling or whistler component), downgoing slow, and downgoing fast in the prime coordinate system. If the eigenvalues are too large in magnitude or if they cannot be ordered IFAIL is set to 1 and an error message results.
SUBROUTINE CIMVER (A, AINV, N, NDIM, ERR)

CIMVER computes the inverse (returned in AINV) of the N×N matrix A. NDIM is an integer variable which must be greater than or equal to N. ERR is the estimated relative error of the inverse matrix.

SUBROUTINE CLINEQ (A, B, X, N, NDIM, IFLAG, ERR)

CLINEQ computes the solution of simultaneous linear equations with complex coefficients. That is, it solves the matrix A * X = B for the vector X of length N, given the matrix A of size N by N and the vector B of length N. The matrix A is destroyed by CLINEQ. NDIM is an integer variable which must be greater than or equal to N. IFLAG is an integer variable normally set to 0. ERR indicates the relative error in the computed solution of vector X.

SUBROUTINE RBARS (C, S, RBAR11, RBAR22)

RBARS calculates the plane wave reflection coefficients \( \bar{R}_{11} \) and \( \bar{R}_{22} \) looking towards the ground from LWSTHT in the input system. Evaluations are in terms of modified Hankel functions of order 1/3. The cosine (C) and sine (S) of the eigenvalue, THETA, are used in the \( \bar{R} \) determination.

SUBROUTINE MDHNKL (Z, H1, H2, HIPRME, H2PRME)

MDHNKL calculates for argument Z two independent solutions (H1 and H2) and their derivatives (HIPRME, H2PRME) of Stokes' equation by methods described in reference [10]. H1 and H2 are the modified Hankel functions of order 1/3 referred to above.

SUBROUTINE WKB

If STPFLG.NE.1, WKB extends calculation of the solution vector from LWSTHT' to WKBHT' by means of the formulas in section II. Integration of the phase factors \( q_j \) and \( \Gamma_{jj} \) (j index for upgoing, in prime system, whistler mode) is performed by Simpson rule routine with fixed step size that can be either input to the program or set internally.
SUBROUTINE WKVAR

If STPFLG.EQ.1, WKBVAR extends calculation of the solution vector from LWSHT' to WKBHT' via the WKB formulas of section II. Although generally requiring more CAU time this is the preferred option since it involves checks on step size. Integration of the phase factors $q_j$ and $\Gamma_{jj}$ ($j$ index for upgoing, in prime system, whistler mode) is performed by a Simpson rule routine which maintains precision by adjustment of the step size much like the Runge-Kutta step size adjustment in WFINTG.

ENTRY INITDT

This is an entry in DDKXMT where all height independent quantities are initialized before extending field calculations from LWSHT' to WKBHT' via the WKB method.

SUBROUTINE Q GAMMA (HT', DELHT, LWSTHT', WKBHT', Q, GAMMA)

Q GAMMA determines the Booker quartic solutions for the least attenuated upgoing (in the prime system) magneto-ionic component at LWSTHT' and computes the $a_i$'s, $b_i$'s, $A_i$, $F_i$ and $\Gamma_{ii}$ according to formulas of section II. Full-wave solutions are matched at LWSTHT' to the WKB solutions. At other heights and up to WKBHT' coefficients of the exponentials in equation (9) of section II are calculated along with coefficients required for calculating the EZ and HZ fields.

SUBROUTINE DDKXMT

DDKXMT calculates susceptibility matrix elements, $M$, the $I$ matrix elements and their derivatives with respect to height in the height range $LWSHT' \leq HT' \leq WKBHT'$.

WF BNDY(B)

This computes coefficients $B_{m1}'$--see equation (21)--for each source and orientation at WKBHT'. The coefficients are used to determine the strength of the full wave solution which has been carried from within the guide to WKBHT' by the combination of full wave and WKB methods.
ENTRY WF LOAD

WF LOAD is an entry in WF SCAL which makes available for the purpose of back substitution into the full wave solutions the orthogonalizing and normalizing coefficients saved in WF STOR.

ENTRY HT GAIN

This is an entry in RBARS where the quantities required to calculate the properly normalized height gains at the receiver altitude are evaluated.
CALLS FOR INPUT DATA VIA XINPUT.

Calls for computation by WAVFLD of excitation factors for the ELF/VLF electric field components beneath the ionosphere produced by electric and magnetic dipoles at satellite altitudes.

**XINPUT**

Input electron and ion species densities and collision frequencies as a function of altitude in the input coordinate system (z positive into the ionosphere). Converts input to the primed coordinate system (z' positive towards the ground).

**WAVFLD**

Calls for Runge-Kutta integration of two independent field solutions from TOPHT' to LWSTHT' at DELHT increments. The two solutions are combined to satisfy the modal polarization condition and decomposed into magneto-ionic components at LWSTHT'. WAVFLD calls for the back substitution of orthogonalizing and normalizing values (saved as data in WFSTOR). It calls for the extension of the fields from LWSTHT' to WKBHT' via Budden's WKB formalism. At WKBHT' the field components are matched to the corresponding components generated by each source. Excitation factors for the ELF/VLF electric field components are then computed for an input receiver altitude which must be within the isotropic space between the ground and ionosphere.

MAIN

Calls for input data via XINPUT.
Calls for computation by WAVFLD of excitation factors for the ELF/VLF electric field components beneath the ionosphere produced by electric and magnetic dipoles at satellite altitudes.
WF INTG

WF INIT
- Sets the starting field vectors, \( P \), at \( 10 \Phi \), subject to the condition of outgoing wave in the prime system.

WF DENS
- Computes the density and collision frequency of each charged species by logarithmic interpolation of the corresponding profile.

WF MIRX
- Calculates \( I \) matrix and susceptibility matrix, \( M \), elements.

XFER(P, PSAVE, 16)
- Transfers the 16 (8 real, 8 imaginary) element array \( P \) to \( P \) SAVE.

WF STEP
- Increments the solution matrix, \( P \), from the last height to the next height using Runge-Kutta integration.

P DERIV
- Calculates height derivatives of the solution vectors, \( P \).

WF STOR
- This is an entry in \( WF \) SCAL. Field values, orthogonalization and normalization factors are saved at heights between \( 10 \Phi \) and \( \Phi \) at DELHT intervals in the primed system.

WF SCAL
- This scales and orthogonalizes the solution vectors contained in \( P \). This scaling must be removed to yield correct (unscaled) solutions.

XFER
This is an entry in DDKXMT. All height independent quantities are initialized before extending field calculations from LWSTHT' to WKBHT' via Budden's WKB formulas for an anisotropic medium.

Beginning at LWSTHT' in the prime system computes $a_j'$s, $b_j'$s, $A_j$, $F_j$, $F_{jj'}$, $a_j''$s, $b_j''$s ($j$ is index for least attenuated outgoing wave in the prime system) according to the formulas of section II. Matches full wave fields of LWSTHT' to WKB solutions.

Calculates susceptibility matrix elements, $M$, the $T$ matrix elements and the derivative of the $M$, $T$ elements with respect to height.

WF DENS
VI. SAMPLE INPUT AND OUTPUT

A. INPUT

Altitude independent parameters, species densities and collision frequencies as a function of altitude are supplied via an input data deck on a standard input unit. Read-in occurs in the subroutine XINPUT. The data deck is divided into several parts, each of which is marked by control cards DATUMFOL, PROFILE, COLLFREQ, QUIT, or STOP. Each of these control cards is described below:

(1) DATUMFOL

DATUMFOL - is a control card signifying that input for the namelist DATUM follows.

&DATUM - initiates reading of input data which do not vary with altitude. The data is read in namelist input format. All altitude related quantities are referenced to the input coordinate system (z positive into the ionosphere, ground at z = 0). These data are:

THETA - complex angle of incidence in degrees as measured at height H. THETA must be applied from a waveguide program (e.g., ref. [5]).

FREQ - frequency in kHz.

TOPHT - height to which full wave integrations are carried (km).

LWSITH - beginning height for full wave integrations (km). Also equivalent to level D where mode equation is evaluated.
WKBHT - lower boundary of homogeneous, anisotropic slab in which the transmitter is imbedded (km). Also height to which field solution is propagated via WKB formalism from TOPHT.

DELHT - height increment in km's at which field solutions are saved.

PRECSN - accuracy to be maintained locally in the numerical integrations. Usually taken to be the default value of 3.0E-5.

AZIM - azimuth of propagation path in degrees, measured clockwise from geomagnetic north.

CODIP - codip of geomagnetic field in degrees.

MAGFLD - geomagnetic field strength in webers/square meter.

COEFNU(5) - coefficient of exponential form of collision frequency (if not specified by profile). Up to five values may be specified. One for each species.

EXPNU(5) - exponent of exponential form of collision frequency (if not specified by profile). Up to five values may be specified. One for each species.

ALPHA - earth curvature coefficient in inverse km. Default is 3.14E-4.

SIGMA - ground conductivity in siemens/meter. Default value is seawater value of 4.64.

EPSLON - ground permittivity in farads/meter. Default value is 7.172015D-10. This corresponds to a dielectric constant of 81 for seawater.
EPSLNO - permittivity of free space (8.8544E-12) in farads/meter.

H - altitude in km at which modified refractive index is unity. Eigenangles are referenced to this altitude. H must be consistent with the H setting in the waveguide program which supplied the eigenangles.

TXHT - transmitter altitude (km).

RXHT - receiver altitude (km).

ITR - integer flag which should be set to zero in this program.

RMAG(4) - magnitude of the plane wave reflection elements of R. \( R(1) = \pi R_u, R(2) = \pi R_\perp, R(3) = \pi R_u, R(4) = \pi R_\perp \). These elements are referenced to level D (or equivalently LWSTHT) in the input coordinate system and must be supplied from a waveguide program such as that of reference [5].

RANG(4) - phase of the plane wave reflection elements of R. These elements are referenced to level D (or equivalently LWSTHT) in the input coordinate system and must be supplied from a waveguide program such as that of reference [5].

XTRMAG - magnitude of the ground based vertical dipole excitation factor for the vertical electric field, \( E_z \). This, along with XTRANG, is used for the purpose of obtaining the derivative of the modal equation, \( df/d\theta \). XTRMAG must be supplied from a waveguide program such as that of reference [5].
XTRANG  - phase of the ground based vertical dipole excitation factor for the vertical electric field, Ez. This, along with XTRMAG, is used for the purpose of obtaining the derivative of the modal equation, dF/dθ. XTRANG must be supplied from a waveguide program such as that of reference [5].

NUMDIV  - number of divisions into which DELHT is divided when using subroutine WKB (i.e., when STPFLG ≠ 1) with fixed input step size. NUMDIV must then be a multiple of 2. If STPFLG ≠ 1 and NUMDIV = 0 then the program will determine a constant step size to be used with the Simpson rule integration. It is repeated here that the preferred option is with STPFLG = 1 because of the checks on the adequacy of the step size.

STPFLG  - integer flag which must be set to 1 if the recommended WKBVAR subroutine is to be used to effect the WKB integrations. Other options are discussed above (NUMDIV).

NRSPEC  - number (integer) of species in the ionosphere. Can take on values up to 5. Default value is 1.

CHARGE(5)- sign of charge of each species in proton units. For an electron, the charge is -1.0. Default values are (-1.0, 1.0, -1.0, 1.0, -1.0).

RATIO(5)- mass of each species relative to mass of an electron. Default values are (1.0, 5.8D4, 5.8D4, 5.8D4, 5.8D4).

&END   - signifies end of the DATUM namelist input.
PROFILE - is a control card initiating reading of the ionospheric profile cards. All altitudes and related quantities are referenced to the input coordinate system (z positive into the ionosphere, ground at \( z = 0 \)). The control card PROFILE is followed by an alpha-numeric card which is used to identify the profile. The profile is input starting at the top of the ionosphere (i.e., WKBHT). The cards must be input in descending order by height. The profile cards contain the height in kilometers and the species densities in particles per cubic centimeter. A maximum of five species can be specified. In the special case of three species, only two are specified on the card. The first is assumed to be electrons and the second is positive ions. The third species, negative ions, is calculated by subtracting the electron density from the positive ion density. All three species are listed on the computer printout. If the value of any species density is less than or equal to zero, it is set in the program to 1.0E-40. The heights are punched in the form xxx.xx with the decimal point in column 5, and the densities are punched in the form x.xxD+xx with the decimal points in column 15, 25, etc. All species data must be specified except for the special case discussed above. The end of the profile is indicated by a dummy height of 999.99.
(iii) COLLFREQ

COLLFREQ - is a control card initiating reading of the collision frequency profile. All altitudes and related quantities are referenced to the input coordinate system (z positive into the ionosphere, ground at z = 0). The control card COLLFREQ allows for using nonexponential collision frequencies. A strictly exponential collision frequency may be specified in namelist input by the variables COEFNU and EXPNU. If a collision frequency profile deck is used it overrides COEFNU and EXPNU. Collision frequencies for all species must be input. The card preparation is just as above with the species density values replaced by collision frequency values (in collisions/sec) on the cards. The end of the profile is indicated by a dummy height of 999.99.

(iv) QUIT

QUIT - is a control card which indicates the end of input data for a call to WAVFLD. After calling WAVFLD, XINPUT can be read and used as data for a subsequent call to WAVFLD. This allows several runs to be made with one input deck. Note that only those data which are changed need be specified after the card QUIT.

(v) STOP

STOP - control card which indicates that there are no more input data and that the run is to be terminated after the next call to WAVFLD. Note that if QUIT is encountered ISTOP is set to zero, but if STOP is encountered ISTOP is set to one.
A schematic of an input deck is shown on page 34, and one actual sample input is shown on pages 35 through 37.

B. OUTPUT

The output shown on pages 38 through 41 corresponds to the first DATUMFOL shown on page 36. The output corresponds to the STPFLG = 1 setting. Output corresponding to the additional DATUMFOL have not been shown simply to cut down on the number of pages.

First come NAMELIST and profile printouts in the input coordinate system (z positive into the ionosphere, ground at z = 0). Natural logarithms of the profiles in the prime system are also printed. Next follows output which gives the number of Runge-Kutta integration steps used in WAVFLD during the course of integrating from TOPHT' to LWSTHT' as well as the smallest and the average step sizes used in the WKB extension from LWSTHT' to WKBHHT'. This is followed by the principal output; two nine element arrays which are the excitation factors for both electric and magnetic dipole transmitters. The column headings indicate excitation factors for the z, y and x electric field components in the input coordinate system. The row labels indicate excitation by vertical (V), horizontal broadside (HB) and horizontal end-on (HE) point dipoles of either electric or magnetic type.

We list below a comparison at 20 kHz between the excitation factors obtained using full wave (FW) methods up to 500 km as opposed to using WKB formalism between 125 km and 500 km. The agreement will be seen to be excellent.

**ELECTRIC DIPOLE TRANSMITTER**

<table>
<thead>
<tr>
<th></th>
<th>Z MAG</th>
<th>ANG (RAD.)</th>
<th>Y MAG</th>
<th>ANG (RAD.)</th>
<th>X MAG</th>
<th>ANG (RAD.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>WKB</td>
<td>.913-4</td>
<td>-1.504</td>
<td>1.18-7</td>
<td>-2.234</td>
<td>.460-7</td>
</tr>
<tr>
<td></td>
<td>FW</td>
<td>.913-4</td>
<td>-1.485</td>
<td>-1.18-7</td>
<td>-2.215</td>
<td>.460-7</td>
</tr>
<tr>
<td>HB</td>
<td>WKB</td>
<td>.809-4</td>
<td>1.358</td>
<td>.104-7</td>
<td>.628</td>
<td>.408-7</td>
</tr>
<tr>
<td></td>
<td>FW</td>
<td>.808-4</td>
<td>1.377</td>
<td>.104-7</td>
<td>.647</td>
<td>.407-7</td>
</tr>
<tr>
<td>HE</td>
<td>WKB</td>
<td>.816-4</td>
<td>-.243</td>
<td>.105-7</td>
<td>-.973</td>
<td>.412-7</td>
</tr>
<tr>
<td></td>
<td>FW</td>
<td>.816-4</td>
<td>-.224</td>
<td>.105-7</td>
<td>-.954</td>
<td>.411-7</td>
</tr>
</tbody>
</table>
### Magnetic Dipole Transmitter

<table>
<thead>
<tr>
<th></th>
<th>Z MAG</th>
<th>ANG (RAD.)</th>
<th>Y MAG</th>
<th>ANG (RAD.)</th>
<th>X MAG</th>
<th>ANG (RAD.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WKB</td>
<td>.785-4</td>
<td>2.928</td>
<td>.101-7</td>
<td>2.198</td>
<td>.396-7</td>
<td>.573</td>
</tr>
<tr>
<td>FW</td>
<td>.785-4</td>
<td>2.946</td>
<td>.101-7</td>
<td>2.216</td>
<td>.396-7</td>
<td>.591</td>
</tr>
<tr>
<td>HB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WKB</td>
<td>.279-2</td>
<td>1.358</td>
<td>.360-6</td>
<td>.628</td>
<td>.141-5</td>
<td>-.996</td>
</tr>
<tr>
<td>FW</td>
<td>.279-2</td>
<td>1.377</td>
<td>.360-6</td>
<td>.647</td>
<td>.141-5</td>
<td>-.978</td>
</tr>
<tr>
<td>HE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WKB</td>
<td>.279-2</td>
<td>-.212</td>
<td>.360-6</td>
<td>-.942</td>
<td>.141-5</td>
<td>-.567</td>
</tr>
<tr>
<td>FW</td>
<td>.279-2</td>
<td>-.194</td>
<td>.360-6</td>
<td>-.924</td>
<td>.141-5</td>
<td>-.549</td>
</tr>
</tbody>
</table>
EXAMPLE OF INPUT DECK

DATUMFØL

&DATUM
(input data)
&END

PROFILE
(profile cards for each specie)

999.99

COLLFREQ
(coll. frequency cards for each specie)

999.99

QUIT (end of input for this run)

DATUMFØL

&DATUM
(changes in input data)
&END

PROFILE
(specify entire new profile deck)

999.99

QUIT (end of input for this run)

DATUMFØL

&DATUM
(changes in input data)
&END

STØP (end of data deck)
SAMPLE INPUT

1       DATAMFILE
2       ADATU:
3       T-H'T-A = (9,300, -6,297),
4       RVV=37, 4848, 79127, 71191,
5       TVV=1 1, 23340, 325, 6664, 256, 47899, 265, 18541,
6       XTRANX = 2.2, XTRANX = 2.3, XTRANX = 1.755,
7       FEX=20.0,
8       ID=27,
9       TSTDHT=125, LWSHT=100, WKBHT=500, DELHHT=25,
10      PRESN=3E-4,
11      AZIM=276.5, IP=39.0, MAGFLD=4.31E-5,
12      SCLON=1.64, FSLON=17.7, 172018.5E-10,
13      EPSIN=88.34D-11,
14      RXHT=500, RXHT=0,
15      HO=NS,
16      ENS1V=0,
17      SPPLG=1,
18      RSPEC=1,
19      KEND
20      PROFILE 1
21      GLOBAL NIGHTTIME IONOSPHERE (SAT. NIGHT ABOVE 99, H'87 BELOW 99)
22      500.00  2.000+05
23      400.00  4.000+05
24      300.00  4.300+05
25      350.00  4.400+05
26      320.00  4.300+05
27      250.00  1.000+05
28      225.00  5.000+03
29      220.00  3.200+03
30      210.00  1.600+03
31      200.00  1.000+03
32      190.00  6.500+02
33      160.00  4.600+02
34      140.00  3.450+02
35      130.00  2.820+02
36      125.00  2.600+02
37      120.00  2.500+02
38      115.00  2.550+02
39      140.00  2.800+02
40      130.00  3.700+02
41      120.00  5.800+02
42      112.00  1.100+03
43      110.00  1.300+03
44      100.00  1.700+03
45      90.00  1.800+03
46      80.00  1.900+03
47      70.00  2.000+03
48      60.00  1.450+03
49      50.00  1.900+03
50      40.00  1.800+02
51      30.00  2.000+02
52      20.00  3.600+02
53      10.00  3.900+02
54      0.00  1.000+02
### Sample Output

| COLFREQ | COLFREQ
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>340.00</td>
<td>6.36E+00</td>
</tr>
<tr>
<td>350.00</td>
<td>5.91E+00</td>
</tr>
<tr>
<td>360.00</td>
<td>5.63E+00</td>
</tr>
<tr>
<td>370.00</td>
<td>5.51E+00</td>
</tr>
<tr>
<td>380.00</td>
<td>5.52E+00</td>
</tr>
<tr>
<td>390.00</td>
<td>5.56E+00</td>
</tr>
<tr>
<td>400.00</td>
<td>5.64E+00</td>
</tr>
<tr>
<td>410.00</td>
<td>5.84E+00</td>
</tr>
<tr>
<td>420.00</td>
<td>6.13E+00</td>
</tr>
<tr>
<td>430.00</td>
<td>6.40E+00</td>
</tr>
<tr>
<td>440.00</td>
<td>6.91E+00</td>
</tr>
<tr>
<td>450.00</td>
<td>7.44E+00</td>
</tr>
<tr>
<td>460.00</td>
<td>8.07E+00</td>
</tr>
<tr>
<td>470.00</td>
<td>8.52E+00</td>
</tr>
<tr>
<td>480.00</td>
<td>1.15E+01</td>
</tr>
<tr>
<td>490.00</td>
<td>1.19E+01</td>
</tr>
<tr>
<td>500.00</td>
<td>1.30E+01</td>
</tr>
<tr>
<td>510.00</td>
<td>1.30E+01</td>
</tr>
<tr>
<td>520.00</td>
<td>1.30E+01</td>
</tr>
<tr>
<td>530.00</td>
<td>1.29E+01</td>
</tr>
<tr>
<td>540.00</td>
<td>1.22E+01</td>
</tr>
</tbody>
</table>

### COLFREQ

| 0.00 | 104.00 |
| 2.00E+02 |
| 3.80E+02 |
| 3.90E+02 |
| 3.80E+02 |
| 3.50E+02 |
| 2.87E+02 |
| 1.05E+02 |
| 3.50E+01 |
| 3.00E+01 |
| 3.30E+01 |
| 4.50E+01 |
| 1.69E+01 |
| 1.00E+01 |
| 3.00E+00 |
| 1.82E+01 |

### QUIT

50 INTEGRATION STEPS, HT = 404.2969
100 INTEGRATION STEPS, HT = 395.0234
150 INTEGRATION STEPS, HT = 385.6563
192 INTEGRATION STEPS USED IN WAVFLO
SAMPLE OUTPUT

SMALLEST INTEGRATION INTERVAL BETWEEN LWDHT' = 375.0 KM AND AKBHT' = .0 KM IS .0444 KM
AVERAGE INTEGRATION INTERVAL BETWEEN LWDHT' = 375.0 KM AND AKBHT' = .0 KM IS .0445 KM
**SAMPLE OUTPUT**

**ELECTRIC DIPOLE TRANSMITTER**

<table>
<thead>
<tr>
<th></th>
<th>Z</th>
<th></th>
<th>Y</th>
<th></th>
<th>X</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>.599227-005</td>
<td></td>
<td>.255878-009</td>
<td></td>
<td>.291676-008</td>
<td></td>
</tr>
<tr>
<td>HB</td>
<td>.713075-005</td>
<td></td>
<td>.304493-009</td>
<td></td>
<td>.347092-008</td>
<td></td>
</tr>
<tr>
<td>HE</td>
<td>.724110-005</td>
<td></td>
<td>.309205-009</td>
<td></td>
<td>.352463-008</td>
<td></td>
</tr>
</tbody>
</table>

**MAGNETIC DIPOLE TRANSMITTER**

<table>
<thead>
<tr>
<th></th>
<th>Z</th>
<th></th>
<th>Y</th>
<th></th>
<th>X</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>.717382-005</td>
<td></td>
<td>.306331-009</td>
<td></td>
<td>.349168-008</td>
<td></td>
</tr>
<tr>
<td>HB</td>
<td>.213118-003</td>
<td></td>
<td>.910042-008</td>
<td></td>
<td>.103736-006</td>
<td></td>
</tr>
<tr>
<td>HE</td>
<td>.213036-003</td>
<td></td>
<td>.909693-008</td>
<td></td>
<td>.103696-006</td>
<td></td>
</tr>
</tbody>
</table>
V. MODE SUMS

The excitation factors defined in the previous section are useful for calculating the strengths of the electric field components in the earth-ionosphere guide. In this section explicit expressions are given for calculating those field components. The formulas given are asymptotic and restricted to the range

\[ |k \sin \theta_N| >> 1 \text{ and } |k(\pi - \rho) \sin \theta_N| >> 1 \]

with \( k \) the free space wave number, \( \theta_N \) the eigenangle for Mode \( N \), \( a \) the earth's radius and \( \rho \) the transmitter receiver great circle separation. The mode sums are

\[
E_{m}^{[\mu \nu]}_{j \mathbf{m}} = \frac{C_{1} I \mu f^{3/2}}{k_{\text{kHz}} \sin(\theta)} \sum_{N} \lambda_{jN}^{m} e^{-i k \sin \theta_N} \quad j = 1, 2, 3 \quad m = 1, 2, 3
\]

\[
E_{m}^{[\mu \nu]}_{j \mathbf{m}} = \frac{C_{2} I \alpha f^{5/2}}{k_{\text{kHz}} \sin(\theta)} \sum_{N} \lambda_{jN}^{m} e^{-i k \sin \theta_N} \quad j = 1, 2, 3 \quad m = 4, 5, 6
\]

where the sum is over the number of modes and

\[ C_{1} = 2.86 \times 10^{-3} \]

\[ C_{2} = 5.98 \times 10^{-8} \]

\( f_{\text{kHz}} \) = frequency in kHz

\( I \ell \) = electric dipole current moment - Amp\( \cdot m \)

\( I \alpha \) = magnetic dipole current moment - Amp\( \cdot m^2 \)

\( \lambda_{jN}^{m} \) = excitation factors defined in previous section.
The field strengths $E_j^m$ are in terms of microvolts per meter. A program for computing and plotting the mode sums is included in appendix B. The latter takes punched output from the excitation factor program given in appendix A. Results of the mode sum program are shown in figures 1 through 3 for electric dipole (ED) and magnetic dipole (MD) excitation of EX, EY and EZ. The dipole moments in this example have been chosen to correspond to one kilowatt free space radiation.
**Fig. 1**

HAWAII TO SAN DIEGO GLOBAL NIGHT PROFILE

FREQ=20,000 KHz  TXHT=500.0 KM  RXHT=10.0 KM
AZIM=58.5 DEGREES  CODIP=59.0 DEGREES
MAGFLO=4.31x10^{-8} W/M  SIGMA=4.64x10^{-2} M  EPSR=81.0
IL=1.69x10^{-4} A-M  IA=4.03x10^{-4} A-M
Fig. 2
HAWAII TO SAN DIEGO GLOBAL NIGHT PROFILE
FREQ-20.900 KHZ TXHT-500.0 KM RXHT-10.0 KM
AZIM-58.5 DEGREES CODIP- 39.0 DEGREES
MAGFLD-4.31x10^-6 W/M^2 SIGMA- 4.64x10^-5 S/M EPSR- 81.0
IL- 1.69x10^4 A-M IA- 4.03x10^4 A-M

ED - V

MD - V

ED - HB

MD - HB

ED - HE

MD - HE
Fig. 3
HAWAII TO SAN DIEGO GLOBAL NIGHT PROFILE
FREQ=20.000 KHZ TXHT=500.0 KM RXHT=10.0 KM
AZIM=38.5 DEGREES CODP= 39.0 DEGREES
MAGFD=4.31 x 10^5 W/M^2 SIGMA= 4.64 x 10^3 S/M EPSR= 31.0
IL= 1.69 x 10^9 A-M IA= 4.03 x 10^7 A-M
REFERENCES


4. Smith, RR, A program to compute ionospheric height gain functions and field strengths, Interim Report No. 711 prepared by the Naval Electronics Laboratory Center (now NOSC) for the Defense Atomic Support Agency (now DNA), December 1970.

5. Sheddy, CH, RA Pappert, Y Gough and WF Moler, A FORTRAN program for mode constants in an earth-ionosphere waveguide of arbitrary electron density distribution. Interim Report No. 683 prepared by the Naval Electronics Laboratory Center (now NOSC) for the Defense Atomic Support Agency (now DNA), May 1968.


7. Pappert, RA, Excitation of the earth-ionosphere waveguide by a vertical dipole at satellite heights, Interim Report No. 723 prepared by the Naval Electronics Laboratory Center (now NOSC) for the Defense Atomic Support Agency (now DNA), June 1972.


APPENDIX A

SATELLITE EXCITATION FACTOR PROGRAM LISTING
C PROGRAM DRIVER
C THE DRIVER PROGRAM ALTERNATELY CALLS
C FOR THE INPUT OF DATA ON THE STANDARD
C INPUT UNIT AND CALLS FOR THE COMPUTATION
C OF HEIGHT GAIN FUNCTIONS BY WAVFLD.
C
C IMPLICIT REAL*8 (A-H,O-Z)
C COMPLEX*16 EX(129,6),EY(129,6),EZ(129,6),
C HX(129,6),HY(129,6),HZ(129,6)
C
C istart = 1
C PRINT 905
C 100 CALL XINPUT (ISTART, ISTOP)
C 150 istart = 0
C 200 DO 200 J = 1, 129
C 210 DO 200 K = 1, 6
C 220 EX(J,K) = 0.0
C 230 EY(J,K) = 0.0
C 240 EZ(J,K) = 0.0
C 250 HX(J,K) = 0.0
C 260 HY(J,K) = 0.0
C 270 HZ(J,K) = 0.0
C 280 CONTINUE
C 290 CALL WAVFLD(EX,EY,EZ,HX,HY,HZ)
C
C 300 IF (ISTOP.EQ.0) GO TO 100
C 310 FORMAT(1,/) GO TO 100
C
C END
SUBROUTINE XINPUT (ISTART, ISTOP)

C INPUT READS IN IONOSPHERIC INPUT
C DATA (via NAMELIST), AS WELL AS
C ELECTRON OR ION DENSITY AND
C COLLISION FREQUENCY PROFILES. THIS
C ROUTINE SHOULD BE CALLED PRIOR TO
C CALLING WAVED. ISTART = 1 INDICATES
C THAT A NEW SET OF INPUT DATA IS TO BE
C READ. ISTART = 0 INDICATES AN UPDATE
C OF EXISTING DATA. ISTOP = 1 INDICATES
C THE END OF INPUT DATA FOR THIS JOB.
C ISTOP = 0 INDICATES MORE INPUT DATA
C APPEARS IN THE DATA DECK.

C INPLICIT REAL *8 (A-H, O-Z)
C DIMENSION ED(5), COLL(5), TEM(100,5), SAVE(100), RMAG(4), RANG(4)
C COMMON/EXTRA/XTRA,N
C COMMON/STOR/STORE
C COMMON/AFINPT/THETA, FREQ, AZIM, CODIP, MAGFLD, COEFNU(5), EXPNU(5),
C $ TOPHT, LWSHT, WBHT, DELHT, H, ALPHA, SIGMA, EPSLON
C COMMON/WF FLAG/PRECSN, ISO, IDBG
C COMMON/WIPROF/ENHT(100), ENLOG(100,5), COLLHT(25), COLLRH(25,5),
C $ LHT, WH, CHARGE(5), RATIOM(5), NRSPEC
C COMMON/EW IN/TXHT, RXHT
C COMMON/DDELTA/NUMDIV, STPFILG
C CHARACTER*B BCD(10)
C INTEGER STPFILG
C REAL*8 MAGFLD, LWSHT
C COMPLEX*16 R(4), XTRA, 1/1.0000, 1.0000/ THETA
C NAMELIST/DATUM/THETA, FREQ, IDBG, TOPHT, LWSHT, WBHT, DELHT, PRECSN,
C $ AZIM, CODIP, MAGFLD, COEFNU, EXPNU, ALPHA, SIGMA, EPSLON, EPSLNO,
C $ TXHT, RXHT, H, RMAG, RANG, XTRAMAG, XTRANG, NUMDIV, STPFILG,
C $ NRSPEC, CHARGE, RATIO
C DATA DEGRAD/1.74532925D-2/

C IDAT = 0
C ISTOP = 0
C ICOLL = 1
C IF (ISTART.NE.0) ICOLL = 0
C 100 READ 900, BCD
C 100 PRINT *'IT', BCD
C IF (BCD(1)) .EQ. 'DATUMFIL' GO TO 200
C IF (BCD(1)) .EQ. 'COLLFEQ' GO TO 400
C IF (BCD(1)) .EQ. 'PROFILE' GO TO 500
C IF (BCD(1)) .EQ. 'STOP' GO TO 120
C IF (BCD(1)) .NE. 'QUIT' GO TO 800
C ISTOP = 0
C 120 IF (IDAT.EQ.0.AND. ISTART.NE.0) GO TO 600
C IF (NRSPEC.EQ.0) NRSPEC = 1
C IF (ICOLL.NE.0) RETURN
COLLHT(1) = STORE
COLLHT(2) = 0.0
DO 150 J = 1, NRSPEC
COLLFR(2, J) = DLOG(COFNU(J) * DEXP(STORE + 1000.0 * EXPNU(J)))
COLLFR(1, J) = COLLFR(2, J) - 1000.0 * STORE * EXPNU(J)
150 CONTINUE
RETURN
C
200 IDAT = 1
PREVNU = COFNU(1)
COFNU(1) = 0.0
READ (5, DATUM)
IF (COFNU(1).NE.0.0) ICOLL = 0
IF (COFNU(1).EQ.0.0) COFNU(1) = PREVNU
IF (NURDIV .NE. 0 .OR. STPFLG.EQ.1) GO TO 249
NUM = DLOG(75.0 * DSQRT(DELHT) * DSQRT(FREQ))
NUM = 2 * NUM
WRITE (6, DATUM)
DO 20 JK = 1, 4
RANG(JK) = RANG(JK) * DEGRAD
R(JK) = RANG(JK) * (DCOS(RANG(JK)) + i * DSIN(RANG(JK)))
XTRA = XTRANG * (DCOS(XTRANG) + i * DSIN(XTRANG))
PUNCH 920
PUNCH 921, THETA
PUNCH 922, FREQ
PUNCH 923, TXHT, RXHT
EPSR = EPSLEN / EPSLNO
PUNCH 925, AZIM, CODIP, MAGFD, SIGMA, EPSR
PUNCH 924
GOTO 100
C
400 ICOLL = 1
L = 1
READ 902, HT, COLL
IF (DABS(HT - 999.99) .LT. 0.001) GO TO 430
IF (HT .LT. 25) GO TO 600
IF (HIL .NE. 1 .AND. HT .GE. COLLHT(L - 1)) GO TO 800
DO 411 M = 1, NRSPEC
IF (ICOLL(M) .LE. 0.0) COLL(M) = 1.00 - 40
411 CONTINUE
PRINT 903, HT, (COLL(M), M = 1, NRSPEC)
COLLHT(L) = HT
DO 415 M = 1, NRSPEC
415 COLLFR(L, M) = DLOG(COLL(M))
L = L + 1
GOTO 430
PRINT 903, HT
LL = L - 1
DO 420 J = 1, LL
DO 420 M = 1, NRSPEC
420 TEM(J, M) = COLLFR(J, M)
DO 425 J = 1, LL
DO 425 M = 1, NRSPEC
425 COLLFR(J, M) = TEM(J, M)
LL = L - 1
DO 440 J=1,LL
440  SAVE(JJ) = COLLHT(J)
DO 450 J=1,LL
COLLHT(J) = SAVE(J)
450  PRINT 903, COLLHT(J), (COLLFR(J,M), M=1,NRSPEC)
GO TO 100
C
500  READ 900,BCD
PUNCH 900,BCD
L=1
510  READ 902,HT,ED
IF(DABS(HT-999.99) .LT. 0.001) GO TO 530
IF (L.GT.100) GO TO 800
IF(L .NE. 1 .AND. HT .GE. ENHT(L-1)) GO TO 800
ENHT(L) = HT
IF(NRSPEC .EQ. 3) ED(3) = ED(2)-ED(1)
DO 511 M=1,NRSPEC
IF(ED(M) .LE. 0.0) ED(M)=1.0D-40
CONTINUE
PRINT 903,HT,(ED(M), M=1,NRSPEC)
DO 515 M=1,NRSPEC
ENLOG(L,M) = DLOG(ED(M))
L = L+1
GO TO 510
530  PRINT 903,HT
LL = L-1
DO 520 J=1,LL
DO 520 M=1,NRSPEC
JJ = LL+1-J
520  TEM(JJ,M) = ENLOG(J,M)
DO 525 J=1,LL
DO 525 M=1,NRSPEC
ENLOG(J,M) = TEM(J,M)
525  TXHT = -TXHT+ENHT(1)
RXHT = -RXHT+ENHT(1)
TOPHT = -TOPHT+ENHT(1)
LWSTHT = -LWSTHT+ENHT(1)
HKBHT = -HKBHT+ENHT(1)
53  TEM = LWSTHT
LWSTHT = TOPHT
TOPHT = TEM
H = -H +ENHT(1)
LL = L-1
STORE = ENHT(1)
DO 540 J=1,LL
540  ENHT(J) = -ENHT(J)+STORE
UJ = LL+1-J
540  SAVE(JU) = ENHT(J)
DO 550 J=1,LL
ENHT(J) = SAVE(J)
550  PRINT 903, ENHT(J), (ENLOG(J,M), M=1,NRSPEC)
GO TO 100
C
C  ERROR EXIT
00  PRINT "110"
170  STOP
171  900  FORMAT (10AH)
172  901  FORMAT (1X,10AH)
173  902  FORMAT (F7.3,5X,F7.3)
174  903  FORMAT (1X,F7.3,5X,F1PE9.2,6X)
175  910  FORMAT ("ERROR IN DATA DECK DETECTED",/)
188  $ ' IN SUBROUTINE XINPUT'
177  920  FORMAT (" &DATE")
178  921  FORMAT (" THETA=("F6.3,"F7.3,"),")
179  922  FORMAT (" FREQ="F7.3,",")
180  923  FORMAT (" TXHT="F8.3," , RXHT="F8.3,",")
181  924  FORMAT (" &END")
182  925  FORMAT (" AZIM="F7.3"," , 'COD1P="F7.3"," , 'MAGFLD="D11.4",",")
183  $ 'SIGMA="D10.3"," , 'EPSR="F4.1",",")
184  END
SUBROUTINE WAVFLD(Ex,Ey,Ez,Hx,Hy,Hz)

C WAVFLD CALLS FOR THE DOWNWARD
C INTEGRATION, AND THEN PERFORMS THE
C BACK SUBSTITUTION OF NORMALIZING
C VALUES (SAVED AS DATA BY WSTOR).
C FIELD STRENGTHS ARE COMPUTED AT
C HEIGHTS FROM TOPHT TO LWSHT AT
C DELHT INCREMENTS, AND ARE RETURNED
C IN THE LISTS EX, EY, EZ, HX, HY, HZ.
C
C IMPLICIT REAL *8 (A-H,O-Z)
C COMMON/XTRA,XTRA,R(4)
C COMMON/STOR/STOR
C COMMON/WF,FLAG,PRECN,ISO,DBG
C COMMON/WF,SAVE,P(4,2),M31,M32,M33,ORTHO,ANORM,BNORM,HT,LEV
C COMMON/CS,C,S,CI,S
C COMMON/WFINPT/THETA,FREQ,AMPHT,CDP1,MAGFLD,CEFNU(5),EXPNU(5),
C $ TOPHT,LWSHT,WKBHT,DELT,H,ALPHA,SIGMA,ESPHON
C COMMON/EXC IN/TXHT,RXHT
C COMMON/D RX TX/Ex,Ey,Ez,DTXHT,ETXHT,EZ TXHT,
C $ EX RXHT,EY RXHT,EZ RXHT,
C $ HX RXHT,HY RXHT,HZ RXHT
C REAL*3 MAGFLD, LWSHT
C COMPLEX*16 1/(0.0001,0.0001),/
C COMPLEX*16 EX(129,6),EY(129,6),EZ(129,6),
C $ HX(129,6),HY(129,6),HZ(129,6),
C $ XTRA,P,M31,M32,M33,ORTHO,C,S,CI,SI,THETA,EYD,EZD,
C $ EXHT,EYHT,EZHT,EXKHT,EYKHT,EZKHT,EXRHT,EXXHT,EXYHT,EXZHT,
C $ HYKHT,HZKHT,BX(12),BY(12),BZ(12),EXT,ETY,
C $ EZI,HTI,HTI,HYD,HYNHT,HYREC,EXC,EFIELD(3,3),
C $ HFIELD(3,3),EYDEE,EYENVHT,EXENVHT,EYREC,EXREC
C DIMENSION EXAG(3,3),EYAG(3,3),EYPAG(3,3),HAG(3,3),
C $ EXAG(3,3),EYAG(3,6),EYPAG(6,3),HAG(3,6),
C $ EXAG(3,3),EYAG(6,3),EYPAG(6,6),HAG(6,6),
C $ EQUIVALENCE(TOPHT,D)
C DATA PI/3.14159265358/ 
C JHT = TOPHT/DELHT+1.01
C TEST = (JHT-1)*DELHT-TOPHT
C IF(DABS(TEST) .GT. 0.0001) GO TO 800
C TOPHT = LWSHT/DELHT+1.01
C TEST = (JHT-1)*DELHT-LWSHT
C IF(DABS(TEST) .GT. 0.0001) GO TO 800
C CALL WFINTG(TOPHT,LWSHT,DELT,0)
C CALL DECOMP(LWSHT)
C CALL WKB
C CALL COSINE SOLUTIONS AT GROUND SO THAT
C THEY SATISFY BOUNDARY CONDITION.
C CALL WF BNDY(0)
C PERFORM BACK SUBSTITUTION OF
C NONORMALIZING VALUES.
0 SLM = 0.0
PROMA = 1.0
PMDB = 1.0
IMT = IMM
CALL WLOAD
GO TO 25
C
21 U SUM = 0 SUM*ANORM*BNORM+ORTHO
PRODA = PROMA*ANORM
IF (PRODA.LT. 1.0D-30) PRODA = 0.0
PRODB = PROMDB*BNORM
CALL WLOAD
DD 23 J=1,4
P(J,2) = P(J,2) - SUM*P(J,1)*PRODB
23 P(J,1) = P(J, 1) * PRODA
C
C COMPUTE FIELD STRENGTHS
C AT PROFILE HEIGHTS.
25 DD 26 N=1,11,2
DD 26 J=1,4
26 W(J,N) = P(J,1)*B(N)+P(J,2)*B(N+1)
DD 27 N=1,11,2
NN = (N+1)/2
EX(JHT,NN) = W(1,N)
EY(JHT,NN) = -W(2,N)
EZ(JHT,NN) = -(5*W(4,N)+M31*W(1,N)-M32*W(2,N))/(1.*M33)
HX(JHT,NN) = W(3,N)
HY(JHT,NN) = W(4,N)
HZ(JHT,NN) = -S*W(2,N)
EXANG(N) = CADABS(EX(JHT,NN))
EYANG(N) = CADABS(EY(JHT,NN))
EZANG(N) = CADABS(EZ(JHT,NN))
HXANG(N) = CADABS(HX(JHT,NN))
HYANG(N) = CADABS(HY(JHT,NN))
HZANG(N) = CADABS(HZ(JHT,NN))
C
27 CONTINUE
C
100 IHT = IHT + 1
101 IF (IHT.LE.JHT) GO TO 21
102 IF (LEVEL .NE. 0) PRINT 903,LEVEL
103 CALL RBARS(C,S,RBAR11,RBAR22)
104 CALL HGAIND(N,EXT,EYT,EZT,HXT,HYT,HZT)
105 HYD = HYT
106 EYDGE = EYT
107 CALL HGAIND(STORE,EXT,EYT,EZT,HXT,HYT,HZT)
108 HYENHT = HYT/HYD
109 EYENHT = EYT/EYDGE
110 EXENHT = EAT/HYD
111 IF (IMT.EQ. STORE) GO TO 50
$COUTAT(S)$

170 DO 100 J=1,3
171 LD 100 K=1,3
172 IMAG(J,K) = CDABS(EFIELD(J,K))
173 ENAG(J,K) = CDANG(EFIELD(J,K))
174 HMAG(J,K) = CDABS(HFIELD(J,K))
175 HANG(J,K) = CDANG(HFIELD(J,K))
176 CONTINUE
177 100 PRINT 920
178 PRINT 921
179 PRINT 922
180 PRINT 923, (EMAG(1,K),EANG(1,K),K=1,3)
181 PRINT 924, (EMAG(2,K),EANG(2,K),K=1,3)
182 PRINT 925, (EMAG(3,K),EANG(3,K),K=1,3)
183 PUNCH 927, (EMAG(1,K),EANG(1,K),K=1,3)
184 PUNCH 928, (EMAG(2,K),EANG(2,K),K=1,3)
185 PUNCH 929, (EMAG(3,K),EANG(3,K),K=1,3)
186 PRINT 930
187 PRINT 921
188 PRINT 922
189 PRINT 923, (HMAG(1,K),HANG(1,K),K=1,3)
190 PRINT 924, (HMAG(2,K),HANG(2,K),K=1,3)
191 PRINT 925, (HMAG(3,K),HANG(3,K),K=1,3)
192 PUNCH 927, (HMAG(1,K),HANG(1,K),K=1,3)
193 PUNCH 928, (HMAG(2,K),HANG(2,K),K=1,3)
194 PUNCH 929, (HMAG(3,K),HANG(3,K),K=1,3)
195 PRINT 925
196 RETURN

C
197 800 PRINT 922
198 STOP
200 C
201 902 FORMAT ('ERROR IN WAVEFLD',/)
202 903 FORMAT ('DELT does not divide TOTHT-LWSTHT evenly')
203 904 FORMAT('LEVEL NOT ZERO. LEVEL = ',I3)
204 920 FORMAT('14X,ELECTRIC DIPOLE TRANSMITTER///')
205 921 FORMAT('13X,Z',25X,'Y',25X,'X/')
206 922 FORMAT('24X,9X,THG',11X)
207 923 FORMAT('14X,V',4X,3(E13.6,2X,F6.3,5X)/)
208 924 FORMAT('13X,HB',4X,3(E13.6,2X,F6.3,5X)/)
209 925 FORMAT('13X,HE',4X,3(E13.6,2X,F6.3,5X)/)
210 926 FORMAT('11')
211 927 FORMAT('V',2X,3(E13.6,2X,F6.3,4X))
212 928 FORMAT('HB',2X,3(E13.6,2X,F6.3,4X))
213 929 FORMAT('HE',2X,3(E13.6,2X,F6.3,4X))
214 930 FORMAT('0',14X,'MAGNETIC DIPOLE TRANSMITTER///')
215 END
SUBROUTINE WF INTG(TOPHT, LWSHT, DELHT, IFLAG)
C
C WF INTG PERFORMS THE INTEGRATION OF THE P MATRIX THROUGH THE
C IONOSPHERE UING THE TECHNIQUES GIVEN BY RITTEWAY. ACCURACY IS
C MAINTAINED BY ADJUSTING THE STEP SIZE SO THAT THE P MATRIX IS
C COMPUTED WITH SUFFICIENT ACCURACY.
C
C IFLAG=0 INTEG FOR THETA ONLY
C IFLAG=1 INTEG FOR THETA AND THETA-DTHETA
C
IMPLICIT REAL *B (A-H,O-Z)
COMMON/WF FLAG,PRCSN,ISO,IDBG
COMMON/P MIX/P(8),PI(8)
COMMON/WF SAVE/P SAVE(8),M31 SAV,M32 SAV,M33 SAV,
$ ORTHO,ANORM,BNORM,H1,LEVEL
COMMON/M MTX/M(3,3)
COMMON/WF PROF/ENLOG(100,5),COLLHT(25),COLLFR(25,5),
$ LHT,MHT,CHARGE(S),RATIOM(S),NRSPEC
INTEGER SVFLAG
REAL*B LWSHT
COMPLEX*16 M31 SAV,M32 SAV,M33 SAV,ORTHOM
COMPLEX*16 P,PI,P SAVE,PREV P,TEMP P,DPDH,PV DPDH,D1DM
DIMENSION PREV(8),TEMPP(8),PV DPDH(8),D1DM(8)
DIMENSION PR(16),TPR(16)
EQUIVALENCE (P,PR),(TEMPP,TPR)
DATA EPSH/5.0D-4/
DATA DMIN/1.0D-3/
C MINIMUM STEP SIZE ALLOWED
C
CALL INIT T
CALL T MTRX(TOP HT)
CALL WF INIT(P)
CALL P DERIV(P,DPDH)
IF(IFLAG..EQ.0) GO TO 11
CALL TI MTRX
CALL WF INIT(PI)
CALL P DERIV(PI,DPDH)
CONTINUE
C
ISTEPS = 0
KMAX = 0
LEVEL = 0
HT = TOPHT
CALL XFER(P,P SAVE,B)
M31 SAV = M(3,1)
M32 SAV = M(3,2)
M33 SAV = M(3,3)
CALL WF STOR
WFHT = TOPHT - DELHT
DELHT = 0.125D0*DELHT
SVFLAG=0
C
C DETERM NEXT STEP SIZE TO USE.
10 IF (SVFLAG.EQ.1) DELH2 = SAVDH2
11 SVFLAG = 0
12 NOCOL = 0
13 HT0 = HT
14 CALL XFER(P,PREVP,B)
15 CALL XFER(P,PDH,PDPH,8)
16 HLIM = CFHT
17 IF (ENHT(LHT+1),GT,HTLIM+EPSHT) $ HLIM = ENHT(LHT+1)
18 IF (COLLHT(LHT+1),GT,HTLIM+EPSHT) $ HLIM = COLLHT(LHT+1)
19 IF (HT0-DELH2.GE,HTLIM+EPSHT) GO TO 50
20 SAVDH2 = DELH2
21 SVFLAG = 1
22 DELH2 = HT0- HTLIM
23 C
24 C PERFORM NEXT INTEGRATION STEP.
25 CALL WF STEP(P,PDH,HT,DELH2,0)
26 CALL XFER(P,PREVP,B)
27 CALL XFER(P,PDH,PDPH,8)
28 IF (PMAX.GT.PRECSN) GO TO 100
29 IF (PVMAX.GT.PRECSN) GO TO 100
30 IF (PMAX.GT.PRECSN) GO TO 100
31 IF (PVMAX.GT.PRECSN) GO TO 100
32 IF (PMAX.GT.PRECSN) GO TO 100
33 C
34 C CHECK ACCURACY OF RESULT.
35 FMAX = 0.0
36 DO BS J=1,16
37 IF (ABS(J) .LT. TPR(J)) PABS = DABS(PR(J)-TPR(J))
38 IF (PMAX.LT.PABS) PMAX = PABS
39 CONTINUE
40 C
41 C ADJUST STEPSIZE IF NECESSARY.
42 IF (PMAX.LT.PRECSN) GO TO 100
43 IF (PVMAX.LT.PRECSN) GO TO 100
44 IF (PMAX.LT.PRECSN) GO TO 100
45 IF (PVMAX.LT.PRECSN) GO TO 100
46 IF (PMAX.LT.PRECSN) GO TO 100
47 C
48 CONTINUE
49 IF (PMAX.LT.10.0*PRECSN) GO TO 99
50 DELH2 = 0.25 * DELH2
51 NOCOL = 0
52 C
53 CALL XFER(PREVP,B)
54 CALL XFER(P,PDH,PDPH,8)
55 SVFLAG = 0
56 GO TO 50
57 C
58 CALL WF SCAL(P,0)
59 CALL XFER(P,PREVP,B)
IF(HT.LT.WFHT+EPSHT) CALL WF STOR
CALL P DERIV(P,DPIPH)
IF(IFLAG .EQ. 0) GO TO 72
HT = H0
CALL WF STEP(PI,DPIPH,HT,DELH,3)
CALL P DERIV(PI,DPIPH)
CALL WF STEP(PI,DPIPH,HT,DELH,4)
CALL WF SCAL(PI,T)
CALL P DERIV(PI,DPIPH)
72 CONTINUE

C

ISTEPS = ISTEPS+1
IF(IDBG .EQ. 0) GO TO 73
IDIV = ISTEPS/50
IF(ISTEPS .EQ. 50*IDIV) PRINT 902,ISTEPS,HT
CONTINUE
IF(NO DBL .EQ. 0 .AND. PMAX .LT. 0.1*PRECSN) DELH2 = 2.0*DELH2

C

CHECK INTEGRATION AND PROFILE HEIGHTS.
IF (HT.LT.LWSTHT+EPSHT) GO TO 80
IF (HT.LT.WFHT+EPSHT) WFHT = WFHT - DELHT
IF (HT.LT.ENHT(LHT+1)+EPSHT) LHT = LHT + 1
IF (HT.LT.COLLHT(MHT+1)+EPSHT) MHT = MHT + 1
GO TO 10

C

80 PRINT 901,ISTEPS
RETURN

C

900 FORMAT ( 'MINIMUM STEPSIZE USED AT HT =',D14.5)
901 FORMAT (1X,13,' INTEGRATION STEPS USED IN WAVFLED',/)
902 FORMAT (1X,14,' INTEGRATION STEPS, HT =',F9.4)
END
SUBROUTINE T MTRX(HT)
C T MTRX COMPUTES THE MATRICES
C M= THE SUSCEPTIBILITY TENSOR
C T= THE COEFFICIENT MATRIX OF
C THE LINEAR SYSTEM OF O.D.E.
C DP/DZ = -IK*T.P.
C NOTE THAT ON CALL TO ENTRY INIT T, VARIOUS IONOSPHERIC CONSTANTS
C ARE COMPUTED.

IMPLICIT REAL *8 (A-H,O-Z)
COMMON:WF,FLAG,PREC,SN,ISO,DBG
COMMON:M,MTX/M11,M21,M31,M12,M22,M32,M13,M23,M33
COMMON:M,MTX/M11,M21,M31,M12,M22,M32,M13,M23,M33
COMMON:TM,MTX/T11,T13,T14,T12,T12,T12,T14,T14,T34,T44
COMMON:TM,MTX/T11,T13,T14,T12,T12,T12,T14,T14,T34,T44
COMMON:CS/C.S.CI.SI
COMMON:W,FINPT/THETA,FREQ,AZMUTH,CODIP,MAGFD,CEFNU(5),EXPNU(5),
$ TOPHT,LWST,T,WKHT,DELTH,M,ALPHA,SIGMA,PSILON
COMMON:W,PROF/ENHT(100),ENLLOG(100,5),COLLHT(25),COLLFR(25,5),
$ LHT,MYHT,CHARGE(5),RATIO(5),NRSPEC
COMMON:W,REAL*B,MAGFD,LWSTHT,
$ LSYQS,M,MSQ,MSQSY,NSQSY,
$ LMSQ,LMNSQ,MSQSY,NU,
$ LY,MY,NY
COMPLEX*16 M(3,3),
$ M11,M21,M31,M12,M22,M32,M13,M23,M33,
$ T11,T13,T14,T12,T12,T12,T14,T14,T34,T44,
$ T11,T13,T14,T12,T12,T12,T14,T14,T34,T44,
$ C.S.CI.SI,CSQ,SSQ,CSQI,SSQI,
$ THETA,DTHETA,
$ D,MJ,MD,
$ D,USQ,DD,1,1,D,T,1B
DIMENSION Y(5),Y0(5),LY(5),MY(5),
$ NY(5),LMYS(5),MNYS(5),EN(5),NU(5),
$ LSQSY(5),MSQSY(5),NSQSY(5),
$ COEF EN(5)

EQUIVALENCE(M11,M)
DATA PI/3.14159265300/
DATA DR/1.7453292520-2/
DATA COEFF/3.182357003/,COEFFY/1.758796011/
DATA I/1.0000,1.0000/
DATA VELT/2.997928005/
DATA DTHETA/(5.0D-2,1.0D-2)/

CALCULATE THE MATRIX M.
M(1,1) = 0.0
M(1,2) = 0.0
M(1,3) = 0.0
M(2,1) = 0.0
M(2,2) = 0.0
M(2,3) = 0.0
C M(1,1) = 0.0
56 M(3,2) = 0.0
57 M(5,3) = 0.0
58 C CALL DF DENS (HT, EN, NU)
59 NFLAG = 0
60 DD 20 K=1,NRSPEC
61 C ADD IN THE CONTRIBUTIONS TO THE
62 C SUSCEPTIBILITY TENSOR M FOR EACH
63 C SPECIE IN THE IGONOSPHERE.
64 C IF(EN(K).LT. 1.0E-3) GO TO 20
65 NFLAG = 1
66 X = COEF EN(K)*EN(K)
67 Z = NU(K)*DV OMGA
68 USQ=US+U
69 DD = -X / (U + (USQ - YSQ(K)))
70 IUD = (Z+1)*DD
71 TA = USQ * DD
72 M(1,1) = M(1,1) + TA
73 M(2,2) = M(2,2) + TA
74 M(3,3) = M(3,3) + TA
75 M(2,2) = M(2,2) - NSQYSQ(K) * DD
76 TA = MY(K)*IUD
77 TB = LNYSQ(K) * DD
78 M(1,1) = M(1,1) + TA - TB
79 M(3,1) = M(3,1) - TA - TB
80 IF (ISO. NE. 0) GO TO 20
81 M(1,1) = M(1,1) - LSQYSQ(K) * DD
82 M(3,3) = M(3,3) - NSQYSQ(K) * DD
83 TA = NY(K)*IUD
84 TB = LNYSQ(K) * DD
85 M(2,1) = M(2,1) + TA - TB
86 M(1,2) = M(1,2) - TA - TB
87 TA = LY(K)*IUD
88 TB = NYYSQ(K) * DD
89 M(3,2) = M(3,2) + TA - TB
90 M(2,3) = M(2,3) - TA - TB
91 C CONTINUE
92 20 C CRVTRM=ALPHA*(H-HT)
93 M1,1 = M(1,1) + CRVTRM
94 M2,2 = M(2,2) + CRVTRM
95 M3,3 = M(3,3) + CRVTRM
96 C CALCULATE THE MATRIX T.
97 D = 1.0/(1.0+M33)
98 TM41 = 1.0*M11
99 TM32 = M22
100 TM14 = D
101 IF(NFLAG .EQ. 0) GO TO 40
102 M3D = M13*D
103 M23D = M23*D
104 TM41 = TM41-M31*M13D
105 TM11 = M31*D
106 TM44 = M13D
107 IF(ISO .NE. 0) GO TO 40
112 TM32 = TM32-M23D
113 TM31 = TM31-M23D-M21
114 TM12 = M12-D
115 TM42 = M12-M13D-M12
116 TM34 = M23D
117 C
118 DO 40 T1 = T41
119 32 T32 = CSO+TM32
120 T14 = 1.0-SSQ*TM14
121 IF(NFLAG.EQ.0) GO TO 70
122 T11 = -S*TM41
123 T44 = -S*TM44
124 IF(ISO .NE. 0) RETURN
125 T31 = TM31
126 T12 = S*TM12
127 T42 = TM42
128 T34 = S*TM34
129 RETURN
130 C
131 ENTRY TI MTRX
132 T41 = TM41
133 T32 = CSO+TM32
134 T14 = 1.0-SSQ*TM14
135 T11 = -S*TM41
136 T44 = -S*TM44
137 IF(ISO .NE. 0) RETURN
138 T31 = TM31
139 T12 = S*TM12
140 T42 = TM42
141 T34 = S*TM34
142 RETURN
143 C
144 C
145 ENTRY INIT T
146 C COMPUTE VARIOUS QUANTITIES
147 C WHICH DO NOT VARY WITH HEIGHT.
148 LMT = 0
149 150 ISO = 0
151 IF(MAGFLD) .EQ. 0.01 GO TO 250
152 IF(DABS(CODIP-90.0).GE.0.15) GO TO 300
153 IF(DABS(AMUTH-90.0).LT.0.15) GO TO 250
154 IF(DABS(AMUTH-270.0).GE.0.15) GO TO 300
155 250 ISO = 1
156 300 OMEGA = 2000.0+PI+FREQ
157 C. DVA = 1.0.OMEGA
158 WAVVR = OMEGA,VELLI
159 SINDIP = OSIN(CODIP-DTR)
160 DGCSL = SINDIP+DCOS(AMUTH*DTR)
161 DGCSM = SINDIP*OSIN(AMUTH*DTR)
162 DGCSN = DCOS(CODIP*DTR)
163 DD 60 K=1,NRSPEC
164 COEF EN(K) = COEFFX*1.0E6*CHARGE(K)*2/(OMEGA**2*RATIO(K))
165 Y1(K) = COEFFY*CHARGE(K)*MAGFLD
166 $ (OMEGA / RATIO(K))
167 YSP(K) = Y1(K)*2
168 L1(K) = DGCSL*Y(K)
MY(K) = DCOSM*Y(K)
NY(K) = DCOSN*Y(K)
LSQYSQ(K) = DCOSL**2*YSQ(K)
MSSQYSQ(K) = DCOSM**2*YSQ(K)
NSQYSQ(K) = DCOSN**2*YSQ(K)
LWSQF(K) = DCOSL*DCOSM*YSQ(K)
LNYSQ(K) = DCOSL*DCOSN*YSQ(K)
MNYSQ(K) = DCOSM*DCOSN*YSQ(K)
60 CONTINUE
C = CDCOS(THETA+DTR)
S = CDSCIN(THETA+DTR)
C50 = C**2
SSQ = S**2
CI = COCOS((THETA-OTHETA)+DTR)
SI = COSIN((THETA-OTHETA)+DTR)
C5Q1 = CI**2
SSQ1 = SI**2
70 T11 = 0.0
T31 = 0.0
T12 = 0.0
T42 = 0.0
T34 = 0.0
T44 = 0.0
RETURN
END
SUBROUTINE WF DENS (MT, EN, COLT)
C
C   WF DENS COMPUTES THE ION DENSITY
C   AND COLLISION FREQUENCY FOR EACH
C   SPECIE BY LOGARITHMIC INTERPOLATION
C   OF THE CORRESPONDING PROFILES.
C   PROFILE VALUES ARE INTERPOLATED BETWEEN
C   ENTRIES MHT AND MHT+1 (LHT AND LHT+1).
C
C   IMPLICIT REAL *8 (A-H,O-Z)
C   COMMON/WFPRF/ENHT(100),ENLOG(100,5),COLLHT(25),COLLFR(25,5).
C   $LHT,MHT,CHARGE(5),RATIOM(5),NRSPEC
C   DIMENSION EN(5), COLL(5), DELE(5), DELC(5)
C   COMMON/PRIMES/DELE,DELC
C   DATA EPSHT/5.0=4/
C
C   LUCKY=0
MUCKY=0
IF (LHT.EQ.0) LHT=1
IF (MHT.EQ.0) MHT=1
10 IF (MT.GE.ENHT(LHT+1)-EPSHT .AND. MT.LT.ENHT(LHT)+EPSHT) GO TO 20
20 IF (LUCKY.EQ.1) GO TO 30
LHT=LHT-1
IF (LHT.EQ.0) LHT=1
IF (LHT.EQ.1) LUCKY=1
GO TO 10
30 LHT=LHT+1
IF (LHT.GT.101) GO TO 899
GO TO 10
20 IF (MT.GE.COLLHT(MHT+1)-EPSHT .AND. MT.LT.COLLHT(MHT)+EPSHT) GO TO 100
30 IF (MHT.EQ.1) GO TO 40
MHT=MHT-1
IF (MHT.EQ.0) MHT=1
IF (MHT.EQ.1) MUCKY=1
GO TO 20
40 MHT=MHT+1
IF (MHT.GT.26) GO TO 899
GO TO 20
C
100 IF (LHT.EQ.LSAVE) GO TO 200
   DO 150 K = 1,NRSPEC
      DELE(K) = (ENLOG(LHT+1,K) - ENLOG(LHT,K))
   $ / (ENHT(LHT+1) - ENHT(LHT))
C
150 CONTINUE
   LSAVE = LHT
200 IF (MHT.EQ.MSAVE) GO TO 300
   DO 250 K = 1,NRSPEC
      DELC(K) = (COLLFR(MHT+1,K) - COLLFR(MHT,K))
   $ / (COLLHT(MHT+1) - COLLHT(MHT))
250 CONTINUE
   MSAVE = MHT
300  D H = M H - E H M ( H M T )

500  CONTINUE

899  PRINT 900

900  C U N T ( E R R O R  I N  P R O F I L E  I N T E R P R O L A T I O N )

END
SUBROUTINE WF_INIT(P)
C
C  WF_INIT COMPUTES THE INITIAL
C  P MATRIX, I.E., THE INITIAL CONDITIONS
C  FOR THE INTEGRATION DP/DZ = -I*K*P.
C
C    COMMON/CS/C,IGNORE(12)
8 COMPLEX*16 P(4,2),C
C
10 P(1,1) = C
11 P(2,1) = 0.
12 P(3,1) = 0.
13 P(4,1) = 1.
14 P(1,2) = 0.
15 P(2,2) = 1.
16 P(3,2) = C
17 P(4,2) = 0.
C
END
SUBROUTINE P_DERIV(P,DPDH)
C P_DERIV COMPUTES THE HEIGHT DERIVATIVES
C OF THE FIELD VECTORS P ACCORDING TO
C EMMOW AND HEDDING (1954).
C EQUATION IS DP/DZ = -IK*T*P.
C MULTIPLICATION BY *-1 IS PERFORMED BY
C OPERATING ON REAL AND IMAG PARTS.
C MULTIPLICATION BY K IS PERFORMED
C IN SUBROUTINE WFL STEP.
C
COMMON/T MTH/ T11,T31,T41,T12,T32,T42,T14,T34,T44
COMPLEX*16 P(4,2),DPDH(4,2),DERIV.
$ T11,T31,T41,T12,T32,T42,T14,T34,T44
C
C DD 11 J=1,2
DERIV = T11*P(1,J)+T12*P(2,J)+T14*P(4,J)
DPDH(1,J) = (0.00,-1.00)*DERIV
DERIV = P(3,J)
DPDH(2,J) = (0.00,-1.00)*DERIV
DERIV = T31*P(1,J)+T32*P(2,J)+T34*P(4,J)
DPDH(3,J) = (0.00,-1.00)*DERIV
DERIV = T41*P(1,J)+T42*P(2,J)+T44*P(4,J)
DPDH(4,J) = (0.00,-1.00)*DERIV
11 CONTINUE
RETURN
C END
SUBROUTINE WF SCAL (PP, IFLAG)

C SCAL SCALES AND ORTHOGONALIZES THE SOLUTION
C VECTORS P. THIS SCALING MUST LATER BE
C REMOVED TO YIELD CORRECT (UNSCALED) SOLUTIONS.

C IMPLICIT REAL*8 (A-H,O-Z)
C COMMON/FSAVE/P SAVE(8),M31 SAV,M32 SAV,M33 SAV,
C SUM,APROD,BPROD,HT,LEVEL
C COMMON/SAVE/P ETF(27,129)
C COMPLEX*16 P(4,2),PP,PSAVE,
C M31 SAV,M32 SAV,M33 SAV,O SUM,ORTHO
C DIMENSION PR(8,2),PP(8),PSTORE(27)
C EQUIVALENCE (PSAVE,PSTORE)
C EQUIVALENCE (P,FR)

C CALL XFER (PP,P,8)
C A NORM = 0.0
C DO 11 J=1,8
11 ANORM = ANORM+PR(J,1)**2
C ORTHO = 0.0
C DO 12 J=1,4
12 ORTHO = ORTHO+DCONJG(P(J,1))*P(J,2)
C ORTHO = ORTHO/ANORM
C DO 13 J=1,4
13 P(J,2) = P(J,2)-ORTH0*P(J,1)
C B NORM = 0.0
C DO 14 J=1,8
14 B NORM = B NORM+PR(J,2)**2
C ANORM = 1.0/D SQRT(ANORM)
C B NORM = 1.0/D SQRT(B NORM)
C DO 15 J=1,8
C PR(J,1) = PR(J,1)*ANORM
C PR(J,2) = PP(J,2)*B NORM
C CALL XFER (PP,PP,8)
1 IF (IFLAG .NE. 0) RETURN
C DO SUM = O SUM+ORTH0*APROD/BPROD
C APROD = APROD+ANORM
C BPROD = BPROD+B NORM
C RETURN
C ENTRY WF STOR
C LEVEL = LEVEL+1
C LD 25 J=1,27
C F ETF(J,LEVEL)=PSTORE(J)
C DO SUM = 0.0
C A PROD = 1.0
C B PROD = 1.0
C RETURN
Li-Li

[IC]

OU
SUBROUTINE WF_STEP(P,DPDH,HT,DELH,IFLAG)

C WF_STEP INCREASES THE SOLUTION OF P
C FROM HT TO HT-DELH, USING
C RUNGE-KUTTA INTEGRATION

C IFLAG=0 ONE LARGE STEP, THETA
C IFLAG=1 FIRST SMALL STEP, THETA
C IFLAG=2 SECOND SMALL STEP, THETA
C IFLAG=3 FIRST SMALL STEP, THETA-DELTA
C IFLAG=4 SECOND SMALL STEP, THETA-DELTA

C IMPLICIT REAL*8 (A-H,O-Z)
COMMON/WF CON/OMEGA.WAVE NR
COMMON/WF FLAG/PRECNO.ISO.IDBG
COMMON/T MTRX/T(9)
COMMON/TM MTRX/TM(9)
COMPLEX*16 P(8),DPDH(8),PO(8),HDELPO(8),DELP1(8),DELP2(8),DELP4
COMPLEX*16 T,SAVE1(9),SAVE2(9),
$ TM,TIMEV1(9),TIMEV2(9),TIMEV3(9),TIMEV4(9)

C

HT0 = HT
DELH K = DELH*WAVE NR
HDELH K = DELH K*0.5
DO 11 J=1,8
PO(J) = P(J)
HDELPO(J) = -DPDH(J)*HDELH K

11 P(J) = PO(J)+HDELPO(J)

C

HT = HT0-0.5*DELH
IF(IFLAG .LE. 2) CALL T MTRX-HT
IF(IFLAG .EQ. 0) CALL XFER(T,SAVE1,9)
IF(IFLAG .EQ. 0) CALL XFER(TM,TM SAV2,9)
IF(IFLAG .EQ. 1) CALL XFER(TM,TM SAV1,9)
IF(IFLAG .EQ. 2) CALL XFER(TM,TM SAV3,9)
IF(IFLAG .EQ. 3) CALL XFER(TM,TM SAV1,TM,9)
IF(IFLAG .EQ. 4) CALL XFER(TM,TM SAV3,TM,9)
IF(IFLAG .GE. 3) CALL T MTRX

C CALL P DERIV(P,DPDH)
DO 12 J=1,8
DELP1(J) = -DPDH(J)*DELH K

12 P(J) = PO(J)+0.5*DELP1(J)

C CALL P DERIV(P,DPDH)
DO 13 J=1,8
DELP2(J) = -DPDH(J)*DELH K

13 P(J) = PC(J)+DELP2(J)

C

HT = HT0-DELH
CALL DERIVP,DPDH
THIRD = 1.000,3.000
DO 14 J=1,8

DELP4 = (HDELP0(J)*DELP1(J)+DELP2(J)-DPDH(J)+HDELH K)*THIRD

P(J) = PC(J)+DELP4
RETURN

END
SUBROUTINE WAB
IMPLICIT REAL*B(A-H,O-Z)
INTEGER STPFLG
REAL*B W A FL0, LWSHT
COMPLEX*16 I, THETA, Q, GAMMA, INTRQ, INTGM, SUMQ, SUMGAM, EXPO,
$ P(1:4), PARTS
$ COMMON/XFINP/I,THETA, FREQ, AZIM, CODIP, MAGFDL, COEFN(5), EXPN(5),
$ TOPH1, LWSHT, WKBHT, DELHT, H, ALPHA, SIGMA, EPSLON
COMMON/XGAM/OMEGA, WAVENR
COMMON/DELTA/NUMDIV, STPFLG
COMMON/ANSWER/PART S(1:4.2)
COMMON/P MTK/P, IGNORR(32)
DATA I/(0.000,1.000)/

IF (WKBHT .GE. LWSHT) RETURN
CALL INIT DI
C STPFLG=0 FIXED INTEGRATION STEPSIZE OF DELHT/NUMDIV
C STPFLG=1 VARIABLE INTEGRATION STEPSIZE
IF (STPFLG.EQ.1) CALL WKBVAR
IF (STPFLG.EQ.1) RETURN

EPSH=DELHT/NUMDIV
IHT=(LWSHT-WKBHT)/EPSH + 1.0D0
SUMQ=(0.000,0.000)
SUMGAM=(0.000,0.000)

DO 90 N=1,IHT-2,2
DO 30 M=1,3
HT=LWSHT-EPSH*(N+M-2)
CALL QGAMMA(HT, DELHT, LWSHT, Q, GAMMA)
IF (M.EQ.2) GO TO 110
INTGRQ=EPSH*Q/3.0D0
INTGAM=EPSH*GAMMA/3.0D0
GO TO 120
110 INTGRQ=4.000*EPSH*Q/3.0D0
120 SUMQ=SUMQ+INTGRQ
INTGAM=INTGAM+SUMGAM

40 NNN=(N+1)/NUMDIV
41 IF ((NNN-1)/NUMDIV .EQ. NNN .NE. 0.0) GO TO 90
EXPO=COEXP(WAVENR+(SUMQ+1-SUMGAM))
DO 150 K=1,4
DO 150 L=1,2
150 F(K,L)=EXPO*PART S(K,L)

C OUTPUT OF INTERMEDIATE P VALUES COULD BE ACCOMPLISHED HERE
90 CONTINUE
RETURN
END
$D$.

$E$

$F$

$G$

$H$

$I$

$J$

$K$

$L$

$M$

$N$

$O$

$P$

$Q$

$R$

$S$

$T$

$U$

$V$

$W$

$X$

$Y$

$Z$

$\alpha$

$\beta$

$\gamma$

$\delta$

$\epsilon$

$\zeta$

$\eta$

$\theta$

$\iota$

$\kappa$

$\lambda$

$\mu$

$\nu$

$\xi$

$\omicron$

$\pi$

$\rho$

$\sigma$

$\tau$

$\upsilon$

$\phi$

$\chi$

$\psi$

$\omega$

$\alpha$

$\beta$

$\gamma$

$\delta$

$\epsilon$

$\zeta$

$\eta$

$\theta$

$\iota$

$\kappa$

$\lambda$

$\mu$

$\nu$

$\xi$

$\omicron$

$\pi$

$\rho$

$\sigma$

$\tau$

$\upsilon$

$\phi$

$\chi$

$\psi$

$\omega$
55    DELH2 = DELH2/2.
56    GO TO 60
57
58    QABS = CDABS(QSAVE-INTGRQ)
59    IF (CDABS(INTGRQ).NE.0.) QABS = QABS/CDABS(INTGRQ)
60    IF (QABS.LT.PRECSN) GO TO 100
61    IF (DELH2.GT.DEHMIN) GO TO 95
62    IF (KMAX.EQ.0) PRINT 900,HT
63
64    FORMAT (' MINIMUM STEPSIZE USED FIRST AT HT ',D14.5)
65    KMAX = 1
66    GO TO 100
67
68    HT = HT+DELH2
69    DELH2 = DELH2/2.
70    NODBL = 1
71    IF (QABS.LT.10.*PRECSN) GO TO 99
72    DELH2 = DELH2/4.
73    NODBL = 0
74
75    99  SVFLAG = 0
76    GO TO 50
77
78    IF (NODBL.EQ.0 .AND. QABS.LT.PRECSN/3.) DELH2 = 2.*DELH2
79    SUMQ = SUMQ+INTGRQ
80    SUMGAM = SUMGAM+INTGAM
81    NUMINT = NUMINT+1
82    IF (HT.GT.WFHT+EPSHT) GO TO 10
83    WSHORT = WFHT-DEHMT
84    EXP1 = CDEXP(WAVENR*(SUMQ+1 - SUMGAM))
85    DO 150 K = 1, 4
86    DO 150 L = 1, 2
87
88    150 P(K,L) = EXP1*PART S(K,L)
89
90    IF (HT.GT.WKBHST+EPSHT) GO TO 10
91    LWFORM = (LWSHT-WKBHST)/(4.*NUMINT)
92    PRINT 909,LWSHT,WKBHST,DEHSV
93    PRINT 910,LWSHT,WKBHST,DEHST
94    RETURN
95
96
97
98
99
100   FORMAT(1X,'SMALLEST INTEGRATION INTERVAL BETWEEN LWSTHT' =  ',F7.1,'  
101   ' , KM AND WKBHST' =  ',F7.1,' KM IS ',F9.4,' KM')
102
103   FORMAT(' AVERAGE INTEGRATION INTERVAL BETWEEN LWSTHT' =  ',F7.1,'  
104   ' , KM AND WKBHST' =  ',F7.1,' KM IS ',F9.4,' KM')
105   END
C
CALCULATE THE MATRIX M.

M(1,1) = 0.0
M(1,2) = 0.0
M(1,3) = 0.0
M(2,1) = 0.0
M(2,2) = 0.0
M(2,3) = 0.0
M(3,1) = 0.0
M(3,2) = 0.0
M(3,3) = 0.0

XPRIME(1,1) = 0.0
XPRIME(1,2) = 0.0
XPRIME(1,3) = 0.0
XPRIME(2,1) = 0.0
XPRIME(2,2) = 0.0
XPRIME(2,3) = 0.0
XPRIME(3,1) = 0.0
XPRIME(3,2) = 0.0
XPRIME(3,3) = 0.0

DATA PI:3.14159265359/ 
DATA DTS:1.71532952502/ 
DATA COEFF5:3.162357003/, COEFFY/1.75879601/ 
DATA 1/10.000, 1.000/ 
DATA VELT:2.997928005/ 
DATA D'THETA/(5.00-2,1.00-2)/
CALL DF DEYS (HT, EN, NU)
GO TO 10, K=1, NSPEC
ENPRIM(K) = C(N(K)) + CE(K)
NUPRIM(K) = NU(K) * DELC(K)
10 CONTINUE
C
NFLAG = 0
DO 20 K = 1, NSPEC
C ADD IN THE CONTRIBUTIONS TO THE
C SUSCEPTIBILITY TENSOR M FOR EACH
C SPECIE IN THE IONOSPHERE.
10 EN(K) .LT. 1.0E-2 GO TO 20
NFLAG = 1
X = COEF EN(K) * EN(K)
XPRIME = CUEFEN(K) * ENPRIM(K) / WAVNR
Z = NU(K) * OMGA
U = 1.0E-2
UPRIME = I + NUPRIM(K) * OMGA / WAVNR
USQ = U * U
DD = X / (U * (USQ - YSQ(K)) )
DDPRIM = (X * (3.0D0 * USQ * UPRIME - YSQ(K) * UPRIME) -
1.0D0 * XPRIME * (U * (USQ - YSQ(K))) ) / (U * (USQ - YSQ(K))) + 2
IUD = Z * I + DD
IUDPRI = NUPRIM(K) * OMGA / WAVNR * DD + (Z * I) * DDPRIM
20 TA = USQ * DD
M(1,1) = M(1,1) + TA
M(2,2) = M(2,2) + TA
M(3,3) = M(3,3) + TA
M(2,2) = M(2,2) - MSQSQ(K) * DD
TAPRIM = 2.0D0 * U * UPRIME * DD + USQ * DDPRIM
MPRIME(1,1) = MPRIME(1,1) + TAPRIM
MPRIME(2,2) = MPRIME(2,2) + TAPRIM
MPRIME(3,3) = MPRIME(3,3) + TAPRIM
MPRIME(2,2) = MPRIME(2,2) - MSQSQ(K) * DDPRIM
30 TA = NY(K) * IUD
TB = LNYSQ(K) * DD
M(1,1) = M(1,1) + TA - TB
M(3,1) = M(3,1) - TA - TB
TAPRIM = NY(K) * IUDPRI
TBM = LMYSQ(K) * DDPRIM
MPRIME(1,3) = MPRIME(1,3) + TAPRIM - TBMPRIM
MPRIME(3,1) = MPRIME(3,1) - TAPRIM - TBMPRIM
101 IF (ISO .NE. 0) GO TO 20
M(1,1) = M(1,1) - LSQSQ(K) * DD
M(3,3) = M(3,3) - NSQSQ(K) * DD
LPRIME(1,1) = LPRIME(1,1) - LSQSQ(K) * DDPRIM
LPRIME(3,3) = LPRIME(3,3) - NSQSQ(K) * DDPRIM
103 TA = NY(K) * IUD
104 TB = LMYSQ(K) * DD
M(2,1) = M(2,1) + TA - TB
M(1,2) = M(1,2) - TA - TB
105 TAPRIM = NY(K) * IUDPRI
106 TBMPRIM = LMBYSQ(K) * DDPRIM
CPRIME(3,2) = MPRIME(2,2) - (MPRIME(2,3) * M(3,2) + M331 +
1 M(2,3) * MPRIME(3,2) + W331 - MPRIME(3,3) * M(2,3) * M(3,2)) / M331**2
CPRIME(4,1) = MPRIME(1,1) - (MPRIME(1,3) * M(3,1) + M331 +
1 M(1,3) * MPRIME(3,1) + W331 - MPRIME(3,3) * M(1,3) * M(3,1)) / M331**2
RETURN

C ENTRY INIT DT
ISO=0
IF (MAGFLD .EQ. 0.0) GO TO 250
IF (DABS(COLAT-90.0).GE.0.15) GO TO 300
IF (DABS(AZMUTH-90.0).LT.0.15) GO TO 250
IF (DABS(AZMUTH-270.0).GE.0.15) GO TO 300
ISO = 1
OMEGA = 2000.0 * PI * FREQ
OMEGA/V = 1.0 / OMEGA
WAVLEN = OMEGA / VELTT
SINDIP = DSIN (CODIP * DTR)
DRCOSL = SINDIP * DCO (AZMUTH * DTR)
DRCOSM = SINDIP * DSIH (AZMUTH * DTR)
DRCOSN = DCO (CODIP * DTR)
DO 60 K=1,NRSPEC
COEFFN(K) = COEFFX + (1.0E6 * CHARGE(K) * 2 / (OMEGA**2 * RATIO(K))
Y(K) = COEFFY + CHARGE(K) * MAGFLD
$ / (OMEGA * RATIO(K))
YQQ(K) = Y(K)**2
LX(K) = DRCOSL * Y(K)
MY(K) = DRCOSM * Y(K)
NY(K) = DRCOSN * Y(K)
LXYSQ(K) = DRCOSL**2 * YQQ(K)
MYYSQ(K) = DRCOSM**2 * YQQ(K)
NYYSQ(K) = DRCOSN**2 * YQQ(K)
LNYYSQ(K) = DRCOSN * DRCOSN * YQQ(K)
LNYYSQ(K) = DRCOSN * DRCOSN * YQQ(K)
LNYYSQ(K) = DRCOSN * DRCOSN * YQQ(K)
LHYYSQ(K) = DRCOSN * DRCOSN * YQQ(K)
LHYYSQ(K) = DRCOSN * DRCOSN * YQQ(K)
LHYYSQ(K) = DRCOSN * DRCOSN * YQQ(K)
LHYYSQ(K) = DRCOSN * DRCOSN * YQQ(K)
60 CONTINUE
C = DCCOS(TETA*DTR)
S = DCSIN(TETA*DTR)
CSQ = C**2
SSQ = S**2
CI = DCCOS((TETA-DTETA)*DTR)
SI = DCSIN((TETA-DTETA)*DTR)
CSQI = CI**2
SSQI = SI**2
RETURN
END
I0H3.0I10.E Q.0.MMA IHT, CI.HT, LSHT, Q. QAMMA)
C0.LL 'GT HEJT-(A-.D.0-Z)
I0H3.0I10.E LSHT
C0.LL 'GT HEJ T
11 L.G.LXTC .C.S.C. 5.1.0194, Q.GAMMA, A.U, F.J, F.DERIV.
12 $ FLX, 1, RENE, (3.4), TPRIME (4.4),
13 $ COEFF, COEFF, COEFF, COEFF, COEFF, COEFF, COEFF, COEFF1, COEFF2,
14 $ A1, A2, A3, A4, A5, A6, B4, B5, B6.
15 $ SAPTV, SAPTV, ASAP, ASAP, SPASAP, BSAPV, BSAPV,
16 $ BSAPV, BSAPV, BSAPV, BSAPV,
17 $ SPVSC, S, NORM(4.2), PART S
18 COMMON /CS.C, CS.S1
19 COMMON ANSVR, PART S(4.2)
20 COMMON S.MTX, SPVSC(4.4)
21 DATA 1.10, 0.000, 1.000/
22 (CALL QOKIMT (HT, LT, M, MPRI M, TPRIME)
23 COEF = 1.000
24 COEF = I(1,1)+I(4,4)
25 COEF = T(1,1)+T(4,4)-T(1,4)+T(4,1)-T(3,2)
26 COEF = T(3,2)*T(1,1)+T(4,4)-T(3,2)*T(4,1)
27 COEF = T(3,4)*T(4,2)-T(3,2)*T(4,4)
28 1 T(1,2)*T(3,1)+T(4,1)
29 2 T(1,4)+T(3,2)+T(4,1)
30 JJ = 7
31 IF (HT.LT.LSHT) GO TO 400
32 CALL Q0ART (COEFFB, COEFFC, COEFFD, COEFFE, QQ)
33 C8 MAX CO(1)
34 JU = 1
35 GO TO 23 J = 2.4
36 QREAL = QQ(J)
37 IF (QREAL.LT.QLR MAX) GO TO 23
38 QMAX = QREAL
39 JU = 0
40 CONTINUE
41 QQ(JU(J2))
42 500 CONTINUE
43 IF (K.GT.21) GO TO 510
44 F = 0.4+COEFFA +Q.*3+COEFFB +Q.*2+COEFFC +Q.*COEFFD +COEFFE
45 DERIV = 4.DO+Q.*3+COEFFA +3.DO+Q.*2+COEFFB +2.DO+Q.*COEFFC +COEFFD
46 REAL +DERIV
47 GINARY = -1/F/DERIV
48 IF ((REAL.GE.1.D-20) .AND. (GINARY.GE.1.D-30)) GO TO 530
49 QQ = F.DERIV
50 M = 1
51 GO TO 500
52 510 IF (JJ.EQ.7) GO TO 570
53 PRINT 520, HT
54 520 FORMAT (10X,'DOES NOT CONVERGE AFTER 20 ITERATIONS AT HEIGHT=',
55 1 F10.5/)
56 530 CONTINUE
57 HEAL+Q
55 GANARY=1*Q
56 IF (REAL*2 .GT. GANARY*2 .AND. REAL.GT.0.) GO TO 540
57 IF (J1.EQ.13) GO TO 550
58 570 JJ=13
59 GO TO 10
60 550 PRINT 'H10. HT
61 560 FORMAT (10X,'DOES NOT CONVERGE TO THE PROPER Q AT HEIGHT=',F10.5)
62 540 CONTINUE
63
64 A1=T(1,1)+T(4,4)
65 A2=T(1,1)+T(4,4)-T(1,4)*T(4,1)
66 A3=t(1,2)
67 A4=T(1,4)+T(4,2)-T(1,2)*T(4,4)
68 A5=T(4,2)
69 A6=T(4,1)+T(1,2)-T(1,1)*T(4,2)
70 B3=T(3,4)
71 B4=T(1,4)+T(3,1)-T(3,4)*T(1,1)
72 B5=T(3,1)
73 B6=T(4,1)+T(3,4)-T(4,4)*T(3,1)
74 A3PRIM=TPRIME(1,2)
75 A4PRIM=TPRIME(1,4)+T(4,2)+T(1,4)*TPRIME(4,2)-
76 1 TPRIME(1,2)*T(4,4)-T(1,2)*TPRIME(4,4)-
77 1 A5PRIM=TPRIME(4,2)
78 A6PRIM=TPRIME(4,1)+T(1,2)+T(4,1)*TPRIME(1,2)-
79 1 TPRIME(1,1)*T(4,2)-T(1,1)*TPRIME(4,2)-
80 B3PRIM=TPRIME(3,4)
81 B4PRIM=TPRIME(1,4)+T(3,1)+T(1,4)*TPRIME(3,1)-
82 1 TPRIME(3,4)*T(1,1)-T(3,4)*TPRIME(1,1)-
83 B5PRIM=TPRIME(3,1)
84 B6PRIM=TPRIME(4,1)+T(3,4)+T(4,1)*TPRIME(3,4)-
85 1 TPRIME(4,4)*T(3,1)-T(4,4)*TPRIME(3,1)
86 AJ=Q*A1+Q*A2
87 FJ=2.000*Q*AJ+(Q*Q-T(3,2))*(2.000+Q*A1)-(A3*B5+B3*A5)
88 COEFF0=A4*B6PRIM+A4PRIM*B6+A6*B4PRIM-A6PRIM*B4
89 COEFF1=A3*B6PRIM+A3PRIM*B6+A4+B5PRIM-A4PRIM*B5+A5*B4PRIM-
90 1 A5PRIM*B4+A5*B3PRIM-A5PRIM*B3
91 COEFF2=A3*BNPRIM*A3PRIM*B5+A5*B3PRIM-A5PRIM*B3
92 GAMMA=Q*Q*COEFF2+Q*COEFF1+COEFF0/(2.000+AJ+FJ)
93 NICE=ILWSTH -HT)/DELHT
94 17 ((LWSTH-.HT)/DELHT - NICE .NE. 0.0) RETURN
95 IF (HT.NE.LWSTHT) GO TO 605
96 DD 600 N=1.2
97 600 S NORM1,2) = (Q*A3*A4)/CDSORT(AJ+FJ) /SPVEC(1,N)
98 S NORM2,2) = (-AJ) /CDSORT(AJ+FJ) /SPVEC(2,N)
99 S NORM3,3) = (Q*A3) /CDSORT(AJ+FJ) /SPVEC(3,N)
100 600 S NORM4,4) = (Q*A5*A6)/CDSORT(AJ+FJ) /SPVEC(4,N)
101 605 DD DO 610 N=1.2
102 610 PART S(1,1) = (Q*A3*A4)/CDSORT(AJ+FJ) /S NORM1(1,N)
103 PART S(2,2) = (-AJ) /CDSORT(AJ+FJ) /S NORM2(2,N)
104 PART S(3,3) = (Q*A3) /CDSORT(AJ+FJ) /S NORM3(3,N)
105 610 PART S(4,4) = (Q*A5*A6)/CDSORT(AJ+FJ) /S NORM4(4,N)
107 RETURN
108 END
SUBROUTINE QUARTIC (FOURB3, SIXB2, FOURB1, B0, Q)
C QUARTIC FINDS THE ROOTS OF A QUARTIC POLYNOMIAL, FROM THE CLOSED FORM.
C
IMPLICIT REAL *8 (A-H,O-Z)
COMPLEX *16
CD, B3, B2, B1, B0, Q, FOURB3, SIXB2, FOURB1, B3SQ, H, I, G, HPRIME, GPRIME

REAL *8 MGPLUS, MGXNUS
DIMENSION PRI(2)
EQUIVALENCE (P, PRI)
DATA OMEGA1/(-5.0D-1, 8.6602540360-1)/
DATA OMEGA2/(-5.0D-1, -8.6602540360-1)/
DATA PRECSN/1.0D-10/

C

C

B3=FOURB3*0.25
B2=SIXB2/6.0
B1=FOURB1*0.25
B3SQ=B3**2
H=B2-B3SQ
I=B0-4.0*B3+B1+3.0*B2+2
G=B1+B3-(3.0*B2+2.0*B3SQ)
HPRIME=1/12.0
GPRIME=G**2/4.0-H*(H**2+3.0*HPRIME)
SQR01 = CSQRT (GPRIME**2 + 4.0 * HPRIME**3)
P=(-GPRIME+SQR01)**0.5
MGPLUS=DABS (PRI(1)) + DABS (PRI(2))
PPLUS=P
P=(-GPRIME-SQR01)**0.5
MGXNUS=DABS (PRI(1)) - DABS (PRI(2))
IF (Y2PLUS.GT.MGXNUS) P=PPLUS
LOGP = CLOG (P)
CBERT0 = CDEXP (LOGP / 3.0)
CBERT1=OMEGA1-CBERT0
CBERT2=OMEGA2-CBERT0
ROOTP = CSQRT (CBERT0 - HPRIME / CBERT0 - H)
ROOTQ = CSQRT (CBERT1 - HPRIME / CBERT1 - H)
ROOTR = CSQRT (CBERT2 - HPRIME / CBERT2 - H)
IF (CDABS(Q1).LE.1.00D-50) GO TO 5
S=SIGN(-ROOTP*ROOTQ*ROOTR-2.0/9)
IF (SIGNLT.0.0) ROOTR=-ROOTR
Q1(-ROOTP*ROOTQ*ROOTR-B3)
Q2(-ROOTP*ROOTQ*ROOTR-B3)
Q3(-ROOTP*ROOTQ*ROOTR-B3)
Q4=-ROOTP*ROOTQ*ROOTR-B3
DO 20 N=1,4
C 200 CONTINUE
ROOTP = Q(N)**4 + FOURB3 * Q(N)**3 + SIXB2 * Q(N)**2 + S * FOURB1 * Q(N) + B0
ROOTQ = 4.0 * Q(N)**3 + 3.0 * FOURB3 + Q(N)**2
55 $ = 2.0 \cdot $SIXB2 $ \cdot Q(N) + $DURB1
56 \text{ROOTR} = R0TP / RQUTQ
57 Q(N) $ = Q(N) - $ROOTR
58 \text{IF (COSR (ROOTR).LT.FRECSN) GO TO 20}
59 \text{ITER} = \text{ITER} + 1
60 \text{IF (ITER.LT.10) GO TO 200}
61 \text{PRINT 900, ITER, Q(N)}
62 \text{20 CONTINUE}
63 \text{RETURN}
64 \text{900 FORMAT (I3,'ITERATIONS, Q = ',}
65 $\$E15.5,E13.5,'FAILS TO CONVERGE')$
66 \text{END}
SUBROUTINE WF_BNDY(B)

C WF_BNDY COMPUTES THE VECTOR B, WHICH DETERMINES HOW TO COMBINE THE SOLUTION VECTORS IN ORDER TO SATISFY THE BOUNDARY CONDITIONS.

C

IMPLICIT REAL*8 (A-H,O-Z)
COMMON T_MTX, T11, T131, T141, T12, T32, T142, T14, T34, T44
COMMON M_MTX
COMMON CS_SINE, SINE, IGNOR2(I)
COMMON WF_COND, IGNOR3(2), WAVENR
COMMON P_MTX/P, IGNOR4(32)
COMMON EXC_AMTNT, RH1
COMMON OF_FLAG, ISKIP(3), IDOG
COMMON EXTRA_XTRA, R(4)
COMMON, A, T_P1, THETA, FREQ, AZIM, CODIP, MAGFLD, COEFFNU(5), EXPNU(5),
$ TOPH1, LWSTHT, WKBHT, DELHT, H, ALPHA, SIGMA, EPSLON
COMMON/ S_MTX/PP(4, 4)
COMPLEX*16 T(4, 4), T11, T131, T141, T12, T32, T142, T14, T34, T44,
$ Q(4), SPVEC(4, 4), ASPVEC(4, 4), BETA(2, 2), P(4, 2),
$ BB(4, 4), T Tình/(0.0, 1.0)/M(3, 3), SOURCE(9, 2), SINE,
$ B(4, 2), AAA(4), BBB(4), B(12)
COMPLEX*16 TM(4, 4), SUM(4, 4, 4), PROD(4, 4), SPVEC(4, 4)
COMPLEX*16 R, RBAR11, RBAR22, XTRA, COSINE, THETA, SMALL F, PP, QQ
REAL*8 MAGFLD, LWSTHT
REAL*4 ERR
DATA QMAX/200.00/

CALL TMTRX(0.00)
T(1, 1) = T11
T(2, 1) = 0.0
T(3, 1) = T131
T(4, 1) = T141
T(1, 2) = T12
T(2, 2) = 0.0
T(3, 2) = T32
T(4, 2) = T142
T(1, 3) = 0.0
T(2, 3) = 1.0
T(3, 3) = 0.0
T(4, 3) = 0.0
T(1, 4) = T14
T(2, 4) = 0.0
T(3, 4) = T34
T(4, 4) = T44
CALL EIGVAL(T, Q)
DQ 10 N=1, 4
QT = CDABS(Q(N))
IF(QT .GT. QMAX) IFAIL=1
CONTINUE
CALL EIGVEC(T, Q, SPVEC)
CALL WFSORT(Q, SPVEC, IFAIL)
C
DO 30 N=1,2
BB(1,1,N) = Q(N)**2*(T(4,4)-Q(N))-T(2,3)*(T(3,2)*(T(4,4)-Q(N))
S = T(4,2)**T(3,4)
BB(1,2,N) = T(1,2)**Q(N)**T(4,4)-Q(N)**T(1,1)**T(4,2)
BB(1,3,N) = T(1,2)**T(2,3)*(T(4,4)-Q(N))-T(1,4)**T(4,2)**T(2,3)
BB(1,4,N) = -T(1,2)**T(2,3)**T(3,4)**T(1,4)**Q(N)**2-T(3,2)**T(2,3)
BB(2,1,N) = T(2,3)**T(3,1)**T(4,4)-Q(N)**T(4,1)**T(3,4)
BB(2,2,N) = -Q(N)**(T(1,1)-Q(N))**T(4,4)-Q(N)**T(1,4)**(T(4,1)**Q(N)
BB(2,3,N) = -T(2,3)**T(1,1)**Q(N)**T(4,4)-Q(N)**T(1,4)**T(4,1)
S = T(2,3)
BB(2,4,N) = T(2,3)**T(3,4)**T(1,1)-Q(N)**T(4,1)**T(3,1)**T(2,3)
BB(3,1,N) = Q(N)**T(3,1)**T(4,4)-Q(N)**T(1,4)**T(3,4)
BB(3,2,N) = -T(1,1)-Q(N)**T(3,2)**T(4,4)-Q(N)**T(4,2)**T(3,4)
S = -T(1,2)**T(3,1)**T(4,4)-Q(N)**T(4,1)**T(3,4)
BB(3,3,N) = -Q(N)**(T(1,1)-Q(N))**T(4,4)-Q(N)**T(1,4)**T(4,1)
BB(3,4,N) = Q(N)**T(3,4)**T(1,1)-Q(N)**Q(N)**T(1,4)**T(3,1)
BB(4,1,N) = -Q(N)**2**T(4,1)**T(2,3)**T(3,1)**T(4,2)**T(4,1)**T(3,2)
BB(4,2,N) = Q(N)**(T(1,1)-Q(N))**T(4,2)**T(1,2)**Q(N)**T(4,1)
BB(4,3,N) = T(4,2)**T(2,3)**T(1,1)-Q(N)**T(1,2)**T(4,1)**T(2,3)
BB(4,4,N) = (T(1,1)-Q(N))*Q(N)**2**T(2,3)**T(2,3)**T(1,2)**T(3,1)
S = +T(2,3)
DO 16 N=1,4
TN(1,1,N) = T(1,1)-Q(N)
TN(2,2,N) = T(2,2)-Q(N)
TN(3,3,N) = T(3,3)-Q(N)
TN(4,4,N) = T(4,4)-Q(N)
TN(1,2,N) = T(1,2)
TN(1,3,N) = T(1,3)
TN(1,4,N) = T(1,4)
TN(2,1,N) = T(2,1)
TN(2,3,N) = T(2,3)
TN(2,4,N) = T(2,4)
TN(3,1,N) = T(3,1)
TN(3,2,N) = T(3,2)
TN(3,4,N) = T(3,4)
TN(4,1,N) = T(4,1)
TN(4,2,N) = T(4,2)
TN(4,3,N) = T(4,3)
DO 17 J=1,4
DO 16 K=1,4
DO 17 N=1,2
SUM(J,K,N) = TN(J,1,N)**BB(1,K,N)+TN(J,2,N)**BB(2,K,N)
CONTINUE
DO 40 N=1,2
TEMP = CDEAP(1+WAVENR*XTMRT)+Q(N))/(1.+M(3,3))
SOURCE(1,N) = -TEMP**SINE**2
SOURCE(2,N) = 0.0
SOURCE(3,N) = TEMP**M(2,3)**SINE
SOURCE(4,N) = -TEMP**M(1,3)**SINE
TEMP = (1.+M(3,3))**TEMP
SOURCE(5,N) = -TEMP**SINE
SOURCE(6,N) = -SOURCE(5,N)
SOURCE(7,N) = -SOURCE(5,N)**SINE
SOURCE(8,N) = SOURCE(5,N)
SOURCE(9,N) = SOURCE(5,N)
IF (*AHT,LWSTM*) GO TO 500
   C GO TO 1,4
111   IF(1,1,1),BB(1,1,1),SOURCE(1,1),BB(1,2,1),SOURCE(2,1)
112   S /((Q1)-Q2)*Q1-Q3*Q1-Q4)
113   S /((Q1)-Q2)*Q1-Q3*Q1-Q4)
114   S /((Q1)-Q2)*Q1-Q3*Q1-Q4)
115   CONTINUE
116   B(1,1) = AAA(1)
117   B(2,1) = BBB(1)
119   BP(1,1) = SOURCE(5,1)*BB(1,2,1)
120   B(1,2) = SOURCE(5,2)*BB(1,2,1)
121   B(1,2) = SOURCE(5,2)*BB(1,2,1)
122   B(1,2) = SOURCE(5,2)*BB(1,2,1)
123   B(1,2) = SOURCE(5,2)*BB(1,2,1)
124   B(1,2) = SOURCE(5,2)*BB(1,2,1)
125   B(1,2) = SOURCE(5,2)*BB(1,2,1)
126   B(1,2) = SOURCE(5,2)*BB(1,2,1)
127   B(1,2) = SOURCE(5,2)*BB(1,2,1)
128   B(1,2) = SOURCE(5,2)*BB(1,2,1)
129   B(1,2) = SOURCE(5,2)*BB(1,2,1)
130   B(1,2) = SOURCE(5,2)*BB(1,2,1)
131   B(1,2) = SOURCE(5,2)*BB(1,2,1)
132   B(1,2) = SOURCE(5,2)*BB(1,2,1)
133   B(1,2) = SOURCE(5,2)*BB(1,2,1)
134   B(1,2) = SOURCE(5,2)*BB(1,2,1)
135   B(1,2) = SOURCE(5,2)*BB(1,2,1)
136   B(1,2) = SOURCE(5,2)*BB(1,2,1)
137   B(1,2) = SOURCE(5,2)*BB(1,2,1)
138   B(1,2) = SOURCE(5,2)*BB(1,2,1)
139   B(1,2) = SOURCE(5,2)*BB(1,2,1)
140   B(1,2) = SOURCE(5,2)*BB(1,2,1)
141   B(1,2) = SOURCE(5,2)*BB(1,2,1)
142   B(1,2) = SOURCE(5,2)*BB(1,2,1)
143   B(1,2) = SOURCE(5,2)*BB(1,2,1)
144   B(1,2) = SOURCE(5,2)*BB(1,2,1)
145   B(1,2) = SOURCE(5,2)*BB(1,2,1)
146   B(1,2) = SOURCE(5,2)*BB(1,2,1)
147   B(1,2) = SOURCE(5,2)*BB(1,2,1)
148   B(1,2) = SOURCE(5,2)*BB(1,2,1)
149   B(1,2) = SOURCE(5,2)*BB(1,2,1)
150   B(1,2) = SOURCE(5,2)*BB(1,2,1)
151   B(1,2) = SOURCE(5,2)*BB(1,2,1)
152   B(1,2) = SOURCE(5,2)*BB(1,2,1)
153   B(1,2) = SOURCE(5,2)*BB(1,2,1)
154   B(1,2) = SOURCE(5,2)*BB(1,2,1)
155   B(1,2) = SOURCE(5,2)*BB(1,2,1)
156   B(1,2) = SOURCE(5,2)*BB(1,2,1)
157   B(1,2) = SOURCE(5,2)*BB(1,2,1)
158   B(1,2) = SOURCE(5,2)*BB(1,2,1)
159   B(1,2) = SOURCE(5,2)*BB(1,2,1)
160   B(1,2) = SOURCE(5,2)*BB(1,2,1)
161   B(1,2) = SOURCE(5,2)*BB(1,2,1)
162   B(1,2) = SOURCE(5,2)*BB(1,2,1)
163   RETURN
164   C GO TO 1,4
165   Q(2) = (Q2)-Q1)
166   Q(2) = (Q2)-Q1)
167   Q(2) = (Q2)-Q1)
168   Q(2) = (Q2)-Q1)
169   Q(2) = (Q2)-Q1)
170   Q(2) = (Q2)-Q1)
171   Q(2) = (Q2)-Q1)
172   Q(2) = (Q2)-Q1)
173   Q(2) = (Q2)-Q1)
174   Q(2) = (Q2)-Q1)
175   Q(2) = (Q2)-Q1)
176   Q(2) = (Q2)-Q1)
177   Q(2) = (Q2)-Q1)
178   Q(2) = (Q2)-Q1)
179   Q(2) = (Q2)-Q1)
180   Q(2) = (Q2)-Q1)
181   Q(2) = (Q2)-Q1)
182   Q(2) = (Q2)-Q1)
183   Q(2) = (Q2)-Q1)
184   Q(2) = (Q2)-Q1)
185   Q(2) = (Q2)-Q1)
186   Q(2) = (Q2)-Q1)
187   Q(2) = (Q2)-Q1)
169  B(1) = B(1,1)/(QO*P(1,1))
170  B(2) = SMALLF*B(1)
171  d(3) = SOURCE(5,2)*BB(1,1,2)/(QO*P(1,2))
172  B(4) = SMALLF*B(3)
173  B(5) = SOURCE(6,2)*BB(1,1,2)/(QO*P(1,2))
174  B(6) = SMALLF*B(5)
175  B(7) = SOURCE(7,2)*BB(1,1,2)/(QO*P(1,2))
176  B(8) = SMALLF*B(7)
177  B(9) = SOURCE(8,2)*BB(1,1,2)/(QO*P(1,2))
178  B(10) = SMALLF*B(9)
179  B(11) = SOURCE(9,2)*BB(1,1,2)/(QO*P(1,2))
180  B(12) = SMALLF*B(11)
181  RETURN
182
183  ENTRY DECOMP(H)
184  CALL IMTRX(H)
185  T(1,1) = T11
186  T(3,1) = T31
187  T(4,1) = T41
188  T(1,2) = T12
189  T(3,2) = T32
190  T(4,2) = T42
191  T(1,4) = T14
192  T(3,4) = T34
193  T(4,4) = T44
194  T(2,1) = 0.0
195  T(2,2) = 0.0
196  T(1,3) = 0.0
197  T(2,3) = 1.0
198  T(3,3) = 0.0
199  T(4,3) = 0.0
200  T(2,4) = 0.0
201  CALL EIGVAL(T,Q)
202  DO 101 N=1,4
203  QT = CDABS(Q(N))
204  IF(QT .GT. QMAX) IFAIL=1
205  CONTINUE
206  CALL EIGVEC(T,Q,SPVEC)
207  CALL WFSORT(Q,SPVEC,IFAIL)
208  DO 311 J=1,4
209  DO 311 K=1,4
210  311  SPVEC(J,K) = SPVEC(J,K)
211  CALL CINVER(SPVEC,ASPVEC,4,4,ERR)
212  DO 215 J=1,4
213  DO 215 K=1,4
214  PROD(J,K) = ASPVEC(J,1)*ASPVEC(1,K)+ASPVEC(J,2)*ASPVEC(2,K)
215  $ +ASPVEC(J,3)*ASPVEC(3,K)+ASPVEC(J,4)*ASPVEC(4,K)
216  CONTINUE
217  DO 220 N=1,2
218  DO 220 M=1,2
219  EETA(N,N) = ASPVEC(N,N,1)*P(1,N)+ASPVEC(N,N,2)*P(2,N)
220  $ +ASPVEC(N,N,3)*P(3.N)+ASPVEC(N,N,4)*P(4,N)
221  CONTINUE
222  CALL RBARS(C,SINE,SINE,RBAR11,RBAR22)
223  IF(CDABS(1.-R(1)*RBAR11) .LT. CDABS(1.-R(2)*RBAR22)) GO TO 230
224  SMALLF = -(1.-R(1)*RBAR11)/(R(3)*R(2))
225  GO TO 240
SUBROUTINE EIGVAL (A, SIG)
IMPLICIT REAL*8 (A-H,O-Z)
COMPLEX*16 A, SIG
DIMENSION A(4,4), SIG(4)
C
C COMPUTES 4 EIGENVALUES OF MATRIX A,
AND RETURNS THEM IN SIG.
THE COEFFICIENTS OF THE CHARACTERISTIC
POLYNOMIAL OF A ARE CALCULATED, AND
IT ROOTS COMPUTED BY QUARTC, A QUARTIC
POLYNOMIAL ROUTINE.
C
COMMON/AF,FLAG,PRECSN,ISO,IDBG
COMPLEX*16 B(4), BX(3), BY(3)
COMPLEX*16 BO, BI, SQROOT, SIGN
C
DIMENSION INDEX(4,4)
DATA INDEX /0, 4, 4, 3,
$ 3, 0, 4, 3, 2, 1, 0, 2, 2, 1, 1, 0/
C
C
IF(ISO .NE. 0) GO TO 600
DO 100 J = 1, 4
100 B(I) = 0.0
DO 500 I = 1, 3
IP = I + 1
DO 500 J = IP, 4
II = INDEX(I, J)
JJ = INDEX(J, I)
BX(1) = A(I,1) * A(J,2) - A(I,2) * A(J,1)
BX(2) = 0.0
BX(3) = 0.0
IF (J, NE. 2) GO TO 200
SX(2) = - A(I,1) - A(2, 2)
SX(3) = 1.0
GO TO 250
210 IF (I.EQ.1) BX(2) = - A(J,2)
IF (I.EQ.2) BX(2) = A(J,1)
C
250 BY(1) = A(I,1) * A(JJ,4) - A(I,4) * A(JJ,3)
BY(2) = 0.0
BY(3) = 0.0
IF (II,NE. 3) GO TO 300
BY(2) = - A(3,3) - A(4,4)
BY(3) = 1.0
GO TO 350
310 IF (JJ.EQ.4) BY(2) = - A(I1,3)
IF (JJ.EQ.3) BY(2) = A(I1,4)
C
350 SIGN = (-1)**(I+J-1)
B(1) = B(1) + SIGN * BX(1) * BY(1)
B(2) = B(2) + SIGN *
$ (BX(2) * BY(1) + BX(1) * BY(2))
B(3) = B(3) + SIGN *
$ (BX(3) * BY(1) + BX(2) * BY(2) + BX(1) * BY(3))
C.

540  M14 = B(4) + SIGN *

545  $ (Bx(2) * BY(3) + BX(3) * BY(2))

550  CONTINUE

555  C

560  CALL QUARTIC (B(4), B(3), B(2), B(1), SIG)

565  RETURN

570  C

575  C COMPUTES THE EIGENVALUES X O O X

580  C OF A AS THE ROOTS OF TWO O X X O

585  C QUADRATIC IF A HAS THE O X X O

590  C SPECIAL FORM ON THE RIGHT. X O O X

595  CONTINUE

600  B1 = 0.5 * (A(1,1) + A(4,4))

605  B0 = A(1,1) * A(4,4) - A(1,4) * A(4,1)

610  SQROOT = CD$ORT (B1**2 - B0)

615  SIG(1) = B1 + SQROOT

620  SIG(4) = B1 - SQROOT

625  B1 = 0.5 * (A(2,2) + A(3,3))

630  B0 = A(2,2) * A(3,3) - A(2,3) * A(3,2)

635  SQROOT = CD$ORT (B1**2 - B0)

640  SIG(2) = B1 + SQROOT

645  SIG(3) = B1 - SQROOT

650  RETURN

655  END
SUBROUTINE EIGVEC(A, SIG, VEC)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION A(4,4), SIG(4), VEC(4,4)
COMPLEX*16 A, SIG, VEC

COMPUTES FOUR EIGENVECTORS OF A,
AND RETURNS THEM IN VEC.
VECTORS ARE COMPUTED AS SOLUTION OF
REDUCED SYSTEM OF LINEAR EQNS.
COMMON::
COMMON::FLAG,PRECSN,ISO,DBG
DIMENSION B(3,3), Y(3), X(3)
COMPLEX*16 B, Y, X
REAL*4 ERR

IF(ISO .NE. 0) GO TO 600

DO 100 J = 1,4
   B(1,1) = A(2,2) - SIG(J)
   B(1,2) = A(2,3)
   B(1,3) = A(2,4)
   Y(1) = - A(1,1) * A(4,1)
  10 B(2,1) = A(3,2)
  11 B(2,2) = A(3,3) - SIG(J)
  12 B(2,3) = A(3,4)
  13 Y(2) = - A(3,1) * A(4,1)
  14 B(3,1) = A(4,2)
  15 B(3,2) = A(4,3)
  16 B(3,3) = A(4,4) - SIG(J)
  17 Y(3) = - A(4,1) * A(4,1)
  18 CALL CLINO(B, Y, X, 3, 3, 0, ERR)
  19 VEC(1,J) = A(1,4)
  20 VEC(2,J) = X(1)
  21 VEC(3,J) = X(2)
  22 VEC(4,J) = X(3)
  23 CONTINUE

COMPUTE VECTORS WITH VEC(2,J) = A(2,3)
DO 200 J = 2,3
   B(1,1) = A(4,1) - SIG(J)
   B(1,2) = A(4,2)
   B(1,3) = A(4,4)
   Y(1) = - A(1,2) * A(4,1)
  25 B(2,1) = A(3,1)
  26 B(2,2) = A(3,3) - SIG(J)
  27 B(2,3) = A(3,4)
  28 Y(2) = - A(3,2) * A(4,1)
  29 B(3,1) = A(4,3)
  30 B(3,2) = A(4,4)
  31 B(3,3) = A(4,4) - SIG(J)
  32 Y(3) = - A(4,1) * A(4,1)
  33 CALL CLINO(B, Y, X, 3, 3, 0, ERR)
  34 VEC(1,J) = A(1,4)
  35 VEC(2,J) = X(1)
  36 VEC(3,J) = X(2)
  37 VEC(4,J) = X(3)
  38 CONTINUE

...
CALL CLINEO (S, Y, X, 3, 3, 0, ERR)

VEC(1,J) = X(1)
VEC(2,J) = A(2,3)
VEC(3,J) = X(2)
VEC(4,J) = X(3)

200 CONTINUE
RETURN

C

C COMPUTE EIGENVECTORS IN X 0 0 X
C SIMPLIFIED FORM IF A HAS 0 X 0
C THE SPECIAL FORM ON 0 X 0
C THE RIGHT. X 0 0 X

C

600 CONTINUE

C COMPUTE VECTORS WITH VEC(1,J) = A(1,4).
DO 300 J = 1,4,3
VEC(1,J) = A(1,4)
VEC(2,J) = 0.0
VEC(3,J) = 0.0
VEC(4,J) = SIG(J) - A(1,1)
300 CONTINUE

C

C COMPUTE VECTORS WITH VEC(2,J) = A(2,3)
DO 400 J = 2,3
VEC(1,J) = 0.0
VEC(2,J) = A(2,3)
VEC(3,J) = SIG(J) - A(2,2)
VEC(4,J) = 0.0
400 CONTINUE

RETURN
END
SUBROUTINE WF_SORT (Q, A, IFAIL)
IMPLICIT REAL*8 (A-H,O-Z)
COMPLEX*16 Q(4), A(4,4)
C
C  WF SORT INITIALLY ARRANGES THE
C  Q VALUES SO THAT THEY OCCUR IN THE
C  ORDER UPWARD FAST (EVANESCENT),
C  UPWARD SLOW (TRAVELLING), DOWNWARD
C  SLOW, AND DOWNWARD FAST.
C
COMPLEX*16 QT(4), B(4,4), I, CT
DIMENSION QI(4)
DATA 1 /(0.000,1.000)/
C
DO 510 J = 1,4
QI(J) = Q(J)
QI(J) = - I * Q(J)
DO 510 K = 1,4
510 B(J,K) = A(J,K)
DO 530 J = 1,4
T = QI(I)
JJ = 1
DO 520 K = 2,4
IF(QI(K).GE.T) GO TO 520
T = QI(K)
JJ = K
520 CONTINUE
QI(J) = QT(JJ)
DO 525 K = 1,4
525 A(K,J) = B(K,JJ)
QI(JJ) = 1.0D50
530 CONTINUE
QI(1) = - I * Q(2)
QI(2) = Q(2)
IF (DABS(QI(1)/QI(2)).LT.0.2) GO TO 550
IF(QI(1).GT.0.0) GO TO 850
QI(1) = - I * Q(3)
IF(QI(1).LT.0.0) GO TO 850
RETURN
550 QI(1) = Q(3)
IF (QI(1).GT.QI(1)) GO TO 560
CT = Q(2)
Q(2) = Q(3)
Q(3) = CT
QI(3) = QI(2)
QI(2) = QI(1)
QI(1) = QI(3)
DO 555 K = 1,4
CT = A(K,2)
A(K,2) = A(K,3)
555 A(K,3) = CT
560 IF(QI(1).GT.0.0) GO TO 850
IF(QI(1).GT.0.0) GO TO 850
C 8501 IFAIL = 1 RETURN
C 8502 PRINT 901, Q
C 8503 PRINT 902, Q, Q2

901 FORMAT (X,, Q = E15.5, E13.5),

902 FORMAT (ERROR IN INITIAL Q VALUES)

905 RETURN
C 906 FORMAT ('Q ELEMENTS CANNOT BE ORDERED')
C 907 RETURN
SUBROUTINE RBARS(C, S, RBAR11, RBAR22)

IMPLICIT REAL *8(A-H,O-Z)
COMMON/WFINP/THETA,FREQ,AZMUTH,COORD,MAGFLD,CEFFNU(5),EXPNU(5),
$ TOPHT,LWSHT,WKBHT,DELHT,H,ALPHA,SIGMA,EPSON
COMMON/STOR/STORE
COMMON/WF,CON/OMEGA,K
COMPLEX +18 THETA,1,NGSQ,C,S,SSQ,S2ROOT,T10RT,IKC,
$ PO,H10,H20,H1PRM0,H2PRM0,CAPH10,CAPH20,
$ PD,H1D,H2D,H1PRM0,H2PRM0,CAPH1D,CAPH2D,
$ PZ,H1Z,H2Z,H1PRMZ,H2PRMZ,
$ AI1T,A2ND,A3RD,A4TH,A1,A2,A3,A4,
$ EXD,EXDQ,EX2,EX2SQ,
$ RBAR11,RBAR22,Z1,Z2,
$ DEN12,DEN34,
$ EX,EX,EZ,HX,HY,HZ
REAL +8 K,KVRA0T,KVRA0T,NOSQ,NOSQ,NOSQ,MAGFLD,LWSHT
EQUIVALENCE(PZ,PD),(H1Z,H1D),(H2Z,H2D),(H1PRMZ,H1PRMD),
$ (H2PRMZ,H2PRMD),(EXD,EXZ),(EXDQ,EX2SQ)
$ DATA I/(0.000,1.000)/
$ DATA TSTTHM/I.0D1/
$ DATA EPSLNO/8.65434D+12/

C

D = TOPHT
SSQ=SSQ+
NGSQ = (EPSLNO-I*SIGMA/OMEGA)/EPSLNO
S2ROOT=CDSSQ(NGSQ-SSQ)
THTIM=I*THETA
IF(THTIM .GT. TSTTHM) GO TO 10

C

KVRA0T=DEXP(DLOG(K/ALPHA)/3.0)
KVRA0T=KVRA0T**2
AVRKT0=1.0/KVRA0T
AVRKT2=AVRKT0**2+0.5
NOSQ1=0.0-ALPHA**(H-STORE)
RT10RT=NOSQ/NOSQ/SSQ
PD=KVRA0T**(NOSQ-SSQ)
CALL MDHNLK (PO,H10,H20,H1PRM0,H2PRM0)
CAPH10=H1PRM0+AVRKT0+H10
CAPH20=H2PRM0+AVRKT0+H20
A1ST=CAPH20+1+RT10RT*KVRA0T+H20
A2ND=CAPH10+1+2*RT10RT*KVRA0T+H10
A3RD=H2PRM0+1*KVRA0T+SSQRT+H20
A4TH=H1PRM0+1*KVRA0T+SSQRT+H10
DEN12 = H2D-A2ND-H10*A1ST
DEN34 = H2D-A4TH-H10+A3RD
IF(D .EQ. 0.0) GO TO 10

C

NOSQ1=0.0-ALPHA**(H-0)
PD=KVRA0T**(NOSQ-SSQ)
CALL MDHNLK (PO,H10,H20,H1PRM0,H2PRM0)
CAPH1D=H1PRMD+AVRKT0+H10
CAPH2D=H2PRMD+AVRKT0+H20
A1 = C*NDSQ*(H2D*A2ND-H1D*A1ST)
A2 = I.AVEGT*(CAPH1D*A1ST-CAPH2D*A2ND)
A3 = I.AVEXT*(H2PRM0*A4TH-H1PRM0*A3RD)
A4 = C*(H2D*A4TH-H1D*A3RD)
REALT1=(A1-A2)/(A1+A2)
REALT2=(A3*A4)/(A4*A3)
RETURN

C FLAT EARTH
10 IKC = I*K+C
20 EXD = CDEXP(-IKC*(STORE-D))
30 EXDSQ = EXD**2
40 Z1 = (NGSQ+C*SQR04)/(NGSQ+C+SQR04)
50 Z2 = (C-SQR04)/(C+SQR04)
60 RBAR11 = Z1*EXDSQ
70 RBAR22 = Z2*EXDSQ
80 RETURN

C ENTRY H1 GAIN(ALT,EX,EX,EZ,HX,HY,HZ)
40 IF(TSTHM .GT. TSTTIM) GO TO 50
50 N2SQ = 1.0-ALPHA*(H-ALT)
60 PZ = KVREQ*(N2SQ-SQG)
70 CALL MUNNL(PZ,H1Z,H2Z,H1PRM,H2PRM)
80 EXPON = DEXP(-0.5*ALPHA*ALT)
90 HY = (H2Z*A2ND-H1Z*A1ST)*EXPON/DEN12
100 EZ = (H2Z*A4TH-H1Z*A3RD)/DEN34
110 EX = I*AVRKT*(H2PRM*A2ND-H1PRM*A1ST)/DEN12
120 S = EXPON+AVRKT*HY/N2SQ
130 EZ = -S/N2SQ*HY
140 HZ = S*EX
150 HX = AVRKT/I*(H2PRM*A4TH-H1PRM*A3RD)/DEN34
160 RETURN

C FLAT EARTH
50 EXZ = CDEXP(-IKC*(STORE-ALT))
60 EXZSQ = EXZ**2
70 HY = (1.0+Z1*EXZSQ)/(1.0+Z1)/EXZ
90 EZ = (1.0+Z2*EXZSQ)/(1.0+Z2)/EXZ
90 EX = -C*(1.0-Z1*EXZSQ)/(1.0+Z1)/EXZ
110 EZ = -S*HY
120 HZ = S*EX
130 HX = C*(1.0-Z2*EXZSQ)/(1.0+Z2)/EXZ
140 RETURN

END
DATA C\$12\$1, 2.588190451025220-01, 9.65925826284067C-01)/
DATA C\$13\$1, -9.65925826284067D-01, 2.588190451025220-01)/
DATA C\$14\$1,(-9.65925826289967D-01, -2.588190451025220-01)/
C
50  ZPOWER=1.0
60  SUM1=0.0
70  SUM2=0.0
80  ZTERM=-Z**3/200.0
90  DO 50 M=1,N
100  SUM1=SUM1+A(M)*ZPOWER
110  SUM2=SUM2+B(M)*ZPOWER
120  SUM3=SUM3+C(M)*ZPOWER
130  SUM4=SUM4+D(M)*ZPOWER
140  ZPOWER=ZPOWER+ZTERM
150  IF(COMPZPOWER),I.E.,1.00>30) GO TO 60
160  CONTINUE
170  GM2F=1.0*(Z+SUM2)**2.0/SUM1/ROOT3
180  GPMFP=1.0*(SUM4+2.0**Z+Z**SUM3)/ROOT3
190  H1=Z+SUM2*GM2F
200  H2=H1-2.0*GM2F
210  H1PRIME=SUM4+GPMFP
220  H2PRIME=H1PRIME-2.0*GPMFP
230  RETURN
C
70  SUM1=1.0
80  SUM2=1.0
90  R1Z=CD5ORT(Z)
100  SQRTZB=PTZ+Z
110  ZTERM=1.0/SQRTZB
120  MPower=1.0
130  TERN=1.5/Z
140  GO 60 M=1,14
150  ZPOWER=ZPOWER+ZTERM
160  SPOWER=MPower+1.0/ZTERM
170  TERN1=CAP1(M)*SPOWER
180  TERN2=CAP2(M)*SPOWER
190  SUM1=SUM1+TERM1
200  SUM2=SUM2+TERM2
210  SUM3=SUM3+TERM1
220  SUM4=SUM4+TERM2
230  CONTINUE
240  SUM3=SUM3+TERM
250  SUM4=SUM4+TERM
260  EXP1=CEXP1(2.0*1*SQRZB, 3.)
270  EXP2=EXP1+CGNST1
280  EXP3=CGNST2/EXP1
112  EXP4=CONST3*EXP1
113  EXP5=CONST4/EXP1
114  BETA=ALL*IA/CD3RT(RTZ)
115  ZREAL=Z
116  ZIMAG=-I*Z
117  IF (ZREAL.GE.0.0.OR.ZIMAG.GE.0.0)GO TO 90
118  H1=BETA*(EXP2+SUM2+EXP5+SUM1)
119  H1PRME=BETA+(EXP2+SUM2+SUM4)+SUM1*(-0.25/Z+I*RTZ)+(SUM1+(-0.25/Z)
120     $               -I*RTZ)+SUM3))
121  GO TO 110
122  90  H1=BETA*EXP2+SUM2
123  H1PRME=BETA+EXP2*(SUM2*(-0.25/Z+I*RTZ)+SUM4)
124  IF (ZREAL.GE.0.0.OR.ZIMAG.LT.0.0)GO TO 120
125  H2=BETA*(EXP3+SUM2+EXP4+SUM2)
126  H2PRME=BETA*(SUM1*(-0.25/Z-I*RTZ)+SUM3)+EXP4*(SUM2*(-0.25/Z
127     $               +I*RTZ)+SUM4))
128  RETURN
129  120  H2=BETA*EXP3+SUM1
130  H2PRME=BETA*EXP3*(SUM1*(-0.25/Z-I*RTZ)+SUM3)
131  RETURN
132  END
SUBROUTINE CLIN EQ (A, B, X, N, 
$ N$ DIM, IFLG, ERR)

CLIN EQ USES L=U DECOMPOSITION TO
FIND THE TRIANGULAR MATRICES L, U
SUCH THAT L * U = A. L AND U ARE
STORED IN A. THIS FORM IS USED WITH
B - BACK-SUBSTITUTION TO FIND THE SOLN
X I F A * X = L * U * X = B.
N IS THE NUMBER OF EQUATIONS AND
N DIM IS THE DIMENSION OF ALL ARRAYS
IN THE PARAMETER LIST.

IF IFLAG = 0, L, U, AND X ARE COMPUTED
IF IFLAG IS NON-ZERO, IT IS ASSUMED
THAT L AND U HAVE BEEN COMPUTED IN
A PREVIOUS CALL AND ARE STILL STORED
IN A. THUS ONLY X IS COMPUTED.
ERR IS THE ESTIMATED RELATIVE
ERROR OF THE SOLUTION VECTOR.

C
C COMPLEX*16 A, B, X, T
INTEGER*4 IROW
DIMENSION A(N DIM, N DIM), 
B(N DIM, X(N DIM)
DIMENSION IROW(50), Q(50)
DATA EPS /1.0E-15/

C
IF (N.GT.50) GO TO 900
IF (IFLAG.NE.0) GO TO 600
DO 50 I = 1,N
Q(I) = 0.0
DO 40 J = 1,N
QQ = CDABS (A(I,J))
IF (Q(I).LT.QQ) Q(I) = QQ
IF (Q(I).EQ.0.0) GO TO 901
CONTINUE
ERR = EPS
PPIV = 0.0
DO 100 I = 1,N
IROW(I) = I

DO 500 L = 1,N
FIVOT = 0.0
K = L - 1
DO 400 I = L,N
IF (K.LT.I) GO TO 230
DO 220 J = 1,K
A(I,J) = A(I,J) - A(J,L) * A(I,J)
DO 230 F = CDABS (A(I,J)) / Q(I)
IF (F.PT.GT.1.0) GO TO 240
FIVOT = F
APIVOT = I
CONTINUE
IF (PIVOT.EQ.0.0) GO TO 901
IF (PPIV.LE.PIVOT) GO TO 250
ERR = ERR + PPIV / PIVOT
IF (ERR.GE.1.0) GO TO 901
250 PPIV = PIVOT
IF (NPIVOT.EQ.L) GO TO 280
Q(NPIVOT) = Q(L)
J = IROW(L)
IROW(L1) = IROW(NPIVOT)
IROW(NPIVOT) = J
DO 260 I = 1,N
T = A(L,I)
A(L,I) = A(NPIVOT,I)
A(NPIVOT,I) = T
260 CONTINUE
280 IF (.LT.1.0) GO TO 500
T = (A(DO0,0.0DO) / A(L,L)
M = L + 1
DO 450 I = K,N
IF (.LT.M) GO TO 400
DO 350 J = 1,M
350 A(L,I) = A(L,I) - A(L,J) * A(J,I)
400 A(L,I) = T * A(L,I)
450 CONTINUE
500 CONTINUE
C 
600 DO 620 I = 2,N
620 X(J) = (DO0,0.0DO)
J = IROW(J)
X(I) = B(J) / A(1,1)
DO 700 I = 2,N
J = IROW(I)
M = I - 1
DO 850 L = 1,K
850 X(I) = X(I) + A(I,L) * X(L)
X(I) = (B(J) - X(I)) / A(1,1)
700 CONTINUE
C 
900 PRINT 999
ERR = 1.0
901 CONTINUE
ERR = 1.0
RETURN
C 
997 FORMAT ('ERROR IN CLIN EQ. MATRIX IS SINGULAR')
998 FORMAT ('CAUTION'.
$' CLIN EQ HAS DECOMPOSED AN ILL-CONDITIONED MATRIX',/,
SUBROUTINE CINV (A, A INV, N, N DIM, ERR)

C CINVER SUCCESSIVELY COMPUTES EACH COLUMN
C OF THE INVERSE MATRIX OF A, USING THE
C SUBROUTINE CLIN EQ TO SOLVE A * X = E
C FOR EACH UNIT VECTOR E.
C N IS THE ORDER OF THE MATRIX A AND
C N DIM IS THE DIMENSION OF ALL ARRAYS
C IN THE PARAMETER LIST.
C ERR IS THE ESTIMATED RELATIVE ERROR
C OF THE INVERTED MATRIX.
C THE INVERSE MATRIX IS
C RETURNED IN A INV.

COMPLEX*16 A, A INV, B, X
DIMENSION A(N DIM, N DIM),
$ A INV(N DIM, N DIM),
DIMENSION B(50), X(50)

C
IF (N.GT.50) GO TO 900
II = 0
DO 100 I = 1, N
100 B(I) = (0.000,0.000)
DO 300 I = 1, N
200 A INV(J,I) = X(J)
300 CONTINUE
IF (ERR.GT.1.0E-5) GO TO 901
RETURN

900 PRINT 998
901 PRINT 999
RETURN
998 FORMAT ('ERROR IN CINV, MATRIX SIZE GREATER THAN 50')
999 FORMAT ('CLIN EQ WAS CALLED BY CINV')
END
FUNCTION CDANG(ARG)

IMPLICIT REAL*8 (A-H,O-Z)

COMPLEX*16 ARG, PRT

DIMENSION PARTS(2)

EQUIVALENCE(ARG, PARTS)

ARG PRT = ARG

ARG RL = PARTS(1)

ARG IM = PARTS(2)

RETURN

END
BLOCK DATA

INITIALIZE THE COMMON BLOCK VALUES.

IMPLICIT REAL *B (A-H,O-Z)
COMMON AINFY, THETA, FREQ, AZMUTH, CODIP, MAGFLO, CEFFNU(5), EXPNU(5), TTOPHT, LSTHT, WKBHT, DELHT, H, ALPHA, SIGMA, EPSLON
COMMON N FLAG, PRECSN, ISJ, IDBG
COMMON AEROF INHT(100), ENLUG(100, 5), COLLHT(25), COLFR(25, 5)
COMMON EXC IN/ TXHT, RXHT
COMMON EXC in/ TXHT, RXHT
REAL *B MAGFLO, LWSHT
COMMON THETA

DATA MAGFLO = 0.000/
DATA CEFFNU = 1.816D11, 4*0.000/
DATA EXPNU = 1.50-4.4*0.000/
DATA TOUHT = 100.000/, LWSHT/0.000/, WKBHT/0.000.000/
DATA H = 0.000/
DATA ALPHA = 3.14D-4/
DATA SIGMA = 4.6400/
DATA EPSLON = 7.172015D-10/
DATA PRECSN = 3.00-5/
DATA IDBG = 1/
DATA CHARGE = 1.000, 1.000, -1.000, 1.000, -1.000/
DATA RATIOH = 1.000, 4*5.804/
DATA NRSPEC = 1/
END
APPENDIX B

MODE SUM AND PLOTTING PROGRAM LISTING
THIS PROGRAM WAS DEVELOPED FOR USE ON THE UNIVAC 1100/21 AT THE
NAVAL OCEAN SYSTEMS CENTER. IT WAS WRITTEN IN ASCII FORTRAN USING
THE LEVEL 101 COMPILER WHICH CONFORMS TO AMERICAN NATIONAL
STANDARDS INSTITUTE FORTRAN-77 STANDARDS.
THE DISPLA PACKAGE WAS USED TO GENERATE THE PLOTS.

THE PROGRAM USES EXCITATION FACTORS GENERATED BY THE
WKB SATELLITE PROGRAM TO CALCULATE MODE SUMS FOR THREE
ORTHOGONAL FIELD COMPONENTS - EX, EY, AND EZ. SIX CURVES ARE
GENERATED FOR EACH FIELD COMPONENT DESIRED:
- ELECTRIC DIPOLE - VERTICAL
- ELECTRIC DIPOLE - HORIZONTAL-BROADSIDE
- ELECTRIC DIPOLE - HORIZONTAL-ENDON
- MAGNETIC DIPOLE - VERTICAL
- MAGNETIC DIPOLE - HORIZONTAL-BROADSIDE
- MAGNETIC DIPOLE - HORIZONTAL-ENDON

NOTE THAT THE EXCITATION FACTORS HAVE THE HEIGHT OF THE RECEIVER
BUILT INTO THEM SO IF THE USER WISHES MODE SUMS FOR DIFFERENT
TRANSMITTER AND/OR RECEIVER HEIGHTS NEW EXCITATION FACTORS MUST
BE GENERATED.

CALL INPUT
CALL FIELDS
CALL OUTPUT
STOP
END
SUBROUTINE INPUT

THIS SUBROUTINE READS IN AN IDENTIFICATION CARD FOR PLOT LABELS, 
THE NAMELIST VARIABLES, AND THE COMPLEX EIGEN ANGLE AND 
EXCITATION FACTORS FOR EACH MODE.

IMPLICIT REAL*8(A-H,O-Z)
REAL*8 IL,IA
REAL*4 AZIM,CODIP,MAGFLD,SIGMA,EPSP,DIST,AMP
COMPLEX*16 IM/(0.000,1.000)/
COMPLEX*16 THETA,EXCIT
CHARACTER*14 LABEL
DIMENSION XTRMAG(3),XTRANG(3)
COMMON/ONE/RHOMIN,RHOMAX,DELRHO,MODES,IPLOT
COMMON/TWO/AZIM,CODIP,MAGFLD,SIGMA,EPSP
COMMON/THREE/THETA(10),EXCIT(3,6,10)
COMMON/FOUR/FREQ,IL,IA,TXHT,RXHT,DIST(500),AMP(6,500),NPTS,
$ IPLOT(20),LABEL
$ NAMELIST/DATUM/FREQ,RHOMIN,RHOMAX,IL,IA,MODES,DELRHO,TXHT,RXHT,
$ IPLOT,AZIM,CODIP,MAGFLD,SIGMA,EPSP

DESCRIPTION OF NAMELIST VARIABLES

FREQ - FREQUENCY IN KHZ
RHOMIN - MINIMUM DISTANCE (IN KM) AT WHICH FIELDS ARE PRINTED 
AND PLOTTED
RHOMAX - MAXIMUM DISTANCE (IN KM) AT WHICH FIELDS ARE PRINTED 
AND PLOTTED
IL - ELECTRIC DIPOLE CURRENT MOMENT IN AMP METERS
IA - MAGNETIC DIPOLE MOMENT IN AMP METERS Squared
MODES - NUMBER OF MODES USED IN MODE SUMS
DELRHO - DISTANCE INCREMENT (IN KM) AT WHICH FIELDS ARE PRINTED 
AND PLOTTED
TXHT - TRANSMITTER HEIGHT IN KM
RXHT - RECEIVER HEIGHT IN KM
IPLOT - PLOTTING FLAG
   IPLOT=0 NO PLOTS
   IPLOT=1 EZ PLOT
   IPLOT=2 EZ AND EZ PLOTS
   IPLOT=3 EX, EY, AND EZ PLOTS
THE FOLLOWING FIVE VARIABLES ARE USED SOLEY FOR IDENTIFICATION
PURPOSES ON THE PLOTS
AZIM - AZIUTH IN DEGREES
CODIP - CODIP IN DEGREES
MAGFLD - MAGNETIC FIELD IN WEBERS PER SQUARE METER
SIGMA - CONDUCTIVITY IN SIEMENS PER METER
EPSR - DIELECTRIC CONSTANT

PRINT 902
READ 900,IPLOT
PRINT 910,IPLOT
READ(5,DATUM,END=999)
55 WRITE(6,DATU)
56 DO 50 M=1,MOIES
57 READ 915,THETA(M)
58 PRINT 917,M,THETA(M)
59 DO 50 L=1,3
60 READ 920,(XTRMAG(J),XTRANG(J),J=1,3)
61 DO 50 J=1,3
62 EXCIT(J,L,M) = XTRMAG(J)*(DCOS(XTRANG(J))+IM*DSIN(XTRANG(J)))
63 50 CONTINUE
64 C
65 900 FORMAT(20A4)
66 902 FORMAT('1')
67 910 FORMAT('0',20A4)
68 915 FORMAT(6X,F6.3,1X,F7.3)
69 917 FORMAT(' MODE ','12.',THETA = ',2F7.3)
70 920 FORMAT(1X,3(4X,D13.6,1X,F7.3))
71 C
72 999 RETURN
73 END
SUBROUTINE FIELDS

THIS SUBROUTINE CALCULATES MODE SUMS FOR THREE ORTHOGONAL FIELD COMPONENTS - EX, EY, AND EZ.

IMPLICIT REAL*8(A-H,O-Z)
COMPLEX*16 IM(0.000,1.000),SUM(3,6),THETA,EXCIT
REAL*8 IL,IA
REAL*4 AZIM,CODIP,MAGFLD,SIGMA,EPSP,DIST,AMP,FLD
CHARACTER*14 LABEL

COMMON/ONE/RHOMIN,RHOMAX,DELPHI,MODES,IPLLOT
COMMON/TWO/AZIM,CODIP,MAGFLD,THETA,EPSP
COMMON/THREE/THETA(10),EXCIT(3,6,10)
COMMON/FOUR/FREQ,IL,IA,TXHT,RXHT,DIST(500),AMP(6,500),NPTS,
$ IDPLOT(20),LABEL
COMMON/FIVE/FLD(3,6,500)

DATA TWPI/6.28318530710/,DEGRAD/1.7453290-02/,VELITE/2.99792802/
DATA A/6.37000/,C1/2.8570-3/,C2/5.9850-8/

C WAVENO = TWPI*FREQ*1.003/VELITE
C CONS1 = C1*IL*DSQRT(FREQ**3)
C CONS2 = C2*IA*DSQRT(FREQ**5)
C
NPTS = 0
RHO = RHOMIN

C LOOP OVER RHO
100 FAC = DSQRT(DSIN(RHO/A))
NPTS = NPTS+1
CONS1 = CONS1/FAC
CONS2 = CONS2/FAC
DIST(NPTS) = RHO
DO 200 J=1,3
DO 200 L=1,6
SUM(J,L) = (0.000,0.000)
200 CONTINUE
DO 400 M=1,MODES
TERM = CDEXP(-IM*WAVENO*RHO*DSIN(THETA(M)*DEGRAD))
DO 300 J=1,3
DO 300 L=1,6
SUM(J,L) = SUM(J,L)*EXCIT(J,L,M)*TERM
300 CONTINUE
400 CONTINUE
DO 500 J=1,3
DO 500 L=1,3
FLD(J,L,NPTS) = 2.001*DLOG10(CDABS(CONS1*SUM(J,L)))
FLD(J,L+3,NPTS) = 2.001*DLOG10(CDABS(CONS2*SUM(J,L+3)))
500 CONTINUE
RHO = RHO+DELPHI
IF(RHO .LE. RHOMAX) GO TO 100
SUBROUTINE OUTPUT

THIS SUBROUTINE PRINTS THE FIELDS AND CALLS THE PLOT ROUTINE.

IMPLICIT REAL*8(A-H,O-Z)
REAL*8 IL,IA
REAL*4 DIST,AMP,FLD
CHARACTER*14 LABEL
COMMON/ONE/RHOMIN,RHOMAX,DELHRO,MODES,IPLOT
COMMON/FOUR/FREQ,IL,IA,TXHT,RXHT,DIST(500),AMP(6,500),NPTS,
$ IOPL(20),LABEL
COMMON/FIVE/FLD(3,6,500)

GO TO (740,730,720,710),IPLOT+1
EX OPTION
710 LABEL = 'EX - DB/+M*V/M'
DO 712 L=1,6
DO 712 N=1,NPTS
AMP(L,N) = FLD(3,L,N)
712 CONTINUE
CALL FLDPLT
PRINT 910
PRINT 940
PRINT 950
DO 714 N=1,NPTS
PRINT 960,DIST(N),(AMP(L,N),L=1,6)
714 CONTINUE
EX OPTION
720 LABEL = 'EY - DB/+M*V/M'
DO 722 L=1,6
DO 722 N=1,NPTS
AMP(L,N) = FLD(2,L,N)
722 CONTINUE
CALL FLDPLT
PRINT 920
PRINT 940
PRINT 950
DO 724 N=1,NPTS
PRINT 960,DIST(N),(AMP(L,N),L=1,6)
724 CONTINUE
EX OPTION
730 LABEL = 'EZ - DB/+M*V/M'
DO 732 L=1,6
DO 732 N=1,NPTS
AMP(L,N) = FLD(1,L,N)
732 CONTINUE
CALL FLDPLT
740 CONTINUE
PRINT 930
PRINT 940
PRINT 950
DO 734 N=1,NPTS

115
55      PRINT 960, DIST(N), (FLD(I, L, N), I=1, 6)
56      CONTINUE
57      C
58      910      FORMAT ('OX COMPONENT')
59      920      FORMAT ('OY COMPONENT')
60      930      FORMAT ('OZ COMPONENT')
61      940      FORMAT ('0', 12X, 'ELECTRIC DIPOLE', 5X, 'MAGNETIC DIPOLE', 5X)
62      950      FORMAT (9X, 2(8X, 'VERTICAL', 6X, 'HORIZONTAL-BROADSIDE', 2X,
63          $ 'HORIZONTAL-ENDON'))
64      960      FORMAT (' ', F6.3, G20.7)
65      RETURN
66      END
SUBROUTINE FLDPLT

C USING THE DISSPLA PLOTTING PACKAGE THIS SUBROUTINE GENERATES
SIX PLOTS ON ONE 8.5 X 11 PIECE OF PAPER,
THE SIX PLOTS ARE:
ELECTRIC DIPOLE - VERTICAL
ELECTRIC DIPOLE - HORIZONTAL-BROADSIDE
ELECTRIC DIPOLE - HORIZONTAL-ENDON
MAGNETIC DIPOLE - VERTICAL
MAGNETIC DIPOLE - HORIZONTAL-BROADSIDE
MAGNETIC DIPOLE - HORIZONTAL-ENDON

REAL*8 FREQ, IL, IA, TXHT, RXHT
REAL*4 AZIM, CODIP, MAGFLD, SIGMA, EPSR
CHARACTER*14 LABEL
DIMENSION Y(500)
COMMON/TWO/AZIM, CODIP, MAGFLD, SIGMA, EPSR
COMMON/FOUR/FREQ, IL, IA, TXHT, RXHT, DIST(500), AMP(6,500), NPTS.

C
SUBROUTINE IDPLOT (20) LABEL
DIMENSION PXOR(6), PYOR(6)
DATA PXOR/3.2.0, 3.5.0/
DATA PYOR/7.0, 4.0, 1.0, 7.0, 4.0, 1.0/

CALL BGNLPL(1)
CALL INTAXS
DO 700 J=1,6
CALL PHYSOR (PXOR(J), PYOR(J))
DO 600 K=1, NPTS
Y(K) = AMP(J, K)
600 CONTINUE
YMIN = Y(1)
YMAX = Y(1)
DO 610 K=2, NPTS
IF(Y(K) .LT. YMIN) YMIN = Y(K)
IF(Y(K) .GT. YMAX) YMAX = Y(K)
610 CONTINUE
CALL AXSPLT(0.10, 2.30, XSTEP, XAXIS)
CALL AXSPLT(YMIN, YMAX, 2.0, YOR, YSTEP, YAXIS)
CALL MXTALF('STAND', '*')
CALL MX2ALF('L/CGRDECL', '+')
CALL TITLE('1', 'T=O(MW), 7, LABEL, 14, XAXIS, YAXIS)
CALL GRAPFXOR, XSTEP, YOR, YSTEP
CALL CURVE(DIST, Y, NPTS, 0)
IF(J .EQ. 1) CALL MESSAGE('ED - V', 6.0, 6.2, 0)
IF(J .EQ. 2) CALL MESSAGE('ED - HB', 7.0, 6.2, 0)
IF(J .EQ. 3) CALL MESSAGE('ED - HE', 7.0, 6.2, 0)
IF(J .EQ. 4) CALL MESSAGE('MD - V', 8.0, 6.2, 0)
IF(J .EQ. 5) CALL MESSAGE('MD - HB', 7.0, 6.2, 0)
IF(J .EQ. 6) CALL MESSAGE('MD - HE', 7.0, 6.2, 0)
IF(J .EQ. 6) THEN!
CALL MESSAGE(IDPLOT, 40, -3.0, 9.5)
CALL MESSAGE('FREQ= KHZ', 15, -3.0, 9.2)
CALL REALNO(SNGL(FREQ), 3, -2.5, 9.2)
MISSION RESEARCH CORPORATION
730 STATE STREET
SANTA BARBARA, CA 93101
R HENDRICK
F FAJEN
M SCHIEBE
J GILBERT
CL LONMI'RE

MITRE CORPORATION
PO BOX 205
BEDFORD, MA 01730
J HARDING

PACIFIC SIERRA RESEARCH CORP
1496 CLOVERFIELD BLVD
SANTA MONICA, CA 90404
EC FIELD, JR

PENNNSYLVANIA STATE UNIVERSITY
IONOSPHERIC RESEARCH LABORATORY
318 ELECTRICAL ENGINEERING EAST
UNIVERSITY PARK, PA 16802
IONOSPHERIC RSCH LAB (2)

R&D ASSOCIATES
PO BOX 9695
MARINA DEL REY, CA 90291
FORREST GILMORE
WILLIAM J KARZAS
PHYLLIS GREIFINGER
CARL GREIFINGER
A ORY
BRYAN GABBARD
AP TURCO
SAUL ALTschuler

R AND CORPORATION
1700 MAIN STREET
SANTA MONICA, CA 90406
TECHNICAL LIBRARY
CULLEN CRAIN (2)

SRI INTERNATIONAL
333 RAVENSWOOD AVENUE
MENLO PARK, CA 94025
DONALD NEILSON
GEORGE CARPENTER
WG CHERNUT
JR PETERSON
GARY PRICE

STANFORD UNIVERSITY
RADIO SCIENCE LABORATORY
STANFORD, CA 94305
RA HELLWELL
FRASER SMITH
J KATSUFKAI

TRW DEFENSE & SPACE RESEARCH GROUP
ONE SPACE PARK
REDONDO BEACH, CA 90278
DIANA ZF

CALIFORNIA INSTITUTE OF TECHNOLOGY
JET PROPULSION LABORATORY
4800 OAK GROVE DRIVE
PARADENÀ, CA 91103
ERNEST K SMITH
MAIL CODE J4-8131

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY
PO BOX 73
LEITCHING, MA 02173
DAVE WHITE
JH PANNELL L 246
DM TOWLE
END

DTIC

8-86