SU-SEL-69-046

D

Analytical and Numerical Treatment of Wave-Propagation in the Lower Ionosphere

by

R. R. Scarabucci

August 1969

Technical Report No. 3412-11

Prepared under
U.S. Air Force Office of Scientific Research
Grants AFOSR 783-68 and 783-69
National Aeronautics and Space Administration
Grant NGR 05-020-008
National Science Foundation, Office of Computer Sciences
Grant NSF GP-948

RADIOSCIENCE LABORATORY

STANFORD ELECTRONICS LABORATORIES

STANFORD UNIVERSITY · STANFORD, CALIFORNIA

SEL-69-046

ANALYTICAL AND NUMERICAL TREATMENT OF WAVE-PROPAGATION IN THE LOWER IONOSPHERE

by

R. R. Scarabucci

August 1969

Technical Report No. 3412-11

Prepared under
U.S. Air Force Office of Scientific Research
Grants AFOSR 783-68 and 783-69
National Aeronautics and Space Administration
Grant NGR 05-020-008
National Science Foundation, Office of Computer Sciences
Grant NSF GP-948

Radioscience Laboratory
Stanford Electronics Laboratories
Stanford University Stanford, California

CONTENTS

CONTENTS (cont.)

ILLUSTRATIONS

ACKNOWLEDGMENTS

This research was supported in part by the Air Force Office of Scientific Research of the U. S. Air Force, in part by the National Aeronautics and Space Administration, and in part by the National Science Foundation, Office of Computer Sciences. The work was carried out during the tenure of a scholarship awarded by C.N.A.E. - Brazil.

I. PRELIMINARIES

A. OBJECTIVES

The purpose of this report is to discuss the equations controlling the propagation of waves in the lower region of the ionosphere. The first part of the report deals with the analytical treatment of the wave equations governing reflection and transmission of waves through a planarly stratified ionosphere. The mathematical treatment includes the effect of positive and negative ions. The computer program for integrating the corresponding wave equations is given in a succeeding chapter. The developed computer program is limited to the case in which the waves are generated below the ionosphere and only the effect of electrons is considered. However, the computer program contains all the relevant features required by the numerical treatment of the wave equations and therefore the program can be easily changed in order to satisfy a specified problem.

B. INTRODUCTION

In the lower region of the ionosphere the electron concentration experiences substantial variation in distances comparable to the local wavelengths of waves whose frequencies are below $\sim$ 500 kHz. For these frequencies and more particularly for very-low-frequency waves that travel inside the D-region of the ionosphere the propagation is dominated by internal reflections, coupling between different modes of propagation, and by collisional absorption. An instantaneous picture of the amplitude of the electric or magnetic field vector of a propagating wave would show a spatial variation that is not sinusoidal, therefore ruling out field solutions of the form $e^{-j\beta z}$. Under the above circumstances a

"full-wave" method of solution must be conceived in which the wave-field solution is constructed point by point inside the ionosphere. When substantial variation occurs in the medium at a distance much greater than the local wavelength of a propagating wave a W.K.B. or "ray-method" may be used (see Budden [1966]). In this case there is no internal reflection and the variations of wave-fields E and H are such that, for a lossless medium, the power flow is conserved and there is only an impedance transformation relating E to H.

The set of differential equations governing the propagation of plane waves inside a planarly stratified anisotropic medium was derived by Clemmow and Heading [1954]. These equations are suitable for the study of wave propagation in the lower ionosphere but the resulting set of differential equations reveals a sort of instability when direct numerical integration is attempted by using standard integration procedures. Because of the instability problem the first numerical methods used indirect approaches for solving the Clemmow-Heading equations. For example, Budden [1955] used a related reflection coefficient matrix $\tilde{R}$ that was integrated along the vertical in the ionosphere. Barron and Budden [1959] developed the above technique by introducing an admittance matrix $\tilde{A}$ which simplified the amount of computational work required. However, both of the above methods were not capable of determining the wave-fields inside the ionosphere and, hence, the power transmitted high in the ionosphere. The first successful numerical treatment to overcome the above limitations was given by Pitteway [1965]. In this case the wave equations are integrated directly by introducing an orthogonalizing procedure which stabilizes the numerical technique of

integration. The method of integration that will be described in this report follows the technique of Pitteway.

The mathematical and the physical basis of a numerical method of solution for the equations governing the propagation of low frequency plane wave-fields inside a planarly stratified anisotropic lossy magneto-ionic medium will be derived in Chapter 2. The related computer program is fully discussed in Chapter 3. This program has been tested and used regularly in the IBM/360 computer of the Stanford University Computer Center since May 1968.

II. FULL-WAVE TREATMENT OF THE EQUATIONS CONTROLLING REFLECTION AND TRANSMISSION OF WAVES THROUGH THE LOWER IONOSPHERE

A. THE WAVE EQUATIONS

Suppose there is an electromagnetic plane wave propagating in free space which is incident upon a planarly stratified ionosphere that varies only in the z-direction as shown in Figure 1. The geometry is such that the planes of different stratification are parallel to the (x-y) plane.

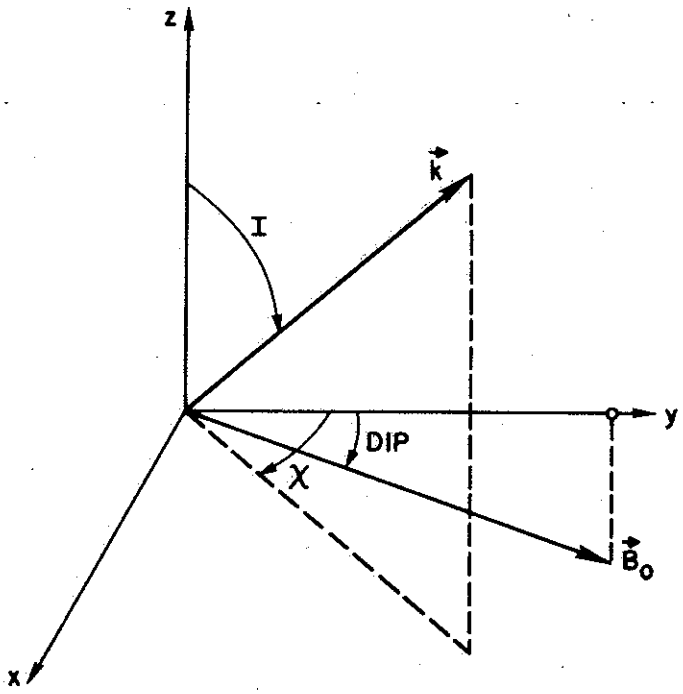

FIGURE 1. The assumed geometry. Planes of constant stratification are parallel to the (x-y) plane. DIP is the angle between the geomagnetic field $\vec{B}_o$ and the y-axis. $\vec{B}_o$ lies in the (y-z) plane. I is the angle between the vertical and the wave-normal vector $\vec{k}$. The azimuthal angle for $\vec{k}$ is χ.

The y axis is parallel to the ground and in the magnetic meridian (plane y-z) with its positive direction pointing northward. The earth's magnetic field is in the y-z plane and has direction cosines $(0,\gamma,\xi)$, i.e.,

$$\gamma = \cos \text{ (DIP)} \tag{2.1}$$

$$\xi = -\sin \text{ (DIP)} \tag{2.2}$$

where DIP is the dip angle of the magnetic field. The wave-normal of the incident wave makes an angle I with the z-axis (angle of incidence) and an angle χ with the magnetic meridian (azimuthal angle). The direction cosines of the incidence wave-normal are (ℓ,m,q_i),

$$\ell = \sin I \sin \chi \tag{2.3}$$

$$m = \sin I \cos \chi \tag{2.4}$$

$$q_i = \cos I \tag{2.5}$$

Next, we repeat the derivation of the four differential equations governing the propagation of plane waves inside the planarly stratified ionosphere first given by Clemmow and Heading [1954].

For sinusoidal wave field excitation with angular frequency ω and for a non-magnetic medium the equations of Maxwell are

$$\nabla \times \vec{E} = -j\omega\mu_o \vec{H} \tag{2.6}$$

$$\nabla \times \vec{H} = j\omega\varepsilon_o (\tilde{I} + \tilde{M}) \cdot \vec{E} \tag{2.7}$$

and two more equations relative to the divergence of $\vec{E}$ and $\vec{H}$ which are not necessary here. The symbols ϵ_o and μ_o stand for the permittivity and the permeability of free space, respectively. In Eq. (2.7) $\tilde{I}$ is the unit matrix and $\tilde{M}$ is the susceptibility matrix, given by the constitutive relation of the medium

$$\vec{P} = \epsilon_o \tilde{M} \cdot \vec{E} \tag{2.8}$$

where $\vec{P}$ is the volumetric polarization of the medium. The susceptibility $\tilde{M}$ will be deduced in section B of this chapter; now it is only necessary to state that $\tilde{M}$ is given by

$$\tilde{M} = \begin{pmatrix} M_{xx} & M_{xy} & M_{xz} \\ M_{yx} & M_{yy} & M_{yz} \\ M_{zx} & M_{zy} & M_{zz} \end{pmatrix} \tag{2.9}$$

The space-time variation of any wave-field of the incident wave is given by

$$\exp\left\{j\omega t - jk(\ell x + my + q_i z)\right\} \tag{2.10}$$

where k is the propagation constant of free space,

$$k = \omega(\epsilon_o \mu_o)^{1/2} = \omega/c \tag{2.11}$$

and c is the velocity of light. The continuity of tangential fields E and H along the successive boundaries in the z-direction is stated by Snell's law:

$$\frac{\partial}{\partial x} = - jk\ell = \text{const.} \tag{2.12}$$

$$\frac{\partial}{\partial y} = - jkm = \text{const.} \tag{2.13}$$

$\partial/\partial z$ is determined by the variational characteristics of the medium along the z direction. Then, from Eq. (2.6) we have

$$- jkm\, E_z - \frac{dE_y}{dz} = - j\omega\mu_o H_x \tag{2.14}$$

$$\frac{dE_x}{dz} + jk\ell\, E_z = - j\omega\mu_o H_y \tag{2.15}$$

$$- jk\ell\, E_y + jkm\, E_x = - j\omega\mu_o H_z \tag{2.16}$$

Equations (2.7) and (2.9) give

$$- jkm\, H_z - \frac{dH_y}{dz} = j\omega\varepsilon_o \left[(1 + M_{xx})E_x + M_{xy}E_y + M_{xz}E_z\right] \tag{2.17}$$

$$\frac{dH_x}{dz} + jk\ell H_z = j\omega\varepsilon_o \left[M_{yx}E_x + (1 + M_{yy})E_y + M_{yz}E_z\right] \tag{2.18}$$

$$- jk\ell H_y + jkmH_x = j\omega\varepsilon_o \left[M_{zx}E_x + M_{zy}E_y + (1 + M_{zz})E_z\right] \tag{2.19}$$

We point out that from Eqs. (2.14) to (2.19) a factor $e^{-jk(\ell x + my)+j\omega t}$ has been omitted for all fields.

We notice that the derivatives of E_z and H_z are not present in Eqs. (2.14) to (2.19). Therefore these fields can be eliminated by the proper combination of the equations. This is easily done and we get

$$\frac{d}{dz}\begin{bmatrix} E_x \\ -E_y \\ Z_o H_x \\ Z_o H_y \end{bmatrix} = -jk\tilde{T}\cdot\begin{bmatrix} E_x \\ -E_y \\ Z_o H_x \\ Z_o H_y \end{bmatrix} \tag{2.20}$$

where Z_o is the characteristic impedance of free-space,

$$Z_o = (\mu_o/\epsilon_o)^{\frac{1}{2}} \tag{2.21}$$

and $\tilde{T}$ is given by

$$\tilde{T} = \begin{bmatrix} \frac{-\ell M_{zx}}{1+M_{zz}} & \frac{\ell M_{zy}}{1+M_{zz}} & \frac{\ell m}{1+M_{zz}} & 1-\frac{\ell^2}{1+M_{zz}} \\ \frac{m M_{zx}}{1+M_{zz}} & \frac{-mM_{zy}}{1+M_{zz}} & 1-\frac{m^2}{1+M_{zz}} & \frac{m\ell}{1+M_{zz}} \\ -M_{yx}-m\ell+\frac{M_{yz}M_{zx}}{1+M_{zz}} & 1+M_{yy}-\ell^2-\frac{M_{yz}M_{zy}}{1+M_{zz}} & \frac{-mM_{yz}}{1+M_{zz}} & \frac{\ell M_{yz}}{1+M_{zz}} \\ 1+M_{xx}-m^2-\frac{M_{xz}M_{zx}}{1+M_{zz}} & \frac{M_{xz}M_{zy}}{1+M_{zz}}-M_{xy}-m\ell & \frac{m M_{xz}}{1+M_{zz}} & \frac{-\ell M_{xz}}{1+M_{zz}} \end{bmatrix} \tag{2.22}$$

Defining the column vector

$$\vec{e} = \begin{bmatrix} E_x \\ -E_y \\ Z_o H_x \\ Z_o H_y \end{bmatrix} \tag{2.23}$$

the set of Eq. (2.18) may be expressed in a more compact form, namely

$$\frac{d\vec{e}}{dz} = - jk \tilde{T} \cdot \vec{e} \qquad (2.24)$$

Equation (2.24) is the set of linear differential wave equations governing the propagation of waves in a planarly stratified and general magnetoionic medium. The elements of $\tilde{T}$ are functions of z because the terms of the constitutive relation M_{ij} vary from point to point inside the inhomogeneous medium. Equation (2.24) is already in a form suitable for numerical integration.

B. THE CONSTITUTIVE RELATION

Consider a magnetoionic medium composed of a mixture of negative and positive ions embedded in a magnetic field $\vec{B}_o$ whose direction cosines are $(0, \gamma, \xi)$ as shown in Figure 1. In order to simplify the following mathematical treatment a lossless cold plasma is considered but the effect of collisional loss will be readily taken into account at the end. The equation of motion for a single particle of species k, charge magnitude $Z_k \cdot e$ and density $N_k(m^{-3})$ is

$$m_k \frac{d\vec{v}_k}{dt} = \epsilon_k Z_k e \, (\vec{E} + \vec{v}_k \times \vec{B}_o) \qquad (2.25)$$

where e is the absolute value of the electron charge and ϵ_k is $+1$ or -1 depending upon whether the k-species has a positive or a negative charge.

Taking $\exp(+j\omega t)$ as the time variation for $\vec{E}$ we get

$$j\vec{v}_k = \frac{\epsilon_k Z_k e}{m_k \omega} \vec{E} + \frac{\epsilon_k}{\omega} \vec{v}_k \times \left(\frac{Z_k e \, \vec{B}_o}{m_k} \right) \qquad (2.26)$$

We now define

$$\Omega_k = \frac{Z_k e\, B_o}{m_k} \tag{2.27}$$

$$Y_k = \Omega_k/\omega \tag{2.28}$$

and

$$X_k = \frac{(Z_k e)^2 N_k}{\epsilon_o m_k} \cdot \frac{1}{\omega^2} \tag{2.29}$$

The gyrofrequency and the plasma frequency for the k^{th} species are respectively

$$\Omega_k/2\pi$$

and

$$\frac{(Z_k e)^2 N_k}{2\pi\epsilon_o m_k}$$

The current density due to this k^{th} species is

$$\vec{J}_k = (Z_k e)\epsilon_k N_k \vec{v}_k \tag{2.30}$$

and then Eq. (2.26) gives

$$j\vec{J}_k = \omega\epsilon_o X_k \vec{E} + \epsilon_k Y_k \vec{J}_k \times \frac{\vec{B}_o}{B_o} \tag{2.31}$$

Working with Eq. (2.31) we get

$$jJ_{kx} = \omega\epsilon_o \frac{X_k}{1-Y_k^2}\left[E_x - j\epsilon_k Y_k E_y + j\epsilon_k Y_k E_z\right] \tag{2.32}$$

$$jJ_{ky} = \omega\epsilon_o \frac{X_k}{1-Y_k^2} \left[j\xi\epsilon_k Y_k E_x + (1-\gamma^2 Y_k^2)\, E_y - \xi\gamma Y_k^2 E_z\right] \tag{2.33}$$

$$jJ_{kz} = \omega\epsilon_o \frac{X_k}{1-Y_k^2} \left[-j\gamma\epsilon_k Y_k E_x - \gamma\xi Y_k^2 E_y + (1-\xi^2 Y_k^2) E_z\right] \tag{2.34}$$

The total current density involves the summation over all the species, that is

$$J = \sum_k \vec{J}_k \tag{2.35}$$

We now follow the notation of Stix [1962] defining

$$R = 1 - \sum_k \frac{X_k}{1+\epsilon_k Y_k} \tag{2.36}$$

$$L = 1 - \sum_k \frac{X_k}{1-\epsilon_k Y_k} \tag{2.37}$$

$$S = \tfrac{1}{2}(R + L) \tag{2.38}$$

$$D = \tfrac{1}{2}(R - L) \tag{2.39}$$

and

$$P = 1 - \sum_k X_k \tag{2.40}$$

Substituting Eqs. (2.35) to (2.40) into Eqs. (2.32) to (2.34) we get

$$\begin{pmatrix} J_x \\ J_y \\ J_z \end{pmatrix} = j\omega\epsilon_o \begin{pmatrix} S-1 & j\xi D & -j\gamma D \\ -j\xi D & \xi^2 S + \gamma^2 P - 1 & \gamma\xi(P-S) \\ j\gamma D & \gamma\xi(P-S) & \gamma^2 S + \xi^2 P - 1 \end{pmatrix} \begin{pmatrix} E_x \\ E_y \\ E_z \end{pmatrix} \tag{2.41}$$

Equation (2.41) corresponds to the constitutive relation of the medium and is related to $\tilde{M}$ by

$$\vec{J} = \frac{d\vec{P}}{dt} = j\omega\varepsilon_o \tilde{M} \cdot \vec{E} \tag{2.42}$$

Therefore,

$$\tilde{M} = \begin{pmatrix} S-1 & j\xi D & -j\gamma D \\ -j\xi D & \xi^2 S + \gamma^2 P - 1 & \gamma\xi(P-S) \\ j\gamma D & \gamma\xi(P-S) & \gamma^2 S + \xi^2 P - 1 \end{pmatrix} \tag{2.43}$$

The effect of collisional losses. Given an effective collision frequency ν_k for each species k the effect of collisional loss is readily taken into account by replacing m_k by $m_k(1-j\frac{\nu_k}{\omega})$ in the definitions of Ω_k and X_k (Eqs. (2.27) and (2.29)). The above replacement corresponds to the effect of a viscous force term that should be present in the equation of motion, Eq. (2.25).

C. MATRIX $\tilde{T}$

With the knowledge of matrix $\tilde{M}$ (Eq. (2.43)) the elements of matrix $\tilde{T}$ can be determined explicitly. Equation (2.43) shows that

$$\begin{aligned} M_{xy} &= -M_{yx} \\ M_{xz} &= -M_{zx} \\ M_{yz} &= M_{zy} \end{aligned} \tag{2.44}$$

and from these relationships we readily obtain for $\tilde{T}$ (Eq. (2.22))

$$\begin{aligned} T_{11} &= -T_{44} \\ T_{12} &= T_{34} \\ T_{13} &= T_{24} \\ T_{21} &= -T_{43} \\ T_{22} &= T_{33} \end{aligned} \tag{2.45}$$

The determination of the elements T_{ij} follows directly from the substitution of the elements of the susceptibility matrix $\tilde{M}$ in Eq. (2.22) giving:

$$
\begin{aligned}
T_{11} &= -j\gamma\ell D/a & T_{21} &= j\gamma m D/a \\
T_{12} &= \gamma\xi\ell(P-S)/a & T_{22} &= -\gamma\xi m(P-S)/a \\
T_{13} &= \ell m/a & T_{23} &= 1 - m^2/a \\
T_{14} &= 1 - \ell^2/a & T_{24} &= T_{13} \\
T_{31} &= j\xi D - m\ell + j\gamma^2\xi D(P-S)/a & T_{41} &= S - m^2 - \gamma^2 D^2/a \\
T_{32} &= \xi^2 D + \gamma^2 P - \ell^2 - \gamma^2\xi^2(P-S)^2/a & T_{42} &= -j\xi D - m\ell - j\gamma^2\xi D(P-S)/a \\
T_{33} &= T_{22} & T_{43} &= -T_{21} \\
T_{34} &= T_{12} & T_{44} &= -T_{11}
\end{aligned}
\qquad (2.46)
$$

where

$$a = \gamma^2 S + \xi^2 S \qquad (2.47)$$

The matrix $\tilde{T}$ when only the effect of electrons is considered.

When only the effect of electrons is taken into account the k-indices of Eqs. (2.27-29) and (2.36-40) are dropped and a new variable is defined,

$$U = 1 - j\nu/\omega \qquad (2.48)$$

where ν is the effective collision frequency for electrons. With the above notation and after some manipulation with Eq. (2.46) we obtain:

$$
\begin{aligned}
T_{11} &= j\ell XU\gamma Y/b & T_{21} &= -\,jmXU\gamma Y/b \\
T_{12} &= \ell X\gamma\xi Y^2/b & T_{22} &= -\,mX\gamma\xi Y^2/b \\
T_{13} &= \ell mU(U^2-Y^2)/b & T_{23} &= 1 - m^2U(U^2-Y^2)/b \\
T_{14} &= 1 - \ell^2U(U^2-Y^2)/b & T_{24} &= T_{13} \\
T_{31} &= -\,m\ell - jX\xi Y(U-X)/b & T_{41} &= 1 - m^2 - XU(U-X)/b \\
T_{32} &= 1 - \ell^2 - [XU(U-X) - X\gamma^2Y^2]/b & T_{42} &= -\,\ell m + jX\xi Y(U-X)/b \\
T_{33} &= T_{22} & T_{43} &= -\,T_{21} \\
T_{34} &= T_{12} & T_{44} &= -\,T_{11}
\end{aligned}
\tag{2.49}
$$

where
$$b = U(U^2-Y^2) - X(U^2-\xi^2Y^2) \tag{2.50}$$

In a loss-free medium the susceptibility matrix $\tilde{M}$ is Hermitian, that is

$$M_{ij} = M_{ji}^{*} \tag{2.51}$$

and by inspection of $\tilde{T}$ we observe that in this case

$$T_{5-j,5-i} = T_{ij}^{*} \tag{2.52}$$

which Budden [1966 - Chapter 18] describes as a Hermitian matrix with respect to the trailing diagonal. Pitteway and Jespersen [1966] have used the above property in order to find the full-wave solution when the incident wave comes from above the ionosphere. This point will be discussed further in Chapter 3.

D. NUMERICAL METHOD FOR SOLVING THE WAVE EQUATION

The set of Eq. (2.24) of linear differential wave equations is already in a form suitable for numerical integration. The method of solution for Eq. (2.24) that will be outlined in this section follows the method introduced by Budden [1955] and more closely the method of Pitteway [1965]. They are direct methods in the sense that the achievement of the solution is based strictly on the physical properties of the wave equations. Methods that we consider indirect approaches to the problem and which introduce new assumptions were developed by Johler and Harper [1962] and more recently by Altman and Cory [1969].

In order to solve the set of Eq. (2.24) the direction and the polarization of an upgoing wave in the air space below the ionosphere are given along with the z-dependent function $N_k(z)$ and $\nu_k(z)$, respectively density and collision frequency of each particle species k. The problem is then to determine all the properties of the wave reflected toward the ground and the properties of the wave transmitted through the ionosphere.

The boundary condition that must be used in order to solve Eq. (2.24) is that the energy of the wave comes from below. It means that there is a height z_1 inside or above the ionosphere where only upgoing waves are allowed to exist. At height z_1 the ionosphere is a slowly-varying medium satisfying the validity criteria required by the W.K.B. method of solution (see Budden [1966]-Chapters 9 and 18), namely that no more partial reflections or couplings occur at z_1. More specifically the medium may be supposed homogeneous in a space of several wavelengths in the neighborhood of z_1. Therefore the matrix $\tilde{T}$ is constant in the vicinity of z_1 and then a particular solution of Eq. (2.24) is given by

$$\vec{e} \sim e^{-jkqz} \tag{2.53}$$

Hence, from Eq. (2.24) we get

$$(\tilde{T} - q\tilde{I}) \cdot \vec{e} = 0 \tag{2.54}$$

The condition for $\vec{e}$ having a non-trivial solution is that

$$\det(\tilde{T} - q\tilde{I}) = 0 \tag{2.55}$$

Equation (2.55) is a characteristic equation and as $\tilde{T}$ is a 4X4 matrix there are 4 eigenvalues q determined by the solution of Eq. (2.55). This is another form of presenting the so-called Booker quartic equation [Booker, 1936, 1939]. Observe that the matrix $\tilde{T}$, Eq. (2.22), depends on the direction of the incident upgoing wave by means of the terms ℓ and m because the differential equations, Eq. (2.24), satisfy Snell's law. Therefore the 4 eigenvalues that come from the solution of Eq. (2.55) at z_1 will produce 4 eigenvectors or characteristic waves $\vec{e}_i$ whose horizontal variation is equal to the one presented by the incident wave, namely

$$e^{-jk(\ell x+my)} \tag{2.56}$$

The coefficients of the quartic equation produced by Eq. (2.55) which will determine the eigenvalues of $\tilde{T}$ are derived in Appendix B. A general solution at z_1 would be given by a linear combination of the 4 eigenvectors, i.e.,

$$\vec{e}(z_1) = a_1\vec{e}_1 + a_2\vec{e}_2 + a_3\vec{e}_3 + a_4\vec{e}_4 \tag{2.57}$$

but because the wave energy comes from below only the eigenvectors corresponding to upgoing waves must be considered. The characteristic upgoing waves are determined by the eigenvalues whose imaginary part is negative. It is pointed out that the definition of "upgoing wave" does not involve the sign of the real part of q .

Suppose then two eigenvalues are selected at z_1 corresponding to upgoing characteristic waves and, from each of them, the related eigenvectors $\vec{e}_1(z_1)$ and $\vec{e}_2(z_1)$. The solution of the proposed problem is then achieved by using the following procedure:

1. Starting with eigenvector $\vec{e}_1(z_1)$ at height z_1 Eq. (2.24) is numerically integrated downward. The integration is stopped at $z = z_n$ below the ionosphere.
2. The same procedure is repeated starting with the other upgoing eigenvector $\vec{e}_2(z_1)$. Observe the meaning of the vector $\vec{e}_1$ (or $\vec{e}_2$) at any height below z_1 inside the inhomogeneous ionosphere: in general $\vec{e}_1(z)$ corresponds to the combination of 4 waves which will produce at z_1 the purely upgoing eigenvector $\vec{e}_1(z_1)$. In particular the vector $\vec{e}_1(z_n)$ below the ionosphere corresponds to the sum of incident and reflected waves, with the polarization of the incident wave being such that only the upgoing characteristic wave $\vec{e}_1(z_1)$ will result at z_1 .
3. A spatial Fourier analysis is made for each solution below the ionosphere yielding to incident and reflected wave-fields corresponding to each solution, i.e., $\vec{e}_1(z_n)$ gives $\vec{U}_1 + \vec{D}_1$, $\vec{e}_2(z_n)$ gives $\vec{U}_2 + \vec{D}_2$ and the Fourier analysis determines the upgoing $\vec{U}_1$, $\vec{U}_2$ and the reflected downgoing $\vec{D}_1$, $\vec{D}_2$ electric wave-fields below the ionosphere.

4. The polarization and the amplitude of the incident wave is now given by (say) supplying its electric field $\vec{U}_o$. Hence the solution is established by the linear combination of $\vec{U}_1$ and $\vec{U}_2$ such that the combination reproduces $\vec{U}_o$, i.e.,

$$U_{ox} = \alpha\, U_{1x} + \beta\, U_{2x} \tag{2.58}$$

$$U_{oy} = \alpha\, U_{1y} + \beta\, U_{2y} \tag{2.59}$$

Eq. (2.58) and Eq. (2.59) determine the complex multiplicative constants α and β. Consequently the total wave-fields originated from the incident-source wave are determined from

$$\vec{e}(z) = \alpha\vec{e}_1(z) + \beta\vec{e}_2(z) \tag{2.60}$$

at any arbitrary height z. The problem is then formally solved.

In Chapter 3 will be discussed how to perform steps 1 to 4 subject to a further complication related to the fact that one of the solutions $\vec{e}_i$ increases much more than the other during the downward integration. The computer program to be described in Chapter 3 is developed for the case where only electrons are taken into account although the technique to be applied when the effect of several ions is also considered is exactly the same. The only changes required in the computer program in this more general case are the determination of $\tilde{T}$ using the set of Eq. (2.46) instead of the set of Eq. (2.49) and the calculation of the eigenvalues at the starting ionospheric height z_1 from a different Booker quartic equation as shown in Appendix A.

III. THE COMPUTER PROGRAM

The purpose of this chapter is to discuss and present the computer program that has been developed for integrating Eq. (2.24) in accordance with the theory introduced in Chapter 2. The computational technique determines the reflected and the transmitted ionospheric wavefields generated by an upgoing incident wave that hits the lower region of the ionosphere.

The problem of solving Eq. (2.24) consists of integrating a set of linear differential equations subject to prescribed boundary conditions. The set is expressed in vector notation by

$$\frac{d\vec{v}}{dz} = \tilde{R}(z) \cdot \vec{v} \tag{3.1}$$

where $\vec{v}$ is the column vector of the dependent variables and $\tilde{R}(z)$ is a square matrix which is a function of the independent variable z. The problem is to integrate Eq. (3.1) through an inhomogeneous region, where $\tilde{R}(z)$ is variable, between two points z_1 and z_n whose neighborhoods are characterized by homogeneous media, i.e. constant $\tilde{R}(z)$. Although some well-known numerical integration procedures might be used for integrating Eq. (3.1) a further complication can arise as is explained below. The solution to Eq. (3.1) is started with one eigenvector $\vec{v}_{ei}$ of $\tilde{R}$ at the point $z = z_1$ and the set of linear differential equations, Eq. (3.1), is numerically integrated from z_1 to z_n, yielding to a solution vector $\vec{v}_i$ at z_n. The above process is repeated for the m eigenvectors of $\tilde{R}$. Therefore, a specific solution $\vec{V}_n$ of Eq. (3.1) at $z = z_n$ is obtained as a combination of the m independent solutions:

$$\vec{V}_n = a_1\vec{v}_1 + a_2\vec{v}_2 + \ldots + a_m\vec{v}_m \qquad (3.2)$$

Suppose now that during the integration the vector $\vec{v}_i$ corresponding to the starting eigenvector $\vec{v}_{ei}$ at z_1 increases much more than the solutions corresponding to the rest of the starting eigenvectors. In addition consider the fact that when an arbitrary solution $\vec{v}_j$ is attempted round-off errors during the integration process continually introduce in $\vec{v}_j$ some small amount of the remaining solution-vectors at all steps $z = z_k$. Round-off errors occur during numerical integration because the number of decimal places is limited in a computer machine. In a stable integration technique the round-off errors are made small. However, because the solution-vector $\vec{v}_i$ increases much more than the others during the numerical integration, the round-off error corresponding to a very small fraction of $\vec{v}_i$ added to $\vec{v}_j$ at an arbitrary integration step will grow during the following steps. After a number of integration steps the attempted solution $\vec{v}_j$ at z is completely masked by the behavior of $\vec{v}_i$. Therefore, it becomes impossible to obtain m independent solutions at $z = z_n$ and Eq. (3.2) cannot be achieved. An algorithmic calculus for handling this general type of problem has been developed by Pitteway (personal communication). A method of solution that overcomes the above "interference" between independent solution-vectors for waves propagating in the lower ionosphere has been introduced by Pitteway [1965] and will be described in Section A of this chapter.

The computer program described in Section B is more restricted in applicability than the one developed by Pitteway because it can only treat the case of upgoing waves as input. However, several improvements

have been made, namely

1. The integration routine uses a more stable integration technique developed by Hamming [1959]. This stable modified predictor-corrector method is specially suited for handling wave equations where the solutions present a sinusoidal-type behavior.
2. Double precision accuracy is used throughout.
3. Relatively small computing time.
4. Information about the relative error committed in each step of integration.

When the incident wave comes from above the ionosphere the boundary conditions must be modified as discussed by Pitteway and Jespersen [1966]. The process by which they separate the internally reflected wave from the downgoing wave uses the fact that $\tilde{T}$ is hermitian about its trailing diagonal when the collision frequency is zero. The collision frequency is made zero where the W.K.B. conditions are valid high in the ionosphere. Under this condition the eigenvectors are related to each other in a way that permits the splitting of the waves in upgoing and downgoing parts.

The unique feature introduced by this treatment of waves incident from above is a reflection coefficient for the internally reflected upgoing wave. The reciprocity theorem proved by Pitteway and Jespersen [1966] shows that the transmission coefficient for waves coming from above with azimuth angle χ_1 is equal to the transmission coefficient of the penetrating mode incident from below with azimuth $\chi_2 = 180^\circ - \chi_1$. The downgoing whistler wave emerges from the ionosphere at an angle I from the vertical which is the same for the corresponding reciprocal penetrating mode. Therefore the reflection coefficient for waves incident from above is the only parameter not determined by the computer program described in the following pages.

The original computer program of Pitteway has been translated to

FORTRAN language by G. H. Smith [Smith and Pitteway, 1969]. Although the mathematical treatment given in Chapter 2 includes the effect of heavy ions, the computer program to be described in this report only includes the effect of electrons. However the amount of work necessary to conceive a more general computer program is relatively small if it is started with the actual program. More specifically it is only necessary to calculate $\tilde{T}$ using Eq. (2.46) and to determine the eigenvalues of $\tilde{T}$ with the more general coefficient given by Eq. (A.11) of Appendix A.

On the other hand Eq. (3.1) is likely to occur in many other branches of physics. For example, problems involving the Schrodinger wave equation in quantum mechanics, problems involving the interaction of waves and atomic structures, etc. Hence, although the computer program is particularized for integrating the Clemmow-Heading equations, the program may also be valuable for people working in other scientific areas.

A. THE ORTHOGONALIZING PROCEDURE OF PITTEWAY

Equation (2.24) represents a set of four linear differential equations. Hence, for obtaining one given field below the ionosphere, four independent solutions would be required. But since the energy comes from below, only two starting eigenvectors corresponding to upgoing waves are necessary at very high altitudes. This means that the field below the ionosphere will be obtained as a combination of two independent solutions.

The integration is started at $z = z_1$ with the upgoing eigenvectors $\vec{e}_1(z_1)$ and $\vec{e}_2(z_1)$ and proceeds downward step by step. At any height z_i the vector $\vec{e}_1(z_i)$ (say) represents the total field

which is the source of $\vec{e}_1(z_1)$. In other words, $\vec{e}_1(z_i)$ is a particular combination of two upgoing and two downgoing waves such that this combination at $z = z_i$ will give rise only to the upgoing eigenvector $\vec{e}_1(z_1)$ at $z = z_1$. If the medium were homogeneous the starting eigenvector would not change as the integration proceeded (only an amplitude factor would be involved if attenuation were present) because in this case no reflection would occur.

When integrating Eq. (2.24) in the lower ionosphere, one of the starting eigenvectors will correspond to one solution which increases very steeply as the integration proceeds downwards. It is called the dominant mode $\vec{e}_1(z)$ which corresponds to the "extraordinary" upgoing wave high in the ionosphere. The other is the non-dominant mode $\vec{e}_2(z)$ related to the propagation of an upgoing "whistler-mode" wave at the top. Suppose the integration of the non-dominant mode $\vec{e}_2(z)$ is started in a computer machine which works to about 16 decimal places. Making the impossible assumption that no error is committed in the integration procedure itself, round-off errors still exist because only 16 decimal places have been used in the computation. Suppose that an error of 10^{-16} has been committed in this step. This is a very small error and in fact it would be very satisfactory if this amount of error would continue during the rest of the integration. Unfortunately the error in $\vec{e}_2(z)$ corresponds in part to introducing in $\vec{e}_2(z)$ some small amount of the dominant mode $\vec{e}_1(z)$. The sum of two independent solutions is itself a solution, so the integration proceeds downward not only with $\vec{e}_2(z)$ but with a sum of solutions. Since the dominant solution increases much more than the other as the integration continues, the polarization of the obtained ionospheric wave-fields changes

gradually from the polarization of $\vec{e}_2(z)$ to a polarization much closer to the dominant solution. Now, in addition to round-off errors there are truncation errors related to the fact that the integration is performed using finite step sizes. Moreover, small errors committed when one particular element of $\vec{e}_2(z)$ has a small value (the solution is of sinusoidal form) may represent an appreciable relative error. Hence, allowing the integration to proceed some wavelengths down does not furnish a second independent solution because the dominant mode solution "swamps" the whistler mode solution. Pitteway [1965] described this phenomenon stating that the traveling wave mode is unstable to such a numerical integration, which converges to the dominant evanescent wave solution. In order to overcome this difficulty Pitteway devised the process described below.

By the Schmidt orthogonalization process a set of mutually orthogonal vectors may be constructed from any set of linearly independent vectors $\vec{e}_1$, $\vec{e}_2$. The construction is as follows [Friedman, 1964]

$$\begin{aligned} \vec{e}_1 &\longrightarrow \vec{e}_1 \\ \vec{e}_2 &\longrightarrow \vec{e}_{20} = \vec{e}_2 + a\vec{e}_1 \end{aligned} \tag{3.3}$$

where

$$a = -\frac{\vec{e}_1^{\,*} \cdot \vec{e}_2}{\vec{e}_1^{\,*} \cdot \vec{e}_1} \tag{3.4}$$

Hence, $\vec{e}_{20}$ is $\vec{e}_2$ minus its projection on $\vec{e}_1$. The symbol $\vec{e}_{20}$ will be used for the vector derived from $\vec{e}_2$ by the above orthogonalizing procedure.

Suppose the above orthogonalizing process is used in the integration procedure at height h. If at $z = h$ we replace $\vec{e}_2(h)$ by

$$\vec{e}_{20}(h) = \vec{e}_2(h) + a_h \vec{e}_1(h) \tag{3.5}$$

we obtain a new solution which is accepted by the integration routine because it is a sum of solutions. Furthermore $\vec{e}_{20}(h)$ has polarization completely different from $\vec{e}_1(h)$. This comes from the fact that

$$\vec{e}_1^*(h) \cdot \vec{e}_{20}(h) = 0 \tag{3.6}$$

if a_h is given by one equation similar to Eq. (3.4).

The integration is allowed to proceed a certain number of steps and then a new orthogonalization is made. This process is carried out throughout the whole interval of integration and, in this fashion, a second solution is obtained which does not attain the polarization of the dominant mode $\vec{e}_1$ and thus does not behave the same as $\vec{e}_1$. Observe that each time the second solution is orthogonalized the part of the error in $\vec{e}_2$ which is parallel to $\vec{e}_1$ is eliminated. Obviously not only this error but all of the part of $\vec{e}_2$ that is parallel to $\vec{e}_1$ is eliminated. The error caused by $\vec{e}_1$ is not allowed to increase during the integration because this error is cut down by the above orthogonalizing process. On the other hand, the numbers produced by the computer do not represent a second pure solution in z because each adjustment changes its polarization abruptly. Hence, in the free space below the ionosphere the second solution is a possible second independent solution and may be combined with $\vec{e}_1$ for finding everything below the ionosphere (reflection coefficients, polarization of reflected wave, height of reflection, etc). But since the correspondence between this second independent solution and the starting eigenvector $\vec{e}_1(z_1)$ at the top is unknown, it is not possible to find the transmitted wave at the top unless a reconstruction of the wave-fields is made starting now with the two independent solutions

below the ionosphere. One possible way of reconstructing the wave-fields is discussed below.

Suppose that from each independent solution below the ionosphere the upgoing and downgoing electric field components are obtained (see Section G)

$$\vec{e}_1(z_n) \longrightarrow \vec{U}_1(z_n), \quad \vec{D}_1(z_n) \tag{3.7}$$

$$\vec{e}_{20}(z_n) \longrightarrow \vec{U}_2(z_n), \quad \vec{D}_2(z_n) \tag{3.8}$$

It is known that the upgoing wave $\vec{U}_2(z_n)$ will be the one which will give rise to the traveling wave at the top. Probably some part of $\vec{U}_2(z)$ will die out inside the ionosphere because of a mismatching of polarization. Hence, instead of obtaining the incident field as a combination of $\vec{U}_1(z_n)$ and $\vec{U}_2(z_n)$ a more suitable technique is to obtain the incident wave as a combination of $\vec{U}_1(z_n)$ - the wave whose energy will be completely reflected or absorbed inside the ionosphere - and a "penetrating" wave $\vec{U}_p(z)$ - the incident wave that maximizes the power at the top. To obtain the penetrating mode the following relationship is set

$$\vec{U}_p(z_n) = \vec{U}_2(z_n) + b\vec{U}_1(z_n) \tag{3.9}$$

such that $\vec{U}_p(z_n)$ and $\vec{U}_1(z_n)$ are mutually orthogonal. In other words,

$$b = - \frac{\vec{U}_1^{\,*}(z_n) \cdot \vec{U}_2(z_n)}{\vec{U}_1^{\,*}(z_n) \cdot \vec{U}_1(z_n)} \tag{3.10}$$

Observe that Eq. (3.10) represents an orthogonalizing condition between two tri-dimensional electric field vectors.

For proving that $\vec{U}_p(z_n)$ is the field which minimizes the input power it is supposed that another vector $\vec{U}_3(z_n)$ would be better, say

$$\vec{U}_3(z_n) = \vec{U}_p(z_n) + b_1\vec{U}_1(z_n) \tag{3.11}$$

Hence the power flux density would be proportional to

$$\vec{U}_3^{\,*}(z_n) \cdot \vec{U}_3(z_n) = |\vec{U}_3(z_n)|^2 = |\vec{U}_p(z_n)|^2 + |b_1|^2 \cdot |\vec{U}_1(z_n)|^2 \tag{3.12}$$

Equation (3.12) is obtained using the fact that $\vec{U}_p(z_n)$ is orthogonal to $\vec{U}_1(z_n)$. Equation (3.12) shows that the minimum power-flux density is achieved for $b_1 = 0$, i.e., $\vec{U}_p(z_n)$ defined by Eq. (3.9) and Eq. (3.10) is the penetrating solution.

Now, linearity requires that if $\vec{U}_p(z_n)$ is chosen as a possible independent second solution for the incident electric field, then

$$\vec{D}_p(z_n) = \vec{D}_1(z_n) + b\vec{D}_2(z_n) \tag{3.13}$$

must also be chosen for the downgoing reflected wave of the penetrating mode. Similarly the total field vector $\vec{e}_{20}(z_n)$ must be replaced by

$$\vec{e}_p(z_n) = \vec{e}_{20}(z_n) + b\vec{e}_1(z_n) \tag{3.14}$$

Obtaining the penetrating wave-fields inside the ionosphere. The integration procedure and the orthogonalizing process will be represented by equations in which the following symbols are used:

1. $\vec{e}_i(j)$ is the vector $\vec{e}_i$ at the height z_j corresponding to the integration step number j, $j = 1, 2, \ldots n$.
2. $\vec{e}_2(j)$ is the non-orthogonalized vector $\vec{e}_2$ obtained from the steps of integration starting with the orthogonalized field $\vec{e}_{20}(j-1)$.
3. a_j is the orthogonalizing factor defined by Eq. (3.4) at the step number j.

The following equations show schematically the integration technique used by the computer program where R means "replaced by" and I means "after a certain number of integration steps yields to".

$$\vec{e}_1(1) \qquad \vec{e}_2(1) \xrightarrow{R} \vec{e}_{20}(1) = \vec{e}_2(1) + a_1\vec{e}_1(1) \tag{3.15}$$

$$\downarrow I \qquad\qquad \swarrow I$$

$$\vec{e}_1(2) \qquad \vec{e}_2(2) \xrightarrow{R} \vec{e}_{20}(2) = \vec{e}_2(2) + a_2\vec{e}_1(2) \tag{3.16}$$

$$\downarrow I \qquad\qquad \swarrow I$$

$$\vec{e}_1(3) \qquad \vec{e}_2(3) \xrightarrow{R} \vec{e}_{20}(3) = \vec{e}_2(3) + a_3\vec{e}_1(3) \tag{3.17}$$

$$\downarrow I \qquad\qquad \swarrow I$$

$$\vdots \qquad\qquad \vdots$$

$$\downarrow I \qquad\qquad \swarrow I$$

$$\vec{e}_1(n-1) \qquad \vec{e}_2(n-1) \xrightarrow{R} \vec{e}_{20}(n-1) = \vec{e}_2(n-1) + a_{n-1}\vec{e}_1(n-1) \tag{3.18}$$

$$\downarrow I \qquad\qquad \swarrow I$$

$$\vec{e}_1(n) \qquad \vec{e}_2(n) \xrightarrow{R} \vec{e}_{20}(n) = \vec{e}_2(n) + a_n\vec{e}_1(n) \tag{3.19}$$

For example, Eqs. (3.15) and (3.16) should be read in the following way: At the height corresponding to step number 1 there are two starting solutions $\vec{e}_1(1)$ and $\vec{e}_2(1)$ -- the two eigenvectors of $\tilde{T}$ corresponding to upgoing waves. The eigenvector $\vec{e}_2(1)$ is replaced by $\vec{e}_{20}(1)$ which is a vector orthogonal to $\vec{e}_1(1)$. After a certain number of integration steps $\vec{e}_1(1)$ and $\vec{e}_{20}(1)$ yields to $\vec{e}_1(2)$ and $\vec{e}_2(2)$ at the height corresponding to step number 2. The whole procedure repeats successively.

At the step number n at z_n in free space the penetrating mode is determined by Eq. (3.14)

$$\vec{e}_p(n) = \vec{e}_{20}(n) + b\vec{e}_1(n) \tag{3.20}$$

Hence, for obtaining the penetrating vector solution at the height corresponding to the step number (n-1) it is first noted from Eq. (3.19) that Eq. (3.20) can be expanded to

$$\vec{e}_p(n) = \vec{e}_2(n) + (b + a_n)\,\vec{e}_1(n) \tag{3.21}$$

which, integrated back gives (see Eq. (3.18)):

$$\begin{aligned}\vec{e}_p(n-1) &= \vec{e}_{20}(n-1) + (b + a_n)\,\vec{e}_1(n-1) \\ &= \vec{e}_2(n-1) + (b + a_n + a_{n-1})\,\vec{e}_1(n-1)\end{aligned} \tag{3.22}$$

Therefore at the step number (n-2) the penetrating mode will be given by

$$\vec{e}_p(n-2) = \vec{e}_{20}(n-2) + (b + a_n + a_{n-1})\,\vec{e}_1(n-2) \tag{3.23}$$

Hence, at any height corresponding to step k the penetrating solution is given by

$$\vec{e}_p(k) = \vec{e}_{20}(k) + \left(b + \sum_{i=k+1}^{n} a_i\right) \cdot \vec{e}_1(k) \tag{3.24}$$

During the integration the stored vectors are $\vec{e}_1(i)$ and the orthogonalized fields $\vec{e}_{20}(i)$ so that the penetrating wave-fields are readily obtained if all the a_i's are stored.

Hence, the penetrating solution is constructed inside the ionosphere and is our second independent solution. It is a possible second solution but, clearly, a different possible second solution could be found if another criterion were used. For example, an independent

solution could be found such that its reflected power is a minimum.

The computer program uses Eq. (3.24) along with an extra scaling procedure for $\vec{e}_1$ for obtaining the penetrating wave-fields. It is necessary to scale-down the dominant mode $\vec{e}_1$ because this field increases too much in comparison with $\vec{e}_2$.

B. DESCRIPTION OF THE COMPUTER PROGRAM

In this section the logistics involved in the full-wave computer program will be described. The duties and the capability of each subroutine will be broadly defined.

The block diagram of FULLWAVE is shown in Figure 2. The program consists of a MAIN PROGRAM and four auxiliary subroutines. They are

subroutine HAMMING

subroutine MATRIX

subroutine BRAIN

and subroutine OUTPUT

The program works under the following plan:

1) Input parameters and data are supplied by a read in statement in the MAIN PROGRAM. Some input parameters are control variables and some are inherent variables of the program such as frequency, angle of incidence, etc. The data consist of the set height, electron density, and collision frequency which is provided in a block of cards. The MAIN PROGRAM then calculates the four eigenvalues q at very high altitudes and select the two of them corresponding to upgoing waves. Finally the eigenvectors relative to this two eigenvalues are formed and subroutine HAMMING is called. At this point all control variables are known and all the duties of the MAIN PROGRAM have been completed. The

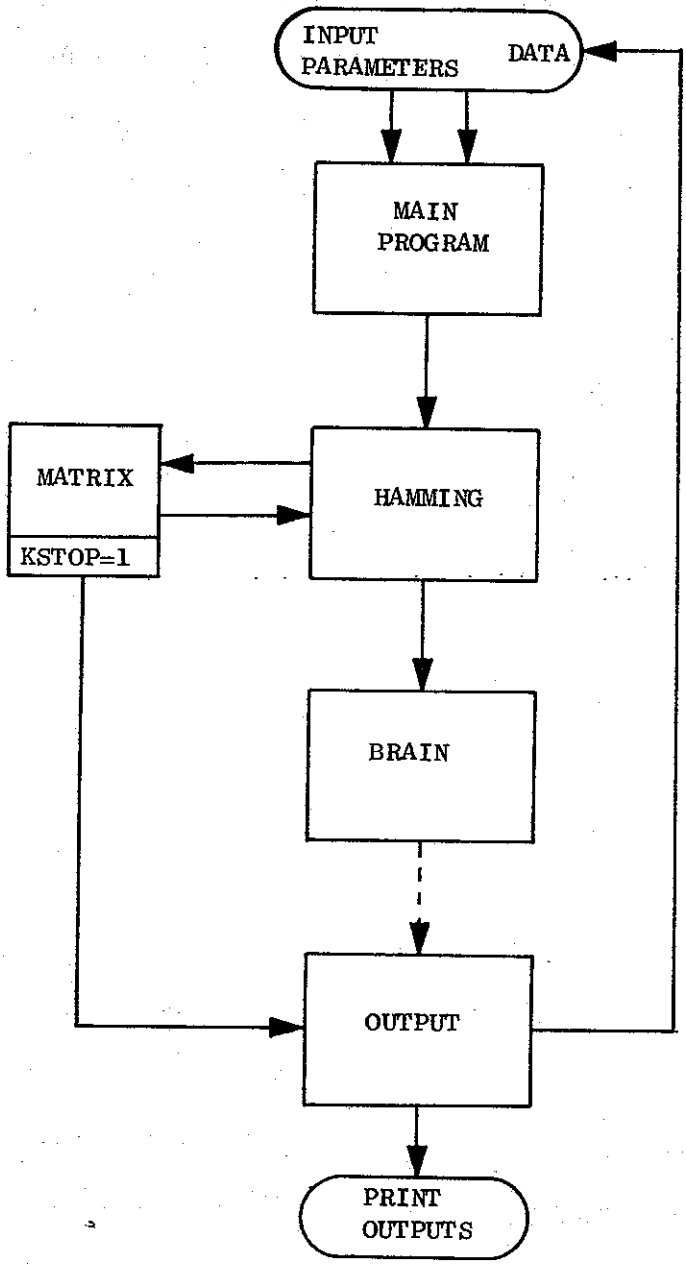

FIGURE 2. Block diagram of "FULLWAVE."

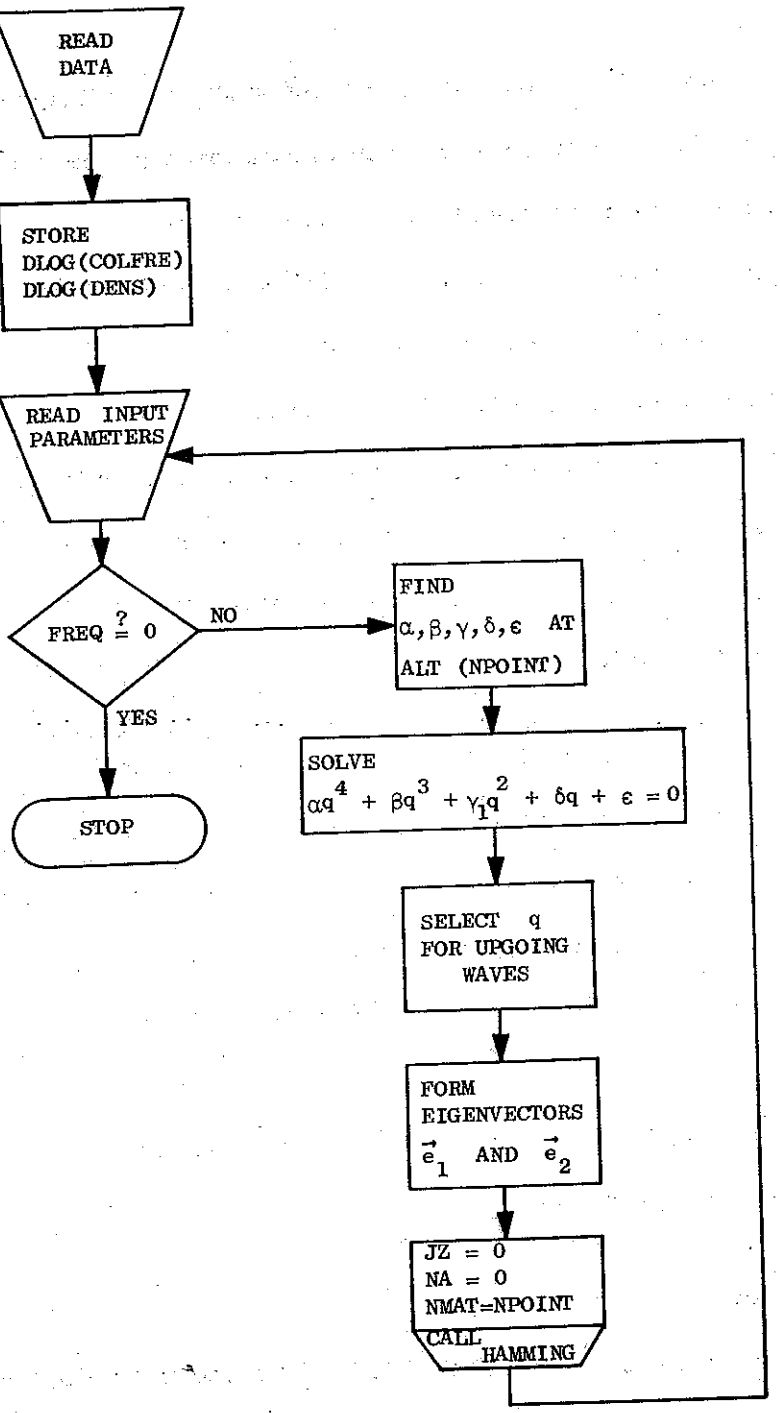

FIGURE 3. Block diagram for the MAIN PROGRAM.

Below the operations performed by the MAIN PROGRAM are described. Figure 3 and the listing of the program are important for a good understanding of the whole procedure.

1. Data input. The program reads NPOINT and NSTEP. NPOINT is the number of points in z where collision frequency and electron density are given. NSTEP is the number of heights where the initial step size of integration will be doubled. The initial step size should be of the order of 1/20 of the local wavelength at the starting height. Based on the author's own experience the present program produces relative errors of the order of 10^{-5} if 1/20 or less of the local wavelength is maintained during the integration. As the density decreases at lower heights the integration step size may be doubled at specified heights, always maintaining a value smaller than 1/20 of the local wavelength. The doubling (or halving) of the step-size is a requirement given by the integration procedure (HAMMING) which does not permit intermediate step sizes.

Next, all the data cards are read in:

ZAXIS, DENS, and COLFRE are height (km), electon density (cm^{-3}) and collision frequency (sec^{-1}) respectively. There are NPOINT cards of this type and ZAXIS is given at equidistant intervals.

NSTEP cards are also read in giving the heights where the integration step-size will be doubled. The name of this control variable is HEIGHT.

Following, DENS and COLFRE are stored in logarithmic form (INS33/35). The reason for doing this is that intermediate points which will be required by the integration procedure will be logarithmically interpolated. Hence it is more practical to store DENS and COLFRE in this form.

2. Input parameters. The next set of input includes the reading of FREQ, FH, ANGI, AZIM and DIP. They are the frequency (Hz), gyro-frequency (Hz), angle I of incidence (degrees), azimuthal angle (degrees), and dip angle of the magnetic field (degrees) respectively (see Figure 1).

Finally the last set of input is read:

HSTART - the height where the integration starts (km)

HEND - the height below the ionosphere where the integration is to be stopped (km)

STEP - the initial step size (km)

HLASTX - the height (km) below which the plasma frequency is made equal to zero. Observe that as DENS and COLFRE are stored in logarithmic form these variables are not allowed to have zero values. Hence these variables are made non-zero near HEND and DENS is effectively made equal to zero for heights lower than HLASTX.

KFORM - a control variable which specifies the outputs to be selected at subroutine OUTPUT. There are available 4 different output formats.

For KFORM = 1 the output will be the penetrating and non-penetrating wave-fields set up by a horizontal electric field of unit amplitude.

For KFORM = 2 the output consists of
- transmission coefficients for the penetrating mode, horizontal, and vertical polarizations
- penetrating and non-penetrating reflection coefficients
- polarization of the penetrating mode
- Budden's reflection coefficients $_{\perp}R_{\perp}$, $_{\perp}R_{\parallel}$, $_{\parallel}R_{\parallel}$, $_{\parallel}R_{\perp}$

For KFORM = 3 the output will be the sum of the outputs for KFORM = 1 and KFORM = 2

For KFORM = 4 the outputs will be
- the output for KFORM = 2
- the envelope of ionospheric x-electric and x-magnetic wave-fields for the penetrating and non-penetrating modes.
- the relative errors committed at each height for the two solutions.

3. Computation of the eigenvalues q at the top. The computer program described here only takes into account the effect of electrons. In this case the coefficients of the quartic of Booker that determines the eigenvalues q of $\tilde{T}$ are given by Eq. (A.14) in Appendix A. In

order to solve the quartic the following symbols are used:

Symbol	Computer Variable
$U = 1-j\nu/\omega$	U
$j = \sqrt{-1}$	AI
ν = collision frequency	COLFRE
$\omega = 2\pi$XFREQ	FAT
Y = FH/FREQ	YY
$X = 8.061 \times 10^7 \times DENS/(FREQ)^2$	XX
$-\xi$ = sin(DIP)	SD
$-\gamma$ = cos(DIP)	CD
q_i = cos I	AN
m = sin I · cos χ	AM
ℓ = sin I · sin χ	AL
α, β, γ_1, δ, ϵ	ALPHA, BETA, GAMMA, DELTA, EPSI

The solution of the quartic equation, Eq. (A.10), is obtained by using a standard method of solving quartic equations. First the resolvent cubic equation is calculated and based on one solution of this cubic the four solutions Q(1), Q(2), Q(3), and Q(4) of the quartic are determined (INS75/110). Next, the eigenvalues corresponding to upgoing waves are selected by choosing the eigenvalues with negative imaginary part (INS112/125).

4. Computation of the eigenvectors at the top. For each eigenvalue q corresponding to upgoing waves the eigenvectors are given by Eqs. (B.13), (B.14) and (B.15) or (B.18) from Appendix B. The elements T_{ij} are computed at the starting height and then the parameters A_1 to A_6 are determined for each upgoing eigenvalue q (INS126/148). The eigenvectors are then computed with E_x chosen arbitrarily equal to one (INS149/170).

Equation (2.24) characterizes a complex vector $\vec{e}$ with four elements. If we separate real and imaginary parts we come up with a new

form of Eq. (2.24);

$$\frac{d\vec{Y}(Z)}{dZ} = \tilde{V}(Z) \cdot \vec{Y}(Z) \tag{3.25}$$

where $\tilde{V}(Z)$ is an 8X8 real matrix related to $\tilde{T}(Z)$ as will be seen in section F. The real column vector $\vec{Y}(Z)$ is related to the elements of $\vec{e}$ in the following way:

$$\vec{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \\ Y_6 \\ Y_7 \\ Y_8 \end{bmatrix} = \begin{bmatrix} \Re(E_x) \\ \Re(-E_y) \\ \Re(Z_o H_x) \\ \Re(Z_o H_y) \\ \Im(E_x) \\ \Im(-E_y) \\ \Im(Z_o H_x) \\ \Im(Z_o H_y) \end{bmatrix} \tag{3.26}$$

Hence, instead of working with a set of four linear differential equations with complex variables, a set of eight linear differential equations with real variables is integrated. INS149 to INS170 are the FORTRAN statements for computing the two eigenvectors $\vec{Y}_1$ and $\vec{Y}_2$ corresponding to $\vec{e}_1$ and $\vec{e}_2$ respectively. $\vec{Y}_1$ corresponds to the dominant evanescent mode and $\vec{Y}_2$ corresponds to the "whistler mode" at the top.

5. Starting the integration. The final statements are concerned with the setting of starting values to some control variables. Subroutine

HAMMING is called and the command continues outside the MAIN PROGRAM. The command will return to the MAIN PROGRAM after all the outputs have been obtained.

D. SUBROUTINE HAMMING

1. General aspects. HAMMING is the numerical procedure for integrating the wave equations, Eq. (2.24), or, actually Eq. (3.25). It uses a modified predictor-corrector method introduced by Hamming [1959]. The purpose is to obtain an approximate solution of a linear system of first-order ordinary differential equations with given initial values. Subroutine HAMMING is a stable fourth-order integration procedure, requiring the evaluation of the right-hand side of Eq. (3.25) only two times per step. The matrix $\tilde{V}$ will be evaluated only once per step. This is a great advantage compared with other methods of the same order of accuracy, especially the Runge-Kutta method, which requires the evaluation of the right-hand side four times per step. Another advantage is that at each step the procedure gives an estimate for the local truncation error.

On the other hand, Hamming's predictor-corrector method is not self-starting; that is, the functional values at a single previous point are not enough to obtain the functional values ahead. Therefore, to get the starting values, a special Runge-Kutta procedure followed by one iteration step is added to the predictor-corrector method.

Given the linear system of first-order ordinary differential equations, Eq. (3.25), and the starting eigenvalue

$$\vec{Y}(Z_o) = \vec{Y}_o \qquad (3.27)$$

the problem is to estimate $\vec{Y}(Z)$ at discrete points Z_i, starting with the knowledge of Eq. (3.27). For stability purposes, the modification by Hamming of Milne's classical modified predictor-corrector method is preferred. Knowing the results at the equidistant points Z_{j-3}, Z_{j-2}, Z_{j-1}, and Z_j, the result at $Z_{j+1} = Z_j + h$ is computed by the following formulas (a prime denotes d/dZ):

Predictor:
$$\vec{P}_{j+1} = \vec{Y}_{j-3} + \frac{4h}{3}\left[2\vec{Y}'_j - \vec{Y}'_{j-1} + 2\vec{Y}'_{j-2}\right] \tag{3.28}$$

Modifier:
$$\vec{M}_{j+1} = \vec{P}_{j+1} - \frac{112}{121}\left(\vec{P}_j - \vec{C}_j\right) \tag{3.29}$$

$$\vec{M}'_{j+1} = \tilde{V}(Z_{j+1}) \cdot \vec{M}_{j+1} \tag{3.30}$$

Corrector:
$$\vec{C}_{j+1} = \frac{1}{8}\left\{9\vec{Y}_j - \vec{Y}_{j-2} + 3h\left[\vec{M}'_{j+1} + 2\vec{Y}'_j - \vec{Y}'_{j-1}\right]\right\} \tag{3.31}$$

Next values:
$$\vec{Y}_{j+1} = \vec{C}_{j+1} + \frac{9}{121}\left[\vec{P}_{j+1} - \vec{C}_{j+1}\right] \tag{3.32}$$

$$\vec{Y}'_{j+1} = \tilde{V}(Z_{j+1}) \cdot \vec{Y}_{j+1} \tag{3.33}$$

In the above formulas $\vec{Y}$, $\vec{Y}'$, $\vec{P}$, $\vec{M}$, $\vec{M}'$, and $\vec{C}$ are all column vectors with eight components, and $\tilde{V}$ is an 8X8 real matrix provided by subroutine MATRIX (section F). The local truncation error committed using Eqs. (3.28) to (3.32) is estimated to be

$$\vec{Tr} \cong \frac{9}{121}\left[\vec{P}_{j+1} - \vec{C}_{j+1}\right] \tag{3.34}$$

Hence, if equal error weight for all eight elements of $\vec{Y}$ is

assumed the evaluation of the local truncation can be estimated by

$$\delta = \sum_{i=1}^{8} \cdot \frac{1}{8} \left| P_{(j+1),i} - C_{(j+1),i} \right| \qquad (3.35)$$

Equation (3.35) could be used in order to control the error by halving or doubling the step-size. This procedure is not followed here because it is time consuming. Instead, δ is obtained and this parameter is printed out for KFORM = 4. The experience shows that for obtaining errors of the order of $10^{-5}/10^{-6}$ the step size should be 1/20 or less of the local wavelength. At specified heights the step size is doubled because the local wavelength becomes larger as the integration proceeds downward. Doubling the step size requires replacing the value of $(\vec{P}_{j+1} - \vec{C}_{j+1})$ to be used in the next step by (see Eq. (3.29):

$$\vec{P}_{j+1} - \vec{C}_{j+1} \cong \frac{242}{27} [\vec{Y}_{j+1} - \vec{Y}_{j-5}] - \frac{121}{36} \cdot 2h[\vec{Y}'_{j+1} + 3\vec{Y}'_{j-1} + 3\vec{Y}'_{j-3} + \vec{Y}'_{j-5}] \qquad (3.36)$$

2. <u>The starting Runge-Kutta procedure</u>. In order to start Hamming's modified predictor-corrector method it is necessary to know the functional and derivative values at four preceding equidistant points Z_o, Z_1, Z_2, Z_3. The values $\vec{Y}_o$ and the derivative $\vec{Y}'_o = \tilde{V}(Z_o) \cdot \vec{Y}_o$ are known because $\vec{Y}_o$ is specified by input and $\tilde{V}(Z_o)$ is provided by subroutine MATRIX. For computation of $\vec{Y}_1$, $\vec{Y}'_1$, $\vec{Y}_2$, $\vec{Y}'_2$, $\vec{Y}_3$, and $\vec{Y}'_3$ a special Runge-Kutta procedure suggested by Ralston [1962] is used. Starting at Z_j the routine computes the vector at $Z_{j+1} = Z_j + h$ using the following formulas

$$\vec{K}_1 = h \cdot \vec{Y}'_j \qquad (3.37)$$

$$\vec{K}_2 = h \cdot \tilde{V}(Z_j + 0.4h) \cdot [\vec{Y}_j + 0.4\vec{K}_1] \tag{3.38}$$

$$\vec{K}_3 = h \cdot \tilde{V}(Z_j + 0.45573725421878943h) \cdot [\vec{Y}_j + 0.29697760924775360\vec{K}_1 + 0.15875964497103583\vec{K}_2] \tag{3.39}$$

$$\vec{K}_4 = h \cdot \tilde{V}(Z_j + h) \cdot [\vec{Y}_j + 0.21810038822592047\vec{K}_1 - 3.05096514869293 08\vec{K}_2 + 3.832864760467010 3\vec{K}_3] \tag{3.40}$$

Next value

$$\vec{Y}_{j+1} = \vec{Y}_j + 0.17476028226269037\vec{K}_1 - 0.55148066287873294\vec{K}_2 + 1.2055355993965235\vec{K}_3 + 0.17118478121951903\vec{K}_4 \tag{3.41}$$

The above formulas are not very stable but this is not very important because they are used only in three successive steps (j = 1,2,3). On the other hand they have the smallest bound of truncation errors of all fourth-order Runge-Kutta procedures and so they are best suited to start a non-self-starting integration method. Furthermore these starting values will be refined by one iteration step using the following fourth-order interpolation formulas:

$$\vec{Y}_1 = \vec{Y}_o + \frac{h}{24}[9\vec{Y}_o' + 19\vec{Y}_1' - 5\vec{Y}_2' + \vec{Y}_3'] \tag{3.42}$$

$$\vec{Y}_2 = \vec{Y}_o + \frac{h}{3}[\vec{Y}_o' + 4\vec{Y}_1' + \vec{Y}_2'] \tag{3.43}$$

$$\vec{Y}_3 = \vec{Y}_o + \frac{3h}{8}[\vec{Y}_o' + 3\vec{Y}_1' + 3\vec{Y}_2' + \vec{Y}_3'] \tag{3.44}$$

which must be considered as an iteration procedure. That is, first the

results of the previous Runge-Kutta method are handed to the right-hand side of Eq. (3.42) to compute a refined $\vec{Y}_1$. After computing the refined $\vec{Y}_1'$, the refined vector $\vec{Y}_2$ is generated using Eq. (3.43). Finally, refined $\vec{Y}_2'$ is used and combined with other values in the right-hand side of Eq. (3.44) to compute the refined vector $\vec{Y}_3$.

Subroutine HAMMING has been derived from some similar procedures described in the System/360 Scientific Subroutine Package, Version II, published by IBM. A more complete mathematical analysis is given by Ralston [1965].

3. Block diagram of subroutine HAMMING. The procedure to be used for integrating Eq. (3.25) has been presented in Chapter 2. Actually it is necessary to integrate two vectors corresponding to the evanescent and to the "whistler mode" upgoing waves. Hence, all the procedures to be followed in the integration of one vector can be duplicated for the other. This process is time saving because $\tilde{V}(Z)$ is calculated only one time for the two vectors at each height. The program operation for one vector is described below although it is understood that actually two vectors are being integrated. The program listing in pages 68 to 86 makes this clear and should be consulted at each stage of the following description.

The block diagram of the subroutine HAMMING is shown in Figure 4. The starting eigenvectors, the upper and the lower bound of the integration interval, the starting step-size, and the heights where the step size will be doubled constitute the set of values required by HAMMING and they are supplied by the MAIN PROGRAM. On the other hand, the allocation of special intermediate-result vectors are stored in a 15X8 auxiliary array AUX. At a height Z_j the stored vectors in AUX are

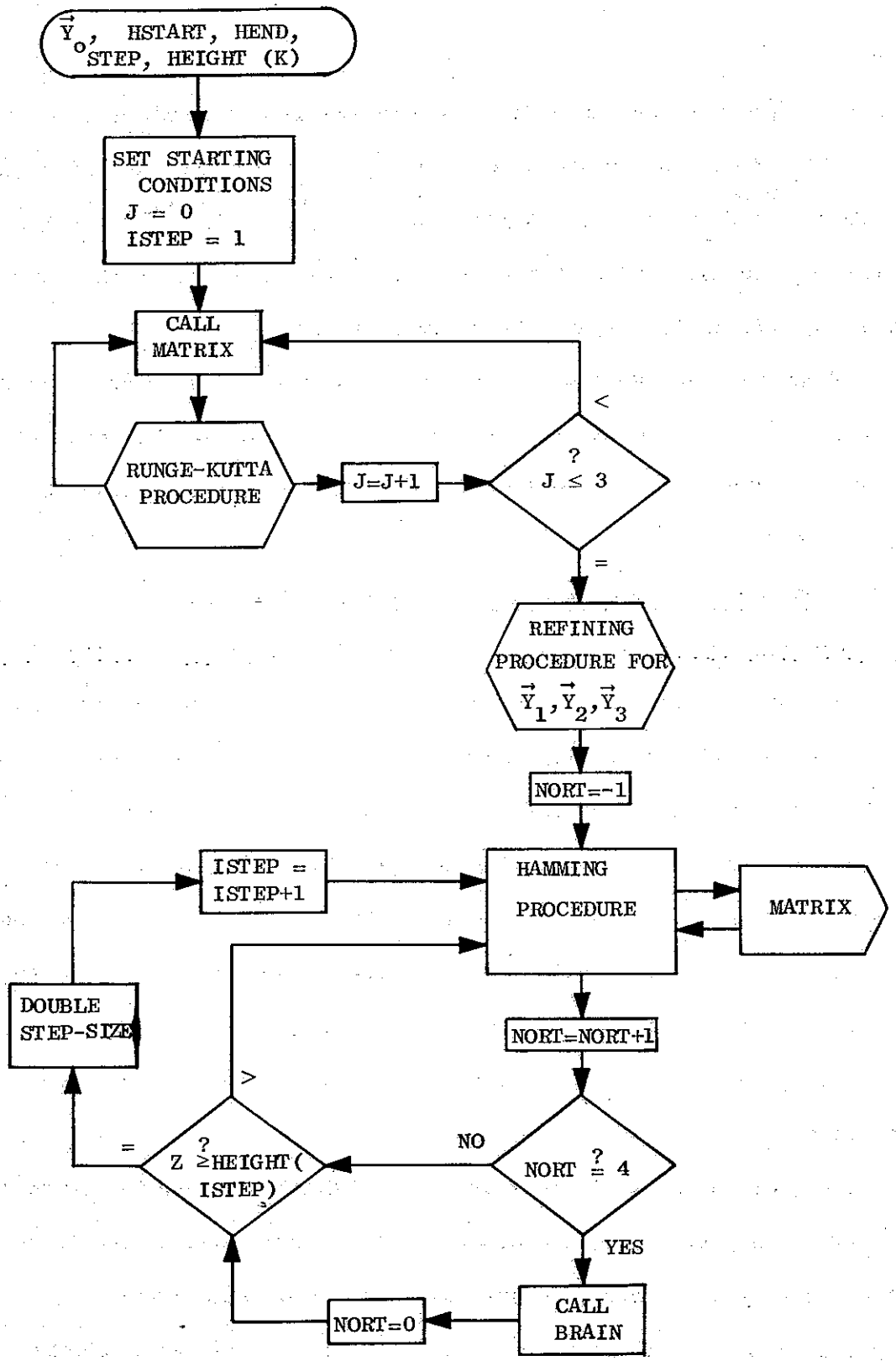

FIGURE 4. Block diagram of subroutine HAMMING.

$$\vec{AUX}(1) = \vec{Y}_{j-6} \qquad \vec{AUX}(8) = \vec{Y}'_{j-6} \tag{3.45}$$

$$\vec{AUX}(2) = \vec{Y}_{j-5} \qquad \vec{AUX}(9) = \vec{Y}'_{j-5} \tag{3.46}$$

$$\vec{AUX}(3) = \vec{Y}_{j-4} \qquad \vec{AUX}(10) = \vec{Y}'_{j-4} \tag{3.47}$$

$$\vec{AUX}(4) = \vec{Y}_{j-3} \qquad \vec{AUX}(11) = \vec{Y}'_{j-3} \tag{3.48}$$

$$\vec{AUX}(5) = \vec{Y}_{j-2} \qquad \vec{AUX}(12) = \vec{Y}'_{j-2} \tag{3.49}$$

$$\vec{AUX}(6) = \vec{Y}_{j-1} \qquad \vec{AUX}(13) = \vec{Y}'_{j-1} \tag{3.50}$$

$$\vec{AUX}(7) = \vec{Y}_{j} \qquad \vec{AUX}(14) = \vec{Y}'_{j} \tag{3.51}$$

and finally,

$$\text{vector } \vec{AUX}(15) = (\vec{P}_j - \vec{C}_j) \tag{3.52}$$

The procedure begins with the program setting

$$\vec{AUX}(1) = \vec{Y}_o \qquad \text{INS210} \tag{3.53}$$

$$Z = \text{HSTART} \qquad \text{INS206} \tag{3.54}$$

and $$H = \text{STEP} \qquad \text{INS207} \tag{3.55}$$

Next, MATRIX is called and the derivative

$$\vec{Y}'_o = \tilde{V}(Z) \cdot \vec{Y}_o \qquad \text{INS222} \tag{3.56}$$

is formed. The statements after the calling of MATRIX (INS193/204) constitute a routine procedure which will be repeated at each calling.

Following each call to MATRIX there is a text to check whether KSTOP has been changed to 1 or not. If KSTOP = 1 the subroutine returns to MAIN.

After the calculation of the initial values the special Runge-Kutta procedure for calculating $\vec{Y}_1$, $\vec{Y}_2$ and $\vec{Y}_3$ is put in operation by the statement "GO TO 200". The Runge-Kutta procedure corresponds to INS288/323. Observe that

INS296	corresponds to Eq. (3.37) for	$\vec{K}_1$
INS306	(3.38) for	$\vec{K}_2$
INS316	(3.39) for	$\vec{K}_3$
INS326	(3.41) for	$\vec{Y}_{j+1}$

Vectors $\vec{Y}_1$, $\vec{Y}_2$ and $\vec{Y}_3$ produced by the above procedure are stored in

INS228	$\vec{AUX}(2) = \vec{Y}_1$
INS233	$\vec{AUX}(9) = \vec{Y}_1'$
INS240	$\vec{AUX}(3) = \vec{Y}_2$
INS245	$\vec{AUX}(10) = \vec{Y}_2'$
INS252	$\vec{AUX}(4) = \vec{Y}_3$
INS257	$\vec{AUX}(11) = \vec{Y}_3'$

The next step is to use the fourth-order interpolation formulas, Eqs. (3.42) to (3.44) to refine the starting values provided by the above Runge-Kutta process. This is accomplished by means of INS259 to INS287. Now the value of the vector $\vec{Y}$ is given at four equidistant points

$$Z = \text{HSTART}, \quad (Z+H), \quad (Z+2H), \quad (Z+3H)$$

and then the integration may continue with Hamming's stable predictor-corrector method. Notice that although $\vec{Y}(Z)$ has been computed at four points no information has been filtered out. In fact results of the integration will come out only when all the elements of AUX have been computed. There is no strong reason for doing this. This is part of our policy of computing the first points with a maximum of accuracy: the first ~ 10 points are computed with a very small step-size, approximately 1/50 of the local wavelength, after which the step size is doubled. For doubling the step size all the elements of AUX are required. Another characteristic of the subroutine, whose reason for being there has been dictated by experience, is explained below. The orthogonalizing procedure previously discussed (section A) must be applied at discrete heights separated by a specified number of integration steps. Here four steps are specified, that is, the fields are integrated in four consecutive steps and only at the last point is the information filtered to BRAIN where the fields will be orthogonalized. BRAIN always returns the orthogonalized fields to HAMMING. The variable controlling the number of steps before each orthogonalization is NORT (see INS378).

The modified predictor-corrector method of Hamming consists of INS300 to INS376. The correspondence between the formulation given previously and program instructions is shown below:

Predictor,	Eq. (3.28)	INS341
Modifier,	Eq. (3.29)	INS343
Corrector,	Eq. (3.30)	INS350
$(\vec{P}_{j+1} - \vec{C}_{j+1})$,	Eq. (3.31)	INS352
Next $\vec{Y}_{j+1}$,	Eq. (3.32)	INS354

δ , Eq. (3.33) INS359

Derivative $\vec{Y}'_{j+1}$ INS367

After the completion of each integration step a test is generated to check whether or not the next step should be doubled. This is done by comparing Z with HEIGHT(ISTEP) as shown at INS375/376. If the step is to be doubled then the vector $(\vec{P}_{j+1} - \vec{C}_{j+1})$ must be changed as given by Eq. (3.36). This is done by INS396.

E. SUBROUTINE BRAIN

In subroutine BRAIN the solution-vectors are scaled, orthogonalized and stored. All these missions are very important in the whole problem of finding the wave-fields inside the ionosphere.

The block diagram of subroutine BRAIN is shown in Figure 5 and the listing of the program in pages 78 and 79.

The first action in BRAIN is to check whether or not a generated variable TEST1 is greater or smaller than 10^3. This variable is a measure of the amplitude of the dominant mode, i.e.,

$$\text{TEST1} = \frac{1}{2}\left\{E_x^2 + (Z_o H_x)^2\right\}^{\frac{1}{2}} \qquad (3.57)$$

If TEST1 is equal to or greater than 10^3 the solution-vector of the dominant mode is scaled down by multiplying its value by 10^{-3} (INS419/425). The height where a particular scaling occurs is stored in HSCALE(NA) = Z. The first scaling occurs in the first calling of BRAIN because the eigen-vector corresponding to the dominant mode is multiplied by 10^3 before calling HAMMING (see INS175). This is done because it is convenient for the first scaling to occur at the first point stored in BRAIN, as will be

apparent when the reconstruction of the wave-fields in subroutine OUTPUT is discussed (Section G).

In subroutine BRAIN the fields will be stored in complex form. Thus $\vec{e}_1$, $\vec{e}_2$, $\frac{d\vec{e}_1}{dZ}$, and $\frac{d\vec{e}_2}{dZ}$ are first formed - INS437/440. Next the orthogonalizing factor AORT(JZ) is generated (INS442/444) and $\vec{e}_2$ is replaced by the orthogonalized field vector

$$\vec{E}2(JZ) \longrightarrow \vec{E}_2(JZ) = \vec{E}2(JZ) + AORT(JZ) * \vec{E}1(JZ) \qquad \text{INS446} \qquad (3.58)$$

There is a corresponding equation for $\frac{d\vec{e}_2}{dZ}$. Finally the related field vector $\vec{Y}_2(Z)$ is restored and BRAIN returns to HAMMING.

For KFORM = 1 one more step is added to BRAIN, namely the calculation of relative errors committed in the integrations of $\vec{e}_1$ and $\vec{e}_2$ (INS432/433).

Summarizing, after each call to subroutine BRAIN the following set of variables is stored:

NA, HSCALE(NA)	if TEST1 $\geq 10^3$
AORT(JZ), ALT(JZ)	
$\vec{E}1(JZ)$, $\vec{E}2(JZ)$	orthogonalized field vectors
$\vec{DE}1(JZ)$, $\vec{DE}2(JZ)$	derivatives
ERROR1(JZ), ERROR2(JZ)	if KFORM = 4

F. SUBROUTINE MATRIX

Subroutine MATRIX performs the following operations:

1) It returns all the 64 elements of $\tilde{V}(Z)$ to HAMMING at each call.

2) It effectively transforms the medium in free space if the height is lower than HLASTX.

3) It turns the command to OUTPUT when the height is lower than HEND.

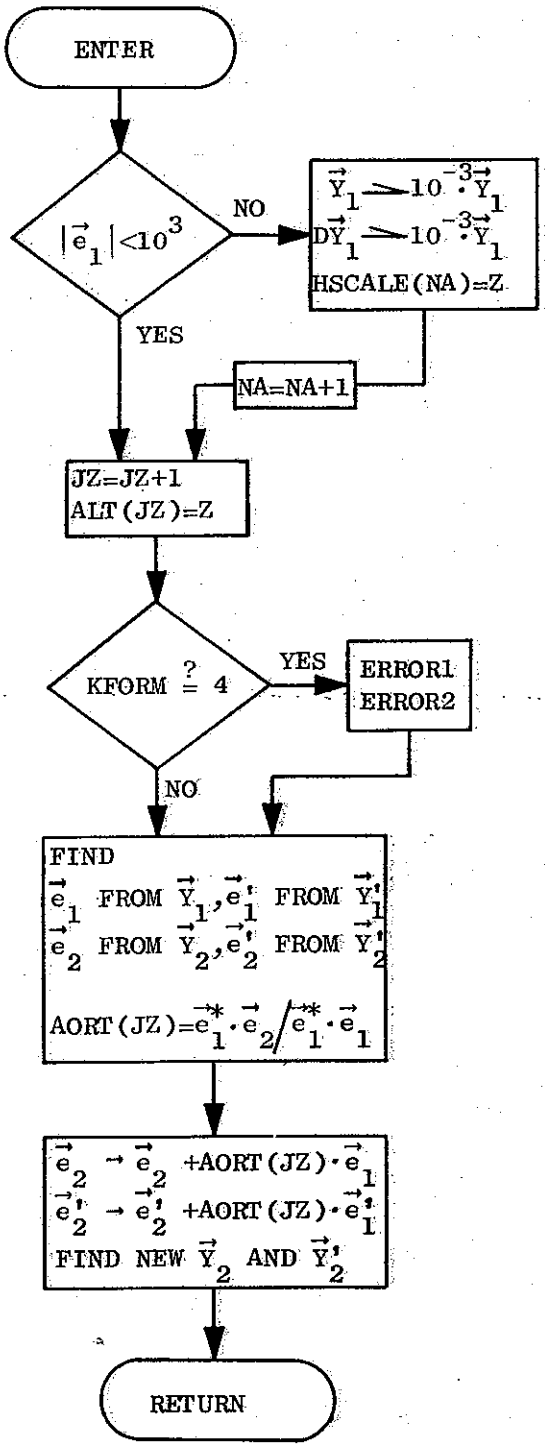

FIGURE 5. Block diagram of subroutine BRAIN.

The block diagram of subroutine MATRIX is shown in Figure 6 and the related program listing can be found on pages 80 to 81.

If the height is greater than HLASTX then the electron density and collision frequency must be found for this particular height. This procedure starts at INS489 where the actual value of Z is compared with ZAXIS(NMAT). Observe that NMAT starts with the value NSTEP set by INS173. If Z coincides with some height ZAXIS this means that no interpolation is necessary because DENS and COLFRE are already available at this height (INS492/493). If Z does not coincide with ZAXIS a linear interpolation is made in the logarithmically stored values of DENS and COLFRE (INS495/493). Hence, the values of electron density and collision frequency are determined at Z and the computation of $\tilde{T}(Z)$ starts.

When only electrons are involved the elements of $\tilde{T}(Z)$ are given by Eqs. (2.49) and (2.50). The whole set T_{ij} is calculated between INS503 and INS524. Notice that matrix $\tilde{T}(Z)$ is columnwise stored.

Relationship between $\tilde{T}(Z)$ and $\tilde{V}(Z)$. Each element of $\vec{\vec{e}}$ is a complex function and therefore:

$$\vec{\vec{e}} = \begin{bmatrix} E_x \\ -E_y \\ Z_o H_x \\ Z_o H_y \end{bmatrix} = \begin{bmatrix} Y_1 + j\,Y_5 \\ Y_2 + j\,Y_6 \\ Y_3 + j\,Y_7 \\ Y_4 + j\,Y_8 \end{bmatrix} \tag{3.59}$$

Now replacing $\tilde{T}(Z)$ by $-jk\tilde{T}(Z)$,

$$\tilde{T}(Z) \longleftrightarrow -jk\tilde{T}(Z) \tag{3.60}$$

Hence Eq. (2.24) is now written

$$\frac{d\vec{\vec{e}}}{dZ} = \tilde{T} \cdot \vec{\vec{e}} \tag{3.61}$$

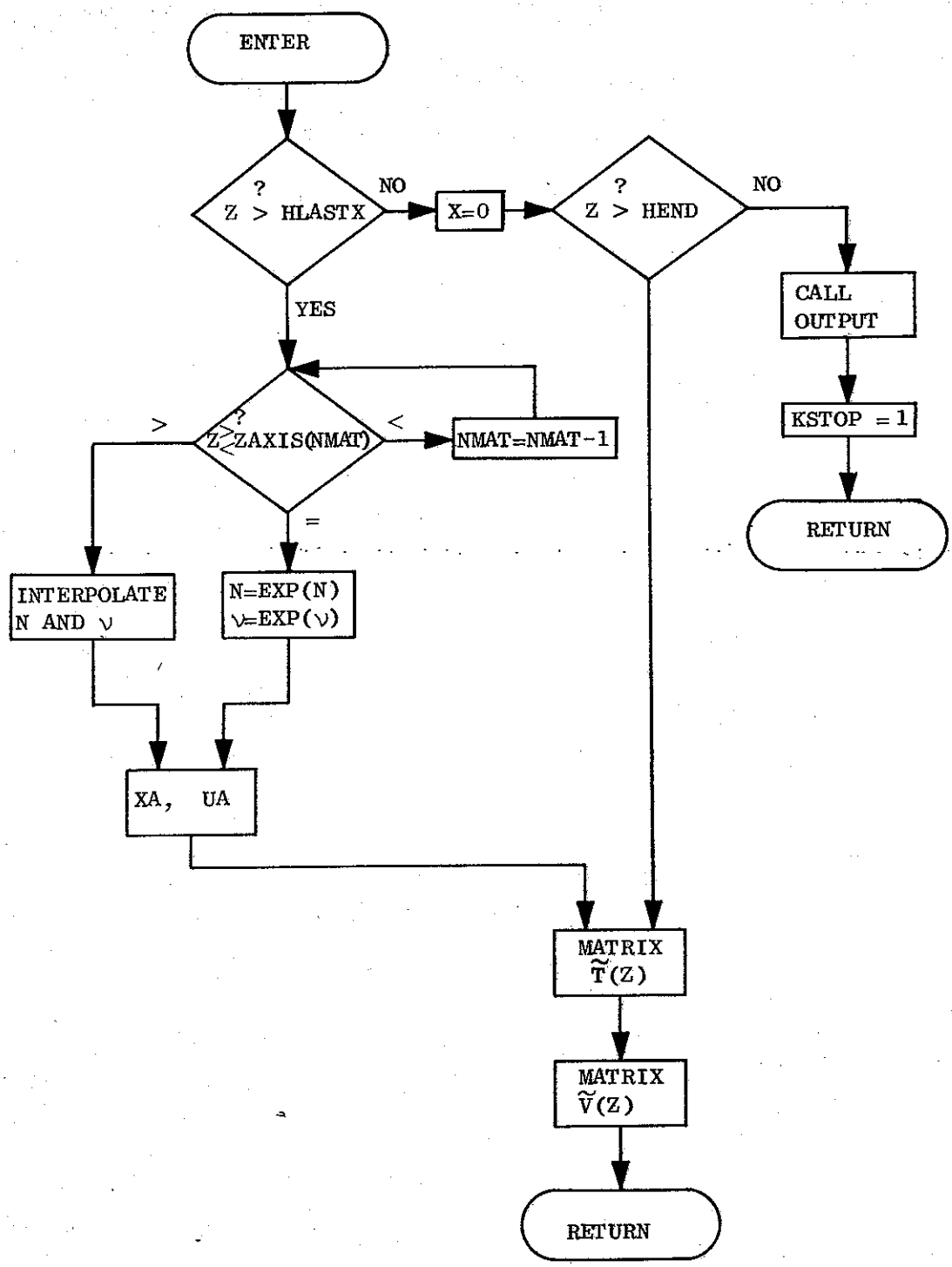

FIGURE 6. Block diagram of subroutine MATRIX.

Next the matrix $\tilde{T}$ is written as it is stored in the computer

$$\tilde{T} = \begin{bmatrix} T_1 & T_5 & T_9 & T_{13} \\ T_2 & T_6 & T_{10} & T_{14} \\ T_3 & T_7 & T_{11} & T_{15} \\ T_4 & T_8 & T_{12} & T_{16} \end{bmatrix} \tag{3.62}$$

All the elements T_i, $i = 1, 16$, are complex in principle and then Eq. (3.62) is fully written in the following form

$$T = \begin{bmatrix} \Re(T_1)+j\Im(T_1) & \Re(T_5)+j\Im(T_5) & \Re(T_9)+j\Im(T_9) & \Re(T_{13})+j\Im(T_{13}) \\ \Re(T_2)+j\Im(T_2) & \Re(T_6)+j\Im(T_6) & \Re(T_{10})+j\Im(T_{10}) & \Re(T_{14})+j\Im(T_{14}) \\ \Re(T_3)+j\Im(T_3) & \Re(T_7)+j\Im(T_7) & \Re(T_{11})+j\Im(T_{11}) & \Re(T_{15})+j\Im(T_{15}) \\ \Re(T_4)+j\Im(T_4) & \Re(T_8)+j\Im(T_8) & \Re(T_{12})+j\Im(T_{12}) & \Re(T_{16})+j\Im(T_{16}) \end{bmatrix} \tag{3.63}$$

Making use of Eqs. (3.59) and (3.63), Eq. (3.61) can be expressed in another form if the real and the imaginary parts of the left-hand side are equated with the corresponding real and imaginary parts of the right-hand side respectively. For example:

$$\frac{dY_1}{dZ} = [\Re(T_1)\ \Re(T_5)\ \Re(T_9)\ \Re(T_{13})]\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix} - [\Im(T_1)\ \Im(T_5)\ \Im(T_9)\ \Im(T_{13})]\begin{bmatrix} Y_5 \\ Y_6 \\ Y_7 \\ Y_8 \end{bmatrix} \tag{3.64}$$

or, better

$$\frac{dY_1}{dZ} = [\Re(T_1)\ \Re(T_5)\ \Re(T_9)\ \Re(T_{13}) - \Im(T_1)\ -\Im(T_5)\ -\Im(T_9)\ -\Im(T_{13})]\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \\ Y_6 \\ Y_7 \\ Y_8 \end{bmatrix} \quad (3.65)$$

Thus, Eq. (2.24) is transformed to

$$\frac{d\vec{Y}}{dZ} = \tilde{V}(Z) \cdot \vec{Y} \quad (3.66)$$

where $\tilde{V}(Z)$ is given by

$$\tilde{V} = \left[\begin{array}{cccc|cccc} \Re(T_1) & \Re(T_5) & \Re(T_9) & \Re(T_{13}) & -\Im(T_1) & -\Im(T_5) & -\Im(T_9) & -\Im(T_{13}) \\ \Re(T_2) & \Re(T_6) & \Re(T_{10}) & \Re(T_{14}) & -\Im(T_2) & -\Im(T_6) & -\Im(T_{10}) & -\Im(T_{14}) \\ \Re(T_3) & \Re(T_7) & \Re(T_{11}) & \Re(T_{15}) & -\Im(T_3) & -\Im(T_7) & -\Im(T_{11}) & -\Im(T_{15}) \\ \Re(T_4) & \Re(T_8) & \Re(T_{12}) & \Re(T_{16}) & -\Im(T_4) & -\Im(T_8) & -\Im(T_{12}) & -\Im(T_{16}) \\ \hline \Im(T_1) & \Im(T_5) & \Im(T_9) & \Im(T_{13}) & \Re(T_1) & \Re(T_5) & \Re(T_9) & \Re(T_{13}) \\ \Im(T_2) & \Im(T_6) & \Im(T_{10}) & \Im(T_{14}) & \Re(T_2) & \Re(T_6) & \Re(T_{10}) & \Re(T_{14}) \\ \Im(T_3) & \Im(T_7) & \Im(T_{11}) & \Im(T_{15}) & \Re(T_3) & \Re(T_7) & \Re(T_{11}) & \Re(T_{15}) \\ \Im(T_4) & \Im(T_8) & \Im(T_{12}) & \Im(T_{16}) & \Re(T_4) & \Re(T_8) & \Re(T_{12}) & \Re(T_{16}) \end{array}\right] \quad (3.67)$$

And notice that if $\tilde{V}$ is columnwise stored the following relationships exist:

$$V_{KK} = \Re(T_K) \quad (3.68)$$

$$V_{MM} = \Im(T_K) \quad (3.69)$$

$$V_{LL} = -\Im(T_K) \tag{3.70}$$

$$V_L = \Re(T_K) \tag{3.71}$$

where

$$K = I + 4(J-1) \tag{3.72}$$

$$KK = I + 8(J-1) \tag{3.73}$$

$$MM = I + 4 + 8(J-1) \tag{3.74}$$

$$LL = I + 32 + 8(J-1) \tag{3.75}$$

$$L = I + 36 + 8(J-1) \tag{3.76}$$

and

$$I = 1, 2, 3, 4 \tag{3.77}$$

$$J = 1, 2, 3, 4 \tag{3.78}$$

The operations determining $\tilde{V}(Z)$ are INS527 to INS538.

Finally one last comment should be made about the possibility of having the data in analytical form instead of equally spaced points (DENS and COLFRE). If the height distribution of ionization and collision frequency are functionally given then the first part of MATRIX must be changed by the respective functions. From INS503 to the end of the subroutine everything can be maintained. Obviously the corresponding read-in statements in the MAIN PROGRAM would not be necessary.

Subroutine MATRIX transfers the command to OUTPUT if the height is less or equal to HEND. Following the return statement from OUTPUT the control variable KSTOP is made equal to one. This will in turn transfer the command to the MAIN PROGRAM.

G. SUBROUTINE OUTPUT

In subroutine OUTPUT all the results provided by FULLWAVE are obtained and printed out. From the standpoint of computational technique the hard job has already been completed and all the results are stored. Now it is only necessary to combine conveniently the stored results in order to get suitable information about the whole process of reflection - absorption - transmission in the given ionosphere. A set of output parameters that can be obtained with the FULLWAVE program is presented. Clearly the capabilities of the program can be extended depending upon the requirements established by the problem at hand.

The block diagram of subroutine OUTPUT is shown in Figure 7 and a listing of the computer program can be found between pages 82 and 86.

1. Obtaining upgoing and downgoing waves. The stored field-vectors $\vec{e}_1$ and $\vec{e}_{20}$ are calculated effectively in free space in the last integration steps. These two vectors are the orthogonalized vector solutions which have been stored in BRAIN. The important feature about them is that they are independent solutions (it is impossible to obtain one of them by multiplying the other by a complex constant). Hence these two solutions can be combined conveniently in order to obtain any specified incident wave. This fact is shown by first observing that it is possible to determine upgoing and downgoing waves for each independent solution as follows:

It is known that the Z-variations of the incident and the reflected wave are $\exp(-jkq_i z)$ and $\exp(jkq_i z)$ respectively. Then at the last calculated point it is known that

$$E_x = U_x + D_x \tag{3.79}$$

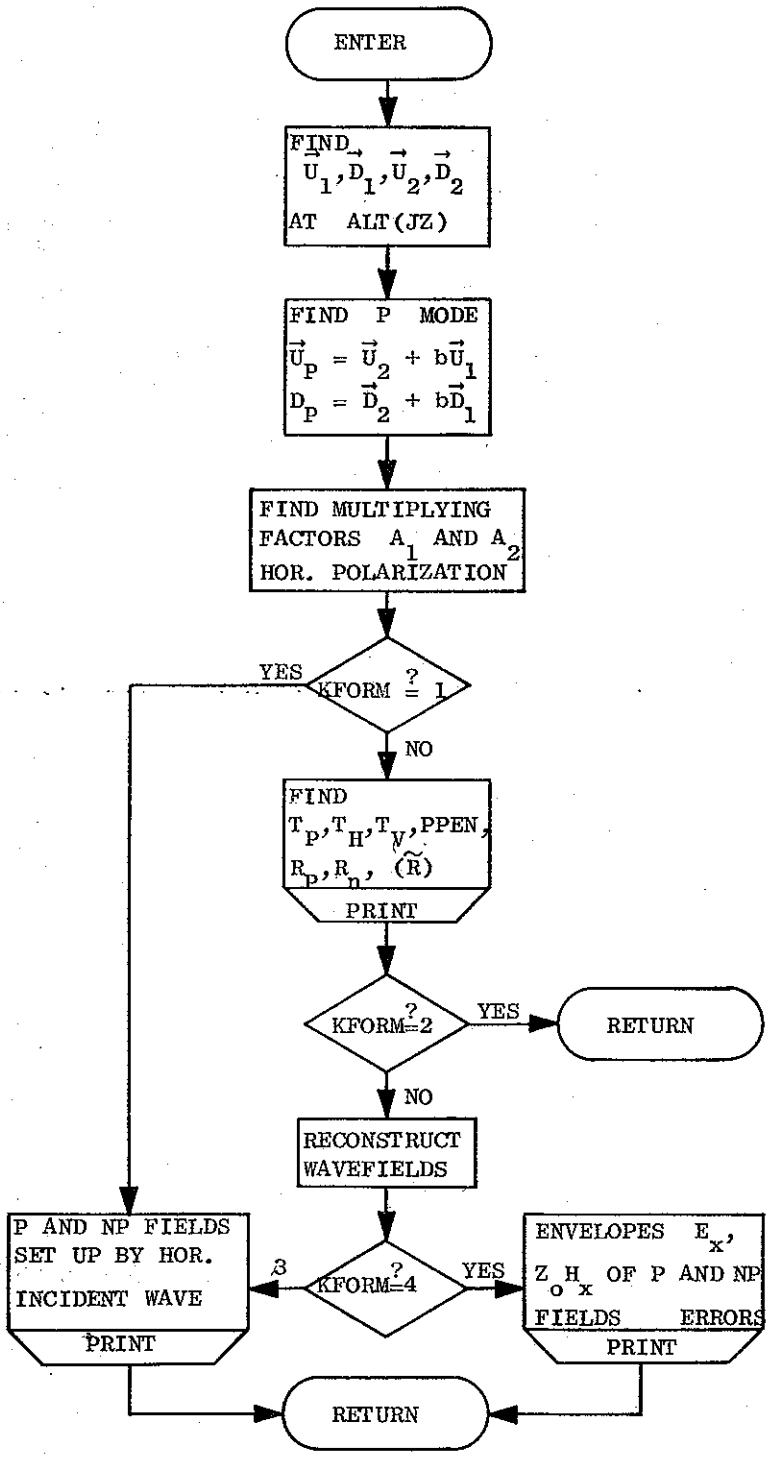

FIGURE 7. Block diagram of subroutine OUTPUT.

and

$$\frac{E'_x}{jkq_i} = - U_x + D_x \tag{3.80}$$

where E_x is the total x-electric field given by the particular $\vec{e}$ vector at HEND, U_x and D_x are the x-electric fields for the upgoing and downgoing waves respectively and E'_x is the z-derivative of E_x.

Hence, U_x and D_x are determined by

$$U_x = \frac{1}{2}\left\{E_x - \frac{E'_x}{jkq_i}\right\} \tag{3.81}$$

$$D_x = \frac{1}{2}\left\{E_x + \frac{E'_x}{jkq_i}\right\} \tag{3.82}$$

It is easy to determine U_x and D_x because both E_x and E'_x are known. Equations similar to Eqs. (3.81) and (3.82) also determine U_y and D_y, the related y-fields. The z-fields come from

$$\ell U_x + mU_y + q_i U_z = 0 \tag{3.83}$$

$$\ell D_x + mD_y - q_i D_z = 0 \tag{3.84}$$

Hence, for each vector-solution $\vec{e}$, the upgoing and the downgoing electric wave-field vectors are obtained

$$\vec{e}_1 \longrightarrow \vec{U}_1 \quad \text{and} \quad \vec{D}_1 \tag{3.85}$$

$$\vec{e}_{20} \longrightarrow \vec{U}_2 \quad \text{and} \quad \vec{D}_2 \tag{3.86}$$

The corresponding FORTRAN instructions are INS570 to INS581.

2. The penetrating mode solution. The penetrating electric wavefields are determined by Eqs. (3.9), (3.10) and (3.13). The corresponding FORTRAN statements are INS582/589. Observe the correspondence

$$-b \text{ of Eq. (3.10)} \longrightarrow \text{B1 of INS583}$$

3. Multiplying factors for obtaining the incident wave. As has been pointed out before, two independent solutions are required to obtain a given incident wave, in fact, any two independent solutions. Hence the penetrating mode previously defined as one of the solutions may be used. Now, if the incident wave is horizontally polarized with the electric field amplitude equal to one, then the geometry of the problem shows that (see Figure 1):

$$E_{xinc} = 1 \times C_A \tag{3.87}$$

$$E_{yinc} = -1 \times S_A \tag{3.88}$$

where

$$C_A = \cos\chi \tag{3.89}$$

$$S_A = \sin\chi \tag{3.90}$$

The right combination of upgoing # 1 and upgoing # p (p for penetrating) are established in order to get the above incident wave

$$E_{xinc} = C_A = a_1 U_{x1} + a_2 U_{xp} \tag{3.91}$$

$$E_{yinc} = -S_A = a_1 U_{y1} + a_2 U_{yp} \tag{3.92}$$

Hence, a_1 and a_2 are readily determined from Eqs. (3.91) and (3.92):

$$a_1 = - (S_A U_{xp} + C_A U_{yp})/\Delta \tag{3.93}$$

$$a_2 = (S_A U_{xl} + C_A U_{yl})/\Delta \tag{3.94}$$

where

$$\Delta = U_{xp} U_{yl} - U_{xl} U_{yp} \tag{3.95}$$

The above multiplying factors will be used later when the output is chosen to be the penetrating and the non-penetrating ionospheric wave-fields set up by a horizontally polarized wave with unit electric field. At each height $\vec{e}_l$ and $\vec{e}_p$ are replaced by

$$\vec{e}_l \longrightarrow a_1 \vec{e}_l \tag{3.96}$$

$$\vec{e}_p \longrightarrow a_2 \vec{e}_p \tag{3.97}$$

Eqs. (3.93) to (3.95) correspond to INS590 to INS592.

4. Polarization, transmission and reflection coefficients for the penetrating and non-penetrating modes. The square of the transmission coefficient is defined as the ratio between the power flow in the z-direction high in the ionosphere and the z-directed incident power flow.

The transmission coefficient for the #1 non-penetrating (np) mode is obviously zero.

On the other hand, the penetrating (p) mode yields to a purely up-going whistler wave at the top. For this mode the vertical component of the cycle average of the Poynting vector is

$$p_z = \frac{1}{2Z_o} \Re[E_x \cdot (Z_o H_y)^* - E_y \cdot (Z_o H_x)^*] \tag{3.98}$$

If the penetrating mode is normalized such that the p wave incident from below has an electric wave-field vector of unit amplitude, Eq. (3.98) can be manipulated to give

$$p_{zin} = \frac{q_i}{2Z_o} \tag{3.99}$$

Hence, the transmission coefficient for the p-mode will be given by

$$T_p^2 = \frac{1}{q_i} \Re[E_x(Z_o H_y)^* - E_y(Z_o H_x)^*] \tag{3.100}$$

The fields at the right-hand side of Eq. (3.100) are p-mode fields set up by one incident p-mode wave of unit electric field.

INS594 to INS598 perform the numerical calculation of T_p. Variable F is the normalizing factor for the incident p-mode (INS594).

On a similar basis the reflection coefficient compares the z-power flow in the reflected and in the incident waves

$$R^2 = \frac{p_{zdown}}{p_{zin}} \tag{3.101}$$

The reflection coefficient for the p-mode and for the np mode are computed by means of INS602/633 and INS604/605 respectively.

The polarization of the incident p-mode wave is defined here as the ratio between the electric field in the plane of incidence (plane $\vec{k}$, z-axis) and the electric field in the horizontal plane (plane x-y).

That is

$$\rho_p = \frac{E_{pi}}{E_{xy}} = \frac{\sin I\, U_{zp} - (S_A U_{xp} + C_A U_{yp}) \cos I}{C_A U_{xp} - S_A U_{yp}} \tag{3.102}$$

Equation (3.102) corresponds to INS599.

It is not necessary to compute the polarization of the incident np mode wave because it is known that

$$\vec{U}_1^* \cdot \vec{U}_p = 0 \tag{3.103}$$

Equation (3.103) says that the polarization of the incident np mode is obtained by interchanging the major and minor axes of the polarization ellipse determined by ρ_p, and reversing the direction of rotation.

5. Transmission coefficients at vertical and horizontal polarizations. If the incident wave has a polarization different than the polarization of the p-mode then it will excite both p and np waves. The transmission of the np mode is zero and is T_p for the p mode. Hence, for calculating transmission coefficient for any incident wave it is only necessary to calculate the amount of p mode excited by the wave. The transmission coefficients for horizontally and vertically polarized waves are computed at INS606/610.

6. Reflection coefficient matrix. The reflection coefficient matrix is another very important result that comes out from the program. The elements of $\tilde{R}$ are $_{\perp}R_{\perp}$, $_{\perp}R_{\parallel}$, $_{\parallel}R_{\parallel}$, and $_{\parallel}R_{\perp}$, such that

$$_{\perp}R_{\perp} = \left.\frac{E_{\perp down}}{E_{\perp inc}}\right|_{E_{\parallel inc}=0} \tag{3.104}$$

$$_{\perp}R_{\parallel} = \left.\frac{E_{\parallel down}}{E_{\perp inc}}\right|_{E_{\parallel inc}=0} \tag{3.105}$$

$$ {}_{\parallel}R_{\parallel} = \left.\frac{E_{\parallel down}}{E_{\parallel inc}}\right|_{E_{\perp inc}=0} \tag{3.106} $$

$$ {}_{\parallel}R_{\perp} = \left.\frac{E_{\perp down}}{E_{\parallel inc}}\right|_{E_{\perp inc}=0} \tag{3.107} $$

where the first symbol preceding R characterizes whether the incident electric field is perpendicular ($\perp$) or parallel ($\parallel$) to the plane of incidence. Similarly, the symbol which follows R characterizes the reflected electric field. In order to find the fields in Eqs. (3.104) to (3.107) any two independent solutions may be combined. For example, here the fields #1 and #2 that come out from the integration are directly combined. After some manipulation it is found that

$$ {}_{\perp}R_{\perp} = \frac{1}{\Delta}\left\{-(C_A D_{x1}-S_A D_{y1})(S_A U_{x2}+C_A U_{y2}) + (C_A D_{x2}-S_A D_{y2})(S_A U_{x1}+C_A U_{y1})\right\} \tag{3.108} $$

$$ {}_{\perp}R_{\parallel} = \frac{1}{\Delta \cos I}\left\{-(S_A U_{x2}+C_A U_{y2})(S_A D_{x1}+C_A D_{y1}) + (S_A D_{x2}+C_A D_{y2})(S_A U_{x1}+C_A U_{y1})\right\} \tag{3.109} $$

$$ {}_{\parallel}R_{\parallel} = \frac{1}{\Delta}\left\{-(C_A U_{x2}-S_A U_{y2})(S_A D_{x1}+C_A D_{y1}) + (S_A D_{x2}+C_A D_{y2})(C_A U_{x1}-S_A U_{y1})\right\} \tag{3.110} $$

$$ {}_{\parallel}R_{\perp} = \frac{\cos I}{\Delta}\left\{-(C_A D_{x1}-S_A D_{y1})(C_A U_{x2}-S_A U_{y2}) + (C_A D_{x2}-S_A D_{y2})(C_A U_{x1}-S_A U_{y1})\right\} \tag{3.111} $$

where

$$ \Delta = U_{y1}\, U_{x2} - U_{x1}\, U_{y2} \tag{3.112} $$

The FORTRAN instructions for computing the above reflection coefficients are INS614 to INS626.

7. Reconstruction of the ionospheric wave-fields. In order to reconstruct the ionospheric penetrating mode wave-fields it is necessary to use Eq. (3.24). Equation (3.24) must now be slightly changed due to the fact that the #1 solution has been constantly scaled down during the integration. Subroutine BRAIN shows that at the heights where scaling took place $\vec{e}_1$ was scaled down first followed by the orthogonalization of the second vector $\vec{e}_2$. Hence for obtaining the penetrating mode at a height k the equation is written

$$\vec{e}_p(k) = \vec{e}_{20}(k) + (b + \sum_{i=k+1}^{n} a_i) \cdot \vec{e}_1(k) \qquad (3.24)$$

if k corresponds to a height equal or less than the lowest value of HSCALE(NA). For the first height above HSCALE(NA) the p-vector will be obtained by the following equation

$$\vec{e}_p(k-1) = \vec{e}_{20}(k-1) + 10^{-3} \cdot (b + \sum_{i=k}^{n} a_i)\vec{e}_1(k-1) \qquad (3.113)$$

and next the value of $\vec{e}_1(k-1)$ is also changed

$$\vec{e}_1(k-1) \longrightarrow 10^{-3}\vec{e}_1(k-1) \qquad (3.114)$$

In order to understand the formation law for $\vec{e}_p$ one further step is shown:

$$\vec{e}_p(k-2) = \vec{e}_{20}(k-2) + 10^{-3}(b + \sum_{i=k}^{n} a_i)\vec{e}_1(k-2) + a_{k-1} \cdot \vec{e}_1(k-2) \qquad (3.115)$$

The last term in Eq. (3.115) comes from the orthogonalizing procedure at the first height above HSCALE(NA). The above reconstruction procedure is then easily generalized for any height yielding to the block diagram of Figure 8 where the whole process is shown.

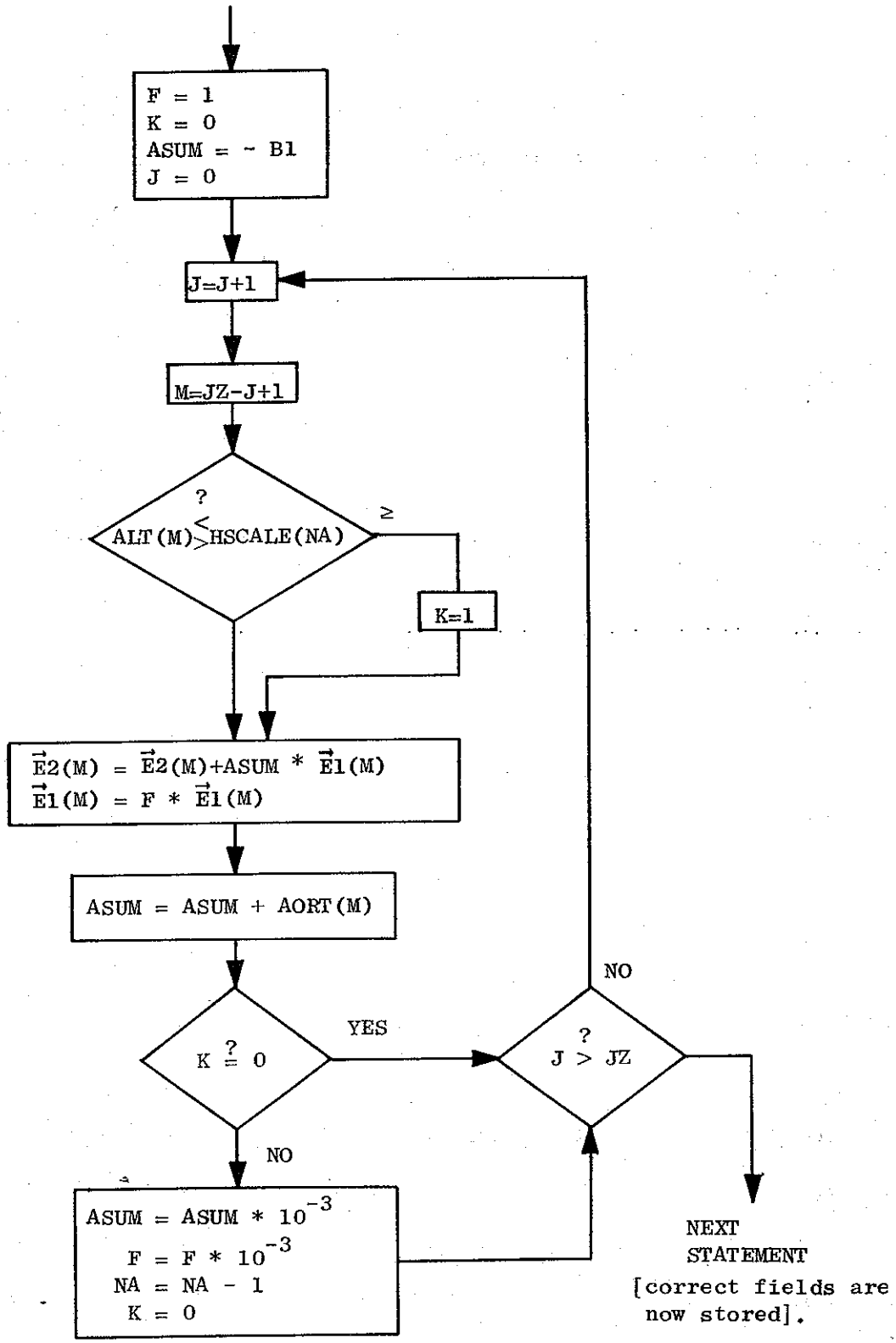

FIGURE 8. Reconstruction of the ionospheric wave-fields.

The FORTRAN instructions corresponding to the reconstruction of the wave-fields are INS641 to INS657.

8. Ionospheric wave-fields set up by a horizontal electric field of unit amplitude - relative errors. At this point the correct solutions $\vec{e}_1$ and $\vec{e}_p$ are known and stored. If an incident horizontally polarized wave is incident upon the lower boundary of the ionosphere it will excite both penetrating and non-penetrating mode waves. Moreover, it is also known that if this incident wave has electric field of unit amplitude it will excite p and np-modes in the following amounts:

$a_1\vec{U}_1$ upgoing np-mode, electric field

$a_2\vec{U}_p$ upgoing p-mode, electric field

The above results come from Eqs. (3.93) to (3.97). Hence, the ionospheric wave-fields excited by the given incident wave will be

$$\text{p mode:} \quad \vec{e}_p \longrightarrow a_2\vec{e}_p \tag{3.116}$$

$$\text{np mode:} \quad \vec{e}_1 \longrightarrow a_1\vec{e}_1 \tag{3.117}$$

Equations (3.116) and (3.117) determine the wave-fields E_x, E_y, $Z_o H_x$ and $Z_o H_y$ for each solution at any height. Fields $Z_o H_x$ and E_z are not printed out in this program but can be obtained immediately from Maxwell's equations plus the knowledge of $\vec{e}$ and $d\vec{e}/dz$.

The relative errors committed at each step of integration are known and stored. Hence they are available for printing at any time.

The ionospheric wave-fields set up by a horizontally polarized wave of unit amplitude are calculated from INS679 to INS698. If KFORM = 4 only the envelope of the x-electric/magnetic fields for the p and the np modes are printed out together with the relative errors

as shown in the listing of the program corresponding to INS699 to INS710.

H. GENERAL CHARACTERISTICS OF THE FULL-WAVE PROGRAM

The FULL-WAVE program has been tested and used regularly in the IBM/360 computer of the Stanford University Computer Center. Some tests corresponded to checking results presented by Pitteway [1965], Piggott et. al. [1965], and Deeks [1966]. Another useful test is the total z-power flow behavior with height. If the collision frequency is zero the power flow in the z direction must be constant due to the continuity of horizontal fields. In a typical case p_z maintained constant within 6 decimal figures.

General Characteristics

- All the variables are double-precision with the exception of the relative errors committed at each step.
- 712 FORTRAN IV-H statements
- 1 main program and 4 subroutines
- Object code: 60224 bytes
- Total array area: 150184 bytes
- Total length: 210408 bytes
- Typical run time: 10/25 seconds for each input set.

In the following pages a listing of the computer program is presented.

```
      C
      C
      C       *.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.
      C
      C       -   MAIN PROGRAM   -                .FULLWAVE.        R.R.SCARABUCCI
      C
      C       *.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.
      C
  1           REAL*8 DEXP.DLOG.DCOS.DSIN.DSQRT.DMAX1.DMIN1.DABS.CDABS
  2           COMPLEX*16 CDEXP,CDLOG,CDSQRT,DCMPLX,DCONJG
  3           REAL*8 YY,YAY,YAZ,AL,AM,AN,C,CA,CD,S,SA,SD,FREQ,ALT(500)
  4           COMMON/BRAZIL/YY.YAY.YAZ.AL.AM.AN.C.CA.CD.S.SA.SD,FREQ,ALT
  5           COMPLEX*16 AI
  6           COMMON/IMAG/AI
  7           REAL*8 Y1(8),Y2(8),DY1(8),DY2(8),AUX1(15,8),AUX2(15,8),HEIGHT(20),
            1   HSTART,HEND,STEP,Z,DELT1,DELT2
  8           COMMON/USA/Y1.Y2.DY1.DY2.AUX1.AUX2.HEIGHT.HSTART.HEND.STEP.Z.DELT1
            1   ,DELT2
  9           REAL*8 ZAXIS(100),DENS(100),COLFRE(100)
 10           COMMON/ITALY/ZAXIS.DENS.COLFRE
 11           REAL*8 V(64),FAT,FAT1,HLASTX
 12           COMMON/FRANCE/V,FAT,FAT1,HLASTX
 13           COMPLEX*16 E1(4,500),E2(4,500),DE1(4,500),DE2(4,500),AORT(500)
 14           COMMON/RUSS/E1,E2,DE1,DE2,AORT
 15           REAL*8 FH,ANGI,AZIM,DIP
 16           COMMON/CANADA/FH,ANGI,AZIM,DIP,JZ,NA,NMAT,KFORM,KSTOP
 17           COMPLEX*16 T(16)
 18           COMMON/MEXICO/T
      C
 19           REAL*8 XX,XY2,QR(4),QI(4),QQ
 20           COMPLEX*16 U.UMX.XA.ALPHA.BETA.GAMA.DELTA.EPSY.P.QO.R,AA,BB,SC,
            1   A1,A2,AT,X1,ABC,DD,EE,Q(4),B1,B2,B3,B4,B5,T11,T12,
            2   T13,T14,T21,T22,T23,T41,T42,A3,A4,A5,A6,FAC,RR.DENOM
 21           EXTERNAL BRAIN,HAMING,MATRIX,OUTPUT
      C
      C              -   F   O   R   M   A   T   S   -
      C
 22     100   FORMAT (2I5)
 23     200   FORMAT (F10.2.2D10.2)
 24     300   FORMAT (2D10.3,3F10.2)
 25     400   FORMAT ('1',' FREQ =',D20.5/'    FH =',D20.5/'   ANGI =',F20.2/
            1 '   AZIM =',F20.2/'    DIP =',F20.2///)
 26     500   FORMAT (4D10.3,I5)
 27     600   FORMAT (' ',' Q1R =',D26.15,'  Q1I =',D26.15/
            1              '  Q2R =',D26.15,'  Q2I =',D26.15/
            2              '  Q3R =',D26.15,'  Q3I =',D26.15/
            3              '  Q4R =',D26.15,'  Q4I =',D26.15///)
 28     700   FORMAT (F10.3)
      C
      C
 29           AI = DCMPLX(0.0D0, 1.0D0)
      C
 30           READ 100, NPOINT, NSTEP
 31           READ (5,200) (ZAXIS(J),DENS(J),COLFRE(J), J=1,NPOINT)
 32           READ (5,700) (HEIGHT(J), J=1,NSTEP)
      C
      C       STORE THE LOGARITM OF DENS AND COLFRE
      C
```

```
33          DO 5 I=1,NPOINT
34          DENS(I) = DLOG(DENS(I))
35        5 COLFRE(I) = DLOG(COLFRE(I))
      C
36       10 READ 300, FREQ,FH,ANGI,AZIM,DIP
37          IF (FREQ.EQ.0) GO TO 55
38          PRINT 400, FREQ,FH,ANGI,AZIM,DIP
      C
39          FAT = 2.0D0*3.14159265358979300*FREQ
40          FAT1 = 8.061D07/(FREQ*FREQ)
41          XX = COLFRE(NPOINT)
42          U = 1.0D0 - AI*DEXP(XX)/FAT
43          XX = DENS(NPOINT)
44          XX = FAT1*DEXP(XX)
45          READ 500, HSTART,HEND,STEP,HLASTX,KFORM
      C
46          ANGI = ANGI*1.7453292519943300-02
47          AZIM = AZIM*1.7453292519943300-02
48          DIP = 1.80D02 - DIP
49          DIP =   DIP*1.7453292519943300-02
50          C = DCOS(ANGI)
51          S = DSIN(ANGI)
52          SA = DSIN(AZIM)
53          CA = DCOS(AZIM)
54          CD = DCOS(DIP)
55          SD = DSIN(DIP)
56          AL = S*SA
57          AM = S*CA
58          AN = C
      C
59          YY = FH/FREQ
60          YAY = YY*CD
61          YAZ = YY*SD
62          UMX = U - XX
63          XA = XX - U*C*C
64          XY2 = XX*YY*YY
      C
      C     *.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*..
      C
      C     START COMPUTATION OF EIGENVALUES AT ALT(NPOINT) -  BOOKER QUARTIC
      C
      C
65          ALPHA = U*U*UMX + YY*YY*(XX*SD*SD - U)
66          BETA = 2.0D0*XY2*AM*SD*CD
67          GAMA = 2.0D0*XA*(U*UMX - YY*YY) + XY2*(1.0D0-(C*SD)**2+(AM*CD)**2)
68          DELTA = -2.0D0*XY2*CD*SD*C*C*AM
69          EPSY = XA*(XA*UMX + YY*YY*C*C) - XY2*(C*CD*AM)**2
      C
70          BETA = BETA/ALPHA
71          GAMA = GAMA/ALPHA
72          DELTA = DELTA/ALPHA
73          EPSY = EPSY/ALPHA
74          ALPHA = 0.75D0*BETA*BETA - 2.0D0*GAMA
      C
75          IF (CDABS(BETA).EQ.0.AND.CDABS(DELTA).EQ.0) GO TO 20
      C
      C     THE RESOLVENT CUBIC
      C
76          P = -GAMA
77          QQ = BETA*DELTA - 4.0D0*EPSY
```

```
      R = -BETA*BETA*EPSY + 4.0DO*GAMA*EPSY - DELTA*DELTA
C
      AA = QC - 3.3333333333333333D-C1*P*P
      BB = (P*P*P/1.35D01) - 3.3333333333333333D-01*P*QO + R
C
      SQ = CDSQRT(0.25DO*BB*BB + (AA*AA*AA/2.70D01))
      A1 = -0.5DO*BB + SQ
      A2 = -0.5DO*BB - SQ
      AT = 3.3333333333333333D-01*CDLOG(A1)
      AT = CDEXP(AT)
      X1 = AT - 3.3333333333333333D-01*((AA/AT) + P)
C
C     THE SOLUTION FOR Q
C
      RR = 0.25DO*BETA*BETA - GAMA + X1
      IF (CDABS(RR).LT.1.0D-70) GO TO 21
      RR = CDSQRT(RR)
      ABC = (BETA*GAMA - 2.0DO*DELTA - 0.25DO*BETA*BETA*BETA)/RR
      DD = CDSQRT(ALPHA - RR*RR + ABC)
      EE = CDSQRT(ALPHA - RR*RR - ABC)
      GO TO 22
   20 RR = 0.0DO + AI*0.0DO
      X1 = GAMA
   21 ABC = CDSQRT(X1*X1 - 4.0DO*EPSY)*2.0DO
      DD = CDSQRT(ALPHA + ABC)
      EE = CDSQRT(ALPHA - ABC)
   22 IF (CDABS(DD).LT.1.0D-70) DD=0.0DO + AI*0.0DO
      IF (CDABS(EE).LT.1.0D-70)  EE= 0.0DO + AI*0.0DO
      A1 = -0.25DO*BETA + 0.5DO*RR
      A2 = -0.25DO*BETA - 0.5DO*RR
C
C     THE FOUR ROOTS
C
      Q(1) = A1 + 0.5DO*DD
      Q(2) = A1 - 0.5DO*DD
      Q(3) = A2 + 0.5DO*EE
      Q(4) = A2 - 0.5DO*EE
C
      DO 23 I=1,4
      A1 = DCONJG(Q(I))
      QR(I) = 0.5DO*(Q(I) + A1)
   23 QI(I) = -AI*0.5DO*(Q(I) - A1)
C
      PRINT 600, QR(1),QI(1),QR(2),QI(2),QR(3),QI(3),QR(4),QI(4)
C
C     *.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.
C
C     CHOOSING EIGENVALUES FOR UPGOING WAVES AT THE TOP
C
C
      J = 0
      DO 24 I=1,4
      IF (QI(I).GT.0) GO TO 24
      J = J + 1
      QR(J) = QR(I)
      QI(J) = QI(I)
   24 CONTINUE
C
      QQ = QI(1) - QI(2)
      IF (QQ) 25,25,26
```

```
   25  Q(1) = QR(1) + AI*QI(1)
       Q(2) = QR(2) + AI*QI(2)
       GO TO 27
   26  Q(1) = QR(2) + AI*QI(2)
       Q(2) = QR(1) + AI*QI(1)
C
C      Q(1) CORRESPONDS TO THE EVANESCENT WAVE EIGENVECTOR AT THE TOP
C      Q(2) CORRESPONDS TO THE TRAVELLING WAVE EIGENVECTOR AT THE TOP
C
C      *.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.
C
C
C      COMPUTATION OF THE EIGENVECTORS AT THE TOP
C
   27  ALPHA = U*(U*U - YY*YY) - XX*(U*U - YAZ*YAZ)
       B1 = XX*U*YAY/ALPHA
       B2 = XX*YAY*YAZ/ALPHA
       B3 = U*(U*U - YY*YY)/ALPHA
       B4 = XX*YAZ*UMX/ALPHA
       B5 = XX*U*UMX/ALPHA
       T11 = -AI*AL*B1
       T12 = AL*B2
       T13 = AL*AM*B3
       T14 = 1.0D0 - B3*AL*AL
       T21 = AI*AM*B1
       T22 = -AM*B2
       T23 = 1.0D0 - B3*AM*AM
       T41 = 1.0D0 - AM*AM - B5
       T42 = -AL*AM - AI*B4
C
       DO 35 I=1,2
       A6 = Q(I) + T11
       A1 = (Q(I) - T11)*A6 - T14*T41
       A2 = (Q(I) - T22)*A6 - T13*T42
       A3 = T12*A6 + T14*T42
       A4 = T21*A6 + T13*T41
       A5 = T23*A6 - T13*T21
       A6 = T13*A6 - T14*T21
C
C      STARTING EIGENVECTORS
C
       Y2(1) = 1.0D0
       Y2(5) = 0.0D0
       DENOM = A3*A5 + A2*A6
       FAC = (A1*A2 - A3*A4)/DENOM
       Y2(3) = 0.5D0*(FAC + DCONJG(FAC))
       Y2(7) = -0.5D0*AI*(FAC - DCONJG(FAC))
       FAC = (A1*A5 + A4*A6)/DENOM
       Y2(2) = 0.5D0*(FAC + DCONJG(FAC))
       Y2(6) = -0.5D0*AI*(FAC - DCONJG(FAC))
       IF (AM.EQ.0) GO TO 30
       FAC = Q(I)*(1.0D0 + (AL/AM)*(Y2(2) + AI*Y2(6))) -
      1   (AL/AM)*(Y2(3) + AI*Y2(7))
       Y2(4) = 0.5D0*(FAC + DCONJG(FAC))
       Y2(8) = -0.5D0*AI*(FAC - DCONJG(FAC))
       GO TO 31
   30  FAC = ALPHA - U*AL*AL*(U*U - YY*YY)
       FAC=(ALPHA*Q(I)+AI*AL*XX*U*YAY-AL*XX*YAY*YAZ*(Y2(2)+AI*Y2(6)))/FAC
       Y2(4) = 0.5D0*(FAC + DCONJG(FAC))
       Y2(8) = -0.5D0*AI*(FAC - DCONJG(FAC))
```

```
 31  IF (I.EQ.2) GO TO 35
     DO 32 J=1,8
 32  Y1(J) = Y2(J)*1.0D03
 35  CONTINUE
C
C    EIGENVECTOR CORRESPONDING TO TRAVELLING WAVE STORED AT Y2(J)
C
C    *.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.
C
C    PREPARING PARAMETERS FOR STARTING INTEGRATION PROCEDURE
C
     JZ = 0
     NA = 0
     NMAT = NPOINT
     CALL HAMING
     GO TO 10
 55  RETURN
     END
```

```
178           SUBROUTINE HAMING
      C\
      C
179           REAL*8 DEXP,DLOG,DCOS,DSIN,DSQRT,DMAX1,DMIN1,DABS,CDABS
180           COMPLEX*16 CDEXP,CDLOG,CDSQRT,DCMPLX,DCONJG
181           REAL*8 Y1(8),Y2(8),DY1(8),DY2(8),AUX1(15,8),AUX2(15,8),HEIGHT(20),
             1   HSTART,HEND,STEP,Z,DELT1,DELT2
182           COMMON/USA/Y1,Y2,DY1,DY2,AUX1,AUX2,HEIGHT,HSTART,HEND,STEP,Z,DELT1
             1   ,DELT2
183           REAL*8 V(64),FAT,FAT1,HLASTX
184           COMMON/FRANCE/V,FAT,FAT1,HLASTX
185           REAL*8 FH,ANGI,AZIM,DIP
186           COMMON/CANADA/FH,ANGI,AZIM,DIP,JZ,NA,NMAT,KFORM,KSTOP
187           REAL*8 HS1,HS2,H,X1,X2,ZO
      C
      C
188    600    FORMAT (' ','  IMPOSSIBLE START INTEGRATING, HSTART=HEND')
189    700    FORMAT (' ','  IMPOSSIBLE START INTEGRATING, STEP HAS WRONG SIGN')
      C
      C
190           ISTEP = 1
      C
191           GO TO 10
      C
      C
192       1   CALL MATRIX
193           IF (KSTOP.EQ.1) GO TO 500
194           DO 3 M=1,8
195           LL = M-8
196           HS1 = 0.0D0
197           HS2 = 0.0D0
198           DO 2 L=1,8
199           LL = LL+8
200           HS1 = HS1 + V(LL)*Y1(L)
201       2   HS2 = HS2 + V(LL)*Y2(L)
202           DY1(M) = HS1
203       3   DY2(M) = HS2
204           GO TO (35,210,220,230,55,75,90,110,335), ISW2
      C
      C
205      10   N = 1
206           Z = HSTART
207           H = STEP
208           KSTOP = 0
209           DO 15 I=1,8
210           AUX1(1,I) = Y1(I)
211           AUX2(1,I) = Y2(I)
212           AUX1(15,I) = 0.0D0
213      15   AUX2(15,I) = 0.0D0
214           IF (H*(HEND - Z)) 25,20,30
215      20   PRINT 600
216           GO TO 500
217      25   PRINT 700
218           GO TO 500
219      30   ISW2 = 1
220           GO TO 1
221      35   DO 40 I=1,8
222           AUX1(8,I) = DY1(I)
223      40   AUX2(8,I) = DY2(I)
      C
```

```
      C     COMPUTATION OF AUX(2,I)
      C
224           ISW1 = 1
225           GO TO 200
226        45 Z = Z + H
227           DO 50 I=1,8
228           AUX1(2,I) = Y1(I)
229        50 AUX2(2,I) = Y2(I)
      C
230           ISW2 = 5
231           GO TO 1
232        55 DO 60 I=1,8
233           AUX1(9,I) = DY1(I)
234        60 AUX2(9,I) = DY2(I)
235           N = 2
236           ISW1 = 2
237           GO TO 200
      C
238        65 Z = Z + H
239           DO 70 I=1,8
240           AUX1(3,I) = Y1(I)
241        70 AUX2(3,I) = Y2(I)
242           ISW2 = 6
243           GO TO 1
244        75 DO 80 I=1,8
245           AUX1(10,I) = DY1(I)
246        80 AUX2(10,I) = DY2(I)
247           N = 3
248           ISW1 = 3
249           GO TO 200
      C
250        85 Z = Z + H
251           DO 87 I=1,8
252           AUX1(4,I) = Y1(I)
253        87 AUX2(4,I) = Y2(I)
254           ISW2 = 7
255           GO TO 1
256        90 DO 95 I=1,8
257           AUX1(11,I) = DY1(I)
258        95 AUX2(11,I) = DY2(I)
      C
      C     FOUR ORDER INTERPOLATION FOR REFINING THE FOUR STARTING POINTS
      C     GIVEN BY THE RUNGE-KUTTA METHOD.
      C
259           N = 1
260           Z = HSTART
261           DO 100 I=1,8
262           Y1(I) = AUX1(1,I) + H*(0.375D0*AUX1(8,I) + 7.916666666666670-01*
             1    AUX1(9,I) - 2.0833333333333330-01*AUX1(10,I) +
             2    4.166666666666667D-02*DY1(I))
263    100    Y2(I) = AUX2(1,I) + H*(0.375D0*AUX2(8,I) + 7.916666666666670-01*
             1    AUX2(9,I) - 2.0833333333333330-01*AUX2(10,I) +
             2    4.166666666666667D-02*DY2(I))
264
265    105    Z = Z + H
266           N = N + 1
267           ISW2 = 8
268           GO TO 1
      C
269    110    IF (N - 4) 115,295,295
```

```
      115  DO 120 I=1,8
           AUX1(N,I) = Y1(I)
           AUX2(N,I) = Y2(I)
           AUX1(N+7,I) = DY1(I)
      120  AUX2(N+7,I) = DY2(I)
           IF (N - 3) 125,135,295
C
      125  DO 130 I=1,8
           X1 = 4.0D0*AUX1(9,I)
           X2 = 4.0D0*AUX2(9,I)
           Y1(I)=AUX1(1,I)+3.3333333333333D-01*H*(AUX1(8,I)+X1+AUX1(10,I))
      130  Y2(I)=AUX2(1,I)+3.3333333333333D-01*H*(AUX2(8,I)+X2+AUX2(10,I))
           GO TO 105
C
      135  DO 140 I=1,8
           X1 = 3.0D0*(AUX1(9,I) + AUX1(10,I))
           X2 = 3.0D0*(AUX2(9,I) + AUX2(10,I))
           Y1(I) = AUX1(1,I) + 0.375D0*H*(AUX1(8,I) + X1 + AUX1(11,I))
      140  Y2(I) = AUX2(1,I) + 0.375D0*H*(AUX2(8,I) + X2 + AUX2(11,I))
           GO TO 105
C
C     .*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*
C
C     RUNGE-KUTTA METHOD FOR STARTING NOT SELF-STARTING PREDICTOR
C     CORRECTOR METHOD
C
C
      200  Z0 = Z
           DO 205 I=1,8
           X1 = H*AUX1(N+7,I)
           X2 = H*AUX2(N+7,I)
           AUX1(5,I) = X1
           AUX2(5,I) = X2
           Y1(I) = AUX1(N,I) + 0.4D0*X1
      205  Y2(I) = AUX2(N,I) + 0.4D0*X2
C
           Z = Z0 + 0.4D0*H
           ISW2 = 2
           GO TO 1
      210  DO 215 I=1,8
           X1 = H*DY1(I)
           X2 = H*DY2(I)
           AUX1(6,I) = X1
           AUX2(6,I) = X2
           Y1(I) = AUX1(N,I) + 2.969776092477536D-01*AUX1(5,I) +
          1   1.587596449710358D-01*X1
      215  Y2(I) = AUX2(N,I) + 2.969776092477536D-01*AUX2(5,I) +
          1   1.587596449710358D-01*X2
C
           Z = Z0 + 4.557372542187894D-01*H
           ISW2 = 3
           GO TO 1
      220  DO 225 I=1,8
           X1 = H*DY1(I)
           X2 = H*DY2(I)
           AUX1(7,I) = X1
           AUX2(7,I) = X2
           Y1(I) = AUX1(N,I) + 2.181003882259205D-01*AUX1(5,I) -
          1   3.050965148692931D0*AUX1(6,I) + 3.832864760467010D0*X1
      225  Y2(I) = AUX2(N,I) + 2.181003882259205D-01*AUX2(5,I) -
```

```
     1   3.050965148692931D0*AUX2(6,I) + 3.832864760467010D0*X2
C
         Z = Z0 + H
         ISW2 = 4
         GO TO 1
  230    DO 235 I=1,8
         Y1(I) = AUX1(N,I) + 1.747602822626904D-01*AUX1(5,I) -
     1    5.514806628787329D-01*AUX1(6,I) + 1.205535599396523D0*AUX1(7,I)
     2    + 1.711847812195190D-01*H*DY1(I)
  235    Y2(I) = AUX2(N,I) + 1.747602822626904D-01*AUX2(5,I) -
     1    5.514806628787329D-01*AUX2(6,I) + 1.205535599396523D0*AUX2(7,I)
     2    + 1.711847812195190D-01*H*DY2(I)
         Z = Z0
         GO TC (45,65,85), ISW1
C
C
C        *.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.
C
C        HAMMING'S MODIFIED PREDICTOR-CORRECTOR METHOD
C
C
  295    NORT = - 1
  300    IF (N - 8) 315,305,315
C
C        N = 8 CAUSES THE ROWS OF AUX TO CHANGE THEIR STORAGE LOCATIONS
C
  305    DO 310 N=2,7
         DO 310 I=1,8
         AUX1(N-1,I) = AUX1(N,I)
         AUX2(N-1,I) = AUX2(N,I)
         AUX1(N+6,I) = AUX1(N+7,I)
  310    AUX2(N+6,I) = AUX2(N+7,I)
         N = 7
  315    N = N + 1
C
C        COMPUTATION OF NEXT VECTOR Y
C
         DO 320 I=1,8
         AUX1(N-1,I) = Y1(I)
         AUX2(N-1,I) = Y2(I)
         AUX1(N+6,I) = DY1(I)
  320    AUX2(N+6,I) = DY2(I)
         Z = Z + H
         DO 330 I=1,8
         X1 = AUX1(N-4,I) + 1.333333333333333D0*H*(AUX1(N+6,I)+AUX1(N+6,I)-
     1     AUX1(N+5,I) + AUX1(N+4,I) + AUX1(N+4,I))
         X2 = AUX2(N-4,I) + 1.333333333333333D0*H*(AUX2(N+6,I)+AUX2(N+6,I)-
     1     AUX2(N+5,I) + AUX2(N+4,I) + AUX2(N+4,I))
         Y1(I) = X1 - 9.256198347107438D-01*AUX1(15,I)
         Y2(I) = X2 - 9.256198347107438D-01*AUX2(15,I)
         AUX1(15,I) = X1
  330    AUX2(15,I) = X2
C
C        PREDICTOR IS NOW GENERATED IN ROW 15 OF AUX.  MODIFIED PREDICTOR
C        IS GENERATED IN Y.  X1 AND X2 ARE AUXILIARY STORAGE.
C
         ISW2 = 9
         GO TO 1
C
C        DERIVATIVE OF MODIFIED PREDICTOR IS GENERATED IN DY
```

```
      C
  349  335  DO 340 I=1,8
  350       X1 = 0.125D0*(9.0D0*AUX1(N-1,I) - AUX1(N-3,I) + 3.0D0*H*(DY1(I) +
           1    AUX1(N+6,I) + AUX1(N+6,I) - AUX1(N+5,I)))
  351       X2 = 0.125D0*(9.0D0*AUX2(N-1,I) - AUX2(N-3,I) + 3.0D0*H*(DY2(I) +
           1    AUX2(N+6,I) + AUX2(N+6,I) - AUX2(N+5,I)))
  352       AUX1(15,I) = AUX1(15,I) - X1
  353       AUX2(15,I) = AUX2(15,I) - X2
  354       Y1(I) = X1 + 7.438016528925620D-02*AUX1(15,I)
  355  340  Y2(I) = X2 + 7.438016528925620D-02*AUX2(15,I)
  356       DELT1 = 0.0D0
  357       DELT2 = 0.0D0
  358       DO 345 I=1,8
  359       DELT1 = DELT1 + 0.125D0*DABS(AUX1(15,I))
  360  345  DELT2 = DELT2 + 0.125D0*DABS(AUX2(15,I))
  361       DO 350 M = 1,8
  362       LL = M - 8
  363       HS1 = 0.0D0
  364       HS2 = 0.0D0
  365       DO 349 L = 1,8
  366       LL = LL + 8
  367       HS1 = HS1 + V(LL)*Y1(L)
  368  349  HS2 = HS2 + V(LL)*Y2(L)
  369       DY1(M) = HS1
  370  350  DY2(M) = HS2
  371       NORT = NORT + 1
  372       IF (NORT.NE.4) GO TO 360
  373       CALL BRAIN
  374       NORT = 0
  375  360  X1 = Z - HEIGHT(ISTEP)
  376       IF (DABS(X1) - 1.0D-06) 365,365,300
      C
      C     H WILL BE DOUBLED
      C
  377  365  Z = HEIGHT(ISTEP)
  378       H = H + H
  379       DO 370 I=1,8
  380       AUX1(7,I) = AUX1(6,I)
  381       AUX2(7,I) = AUX2(6,I)
  382       AUX1(6,I) = AUX1(4,I)
  383       AUX2(6,I) = AUX2(4,I)
  384       AUX1(5,I) = AUX1(2,I)
  385       AUX2(5,I) = AUX2(2,I)
  386       AUX1(14,I) = AUX1(13,I)
  387       AUX2(14,I) = AUX2(13,I)
  388       AUX1(13,I) = AUX1(11,I)
  389       AUX2(13,I) = AUX2(11,I)
  390       AUX1(12,I) = AUX1(9,I)
  391       AUX2(12,I) = AUX2(9,I)
  392       X1 = AUX1(14,I) + AUX1(13,I)
  393       X2 = AUX2(14,I) + AUX2(13,I)
  394       X1 = X1 + X1 + X1
  395       X2 = X2 + X2 + X2
  396       AUX1(15,I) = 8.962962962962963D0*(Y1(I) - AUX1(5,I)) -
           1    3.361111111111111D0*H*(DY1(I) + X1 + AUX1(12,I))
  397  370  AUX2(15,I) = 8.962962962962963D0*(Y2(I) - AUX2(5,I)) -
           1    3.361111111111111D0*H*(DY2(I) + X2 + AUX2(12,I))
  398       ISTEP = ISTEP + 1
  399       GO TO 300
  400  500  RETURN
  401       END
```

```
402           SUBROUTINE BRAIN
      C
      C
403           REAL*8 DEXP,DLOG,DCOS,DSIN,DSQRT,DMAX1,DMIN1,DABS,CDABS
404           COMPLEX*16 CDEXP,CDLOG,CDSQRT,DCMPLX,DCONJG
      C
405           REAL*8 YY,YAY,YAZ,AL,AM,AN,C,CA,CD,S,SA,SD,FREQ,ALT(500)
406           COMMON/BRAZIL/YY,YAY,YAZ,AL,AM,AN,C,CA,CD,S,SA,SD,FREQ,ALT
407           COMPLEX*16 AI
408           COMMON/IMAG/AI
409           REAL*8 Y1(8),Y2(8),DY1(8),DY2(8),AUX1(15,8),AUX2(15,8),HEIGHT(20),
             1    HSTART,HEND,STEP,Z,DELT1,DELT2
410           COMMON/USA/Y1,Y2,DY1,DY2,AUX1,AUX2,HEIGHT,HSTART,HEND,STEP,Z,DELT1
             1    ,DELT2
411           COMPLEX*16 E1(4,500),E2(4,500),DE1(4,500),DE2(4,500),AORT(500)
412           COMMON/RUSS/E1,E2,DE1,DE2,AORT
413           REAL*8 FH,ANGI,AZIM,DIP
414           COMMON/CANADA/FH,ANGI,AZIM,DIP,JZ,NA,NMAT,KFORM,KSTOP
415           DIMENSION ERROR1(500),ERROR2(500)
416           REAL*8 TEST1,TEST2,HSCALE(50),DENOM
417           COMPLEX*16 ANUM,E1C,HELP,HELPC
418           COMMON/OUTP/HSCALE,ERROR1,ERROR2
      C
      C
      C
      C
419           TEST1 = 0.50D0*DSQRT(Y1(1)**2 + Y1(5)**2 + Y1(3)**2 + Y1(7)**2)
420           IF (TEST1.LT.1.0D03) GO TO 5
421           DO 1 I=1,8
422           Y1(I) = Y1(I)*1.0D-03
423           DY1(I) = DY1(I)*1.0D-03
424           DO 1 J=1,15
425         1 AUX1(J,I) = AUX1(J,I)*1.0D-03
426           NA = NA + 1
427           HSCALE(NA) = Z
      C
      C
428         5 JZ = JZ + 1
429           ALT(JZ) = Z
430           IF (KFORM.NE.4) GO TO 7
431           TEST2 = 0.50D0*DSQRT(Y2(1)**2 + Y2(5)**2 + Y2(3)**2 + Y2(7)**2)
432           ERROR1(JZ) = DELT1/TEST1
433           ERROR2(JZ) = DELT2/TEST2
434         7 ANUM = 0.0D0 + AI*0.0D0
435           DENOM = C.CDC
436           DO 10 I=1,4
437           E1(I,JZ) = Y1(I) + AI*Y1(I+4)
438           DE1(I,JZ) = DY1(I) + AI*DY1(I+4)
439           E2(I,JZ) = Y2(I) + AI*Y2(I+4)
440           DE2(I,JZ) = DY2(I) + AI*DY2(I+4)
441           E1C = Y1(I) - AI*Y1(I+4)
442           ANUM = ANUM + E1C*E2(I,JZ)
443        10 DENOM = DENOM + Y1(I)*Y1(I) + Y1(I+4)*Y1(I+4)
444           AORT(JZ) = - ANUM/DENOM
      C
      C
445           DO 15 I=1,4
446           HELP = E2(I,JZ) + AORT(JZ)*E1(I,JZ)
447           HELPC = DCONJG(HELP)
448           Y2(I) = 0.5D0*(HELP + HELPC)
```

```
449           Y2(I+4) = -0.5D0*AI*(HELP - HELPC)
450           E2(I,JZ) = HELP
451           HELP = DE2(I,JZ) + ACRT(JZ)*DE1(I,JZ)
452           DE2(I,JZ) = HELP
453           HELPC = DCONJG(HELP)
454           DY2(I) = 0.5D0*(HELP + HELPC)
455        15 DY2(I+4) = -0.5D0*AI*(HELP - HELPC)
      C
      C     FIELDS ARE ORTHOGONALIZED AND STORED
      C
456        25 RETURN
457           END
```

```
458           SUBRCUTINE MATRIX
459
      C
460           REAL*8 DEXP,DLCG,DCOS,DSIN,DSQRT,DMAX1,DMIN1,DABS,CDABS
461           COMPLEX*16 CDEXP,CDLOG,CDSQRT,DCMPLX,DCONJG
462           REAL*8 Y1(8),Y2(8),DY1(8),DY2(8),AUX1(15,8),AUX2(15,8),HEIGHT(20),
             1    HSTART,HEND,STEP,Z,DELT1,DELT2
463           COMMON/USA/Y1,Y2,DY1,DY2,AUX1,AUX2,HEIGHT,HSTART,HEND,STEP,Z,DELT1
             1    ,DELT2
464           REAL*8 YY,YAY,YAZ,AL,AM,AN,C,CA,CD,S,SA,SD,FREQ,ALT(500)
465           COMMON/BRAZIL/YY,YAY,YAZ,AL,AM,AN,C,CA,CD,S,SA,SD,FREQ,ALT
466           COMPLEX*16 AI
467           COMMON/IMAG/AI
468           REAL*8 ZAXIS(100),DENS(100),COLFRE(100)
469           COMMON/ITALY/ZAXIS,DENS,COLFRE
470           REAL*8 V(64),FAT,FAT1,HLASTX
471           COMMON/FRANCE/V,FAT,FAT1,HLASTX
472           REAL*8 FH,ANGI,AZIM,DIP
473           COMMON/CANADA/FH,ANGI,AZIM,DIP,JZ,NA,NMAT,KFORM,KSTOP
474           COMPLEX*16 T(16)
475           COMMON/MEXICO/T
476           REAL*8 ELDENS,FCOL,XA,AA,ZA
477           COMPLEX*16 UA,ALPHA,B1,B2,B3,B4,B5
      C
      C
      C
      C
478           IF (Z.GT.HLASTX) GO TO 1
479           B1 = 0.0D0 + AI*0.0D0
480           B2 = 0.0D0 + AI*0.0D0
481           B4 = 0.0D0 + AI*0.0D0
482           B5 = 0.0D0 + AI*0.0D0
483           B3 = 1.0D0 + AI*0.0D0
484           ALPHA = 1.0D0 + AI*0.0D0
485           IF (Z.GT.HEND) GO TO 30
486           CALL OUTPUT
487           KSTOP = 1
488           RETURN
489         1 IF (Z - ZAXIS(NMAT)) 5,10,15
490         5 NMAT = NMAT - 1
491           GO TO 1
492        10 ELDENS = DEXP(DENS(NMAT))
493           FCOL = DEXP(COLFRE(NMAT))
494           GO TO 20
      C
495        15 AA = (ZAXIS(NMAT) - Z)/(ZAXIS(5) - ZAXIS(4))
496           ELDENS = DENS(NMAT) + AA*(DENS(NMAT-1) - DENS(NMAT))
497           FCOL = COLFRE(NMAT) + AA*(COLFRE(NMAT-1) - COLFRE(NMAT))
498           ELDENS = DEXP(ELDENS)
499           FCOL = DEXP(FCOL)
      C
500        20 XA = FAT1*ELDENS
501           ZA = FCOL/FAT
502           UA = 1.0D0 - AI*ZA
503           ALPHA = UA*(UA**2 - YY**2) - XA*(UA**2 - YAZ**2)
504           B1 = XA*UA*YAY/ALPHA
505           B2 = XA*YAY*YAZ/ALPHA
506           B3 = UA*(UA**2 - YY**2)/ALPHA
507           B4 = XA*YAZ*(UA - XA)/ALPHA
508           B5 = XA*UA*(UA - XA)/ALPHA
```

```
C
C      MATRIX T IS COLUMNWISE STORED
C     /
   30  T(1) = -AI*AL*B1
       T(2) = AI*AM*B1
       T(3) = -AL*AM + AI*B4
       T(4) = 1.0D0 - (AM*AM) - B5
       T(5) = AL*B2
       T(6) = -AM*B2
       T(7) = 1.0D0 - AL*AL - B5 + XA*YAY*YAY/ALPHA
       T(8) = -AL*AM - AI*B4
       T(9) = AL*AM*B3
       T(10) = 1.0D0 - B3*AM*AM
       T(11) = -AM*B2
       T(12) = -AI*AM*B1
       T(13) = 1.0D0 - B3*AL*AL
       T(14) = AL*AM*B3
       T(15) = AL*B2
       T(16) = AI*AL*B1
       DO 35 J=1,16
   35  T(J) = -AI*FAT*T(J)*3.333333333333333D-06
       DO 40 J=1,4
       DO 40 I=1,4
       K = I + 4*(J-1)
       KK = I + 8*(J-1)
       MM = I + 4 + 8*(J-1)
       LL = I + 32 + 8*(J-1)
       L = I + 36 + 8*(J-1)
       B1 = DCONJG(T(K))
       V(KK) = 0.50D0*(T(K) + B1)
       V(MM) = -0.50D0*AI*(T(K) - B1)
       V(LL) = 0.50D0*AI*(T(K) - B1)
   40  V(L) = 0.50D0*(T(K) + B1)
       RETURN
       END
```

```
541           SUBROUTINE OUTPUT
      C
      C
      C
542           REAL*8 DEXP,DLOG,DCOS,DSIN,DSQRT,DMAX1,DMIN1,DABS,CDABS
543           COMPLEX*16 CDEXP,CDLOG,CDSQRT,DCMPLX,DCONJG
544           REAL*8 YY,YAY,YAZ,AL,AM,AN,C,CA,CD,S,SA,SD,FREQ,ALT(500)
545           COMMON/BRAZIL/YY,YAY,YAZ,AL,AM,AN,C,CA,CD,S,SA,SD,FREQ,ALT
546           REAL*8 FH,ANGI,AZIM,DIP
547           COMMON/CANADA/FH,ANGI,AZIM,DIP,JZ,NA,NMAT,KFORM,KSTOP
548           COMPLEX*16 AI
549           COMMON/IMAG/AI
550           COMPLEX*16 E1(4,500),E2(4,500),DE1(4,500),DE2(4,500),AORT(500)
551           COMMON/RUSS/E1,E2,DE1,DE2,AORT
552           REAL*8 AA,B,F,TP,RP,RN,TH,TV,PPENR,PPENI,TRTR,TRTI,
             1    TRPR,TRPI,PRTR,PRTI,PRPR,PRPI,HSCALE(50),ABSPRP,ABSPRT,ABSTRT,
             2    ABSTRP
553           COMPLEX*16 A1,UPX1,DOX1,UPY1,DOY1,UPZ1,DOZ1,UPX2,DOX2,UPY2,DOY2,
             1    UPZ2,DOZ2,B1,UPPX,UPPY,UPPZ,DOPX,DOPY,DOPZ,ANUM,PPEN,PTW,A2,
             2 DELTA,A11,A12,A21,A22,B11,B12,B21,B22,TRT,TRP,PRT,PRP,F1,A2V,ASUM
554           DIMENSION ERROR1(500),ERROR2(500)
555           COMMON/OUTP/HSCALE,ERROR1,ERROR2
556           COMPLEX EX,EY,HY,HX,EX1,EX2,HX1,HX2
557           REAL AIMAG,REAL,CABS
      C
      C
      C
558    100    FORMAT (' ',' TP =',F20.6/'  TH =',F20.6/'  TV =',F20.6//)
559    200    FORMAT (' ',' RP =',F20.6/'  RN =',F20.6//)
560    300    FORMAT (' ',' POL.PENETRATING MODE =',F20.6,F20.6/)
561    400    FORMAT (' ',T13,'REAL',T31,'IMAGINARY',T55,'ABS.VALUE'/
             1    '  TRT =',F20.6,F20.6,F20.6/
             2    '  TRP =',F20.6,F20.6,F20.6/
             3    '  PRT =',F20.6,F20.6,F20.6/
             4    '  PRP =',F20.6,F20.6,F20.6//)
562    500    FORMAT ('1',' NON-PENETRATING WAVEFIELDS SET UP BY A HORIZONTAL
             1ELECTRIC FIELD OF UNIT AMPLITUDE'//)
563    600    FORMAT (' ',T3,'HEIGHT',T13,' EX ',T23,'R(EX)',T33,'I(EX)',T43,
             1' EY ',T53,'R(EY)',T63,'I(EY)',T73,' HX ',T83,'R(HX)',T93,'I(HX)',
             2   T103,' HY ',T113,'R(HY)',T123,'I(HY)'/)
564    700    FORMAT (F10.3,12E10.3)
565    800    FORMAT ('1',' PENETRATING WAVEFIELDS SET UP BY A HORIZONTAL ELECT
             1RIC FIELD OF UNIT AMPLITUDE'//)
566    900    FORMAT ('1',' ENVELOPE OF X-WAVEFIELDS SET UP BY A HORIZONTAL ELE
             1CTRIC FIELD OF UNIT AMPLITUDE'//'  PENETRATING MODE ',T80,'NON-PEN
             2ETRATING MODE'/T5,'ABS(EX)',T20,'ABS(HX)',T35,'ERROR',T58,'HEIGHT'
             3   ,T80,'ABS(EX)',T95,'ABS(HX)',T110,'ERROR'/)
567    1000   FORMAT (3E15.5,T55,F10.3,T76,3E15.5)
      C
      C
      C
568           AA = 3.14159265358979300*6.666666666666670-06*FREQ*AN
569           A1 = -AI/AA
      C
      C
570           UPX1 = 0.500*(E1(1,JZ) - A1*DE1(1,JZ))
571           DOX1 = 0.500*(E1(1,JZ) + A1*DE1(1,JZ))
572           UPY1 = - 0.500*(E1(2,JZ) - A1*DE1(2,JZ))
573           DOY1 = - 0.500*(E1(2,JZ) + A1*DE1(2,JZ))
574           UPZ1 = -(AL*UPX1 + AM*UPY1)/AN
```

```
575           DOZ1 = (AL*DOX1 + AM*DOY1)/AN
      C
576           UPX2 = 0.5D0*(E2(1,JZ) - A1*DE2(1,JZ))
577           DOX2 = 0.5D0*(E2(1,JZ) + A1*DE2(1,JZ))
578           UPY2 = - 0.5D0*(E2(2,JZ) - A1*DE2(2,JZ))
579           DOY2 = - 0.5D0*(E2(2,JZ) + A1*DE2(2,JZ))
580           UPZ2 = -(AL*UPX2 + AM*UPY2)/AN
581           DOZ2 = (AL*DOX2 + AM*DOY2)/AN
      C
      C     THE PENETRATING MODE
      C
582           B = UPX1*DCONJG(UPX1) + UPY1*DCONJG(UPY1) + UPZ1*DCONJG(UPZ1)
583           B1 = (UPX2*DCONJG(UPX1) + UPY2*DCONJG(UPY1) + UPZ2*DCONJG(UPZ1))/B
      C
584           UPPX = UPX2 - B1*UPX1
585           UPPY = UPY2 - B1*UPY1
586           UPPZ = UPZ2 - B1*UPZ1
      C
587           DOPX = DOX2 - B1*DOX1
588           DOPY = DOY2 - B1*DOY1
589           DOPZ = DOZ2 - B1*DOZ1
      C
590           DELTA = UPPX*UPY1 - UPX1*UPPY
591           A1 = -(CA*UPPY + SA*UPPX)/DELTA
592           A2 = (SA*UPX1 + CA*UPY1)/DELTA
593           IF (KFORM.EQ.1) GO TO 20
      C
      C     TRANSMISSION COEFFICIENT FOR THE PENETRATING MODE (PITTEWAY DEF.)
      C
594           F = UPPX*DCONJG(UPPX) + UPPY*DCONJG(UPPY) + UPPZ*DCONJG(UPPZ)
595           ANUM = E2(1,1)*DCONJG(E2(4,1)) + E2(2,1)*DCONJG(E2(3,1))
596           F1 = ANUM/(F*AN)
597           TP = 0.5D0*(F1 + DCONJG(F1))
598           TP = DSQRT(TP)
      C
      C     POLARIZATION OF THE PENETRATING MODE
      C
599           PPEN = (S*UPPZ - CA*C*UPPY - C*SA*UPPX)/(CA*UPPX - SA*UPPY)
600           PPENR = 0.5D0*(PPEN + DCONJG(PPEN))
601           PPENI = -AI*0.5D0*(PPEN - DCONJG(PPEN))
      C
      C     REFLECTION COEFFICIENT FOR THE PENETRATING MODE
602           RP = DOPX*DCONJG(DOPX) + DOPY*DCONJG(DOPY) + DOPZ*DCONJG(DOPZ)
603           RP = DSQRT(RP/F)
      C
      C     REFLECTION COEFFICIENT FOR THE NON-PENETRATING MODE
      C
604           RN = DOX1*DCONJG(DOX1) + DOY1*DCONJG(DOY1) + DOZ1*DCONJG(DOZ1)
605           RN = DSQRT(RN/B)
      C
      C     TRANSMISSION COEFFICIENT AT HORIZONTAL POLARIZATION
      C
606           TH = A2*DCONJG(A2)
607           TH = TP*DSQRT(TH*F)
      C
      C     TRANSMISSION COEFFICIENT AT VERTICAL POLARIZATION
      C
608           A2V = -C*(SA*UPY1 - CA*UPX1)/DELTA
609           TV = A2V*DCONJG(A2V)
610           TV = TP*DSQRT(TV*F)
```

```
      C
      C     FIRST SET OF OUTPUTS - REFLECTION COEFFICIENTS
      C                            TRANSMISSION COEFFICIENTS
      C                            POLARIZATIONS
      C
  611         PRINT 100, TP,TH,TV
  612         PRINT 200, RP,RN
  613         PRINT 300, PPENR,PPENI
      C
      C     COMPUTATION OF BUDDEN'S REFLECTION COEFFICIENTS
      C
  614         DELTA = UPX2*UPY1 - UPX1*UPY2
  615         A11 = CA*UPY2 + SA*UPX2
  616         A12 = CA*UPY1 + SA*UPX1
  617         A21 = CA*UPX2 - SA*UPY2
  618         A22 = CA*UPX1 - SA*UPY1
  619         B11 = CA*DCX1 - SA*DCY1
  620         B12 = CA*DCY1 + SA*DCX1
  621         B21 = CA*DCX2 - SA*DCY2
  622         B22 = CA*DCY2 + SA*DCX2
      C
  623         TRT = (A12*B21 - A11*B11)/DELTA
  624         TRP = (A12*B22 - A11*B12)/(DELTA*AN)
  625         PRT = AN*(A22*B21 - A21*B11)/DELTA
  626         PRP = (A22*B22 - A21*B12)/DELTA
      C
  627         TRTR = 0.5D0*(TRT + DCONJG(TRT))
  628         TRTI = -AI*0.5D0*(TRT - DCONJG(TRT))
  629         TRPR = 0.5D0*(TRP + DCONJG(TRP))
  630         TRPI = -AI*0.5D0*(TRP - DCONJG(TRP))
  631         PRTR = 0.5D0*(PRT + DCONJG(PRT))
  632         PRTI = -AI*0.5D0*(PRT - DCONJG(PRT))
  633         PRPR = 0.5D0*(PRP + DCONJG(PRP))
  634         PRPI = -AI*0.5D0*(PRP - DCONJG(PRP))
  635         ABSTRT = CDABS(TRT)
  636         ABSTRP = CDABS(TRP)
  637         ABSPRT = CDABS(PRT)
  638         ABSPRP = CDABS(PRP)
  639         PRINT 400, TRTR,TRTI,ABSTRT,TRPR,TRPI,ABSTRP,PRTR,PRTI,ABSPRT,
             1   PRPR,PRPI,ABSPRP
      C
  640    15   IF (KFORM.EQ.2) GO TO 65
      C
      C     RECONSTRUCTION OF THE WAVEFIELDS
      C
  641    20   F = 1.0D0
  642         K = 0
  643         ASUM = -B1
  644         DO 50 J = 1,JZ
  645         M = JZ - J + 1
  646         IF (ALT(M) - HSCALE(NA)) 35,30,30
  647    30   K = 1
  648    35   DO 40 I=1,4
  649         E2(I,M) = E2(I,M) + ASUM*E1(I,M)
  650    40   E1(I,M) = F*E1(I,M)
  651         ASUM = ASUM + AORT(M)
  652         IF (K.EQ.0) GO TO 50
  653         ASUM = ASUM*1.0D-03
  654         F = F*1.0D-03
  655         NA = NA - 1
```

```
656           K = 0
657        50 CONTINUE
      C
      C     CORRECT FIELDS ARE NOW STORED.  PENETRATING MODE STORED IN E2.
      C
658           IF (KFORM.EQ.4) GO TO 70
      C
      C     FIELDS SET UP BY A HORIZONTAL ELECTRIC FIELD OF UNIT AMPLITUDE.
      C
659           PRINT 500
660           PRINT 600
661           DO 55 J=1,JZ
662           EX = E1(1,J)*A1
663           EY = -E1(2,J)*A1
664           HX = E1(3,J)*A1
665           HY = E1(4,J)*A1
666           ABSEX = CABS(EX)
667           EXR = REAL(EX)
668           EXI = AIMAG(EX)
669           ABSEY = CABS(EY)
670           EYR = REAL(EY)
671           EYI = AIMAG(EY)
672           ABSHX = CABS(HX)
673           HXR = REAL(HX)
674           HXI = AIMAG(HX)
675           ABSHY = CABS(HY)
676           HYR = REAL(HY)
677           HYI = AIMAG(HY)
678        55 PRINT 700, ALT(J),ABSEX,EXR,EXI,ABSEY,EYR,EYI,ABSHX,HXR,HXI,
          1    ABSHY,HYR,HYI
      C
679           PRINT 800
680           PRINT 600
681           DO 60 J = 1,JZ
682           EX = E2(1,J)*A2
683           EY = -E2(2,J)*A2
684           HX = E2(3,J)*A2
685           HY = E2(4,J)*A2
686           ABSEX = CABS(EX)
687           EXR = REAL(EX)
688           EXI = AIMAG(EX)
689           ABSEY = CABS(EY)
690           EYR = REAL(EY)
691           EYI = AIMAG(EY)
692           ABSHX = CABS(HX)
693           HXR = REAL(HX)
694           HXI = AIMAG(HX)
695           ABSHY = CABS(HY)
696           HYR = REAL(HY)
697           HYI = AIMAG(HY)
698        60 PRINT 700, ALT(J),ABSEX,EXR,EXI,ABSEY,EYR,EYI,ABSHX,HXR,HXI,
          1    ABSHY,HYR,HYI
      C
      C
699        65 RETURN
700        70 PRINT 900
701           DO 75 J = 1,JZ
702           EX1 = E1(1,J)*A1
703           EX2 = E2(1,J)*A2
704           HX1 = E1(3,J)*A1
```

```
705           HX2 = E2(3,J)*A2
706           ABSEX1 = CABS(EX1)
707           ABSEX2 = CABS(EX2)
708           ABSHX1 = CABS(HX1)
709           ABSHX2 = CABS(HX2)
710       75  PRINT 1000, ABSEX2,ABSHX2,ERROR2(J),ALT(J),ABSEX1,ABSHX1,ERROR1(J)
711           RETURN
712           END
```

APPENDIX A. THE GENERALIZED QUARTIC OF BOOKER

In order to find the eigenvalues of the matrix $\tilde{T}$ Eq. (2.22) it is necessary to solve the characteristic equation

$$\det(\tilde{T} - q\tilde{I}) = 0 \tag{2.55}$$

which is known as the Booker quartic equation. A simple inspection of the equations that determine the elements T_{ij} Eq. (2.46) shows that for obtaining the coefficients of the quartic by direct use of Eq. (2.55) some extensive manipulation is required. Instead of working with Eq. (2.55) an easier and more general process is followed here.

Consider the geometry shown in Figure 1 in which the homogeneous medium where the incident wave exists is also allowed to be a general magnetoionic medium. The refractive index of the incident wave is n_1. The components of the refractive index vector are given by

$$\ell = n_1 \sin I \sin\chi \tag{A.1}$$

$$m = n_1 \sin I \cos\chi \tag{A.2}$$

$$q_i = n_1 \cos I \tag{A.3}$$

The above equations coincide with the definitions previously given to ℓ, m, and q_i when $n_1 = 1$. In order to satisfy Snell's law it is necessary that the horizontal projection of the refractive index be maintained constant at any height. In particular at the height z_1 where the medium is again supposed to be homogeneous the projections of the refractive index n are $n_1 \sin I$ and q such that

$$n^2 = (n_1 \sin I)^2 + q^2 \qquad (A.4)$$

q is the vertical projection of the refractive index at z_1, i.e., an arbitrary eigenvalue of $\tilde{T}$ at z_1.

On the other hand the equation that determines the refractive index is given by [Stix, 1962]

$$An^4 - Bn^2 + PRL = 0 \qquad (A.5)$$

where

$$A = S \sin^2\psi + P \cos^2\psi \qquad (A.6)$$

$$B = RL \sin^2\psi + PS(1 + \cos^2\psi) \qquad (A.7)$$

and ψ is the angle between the magnetic field $\vec{B}_o$ and the wave refractive index vector $\vec{n}$. R, L, P, and S are given by Eqs. (2.36) to (2.40). The angle ψ is related to q by

$$\cos\psi = \frac{\vec{B}_o \cdot \vec{n}}{B_o\, n} = \frac{\gamma n_1 \sin I \cos\chi + \xi q}{(n_1^2 \sin^2 I + q^2)^{1/2}} \qquad (A.8)$$

With the value of $\cos\psi$ given by Eq. (A.8) substituted in Eq. (A.6) and Eq. (A.7) plus Eq. (A.4) replacing n in Eq. (A.5) a new equation in q is obtained from Eq. (A.5):

$$(n_1^2 \sin^2 I + q^2)^2 S + (n_1^2 \sin^2 I + q^2)(P-S)(\gamma n_1 \sin I \cos\chi + \xi q)^2 - \qquad (A.9)$$
$$- (RL + PS)(n_1^2 \sin^2 I + q^2) - (PS-RL)(\gamma n_1 \sin I \cos\chi + \xi q)^2 + PRL = 0.$$

Equation (A.9) determines the coefficients of the quartic equation

$$\alpha q^4 + \beta q^3 + \gamma_1 q^2 + \delta q + \epsilon = 0 \qquad (A.10)$$

where

$$\alpha = S\gamma^2 + P\xi^2 \qquad (A.11)$$

$$\beta = 2m\xi\,\gamma(P-S)$$

$$\gamma_1 = n_1^2 \sin^2 I \left\{S\,[1 + \gamma^2(1-\cos^2\chi)] + P(\xi^2 + \gamma^2\cos^2\chi)\right\} - RL\,\gamma^2 - PS(1 + \xi^2)$$

$$\delta = -2m\xi\gamma\left[(PS-RL) - n_1^2\sin^2 I(P-S)\right]$$

$$\epsilon = PRL + n_1^2\sin^2 I\left\{n_1^2\sin^2 I\,[P\gamma^2\cos^2\chi + S(1-\gamma^2\cos^2\chi)] - RL(1-\gamma^2\cos^2\chi) - \right.$$
$$\left. - PS(1 + \gamma^2\cos^2\chi)\right\}$$

and,

$$\gamma = \cos(DIP) \qquad (2.1)$$

$$\xi = -\sin(DIP) \qquad (2.2)$$

Coefficients of the quartic equation when only electrons are considered. In this case the index k is dropped from the equations that define Y_k and X_k (Eqs. (2.28) and (2.29)) and the collisional variable U given by Eq. (2.48) is used. Equations (2.36) to (2.40) give

$$S = 1 - XU/(U^2-Y^2) \qquad (A.12)$$

$$P = (U-X)/U$$

$$RL = [(U-X)^2-Y^2]/(U^2-Y^2)$$

$$PS - RL = P - S = XY^2/U(U^2-Y^2)$$

It is also defined

$$C^2 = 1 - n_1^2 \sin^2 I \tag{A.13}$$

so that $C = \cos I$ when $n_1 = 1$.

When Eq. (A.12) and Eq. (A.13) are substituted in Eq. (A.11) a factor $U(U^2-Y^2)$ appears dividing all coefficients and is dropped. The new coefficients of the quartic equation are

$$\alpha = U(U^2-Y^2) - X(U^2-\xi^2 Y^2) \tag{A.14}$$

$$\beta = 2m\gamma\xi XY^2$$

$$\gamma_1 = 2(X-UC^2)[U(U-X)-Y^2] + XY^2(1 + m^2\gamma^2 - C^2\xi^2)$$

$$\delta = -2m\gamma\xi C^2 XY^2$$

$$\epsilon = (U-X)(X-UC^2)^2 + C^2Y^2(X-UC^2) - (m\gamma C)^2 XY^2$$

Equations (A.14) reduce to the coefficients found in the literature (see Budden [1966]-Ch.8) when $n_1 = 1$ and this also constitutes a check for the more general coefficients, Eq. (A.11). The coefficients of the Booker quartic when heavy ions are taken into account was derived by Walker [1968] who supplied very complicated expressions for the quartic coefficients formulated as functions of the elements of the susceptibility matrix $\tilde{M}$. The coefficients given here by Eq. (A.11) are much simpler.

APPENDIX B. THE EIGENVECTORS OF THE MATRIX $\tilde{T}$

The eigenvalues q of the matrix $\tilde{T}$ are given by the solution of Eq. (A.10). If one eigenvalue of $\tilde{T}$ is known the characteristic equation, Eq. (2.55), can be written as

$$(q-T_{11})E_x = -T_{12}E_y + T_{13}Z_oH_x + T_{14}Z_oH_y \tag{B.1}$$

$$-(q-T_{22})E_y = T_{21}E_x + T_{23}Z_oH_x + T_{24}Z_oH_y \tag{B.2}$$

$$(q-T_{33})Z_oH_x = T_{31}E_x - T_{32}E_y + T_{34}Z_oH_y \tag{B.3}$$

$$(q-T_{44})Z_oH_y = T_{41}E_x - T_{42}E_y + T_{43}Z_oH_x \tag{B.4}$$

Next, Eqs. (B.1) to (B.4) are manipulated in order to find an eigenvector $\vec{e}$ corresponding to the eigenvalue q. This will be done by relating all the eigenvector components to an arbitrary field amplitude E_x.

Equations (B.1) and (B.2) are multiplied by $(q-T_{44})$ which permits the elimination of Z_oH_y from Eqs. (B.1) and (B.2):

$$A_1E_x = -A_3E_y + A_6Z_oH_x \tag{B.5}$$

$$-A_2E_y = A_4E_x + A_5Z_oH_x \tag{B.6}$$

where

$$A_1 = (q-T_{11})(q-T_{44}) - T_{14}T_{41} \tag{B.7}$$

$$A_2 = (q-T_{22})(q-T_{44}) - T_{24}T_{42} \tag{B.8}$$

$$A_3 = T_{12}(q-T_{44}) + T_{14}T_{42} \tag{B.9}$$

$$A_4 = T_{21}(q-T_{44}) + T_{24}T_{41} \tag{B.10}$$

$$A_5 = T_{23}(q-T_{44}) + T_{24}T_{43} \tag{B.11}$$

$$A_6 = T_{13}(q-T_{44}) + T_{14}T_{43} \tag{B.12}$$

E_y and $Z_o H_x$ are determined from Eq. (B.5) and Eq. (B.6):

$$-E_y = \left(\frac{A_1A_5 + A_4A_6}{A_3A_5 + A_2A_6}\right) E_x \tag{B.13}$$

$$Z_o H_x = \left(\frac{A_1A_2 - A_3A_4}{A_3A_5 + A_2A_6}\right) E_x \tag{B.14}$$

$Z_o H_y$ can be determined directly from Maxwell's equations, Eqs. (2.14) and (2.15):

$$Z_o H_y = q\left\{E_x - \frac{\ell}{m} E_y\right\} - \frac{\ell}{m} Z_o H_x \tag{B.15}$$

with E_y and $Z_o H_x$ given by Eqs. (B.13) and (B.14).

Therefore, given one eigenvalue q of $\tilde{T}$ the corresponding eigenvector is given by

$$\vec{e} = \begin{bmatrix} E_x \\ -E_y \\ Z_o H_x \\ Z_o H_y \end{bmatrix} \tag{B.16}$$

where the elements of $\vec{e}$ are given as functions of E_x by Eqs. (B.13) through (B.15).

When the propagation is from west to east or vice-versa then $m = 0$ and therefore Eq. (B.15) cannot be used. In this case $Z_o H_y$ is determined from Eqs. (2.15) and (2.19):

$$Z_o H_y = \frac{\ell M_{zy} E_y + E_x [\ell M_{zx} + (1 + M_{zz})q]}{M_{zz} + 1 - \ell^2} \tag{B.17}$$

The elements of the susceptibility matrix $\tilde{M}$ are given by Eq. (2.43). When only the effect of electrons is considered Eq. (B.17) yields to

$$Z_o H_y = \frac{\ell \gamma \xi XY^2 E_y + (bq - j\ell\gamma UXY) E_x}{b - \ell^2 U(U^2 - Y^2)} \tag{B.18}$$

where b is given by Eq. (2.50).

REFERENCES

Altman, C. and H. Cory, The generalized thin-film optical method in electromagnetic wave propagation, Radio Science, 4, 459, 1969.

Barron, D. W. and K. G. Budden, The numerical solution of differential equations governing the reflexion of long radio waves from the ionosphere, III., Proc. Roy. Soc. A, 249, 387, 1959.

Booker, H. G., Oblique propagation of electromagnetic waves in a slowly varying nonisotropic medium, Proc. Roy. Soc. A., 155, 235, 1936.

Booker, H. G., The propagation of wave packets incident obliquely on a stratified doubly refracting ionosphere, Phil. Trans. A., 237, 411, 1939.

Budden, K. G., Radio Waves in the Ionosphere, Cambridge University Press, Cambridge, England, 1966.

Budden, K. G., The numerical solution of differential equations governing reflexion of long radio waves from the ionosphere, Proc. Roy. Soc. A., 227, 516, 1955.

Clemmow, P. C. and J. Heading, Coupled forms of the differential equations governing radio propagation in the ionosphere, Proc. Camb. Phil. Soc., 50, 319, 1954.

Deeks, D. G., D-region electron distributions in middle latitudes deduced from the reflexion of long radio waves, Proc. Roy. Soc. A., 291, 413, 1966.

Friedman, B., Principles and Techniques of Applied Mathematics, John Wiley & Sons, Inc. (6th printing), New York, 1964.

Hamming, R. A., Stable predictor-corrector methods for ordinary differential equations, J. Assoc. Comput. Mach., 6, 37, 1959.

Johler, J. R. and J. D. Harper, Reflection and transmission of radio waves at a continuously stratified plasma with arbitrary magnetic induction, J. Res. NBS, 66D, 81, 1962.

Piggott, W. R., M. L. V. Pitteway and E. V. Thrane, The numerical calculation of wave fields, reflexion coefficients and polarizations for long radio waves in the lower ionosphere, II., Phil. Trans. Roy. Soc., London A., 257, 243, 1965.

Pitteway, M. L. V., The numerical calculation of wave fields, reflexion coefficients and polarizations for long radio waves in the lower ionosphere, I., Phil. Trans. Roy. Soc., London, A., 257, 219, 1965.

Pitteway, M. L. V. and J. L. Jespersen, A numerical study of the excitation, internal reflection and limiting polarization of whistler waves in the lower ionosphere, J. Atmos. & Terr. Phys., 28, 17, 1966.

Ralston, A., Runge-Kutta methods with minimum error bounds, MTAC, 16, 431, 1962.

Ralston, A., A First Course in Numerical Analysis, McGraw Hill Book Co., New York, 1965.

Smith, G. H. and M. L. V. Pitteway, Fortran program for obtaining wave-fields of penetrating, non-penetrating, and whistler modes of radio waves in the ionosphere, (submitted to Radio Science, 1969).

Stix, T. H., The Theory of Plasma Waves, McGraw Hill Book Co., New York, 1962.

Walker, A. D. M., Ray tracing in the ionosphere at VLF-I, J. Atmos. & Terr. Phys., 30, 403, 1968.