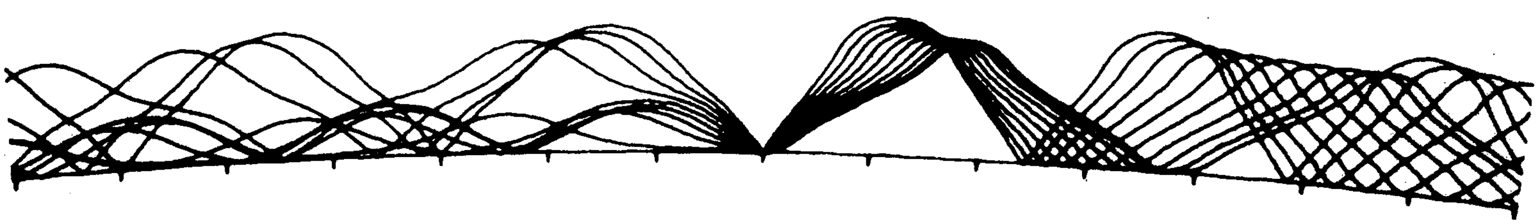
HARPA

A Versatile Three-Dimensional Hamiltonian Ray-Tracing Program for Acoustic Waves in the Atmosphere Above Irregular Terrain

R. Michael Jones
J.P. Riley
T.M. Georges



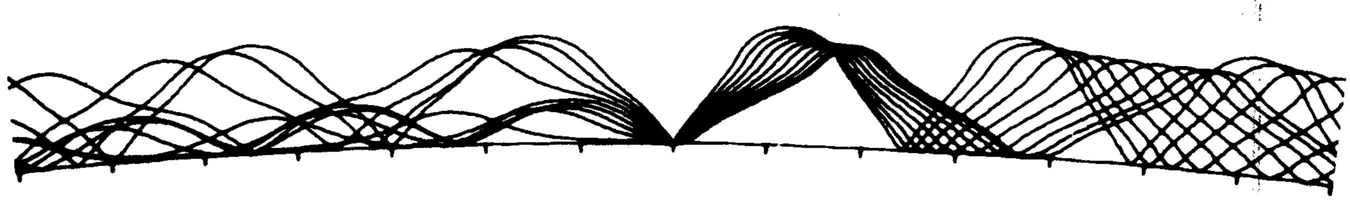
U.S. DEPARTMENT OF COMMERCE
National Oceanic and Atmospheric Administration
Environmental Research Laboratories

HARPA

A Versatile Three-Dimensional Hamiltonian Ray-Tracing Program for Acoustic Waves in the Atmosphere Above Irregular Terrain

R. Michael Jones
J.P. Riley
T.M. Georges

Wave Propagation Laboratory
Boulder, Colorado
August 1986



U.S. DEPARTMENT OF COMMERCE
National Oceanic and Atmospheric Administration
Environmental Research Laboratories

NOTICE

“HARPA: A versatile three-dimensional Hamiltonian ray-tracing program for acoustic waves in the atmosphere above irregular terrain”, by R. Michael Jones, J. P. Riley, and T. M. Georges, NOAA special report, August 1986

Errata, August 3, 2009

1 Documentation Errata

page 31: The Transmitter latitude (W4) should have “km” circled.

pages 31 and 199: On the “form to specify input data”, “stop frequency stepping” should be changed to “stop elevation angle stepping,” and W30, W31, and W32 should be changed to W278, W279, and W280, respectively.

pages 33 and 221: The model check number in the “Form to specify input data for receiver-surface model RTERR” should be 2.0 instead of 3.0

page 49, Fig. 2.23: The implied decimal point for latitude of transmitter should be between card columns 16 and 17. The implied decimal point for longitude of transmitter should be between card columns 22 and 23. The implied decimal point for imaginary part of wave polarization at transmitter should be between card columns 75 and 76.

page 50, Fig. 2.24: The imaginary part of wave polarization should be in card columns 74-77, and the implied decimal point should be between card columns 75 and 76.

page 69: The last two lines should read:

*** Format type 1 implies format number A (see Table 5.3).
*** Format type 2 implies format number 1, 2, or 3 (see Table 5.3).

pages 94 through 98: The figure captions for Figures 6.1 through 6.5 should have the following added:

Circled block numbers correspond to program statement numbers.

page 98: In Figure 6.5, the lower branch on “Test mode.” should read:

MODE = 4 and $y_{i,1} \neq 0$

page 101: The last sentence in Section 6.4 should refer to Table 7.17 instead of 7.9.

page 102: The second line of the first full paragraph should refer to equation (6.30) rather than (4.1).

page 136: The first note in the caption to Figure 7.10 should read:

* See Equation (6.83) to estimate the time of the nearest closest approach to the specified surface.

pages 177 through 196: The calculation of absorption in the sample printout and sample raysets is incorrect. The correct values are in the files dinp.sam and punch.sam

page 200: The sentence "Superimpose these raypath plots on the graph of the previous runset:" should read "Superimpose these raypath plots on the graph of the next runset:"

page 221: The model check number in the "Form to specify input data for receiver-surface model RTERR" should be 2.0 instead of 3.0

page 222: The model check number in the "Form to specify input data for receiver-surface model RVERT" should be 3.0 instead of 2.0

2 Program Errata

page 251: Following line "UCON 30" in LOGICAL FUNCTION UCON, insert the line:

```
      IF(CONV.EQ.-1.0) CONV = 1.0/EARTHR                    UCON305
```

page 251: Line "UCON 38" in LOGICAL FUNCTION UCON should be replaced by:

```
      CNVV(1,3) = -1.0                                      UCON380
```

page 329: The variable OWI in line ANWWL 70 in SUBROUTINE ANWWL should be OW.

page 332: The variable OW in line AWWWL 76 in SUBROUTINE AWWWL should be OWI.

page 344: Line WGAUSS18 in SUBROUTINE WGAUSS2 should be

```
      DATA RECOGU/8.0/                                      WGAUSS18
```

page 395: Line "RVERT 21" in SUBROUTINE RVERT should be replaced by:

```
      DATA RECORR/3.0/                                      RVERT21
```

3 Additional Errata from Appendix E of HARPO Report follows.

APPENDIX E. ERRATA FOR HARPA REPORT

HARPA: A versatile Three-Dimensional Hamiltonian Ray-Tracing Program for Acoustic Waves in the Atmosphere Above Irregular Terrain" by R. Michael Jones, J. P. Riley, and T. M. Georges

2 February 1987

Page xi: change line 12 to:
Table 7.23 Definitions of the parameters in common block /HDRC/....157

Page 21: Following "The profile:" circle the units "km" in the columns labeled z_i and δ_i.

Page 31 and 199: At mid-page, change "stop frequency stepping" to "stop elevation-angle stepping," and change W30, W31, and W32 to W278, W279, and W280, respectively.

Page 33 and 221: Change the Model Check Number from 3.0 to 2.0.

Page 50: Change "Phase path, km" to Phase time, sec" and "Group path, km" to "Pulse travel time, sec."

Page 59: In Table 4.1, change "NPABS" to "NPABSR".

Page 69: Change the last two lines to read:
*** Format type 1 implies format number A (see Table 5.3).
*** Format type 2 implies format number 1, 2, or 3 (see Table 5.3).

Page 79: Change description following W(21) to read "Set = 1 to stop elevation-angle increment when the ray goes out of bounds."

Page 94-98: Add the following to the captions for Figures 6.1 through 6.5: "Circled block numbers correspond to program statement numbers."

Page 98: Change the comment near the lower branch of the "Test Mode" block to read: "MODE = 4 and $Y_{i,1} \neq 0$".

Page 101: In the last sentence of Section 6.4 change the table mentioned from Table 7.9 to Table 7.17.

Page 102: In the second line of the first full paragraph change the equation mentioned from Eq. (4.1) to Eq. (6.30).

Page 126 and 128: Change the captions so that the parenthetical expressions following ANWNL and AWWNL begin "(Acoustic, No Winds..." and "(Acoustic, With Winds...").

Page 127: Change the name of PROGRAM NITIAL to PROGRAM RAYTRC in the second block down.

Page 136: Change the first note in the caption of Figure 7.10 to read: "* See Equation (6.83) to estimate the time of nearest closest approach to the specified surface."

Page 155: Add the variable names NDEVGRP and NDEVBIN to Table 7.19.

Page 158: Replace Table 7.23 by:

Table 7.23--Definitions of the parameters in common block /HDRC/

Position in common	Variable name	Definition
1	INITID	Character string for user name and phone number identifier for graphics output
2	DAT	Character string for the date of the computer run
3	TOD	Character string for the time of day of the computer run

Page 168: In line 9, replace PGRKPH with PGRPH.
In line 11, replace $\partial g/\partial \theta$ by $\partial g/\partial \phi$.

Page 222: Change the Model Check Number from 2.0 to 3.0.

Make the following changes in <u>both</u> the source-code listing (Appendix D) and in the program itself:

Page 251: Following the line "UCON 30" in LOGICAL FUNCTION UCON, insert the line:

```
      IF(CONV.EQ.-1.0) CONV = 1.0/EARTHR                             UCON305
```

Page 251: Replace line "UCON 38" in LOGICAL FUNCTION UCON by:

```
      CNVV(1,3) = -1.0                                               UCON380
```

Page 361: Replace line "TTANH554" in SUBROUTINE TTANH5 by:

```
      ZIM1 = Z0                                                      TTANH554
```

Page 395: Replace line "RVERT 21" in SUBROUTINE RVERT by:

```
      DATA RECORR/3.0/                                               RVERT21
```

Add the following routine:

```
      FUNCTION ITOC(N)                                               ITOC0020
C     RETURN 7 CHARACTER STRING REPRESENTATION OF INTEGER N          ITOC0030
C     IF NUMBER IS TOO LARGE OR SMALL USE FLOATING POINT FORMAT      ITOC0040
      CHARACTER ITOC*7                                               ITOC0050
      IF(N.LT.-9999.OR.N.GT.99999) GO TO 100                         ITOC0060
      ITOC=' '                                                       ITOC0070
      WRITE(ITOC,'(I7)',ERR=100) N                                   ITOC0080
      RETURN                                                         ITOC0090
100   WRITE(ITOC,'(2PG7.0)') FLOAT(N)                                ITOC0100
      END                                                            ITOC0110
```

CONTENTS

FIGURES

TABLES

ACKNOWLEDGMENTS

Part of the organization of this program into subroutines follows that of the program of Dudziak (1961). Also, the coordinate transformation in subroutine PRINTR and the method for data input via the W array are taken from the program of Dudziak (1961). The term "rayset," the idea of outputting computer-readable results of each hop for each ray trace, and the idea of automatically plotting raypaths come from the program of Croft and Gregory (1963). Subroutine RKAM1 is a modification of subroutine RKAMSUB, written by G. J. Lastman, and is available through the CDC CO-OP library (the CO-OP identification is D2 UTEX RKAMSUB). Subroutine GAUSEL was written by L. David Lewis, NOAA Space Environment Laboratory. Judith Stephenson wrote much of the code for the original ionospheric ray-tracing program, upon which HARPA is based. Richard Lindzen devised the method upon which models TTANH5, CSTANH, and GTANH are based. We also thank the many users of earlier versions of the program who provided helpful feedback. Special thanks go to the Editorial Staff of Publication Services for extensive help in clarifying the expression of our ideas, and to Ms. Mildred Birchfield for her excellent typing and layout of the manuscript.

HARPA -- A VERSATILE THREE-DIMENSIONAL HAMILTONIAN RAY-TRACING PROGRAM FOR ACOUSTIC WAVES IN THE ATMOSPHERE ABOVE IRREGULAR TERRAIN

R. Michael Jones, J. P. Riley, and T. M. Georges

ABSTRACT

HARPA stands for Hamiltonian Acoustic Ray-tracing Program for the Atmosphere. This FORTRAN computer program traces the three-dimensional paths of acoustic rays through model atmospheres by numerically integrating Hamilton's equations, which are a differential expression of Fermat's principle. The user specifies an atmospheric model by writing closed-form formulas for its three-dimensional wind and temperature (or sound-speed) distribution, and by defining the height of the reflecting terrain as a function of geographic latitude and longitude. Some general-purpose models are provided, or users can easily design their own.

Because it uses continuous models, the Hamiltonian method avoids the false caustics and discontinuous raypath properties encountered in conventional ray-tracing methods, which use layers or cells where each acoustic-raypath segment can be computed in closed form. Furthermore, computational speed can be traded for accuracy, without changing the model of the medium, by specifying the maximum allowable integration error per step.

In addition to computing the geometry of each raypath, the program can calculate pulse travel time, phase time, Doppler shift (if the medium varies in time), absorption, and geometrical path length. Amplitude is not explicitly computed, but the contributions by absorption, reflection losses, and focusing are separately available for each ray. Only geometrical effects are accounted for; that is, no diffraction or partial-reflection corrections are applied. The program prints out a step-by-step account of each ray's progress, and it can plot the

projection of a set of rays on any vertical plane or on the ground. Furthermore, it can output each ray's properties in machine-readable form for further processing (amplitude calculations, for example).

This report describes the ray-tracing equations and the structure of the program and provides complete instructions for using it, illustrated by a sample case. The program is modular and can be adapted to model propagation through other media by changing the routine that defines the medium's dispersion relation.

PART I: WHAT THIS RAY-TRACING PROGRAM CAN DO

1. Introduction to Hamiltonian Ray Tracing

1.1 Rationale

Many practical problems in atmospheric acoustics submit to a straightforward application of geometrical acoustics, or ray theory. No other propagation-modeling tool provides such an intuitive and graphic portrayal of the paths that acoustic energy follows through inhomogeneous media. Even in situations where ray theory does not strictly apply, a picture of the acoustic raypaths often provides a useful first look at the way the waves and the medium interact, and it gives insight into where higher order computations are required. Some calculations cannot be easily made in any other way, for example, computing multipath pulse travel time or showing which parts of the medium affect each pulse arrival.

Yet most of the ray-tracing computer programs in common use fail to take full advantage of the power of geometrical acoustics. Many are essentially automated versions of graphical techniques that patch together closed-form raypath solutions for layers or cells with simple refractive-index gradients (Roberts, 1974; Cornyn, 1973). In such models, gradient discontinuities at cell boundaries can introduce false caustics and cause discontinuous behavior of ray properties as launch angle varies (Pederson, 1961). Furthermore, it is difficult to extend such models to three-dimensional media, to account for winds, and to compute reflections from complicated terrain models.

This report describes a general-purpose atmospheric acoustic ray-tracing program called HARPA -- for Hamiltonian Acoustic Ray-tracing Program for the Atmosphere -- that we have designed to overcome these limitations. It computes acoustic raypaths by numerically integrating Hamilton's equations, which are a differential expression of Fermat's principle. The user defines an atmospheric model by writing closed-form expressions for its temperature (or sound-speed) and wind distribution in three dimensions, and by defining the terrain height as a function of latitude and longitude. Several simple but generally useful models with user-definable parameters are described in this report; users can pattern their own models after them.

HARPA is the companion to a similar program we have developed for the ocean, known as HARPO. The main differences between the two programs are in the models available for the two media, in provisions for reflections from an upper boundary (in the ocean case), and in the program module that describes the media dispersion relations. HARPO is documented in a separate report (Jones et al., 1986).

1.2 What Is Ray Tracing?

Although ray tracing has a long history, many people outside the field do not know what ray tracing is or what it can do. In ray theory, waves are treated like particles (photons of light, phonons of sound) that travel along geometric trajectories called rays. In material media, the particles travel at a speed determined by the medium's "refractive index." Gradients in refractive index bend rays, giving rise to the problem of computing ray trajectories through a known spatial distribution of refractive index. Ray tracing is any method, graphical or numerical, for solving that problem.

Originally, lensmakers used ray tracing to find out how light rays travel through optical systems. They used graphical techniques based on Snell's law to compute the bending that light rays suffer when they encounter abrupt changes in refractive index, as at the surface of a lens. By constructing bundles of such rays, lensmakers could simulate the magnification, reduction, and focusing of their lens designs without actually building them.

Modern ray-tracing applications, whether acoustic or electromagnetic, serve basically the same function: they allow one to simulate the propagation of waves through media whose refractive-index structure varies in a complicated way, without actually performing the physical measurement. Modern ray-tracing computations are usually performed by programs written for digital computers that can graphically display the results of their computations in various informative ways.

Today's ray-tracing programs do much more than compute the bending of rays as they cross interfaces; they can model media whose refractive index varies continuously in space and even with time. In dissipative media, they integrate absorption along the raypath. They can also integrate phase and pulse travel time, as well as wave amplitude. In time-varying media, they can integrate the rate of change of phase, or Doppler shift. Some programs (including HARPA) produce machine-readable output so that the results of many raypath computations can be processed by other programs to display field observables, such as amplitude.

The most advanced applications of ray-tracing computer programs have been to the fields of ionospheric radio propagation, seismic wave propagation, and the propagation of acoustic or sound waves in the ocean and the atmosphere. In the Hamiltonian formalism, the ray-tracing equations for acoustic, seismic, and electromagnetic waves are identical. General-purpose programs can thus be constructed in which only the modules that describe the wave dispersion relation and how the medium varies in space need be changed to go from one kind of ray-tracing program to another.

1.3 What Approximations Are Involved in Ray Tracing?

Solving a wave equation with arbitrary boundary conditions is still an impractical task, even for the most modern computers. Therefore, practical problems in wave propagation are often solved by making simplifying approximations to the wave equation. Examples of such approximations are the parabolic-equation (P.E.) method (Tappert, 1977), normal-mode theory (Tolstoy and Clay, 1966; Pierce, 1965), fast-field methods for numerical integration of the wave equation in range-independent environments (Raspet et al., 1985), and ray theory.

Ray theory, sometimes called the WKB or eikonal method, results from making a high-frequency approximation in the solution of arbitrary elliptic or hyperbolic partial differential equations (Budden, 1961). Ray tracing is related to the "method of characteristics" for solving such equations because the raypaths are the bi-characteristic rays of the differential equations in the infinite frequency, infinite wave-number limits. In some fields, ray tracing is called the "shooting method" because (as with shooting a gun) the location of the end point is found by trial and error while the initial conditions of a ray are varied.

In the case of the wave equation, the approximation gives rise to the fields of geometrical acoustics or geometrical optics, which are concerned with the trajectories of bundles of acoustic or electromagnetic energy radiated in infinitesimal angular beams. Such rays experience no diffraction but produce sharp shadow boundaries when they encounter solid objects. Ray theory can be extended to include the effects of diffraction, for example, by using the Geometrical Theory of Diffraction (GTD) (Keller, 1962).

In ray theory, one assumes conservation of energy within a bundle of rays called a flux tube so that wave intensity is inversely proportional to the cross-sectional area of the flux tube. When that cross-sectional area becomes zero, ray theory predicts infinite energy density. At such "caustics," higher order corrections to ray theory can give more accurate field estimates when needed. For example, the field near a surface caustic can be calculated in terms of Airy functions (Ludwig, 1966; White and Pedersen, 1981).

Without such corrections, ray tracing accounts only for refraction by large-scale gradients in the medium and not for diffraction and scattering by changes in the medium over scales that are small compared to a Fresnel zone. Even so, ray theory provides a useful first look at many complicated propagation problems and gives a kind of graphical insight lacking in other propagation models.

1.4 When Should You Use Ray Tracing?

Ray tracing is best suited to modeling acoustic propagation in environments where the medium's refractive index can be described deterministically in one, two, or three dimensions, and where changes in refractive index are small in the WKB sense (roughly speaking, within an acoustic Fresnel zone). (This means that ray models are most accurate at high frequencies.) In such environments, ray tracing gives accurate information about the geometrical paths followed by acoustic rays (energy), about shadow boundaries and reflections from surfaces, and about phase, intensity, pulse travel time, absorption and Doppler shift (for time-varying media) integrated along those paths.

In environments where multiple rays reach a receiver location of interest, additional computations, external to the ray tracing, may be required to combine field information from multiple rays. When the number of multipath rays becomes large, alternative formulations of the problem (P.E. or normal-mode theory, for example) are more appropriate for continuous-wave amplitude calculations. For pulse transmissions, ray theory is useful for describing the distinct geometric paths corresponding to each pulse arrival and for computing multipath travel times.

In situations where the applicability of ray theory is doubtful, a raypath picture can tell which regions must be treated with higher order methods, such as GTD or the Airy-function approximation to the field near a caustic. Furthermore, there are standard formulas to estimate how close to a caustic amplitude calculations are accurate (Budden, 1961; 1972).

Even when ray calculations of one wave quantity become inaccurate, they can give useful estimates of others. For example, when amplitude estimates break down (as at surface caustics), other information, such as travel time or phase, may still be reliable and can be tracked through caustics. Furthermore, Budden (1961, pp. 325-326) shows that the ray-computed phase must be advanced by 90° every time a ray passes through a line caustic.

1.5 What Is Hamiltonian Ray Tracing?

An alternative to cellular methods requires the medium to be modeled as a continuous three-dimensional function with continuous gradients and computes each raypath by numerically integrating Hamilton's equations with a different set of initial conditions. This method has been called Hamiltonian ray tracing. Hamilton's equations are the same for all kinds of wave propagation; only the definition of the Hamiltonian varies when going from one wave type to another.

Although Hamilton's equations are more familiar in mechanics, they have a long history of application to more general problems, including wave propagation. There, the point of view is that in a high-frequency limit, waves behave like particles and travel along rays, according to equations that exactly parallel those governing changes of position and momentum in mechanical systems (Lighthill, 1978, Sec. 4.5). Two steps show that integrating Hamilton's equations can lead to approximate solutions of a wave equation:

(1) The first step is to show that solutions to the wave equation are related to paths that satisfy a particular stationary principle, usually called

Fermat's principle. There are at least two standard methods for demonstrating that relation.

(a) First is the method of characteristics (see, for example, Courant and Hilbert, 1962; Garabedian, 1964), in which the solution is related to initial-value data chosen on some appropriate surface. Specifying a surface and constructing a solution requires first constructing the characteristic surfaces that are wave fronts of the wave. These characteristic surfaces can be constructed by first constructing bi-characteristic rays that satisfy a stationary principle. The bi-characteristic rays are the same as the geometrical raypaths whenever all terms in the wave equation are proportional to a derivative of the wave function, or in the limit of infinite frequency and wave number.

(b) Second is the path-integral method (see, for example, Feynman and Hibbs, 1965), in which a solution to the wave equation is constructed as an integral over all possible paths (not just raypaths) that connect the source and observer. Making a saddlepoint (or stationary phase) approximation to the path integral finds the paths that contribute most to the path integral. Such paths are those for which the action (phase) is stationary for variations of the path; that is, they satisfy Fermat's principle.

(2) The second step is to show that Hamilton's equations can be integrated to construct paths that satisfy a variational principle, such as Fermat's principle. This is done in standard texts (for example, Lighthill, 1978). First, the variational principle is expressed as an integral of a Lagrangian along the path (specified in terms of generalized coordinates, q_i). This determines the form of the Lagrangian for the problem, which for the wave equation is usually some simple function of the phase refractive index. Then the generalized momenta p_i are defined, which for the wave equation correspond to components k_i of the wave number vector. Then a Hamiltonian $H(q_i, p_i)$ is constructed from the Lagrangian. For the wave equation, the Hamiltonian is usually a function that gives the dispersion relation for the wave in question when it is set to zero. Integrating Hamilton's equations then gives a path that satisfies the variational (Fermat's) principle.

In Cartesian coordinates, Hamilton's equations take the particularly simple form (Lighthill, 1978)

$$\frac{dx_i}{d\tau} = \frac{\partial H}{\partial k_i} \quad ; \quad \frac{dk_i}{d\tau} = - \frac{\partial H}{\partial x_i} \quad , \quad i = 1 \text{ to } 3 \quad , \tag{1.1}$$

where τ is a parameter (sometimes proportional to time) whose physical meaning depends on the how the Hamiltonian, H, is defined, k_i are the wave-number components, and x_i are the coordinates of a point on the raypath. Transforming to spherical polar coordinates complicates the equations considerably. The full set of equations for spherical coordinates can be found in Chapter 6.

To solve (1.1) for the raypath, one chooses initial values for the six quantities x_i and k_i and performs a numerical integration of the system (1.1) of six total differential equations. For acoustic waves in the atmosphere, the Hamiltonian (which is constant along a raypath) is defined as

$$H(x_i, k_j) = [\omega - \vec{k}\cdot\vec{V}(x_i)]^2 - C^2(x_i)\, k^2 = 0 \ , \tag{1.2}$$

where $\vec{V}(x_i)$ is the wind field, $C(x_i)$ is the sound-speed field, and ω is the angular wave frequency ($\vec{V}$ and C may also depend on time). Thus, the effects of a three-dimensional vector-wind field are automatically included in the definition of the Hamiltonian.

There is an alternative to Hamilton's equations for a differential form of the ray equation, namely the eikonal equation (see, for example, Garabedian, 1964, p. 166; Felsen and Marcuvitz, 1973, p. 126). The eikonal equation is derived by first assuming an approximate solution to the wave equation in terms of an asymptotic series. Substituting the asymptotic series into the wave equation leads to the eikonal equation, which determines the raypaths, and a transport equation, which determines an approximate solution to the wave equation. The eikonal equation is equivalent to Hamilton's equations for determining the raypath. The transport equation is equivalent to methods mentioned above for determining an approximate solution to the wave equation.

In addition to allowing continuous three-dimensional models of the refractive-index field and two-dimensional models of reflecting surfaces, Hamiltonian ray tracing by numerical integration permits the user to trade computing speed for accuracy by specifying the maximum allowed integration error per step. In other words, you can have a fast but crude ray trace or a slower and more accurate one. The program automatically adjusts the integration step length along the raypath to keep the error within specified bounds. In regions where the refractive index varies quickly, small steps are required, but in regions where it varies slowly, large steps save computation. If the quantity being integrated varies monotonically along the raypath, the specified relative accuracy will be preserved in integrated quantities, such as travel time.

1.6 What This Program Does

HARPA computes the paths of acoustic rays, one at a time, through a user-defined model of the atmosphere, given initial conditions that include the source location (latitude, longitude, and height above the ground), wave frequency, direction of transmission (elevation and azimuth), the receiver-surface model, and the maximum number of hops (intersections with the receiver surface). The input data specification forms in Chapter 2 illustrate the generality of acceptable input.

The mechanics of the raypath calculation have been completely separated from the modeling of the medium (sound-speed, wind velocity, and terrain models). This allows the user to select models from those we have developed or to develop new models simply by writing new (or altering existing) subroutines to define those models.

The modular structure of the program allows the user to extend the program easily to other types of geophysical ray tracing (underwater acoustics, for example) simply by substituting new subroutines for defining the Hamiltonian and the model of the medium.

The method for inputting data into the program is easy to learn. The user simply specifies the magnitude and units of the elements of an Input Data File which correspond to physical or mathematical quantities that tell the program what models to use, what rays to trace, and in what form to present its results. We provide input parameter forms for making sure that all the required quantities are specified.

At the user's option, HARPA produces three kinds of output: (1) The printout reproduces the input data set and gives detailed information about each raypath computed, in columnar form, with each line corresponding to a "snapshot" of the ray's progress after a specified number of integration steps. (2) Computer-readable output permits further processing of raypath data by supplementary programs, without recomputing the raypaths. (3) The raypath plots show projections of any part of the raypaths on any vertical plane or on any part of the ground, with any desired magnification. These plots give the user a quick view of the raypath geometries.

Chapter 2 illustrates more fully what the program does by going through the setup and execution of a representative application.

1.7 What This Program Does Not Do

HARPA's computations lie entirely within the scope of geometrical acoustics (ray theory). It applies no corrections for diffraction or partial reflections. The atmospheric model must be deterministic, not stochastic.

There are no provisions built into HARPA for explicit computations of acoustic amplitude. This would normally be done with a supplementary program that processes HARPA's machine-readable output. Total amplitude at a receiver would be computed by combining flux-tube focusing, reflection losses, and absorption, and the user would normally decide whether to add coherently or incoherently the contributions of multipath rays. Because there are so many ways to compute amplitude, we think it is best to keep the various factors separate and let the users combine them however they wish.

Because the numerical integration of Hamilton's equations requires media models with continuous gradients, HARPA cannot presently handle refraction at discontinuities of refractive index or its gradients. If such discontinuities are included in a model, the integration routine will attempt to handle them by taking extremely small steps when a ray encounters a discontinuity, and the results may not be reliable. In general, one can approximate discontinuous functions with continuous functions to any desired accuracy, and HARPA will adjust its step length to accommodate them. Our algorithms for reflecting rays from arbitrary terrain surfaces could be generalized to compute refraction at discontinuities in refractive index.

HARPA is not currently equipped to model penetration of rays into the ground or to account for partial reflections from subsurface layers. However, the user can specify a complex (to account for phase and amplitude) ground-reflection coefficient that is a function of frequency and angle of incidence. Since reflection coefficients do not affect the raypaths, their effects can be added after raypath calculations.

HARPA cannot directly compute the raypaths that connect a specified source and receiver. To find such "eigenrays," one usually launches a fan of rays at small increments in elevation and/or azimuth angle and linearly interpolates for the rays that reach the desired receiver location (range, azimuth and height). We have developed an eigenray program that processes the "rayset" output of HARPA, interpolating in elevation angle only, and that will be documented elsewhere. Some shortcuts for finding three-dimensional eigenrays when azimuthal deflections are small are described by Georges et al. (1986).

HARPA makes no checks to see if atmospheric models satisfy physical conservation laws and boundary conditions, or that wind and temperature models are geostrophically consistent. (Accurate raypaths can be computed through physically impossible models.) Therefore, users should make their models as physically realistic as their application demands.

1.8 History of the Program

HARPA has a long history of development. We started by learning from the programs of Dudziak (1961) and Croft and Gregory (1963). Jones (1966) documented our first version of a three-dimensional ray-tracing program for radio waves in the ionosphere, which included anisotropy caused by the earth's magnetic field. Jones (1968) documented improvements in the original program. Georges (1971) converted the ionospheric radio program to trace raypaths for acoustic-gravity waves in an atmosphere with winds and changed the ray-tracing equations into Hamiltonian form. Jones and Stephenson (1975) documented further significant improvements in the ionospheric program. Jones et al. (1982) documented a Hamiltonian acoustic ray-tracing program for an atmosphere over a spherical earth.

Through its history, HARPA and its predecessors have found application in the propagation of radio waves through the ionosphere (Georges, 1967; 1970;

Georges and Stephenson, 1968; Stephenson and Georges, 1969), acoustic propagation through the atmosphere (Georges, 1972; Georges and Beasley, 1977) and acoustic propagation in the ocean (Georges et al., 1986; Jones et al., 1984). In extending the utility of ray theory, Jones (1970) has treated ray propagation in lossy media (ray tracing in complex space), bending of rays in random, inhomogeneous media (Jones, 1981a), and the frequency shift suffered by pulses propagating in dispersive media (Jones, 1981b). Jones (1983) has also surveyed existing techniques for underwater acoustic ray tracing.

HARPA combines the improvements made by Jones and Stephenson with the acoustic-wave capability and atmospheric models developed by Georges. Although it does not include the capability for tracing acoustic-gravity-wave raypaths, it includes modularity features that make it easier to convert the program to trace rays in other media. It also includes algorithms developed by Jones (1982) for reflecting rays from arbitrary terrain surfaces. In addition, we have developed real-time graphics routines and facilitated operation from time-share graphics terminals as that technology has advanced.

1.9 Scope of This Report

This report documents only the ray-tracing program HARPA, its supporting subroutines, and its various forms of input and output. The main intent of this report is to show what HARPA can do and to explain how to use it. We illustrate its capabilities with a comprehensive sample case. We also show how to extend and modify the program to the user's specific needs.

Not documented here are supplementary programs that we have designed to plot properties of the atmospheric models and to process the computer-readable output of HARPA. Examples of such programs are packages to plot range vs. elevation angle of transmission, range vs. travel time, and amplitude calculations (Jones et al., 1984). We have not documented here our programs for editing input to HARPA or our procedure files for running it on our computer. Nevertheless, the package documented here is self-contained and has everything needed to compute and display raypaths through arbitrary three-dimensional model atmospheres.

Figure 1.1 shows an organization chart of HARPA in relation to its supporting modules. The dotted line encloses the portion documented in this report. Separate reports will document the remaining modules, which are the same for both HARPA and HARPO.

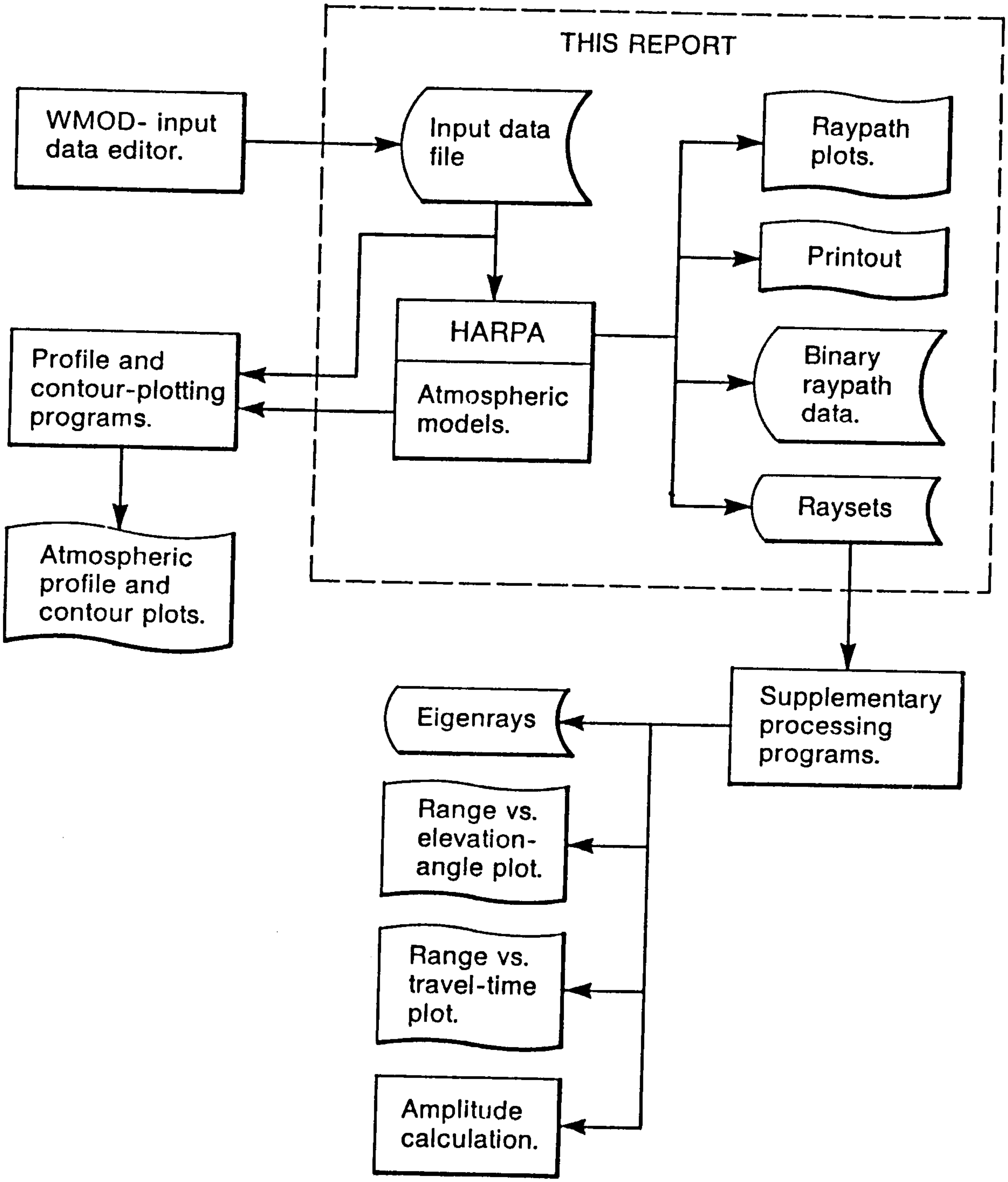


Figure 1.1. Relation of HARPA to its inputs and outputs, as well as its supporting and supplementary programs. The dashed line shows the scope of this report.

2. A Sample Run Illustrating an Application of HARPA

A sample case serves several purposes: it introduces new users to the capabilities of the program in terms of physical models they can understand; it shows new users how to set up and run the program and what output to expect; and it provides a comprehensive test case to exercise the program and make sure everything works on a new machine. New users should run the sample case (provided with the program) first and make sure that the program's output is identical to that shown in this report. Varying the input parameters, one by one, from the sample case is an instructive way to explore the program's capabilities.

The usual procedure for defining models is to fill out "order forms" corresponding to the wind, sound speed, temperature, molecular weight, viscosity/conductivity and terrain models you want to use, then create an "Input Data File" from the information on the order forms. Models can be selected from the general-purpose ones we have created, or you can design your own. The following pages show filled-out forms for the models used by the sample case; blank order forms for all our models are supplied in Appendix B. Using these forms is recommended, even for advanced users, because they make sure that you specify all the required model parameters. They also help you keep track of the models you create.

2.1 The Atmospheric Model for the Sample Case

The atmospheric model described here is designed more to exercise features of the program than to represent any physically realistic situation. It combines a three-dimensional wind and temperature field with a terrain model containing a Lorentzian-shaped ridge. The sample case also includes a simple absorption model based on models of atmospheric viscosity and thermal conductivity.

Refer now to the FORM TO SPECIFY AN ATMOSPHERIC MODEL (Figure 2.1). This form is filled in with the names of all the subroutines required to specify the atmospheric model, for the sample case, including the terrain model, viscosity/ conductivity model and receiver-surface model. Data-set ID numbers uniquely identify the particular set of parameter values used by each subroutine for the sample case. The entire set of models and parameters that constitute the atmospheric model for the sample case is also given a unique ID number, which is S03. The references to W followed by numbers in these forms correspond to specific input data parameters, as described in Section 5.3.

The first subroutine name on this form specifies the acoustic-wave dispersion relation to be used. In the sample case, we specify AWWWL, which means "Acoustic, With Winds, With Losses." This means that we will use a model atmosphere with winds and will calculate acoustic absorption. More efficient versions of the dispersion relation should be selected when wind or absorption models are not used. The remaining subroutine names filled in on this form refer to the atmospheric model subroutines, to be discussed next.

Refer next to the FORM TO SPECIFY INPUT DATA FOR WIND VELOCITY MODEL ULOGZ2 (Figure 2.2). This wind-field model represents a wind profile for the atmospheric boundary layer, neglecting Coriolis forces. The wind has only an eastward component whose magnitude depends only on height above the terrain surface, according to the formula given on the ULOGZ2 order form.

ULOGZ2 requires the user to specify three parameters, z_0, u_*, and k. For the sample case, we choose k=0.35, z_0=1.0 km, and u_*=5.0 m/s. The resulting profile of wind speed is shown in Figure 2.1a. (The program that provided this plot is part of a set of peripheral programs that will be documented in another report.) Because we use no wind-perturbation model, we select the do-nothing wind-perturbation NPWIND.

Refer next to the FORM TO SPECIFY SOUND-SPEED MODEL GAMRTDM (Figure 2.3). We use this sound-speed model because we want to specify a background temperature distribution instead of a sound-speed model directly. GAMRTDM requires no input parameters, but it requires a molecular weight model, discussed next.

Refer next to the FORM TO SPECIFY INPUT DATA FOR ATMOSPHERIC MOLECULAR WEIGHT MODEL MCONST (Figure 2.4). For the sample case, we select the simplest possible model, namely a constant value of 29. Molecular weight is required to convert temperature to sound speed when temperature models are specified.

FORM TO SPECIFY AN ATMOSPHERIC MODEL
(including terrain model)

Name GEORGES RB3 X6437 Date 2-10-86

Atmospheric ID (3 characters) S03

Coordinates of the north pole of the computational coordinate system:

North geographic latitude: 90 rad, km, (deg) (W24)

East geographic longitude: 0 rad, km, deg (W25)

Models:

	Subroutine Name	Data set ID	
Dispersion relation	AWWWL		
Wind velocity	ULOGZ2	3.0	(W102)
Wind-velocity perturbation	NPWIND	0.0	(W127)
Sound speed	GAMRTDM	0.0	(W152)
Sound-speed perturbation	CBLOB2	2.0	(W177)
Temperature	TTANH5	1.0	(W202)
Temperature perturbation	TBLOB2	2.0	(W227)
Molecular weight	MCONST	29.0	(W252)
Terrain	GLORENZ	2.0	(W302)
Terrain perturbation	NPTERR	0.0	(W327)
Viscosity/conductivity	MUARDC	1.0	(W502)
Viscosity/conductivity perturbation	NPABSR	0.0	(W527)
Pressure	PEXP	1.0	(W552)
Pressure perturbation	NPPRES	0.0	(W557)
Receiver surface*	RTERR		
Plot-annotation model*	FULANN		

*The receiver-surface model and plot-annotation model are not considered part of the atmospheric ID

Figure 2.1. Sample of completed form to specify an atmospheric model (including terrain model).

FORM TO SPECIFY INPUT DATA
FOR WIND-VELOCITY MODEL ULOGZ2

A logarithmic wind profile of the atmospheric boundary layer neglecting Coriolis forces. The eastward wind is given by

$$u_\phi = \frac{u_*}{k} \ln \frac{z}{z_0} \qquad \text{for} \qquad z > z_0 e$$

$$u_\phi = \frac{u_*}{k} \frac{z}{z_0 e} \qquad \text{for} \qquad z \leqslant z_0 e \quad ,$$

where $z = G(r,\theta,\phi)$ is determined by the terrain model and is the height above or some kind of distance from the terrain, depending on the terrain model, and r is the radial coordinate of the ray point.

Specify--

the model check for ULOGZ2 = 6.0 (W100)

the input data-format code = ________ (W101)

an input data-set identification number = 3.0 (W102)

an 80-character description of the wind velocity profile:

LOGARITHMIC EASTWARD WIND PROFILE, U* = .5 M/S, ZØ = 1 KM

the reference wind speed, u_* = 5 km/s, (m/s) (W103)

von Kármán's constant, k = .35 (W104) (.35 recommended)

the roughness height, z_0 = 1 km (W105)

OTHER MODELS REQUIRED: Any wind-perturbation model. Use NPWIND if no perturbation is desired.

($\overline{uw} = -u_*^2$ is the surface stress at the ground.)

Figure 2.2. Sample of completed form to specify input data for wind-velocity model ULOGZ2.

FORM TO SPECIFY
SOUND-SPEED MODEL GAMRTDM

This model specifies sound speed in terms of a background temperature model using

$$C^2 = \frac{\gamma RT}{M} ,$$

where $\gamma = 1.4$, R is the universal gas constant, T is the absolute temperature in Kelvins, and $M(r,\theta,\phi)$ is a model of the mean molecular weight of the atmosphere. See Sec. 6.3 for further description of this model.

Specify --

The model check for GAMRTDM = ____1.0____(W150)

OTHER MODELS REQUIRED: Any background temperature model; any molecular weight model.

Figure 2.3. Sample of completed form to specify sound-speed model GAMRTDM.

FORM TO SPECIFY INPUT DATA FOR
ATMOSPHERIC MOLECULAR-WEIGHT MODEL MCONST

A constant molecular weight (independent of height, longitude, latitude, and time)

Specify--

the model check for MCONST = 1.0 (W250)

the input data-format code = ______ (W251)

an input data-set identification number = 29 (W252)

an 80-character description of the molecular weight:

MOLECULAR WEIGHT = 29

the value of the constant molecular weight, M = 29 (W253)

OTHER MODELS REQUIRED: none.

Figure 2.4. Sample of completed form to specify input data for atmospheric molecular-weight model MCONST.

FORM TO SPECIFY INPUT DATA FOR
TEMPERATURE MODEL TTANH5

This model represents the temperature profile by a sequence of linear segments that are smoothly joined by hyperbolic functions:

$$T = T_o + \frac{c_1}{2}(z-z_o) + \sum_{i=1}^{n} \delta_i \left(\frac{c_{i+1} - c_i}{2}\right) \ln \left\{ \frac{\cosh\left(\frac{z-z_i}{\delta_i}\right)}{\cosh\left(\frac{z_i-z_o}{\delta_i}\right)} \right\} + \frac{c_{n+1}}{2}(z-z_o)$$

$$\frac{dT}{dz} = c_1 + \sum_{i=1}^{n} \left(\frac{c_{i+1} - c_i}{2}\right) \left\{\tanh\left(\frac{z-z_i}{\delta_i}\right) + 1\right\}$$

$$c_i = (T_i - T_{i-1})/(z_i - z_{i-1}) \quad .$$

$z = r - r_e$, where r_e is the Earth radius, and r is the radial coordinate of the ray point. Thus, δ_i is the half-thickness of a region centered at approximately z_i km, in which dT/dz changes from c_i to c_{i+1}. Start by drawing a profile using linear segments and get T_i and z_i from the corners. Then select δ_i to round the corners. The final profile will not go through (T_i, z_i).

Specify--

the model check for TTANH5 = 7.0 (W200)

the input data-format code = ____________ (W201)

an input data-set identification number = 1.0 (W202)

an 80-character description of the model with parameters:

U.S. STANDARD ATMOSPHERE 1962 TEMPERATURE PROFILE

and the profile values:

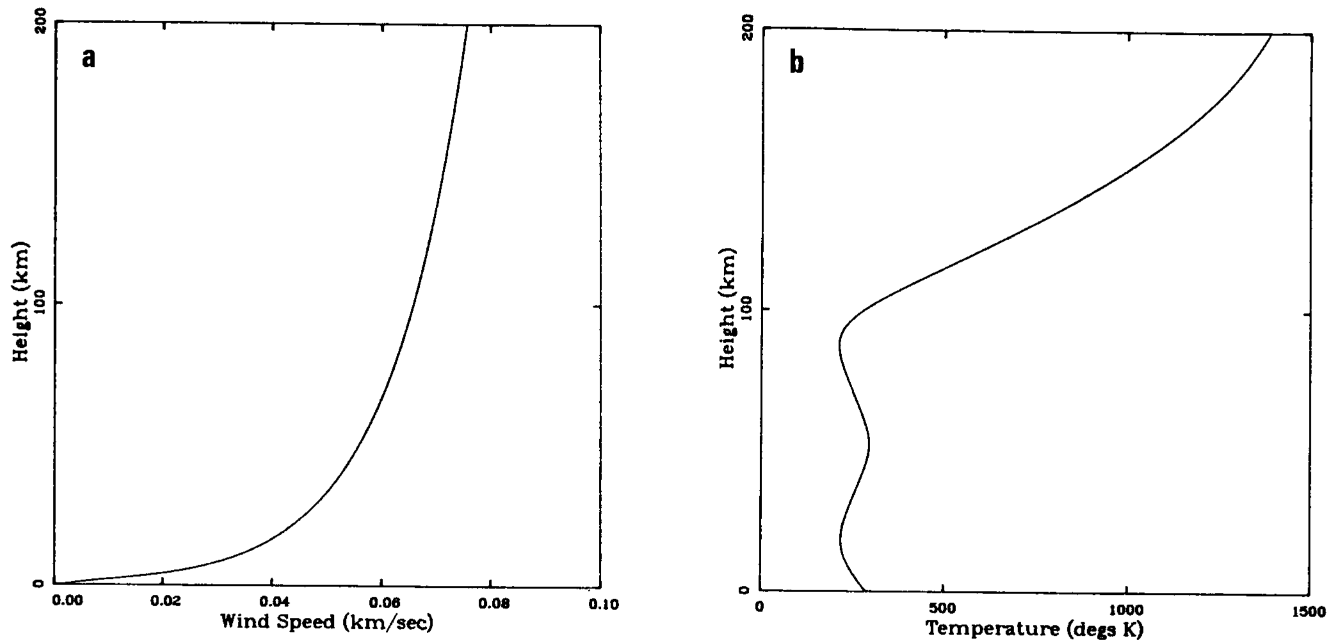
the number of points in the profile -2 = n = 4

the profile:

i	z_i (km,m)	T_i (°K)	δ_i (km,m)
0	0	288	0
1	15	190.5	10
2	52	320	7.5
3	95	191	10
4	165	1451	50
5	300	1586	0

OTHER MODELS REQUIRED: Any temperature-perturbation model. Use NPTEMP if no perturbations are desired. FUNCTION ALCOSH.

Figure 2.5. Sample of completed form to specify input data for temperature model TTANH5.



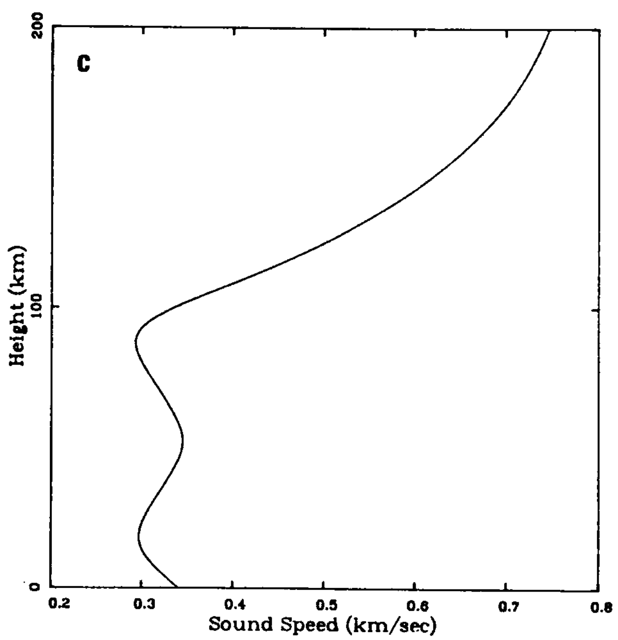


Figure 2.6. Profiles of background (a) eastward wind speed, (b) temperature, and (c) sound speed used in the sample case. The program for plotting these profiles is not supplied with HARPA but is part of a set of supplementary programs documented elsewhere.

Refer next to the FORM TO SPECIFY INPUT DATA FOR TEMPERATURE MODEL TTANH5 (Figure 2.5). This "background" temperature model is a continuous approximation to the 1962 U.S. Standard Atmosphere (USSA) temperature profile (Valley, 1965). The model's parameters, shown on the TTANH5 form, have been selected to give a smooth representation of the USSA profile, which actually has "corners." Figures 2.6b and 2.6c show the resulting temperature and sound-speed profiles. TTANH5 is a very flexible model that can be used to match virtually any temperature profile with linear segments that join smoothly.

Superimposed on the height-dependent background temperature model (TTANH5) are two "perturbations." One is expressed as a temperature perturbation (TBLOB2), and the other is expressed as a sound-speed perturbation (CBLOB2).

Refer now to the FORM TO SPECIFY INPUT DATA FOR ATMOSPHERIC TEMPERATURE-PERTURBATION MODEL TBLOB2 (Figure 2.7). This temperature-perturbation model is in general a three-dimensional blob with Gaussian cross sections in all three dimensions, centered at any latitude, longitude, and height. The formula is given on the TBLOB2 order form.

For the sample case, we suppress the vertical temperature dependence (by setting W_z=0), making the perturbation a vertical cylinder with Gaussian cross sections in the two horizontal dimensions. For the sample case, we locate the cylinder at longitude 50 km east and latitude 105 km north, and set its E-W (zonal) Gaussian width at 25 km and its N-S (meridional) width at 50 km. Its maximum fractional temperature perturbation is 0.5 (50%). Notice that the form allows us to specify some of the model parameters in various units (such as latitude in kilometers); the program will automatically convert to the units (radians, in this example) it uses for computations.

Refer next to the FORM TO SPECIFY INPUT DATA FOR SOUND-SPEED PERTURBATION MODEL CBLOB2 (Figure 2.8). This sound-speed perturbation model is of the same form as TBLOB2, and its formula is shown on the CBLOB2 order from. For the sample case, we center a 50% increase in sound speed at 125 km height, longitude 250 km east, and latitude 335 km north. The Gaussian widths are 25 km in height, 50 km N-S, and 25 km E-W.

Figure 2.9 shows sound-speed contours in a horizontal slice at 125 km height, and Figure 2.10 shows sound-speed contours in a vertical slice near the centers of the two perturbations. (The contour-plotting routine is also part of the supplementary program set.)

The final part of the atmospheric model specifies the method used to compute acoustic absorption parameters required by the dispersion-relation subroutine AWWWL. In the sample case, we use a viscosity/conductivity subroutine called MUARDC, which requires a model of pressure.

FORM TO SPECIFY INPUT DATA FOR
ATMOSPHERIC TEMPERATURE-PERTURBATION MODEL TBLOB2

An increase (or decrease) in temperature in a localized region that decays in a Gaussian manner in all three spatial directions.

$$T(r,\theta,\phi) = T_0(r,\theta,\phi)\ \{1 + \Delta\ \exp[-(\frac{z-z_0}{W_z})^2 - (\frac{\theta-\theta_0}{W_\theta})^2 - (\frac{\phi-\phi_0}{W_\phi})^2]\}$$

$T_0(r,\theta,\phi)$ is the temperature specified by a temperature model. (r,θ,ϕ) are the coordinates of the ray point in an Earth-centered spherical polar coordinate system. $\theta_0 = \pi/2 - \lambda_0$ and $z = r - r_e$, where r_e is the Earth radius. Specify--

the model check for subroutine TBLOB2 = 2.0 (W225)

the input data-format code = ________ (W226)

an input data-set identification number = 2.0 (W227)

an 80-character description for the temperature-perturbation model, including description of parameter values:

50% CYLINDRICAL INCREASE IN TEMPERATURE AT 105 KM N., 105 KM W

the strength of the increase (or decrease), Δ = 0.5 (W228)

the height of maximum effect, z_0 = 100 km (W229)

the latitude of maximum effect, λ_0 = 105 rad, deg, (km) N (W230)

the longitude of maximum effect, ϕ_0 = -105 rad, deg, (km) E (W231)

the Gaussian width in height of the effect, W_z = 0 km (W232)*

the meridional width of the effect, W_θ = 50 rad, deg, (km) (W233)*

the zonal width of the effect, W_ϕ = 25 rad, deg, (km) (W234)*

OTHER MODELS REQUIRED: none.

* Setting W_z, W_θ, or W_ϕ = zero results in no space variation in that direction.

Figure 2.7. Sample of completed form to specify input data for atmospheric temperature perturbation model TBLOB2.

FORM TO SPECIFY INPUT DATA FOR SOUND-SPEED
PERTURBATION MODEL CBLOB2

An increase (or decrease) in sound speed in a localized region that decays in a Gaussian manner in all three spatial directions.

$$C^2(r,\theta,\phi) = C_o^2(r,\theta,\phi)\left(1 + \Delta \exp\left\{-\left(\frac{z-z_o}{W_z}\right)^2 - \left(\frac{\theta-\theta_o}{W_\theta}\right)^2 - \left(\frac{\phi-\phi_o}{W_\phi}\right)^2\right\}\right)$$

$C_o^2(r,\theta,\phi)$ is the square of the sound speed specified by a sound-speed model. (r,θ,ϕ) are the coordinates of the ray point in an Earth-centered spherical polar-coordinate system. $\theta_o = \pi/2 - \lambda_o$ and $z = r - r_e$, where r_e is the Earth radius.

Specify--

the model check for subroutine CBLOB2 = 2.0 (W175)

the input data-format code = ______ (W176)

an input data-set identification number = 2.0 (W177)

an 80-character description for the sound-speed perturbation model, including description of parameter values:

50% INCREASE IN SQ. SOUND SPEED AT 125 KM HT, 335 KM N, 125 KM E

the strength of the fractional increase (or decrease), Δ = 0.5 (W178)

the height of maximum effect, z_o = 125 km (W179)

the latitude of maximum effect, λ_o = 335 rad, deg, (km) N (W180)

the longitude of maximum effect, ϕ_o = 125 rad, deg, (km) E (W181)

the Gaussian width in height of the effect, W_z = 25 km (W182)*

the meridional width of the effect, W_θ = 50 rad, deg, (km) (W183)*

the zonal width of the effect, W_ϕ = 25 rad, deg, (km) (W184)*

OTHER MODELS REQUIRED: none.

* Setting W_z, W_θ, or W_ϕ = zero results in no space variation in that direction.

Figure 2.8. Sample of completed form to specify input data for sound-speed perturbation model CBLOB2.

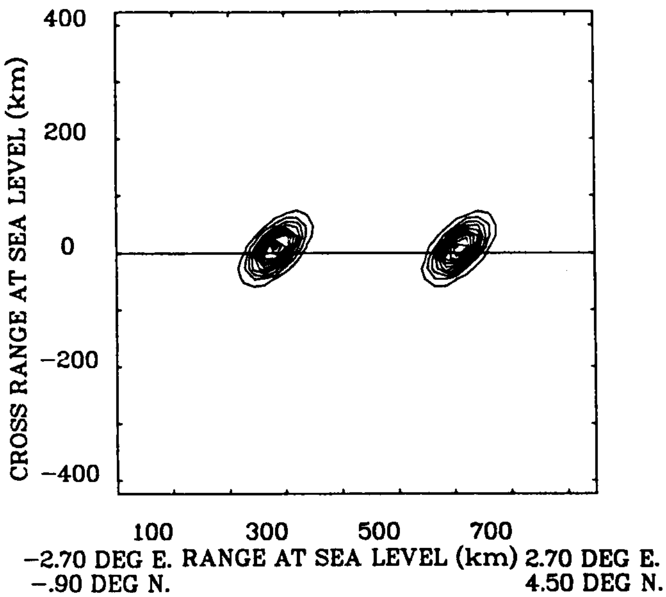


Figure 2.9. A plan view of the sound-speed contours in a horizontal slice through the sample-case atmospheric model at an altitude of 125 km above sea level. The contours show the perturbations caused by TBLOB2 (left) and CBLOB2 (right). The horizontal line across the center of the plot corresponds to the line L-R in Figure 2.18.

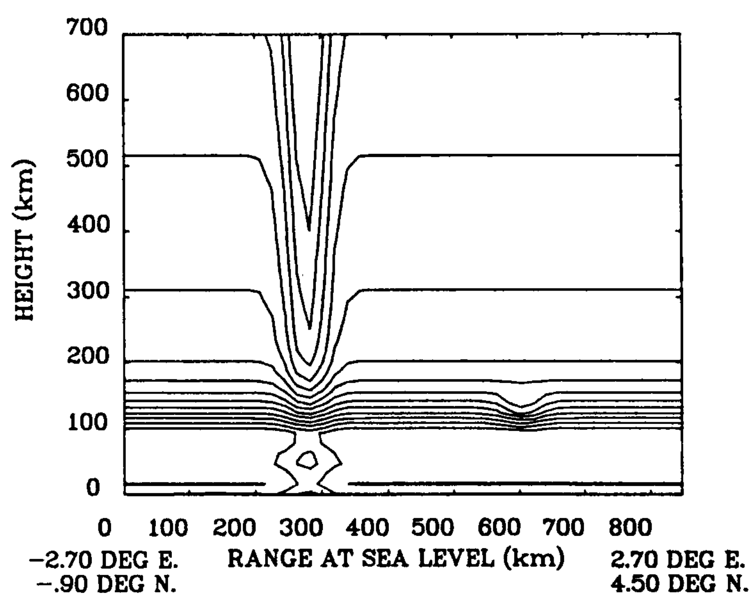


Figure 2.10. Sound-speed contours in a vertical slice through the sample-case model, showing the perturbations caused by models TBLOB2 (left) and CBLOB2 (right) to an otherwise horizontally stratified atmosphere. The plane of the figure corresponds to the line L-R in Figure 2.18.

Refer next to the FORM TO SPECIFY INPUT DATA FOR VISCOSITY/CONDUCTIVITY MODEL MUARDC (Figure 2.11). This model gives a formula (shown on the form) devised by the Air Research and Development Command (ARDC) (NOAA et al., 1976) for atmospheric viscosity and thermal conductivity. Its variable parameters are viscosity constant (β), Sutherland's constant (S), and Prandtl number (P_n). For the sample case, $\beta = 1.45 \times 10^{-6}$, S = 110.4, and P_n=.733.

Refer next to the FORM TO SPECIFY INPUT DATA FOR PRESSURE MODEL PEXP (Figure 2.12). This model specifies an exponential decrease of pressure with height. The variable parameters are the pressure p_0 at sea level (in Newtons per square meter) and the pressure scale height H (in kilometers). For the sample case, $p_0 = 1.01328 \times 10^5$, and H=8.5 km.

Refer next to the FORM TO SPECIFY INPUT DATA FOR TERRAIN MODEL GLORENZ (Figure 2.13). This terrain model superimposes a Lorentzian-shaped ridge on a spherical earth. The ridge, defined by the formula on the GLORENZ order form, runs along a latitude line chosen to be the equator for the sample case. The half width of the ridge is 30 km, and its height is 2 km.

This completes the specification of the atmosphere (and terrain) model for the sample case. Now we specify what raypaths we want to calculate through this model.

2.2 The Ray-Tracing Order Form for the Sample Case

Refer now to the FORM TO SPECIFY INPUT DATA FOR A THREE-DIMENSIONAL RAYPATH CALCULATION (Figure 2.14). The form has been filled out with the values for the sample case. We want to transmit rays from a height 13 km above the earth's surface (as specified in the terrain model), at a latitude of 200 km north, and longitude of zero. The acoustic frequency is 0.05 Hz (infrasound), with no stepping in frequency. The azimuth angle of transmission is 45° (northeastward), with no stepping in azimuth. The elevation angle is stepped from -20° to +140° in steps of 5°. (If azimuth and frequency stepping were used, elevation-angle stepping would be performed first, then azimuth angle, then frequency.)

We want to keep track of ray intersections with a receiver surface 5 km above the terrain, and we want to stop tracing rays that go above 500 km alti-

FORM TO SPECIFY INPUT DATA FOR
VISCOSITY/CONDUCTIVITY MODEL MUARDC

This subroutine calculates the atmospheric molecular viscosity using the ARDC formula for viscosity and calculates atmospheric thermal conductivity from the value of viscosity using a Prandtl number specified by the user. This model is used only to calculate acoustic absorption when either AWWWL or ANWWL is used.

The ARDC formula for viscosity is (U.S. Standard Atmosphere, 1976, p. 19, NOAA, NASA, USAF, U.S. Government Printing Office, Washington, D.C., October 1976)

$$\mu = \beta\ T^{3/2}/(S+T) \ ,$$

where T is the atmospheric temperature in Kelvins.

The atmospheric thermal conductivity using the Prandtl approximation (e.g., Francis Weston Sears, Thermodynamics, Addison-Wesley, 1956, pp. 267-9) is

$$\kappa = \gamma\ R\mu/((\gamma-1)M\ Pr),$$

where γ is the ratio specific heats = 1.4,
R is the universal gas constant,
and M is the mean atmospheric molecular weight.
Specify --

the model check for subroutine MUARDC = 1.0 (W500)

the input data-format code = ________ (W501)

an input data-set identification number = 1.0 (W502)

an 80-character description for the absorption model, including description of parameter values:

ARDC VISCOSITY/THERMAL CONDUCTIVITY MODEL

the viscosity constant, β = 1.458×10^{-6} kg s^{-1} m^{-1} $K^{-1/2}$ (W503)

(1.458×10^{-6} kg s^{-1} m^{-1} $K^{-1/2}$ suggested)

Sutherland's constant, S = 110.4 Kelvins (W504)

(110.4 Kelvins suggested)

Prandtl number, Pr = .733 (W505) (0.733 suggested)

OTHER MODELS REQUIRED: Any atmospheric temperature model and any atmospheric molecular weight model.

Figure 2.11. Sample of completed form to specify input data for viscosity/conductivity model MUARDC.

FORM TO SPECIFY INPUT DATA FOR
PRESSURE MODEL PEXP

This model is used only to calculate absorption when either AWWWL or ANWWL is used. The pressure is given by

$$P = P_0 \exp(-z/H),$$

where z is the height above sea level.

Specify --

the model check for subroutine PEXP = 1.0 (W550)

the input data-format code = ________ (W551)

an input data-set identification number = 1.0 (W552)

an 80-character description for the pressure model, including description of parameter values:

EXPONENTIAL PRESSURE MODEL, SCALE HEIGHT = 8.5 KM

the pressure at sea level, P_0 = 101328. Newtons/m^2 (W553)

(1.01328×10^5 Newtons/m^2 suggested)

the pressure scale height, H = 8.5 (km) m (W554)

OTHER MODELS REQUIRED: Any pressure-perturbation model. Use NPPRES if no perturbation is desired.

Figure 2.12. Sample of completed form to specify input data for pressure model PEXP.

FORM TO SPECIFY INPUT DATA FOR
TERRAIN MODEL GLORENZ

An east-west Lorentzian-shaped ridge.

$$g(r,\theta,\phi) = h - z ,$$

where

$$h = r - r_e ,$$

$$z = z_0/(1 + ((\theta-\theta_0)/\Delta\theta)^2) + z_B ,$$

$$\theta_0 = \pi/2 - \lambda_0 ,$$

and r_e is the radius of the Earth.

Specify--

the model check number for GLORENZ = 4.0 (W300)

the input data-format code number = ______ (W301)

the input data-set identification number = 2.0 (W302)

an 80-character description of the model including parameters:

RIDGE 2-KM HIGH, 30-KM WIDE ALONG EQUATOR

the height of the ridge, z_0 = 2 (km), m (W303)

the latitude of the ridge center, λ_0 = 0 rad, deg, km (W304)

the half-width of the ridge, $\Delta\theta$ = 30 rad, deg, (km) (W305)

base of the ridge (negative if below sea level) z_B = 0 m, km (W306)

OTHER MODELS REQUIRED: Any terrain-perturbation model. Use NPTERR if no perturbation is desired.

Figure 2.13. Sample of completed form to specify input data for terrain model GLORENZ.

FORM TO SPECIFY INPUT DATA FOR A
THREE-DIMENSIONAL RAYPATH CALCULATION

Name________________

Atmospheric ID (3 characters) S03   Date 2-10-86

Title (77 characters) SAMPLE CASE FOR HARPA DOCUMENTATION

Transmitter: Height	13	(km) nm, ft (W3)
	✓	above terrain
	___	above sea level
Latitude	200	rad, deg, km (W4)
Longitude	0	rad, deg, km (W5)
Frequency, initial	.05	rad/s, Hz, s (W7)
final	___	(W8)
step	___	(W9)
Azimuth angle, initial	45	rad, (deg) clockwise of north (W11)
final	___	(W12)
step	___	(W13)
Elevation angle, initial	-20	rad, (deg) (W15)
final	140	(W16)
step	5	(W17)
Receiver: Height	5	(km), nm, ft (W20)
	___	above sea level (rcvr model RHORIZ)
	✓	above terrain (rcvr model RTERR)
Distance from origin	___	rad, deg, km (W30) (rcvr model RVERT)
Latitude of origin	___	rad, deg, km (W31) (rcvr model RVERT)
Longitude of origin	___	rad, deg, km (W32) (rcvr model RVERT)
Stop frequency stepping when ray goes out of bounds	0	(W21 = 1.)
Maximum height	500	km (W26)
Minimum height	-1	km (W27)
Maximum range	1000	km (W28)
Maximum number of hops	3	(W22)
Maximum number of steps per hop	1000	(W23)
Maximum allowable error per step	10^{-6}	(W42)
Additional calculations:		= 1. to integrate = 2. to integrate and print
Phase path	2	(W57)
Absorption	2	(W58)
Doppler shift	___	(W59)
Path length	2	(W60)
Printout:	Every 50	steps of the ray trace (W71)
Computer readable output (raysets):	1	(W72 = 1.)
Diagnostic printing:	___	(W73 = 1.)
Suppress all printout	___	(W74 = 1.)

Figure 2.14. Sample of completed form to specify input data for a three-dimensional raypath calculation.

tude, 1 km below the terrain (all rays should reflect from the terrain surface, however), or beyond 1000-km range. We want to stop the ray trace after 3 hops, a hop being defined as an intersection with the receiver surface.

Because we have selected a receiver surface at a fixed height above the terrain, we also fill out the FORM TO SPECIFY INPUT DATA FOR RECEIVER-SURFACE MODEL RTERR, as shown in Figure 2.15.

We set the maximum number of steps per hop to 1000 (usually a large number that we don't expect to be exceeded under normal conditions but which guards against accidents). We set the maximum allowable integration error per step to 10^{-6}, which means that integrated quantities (that vary monotonically) are computed with at least that relative accuracy. We want to integrate and print phase path, path length, and absorption, but not to calculate Doppler shift (zero for the sample case, which has no time-dependent models). The printed output will display the raypath status every 50th step, in addition to printing at special events, such as reflections, apogees (turnovers), and perigees. Machine-readable "raysets" will also be produced.

2.3 Rayplot Order Form for the Sample Case

Two kinds of raypath plots are available and are specified for the sample case on the FORM TO SPECIFY INPUT PARAMETERS FOR PLOTTING A PROJECTION OF THE RAYPATH (Figures 2.16 and 2.17). The same form is used twice: once for a projection of raypaths on a vertical plane and once for a projection on a horizontal plane.

The vertical plane is specified by the geographic coordinates of its left and right edges and the height above ground of the bottom of the graph. In the sample case, we want the left edge to be at latitude -100 km (south) and longitude -300 km (west); the right edge is to be at latitude 500 km (north), longitude 300 km (east). Thus, the plane of the vertical projection coincides with the plane of initial ray transmission. The bottom of the graph is to be at ground level. We specify tick marks every 100 km, and we want registration marks (the top of the graph) at 300 km height.

The horizontal projection plane is specified by the location of the centers of its left and right edges. For the sample case, we select the left and

FORM TO SPECIFY INPUT DATA
FOR RECEIVER-SURFACE MODEL RTERR

A receiver-surface model in which the receiver surface is a fixed height above the terrain surface.

$$f(r,\theta,\phi) = g(r,\theta,\phi) + z_R$$

$$\frac{\partial f}{\partial r} = \frac{\partial g}{\partial r}, \quad \frac{\partial f}{\partial \theta} = \frac{\partial g}{\partial \phi}, \quad \frac{\partial f}{\partial \phi} = \frac{\partial g}{\partial \phi},$$

where $g(r,\theta,\phi)$ and its derivatives are specified in common block /GG/ by the terrain model.

Specify--

the model check number for subroutine RTERR = ~~3.0~~ 2.0 (W275)

the input data-format code number = ____________ (W276)

an 80-character description of the model including parameters:

RECEIVER SURFACE 5 KM ABOVE TERRAIN

the height of the receiver surface above the terrain, z_R = 5 km (W20)

OTHER MODELS REQUIRED: Any terrain model.

Figure 2.15. Sample of completed form to specify input data for receiver-surface model RTERR.

FORM TO SPECIFY INPUT PARAMETERS FOR PLOTTING A
PROJECTION OF THE RAYPATH

Model ID: S03

Plot directly during raypath calculations ✓, or

plot from precomputed raypaths ______

in disk file ______

Normal or apogee plots:

Normal ✓ (W80=0.0)

Plot apogees only ______ (W80=1.0)

Projection:

Vertical plane, polar plot, rectangular expansion ✓ (W81=1.0)

Horizontal plane, lateral expansion ______ (W81=2.0)

Vertical plane, polar plot, radial expansion ______ (W81=3.0)

Vertical plane, rectangular plot ______ (W81=4.0)

Superimpose these raypath plots on the graph of the previous runset:

Yes ______ (W81 negative.)

No ✓ (W81 positive.)

Vertical or lateral expansion factor 1 (W82)

Coordinates of the left edge of the graph:

Latitude = -100 (rad, deg, km) north (W83)

Longitude = -300 (rad, deg, km) east (W84)

Coordinates of the right edge of the graph:

Latitude = 500 (rad, deg, km) north (W85)

Longitude = 300 (rad, deg, km) east (W86)

Distance between horizontal tick marks = 100 rad, deg, km (W87)

Height above sea level of bottom of graph = 0 km (W88)

Height above sea level of top of graph = 300 km (W89)

Distance between vertical tick marks = 100 km (W96)

Figure 2.16. Sample of completed form to specify input parameters for plotting a projection of the raypath.

FORM TO SPECIFY INPUT PARAMETERS FOR PLOTTING A PROJECTION OF THE RAYPATH

Model ID: S03

Plot directly during raypath calculations ✓, or

plot from precomputed raypaths ______

in disk file ______

Normal or apogee plots:

Normal ✓ (W80=0.0)

Plot apogees only ______ (W80=1.0)

Projection:

Vertical plane, polar plot, rectangular expansion ______ (W81=1.0)

Horizontal plane, lateral expansion ✓ (W81=2.0)

Vertical plane, polar plot, radial expansion ______ (W81=3.0)

Vertical plane, rectangular plot ______ (W81=4.0)

Superimpose these raypath plots on the graph of the previous runset:

Yes ______ (W81 negative.)

No ✓ (W81 positive.)

Vertical or lateral expansion factor 3 (W82)

Coordinates of the left edge of the graph:

Latitude = -100 (rad, deg, (km)) north (W83)

Longitude = -300 (rad, deg, (km)) east (W84)

Coordinates of the right edge of the graph:

Latitude = 500 (rad, deg, (km)) north (W85)

Longitude = 300 (rad, deg, (km)) east (W86)

Distance between horizontal tick marks = 100 rad, deg, (km) (W87)

Height above sea level of bottom of graph = ______ km (W88)

Height above sea level of top of graph = ______ km (W89)

Distance between vertical tick marks = ______ km (W96)

Figure 2.17. Sample of completed form to specify input parameters for plotting a projection of the raypath.

right edges of the horizontal plot to coincide with the coordinates of the left and right edges of the vertical plot, as specified above.

Figure 2.18 shows a plan view of the region of the earth's surface near latitude zero, longitude zero, including the transmitter location, the ray-launch azimuth, the locations of the centers of the CBLOB2 and TBLOB2 perturbations, the ridge along the equator, and the locations of the left and right edges of the two plot projections.

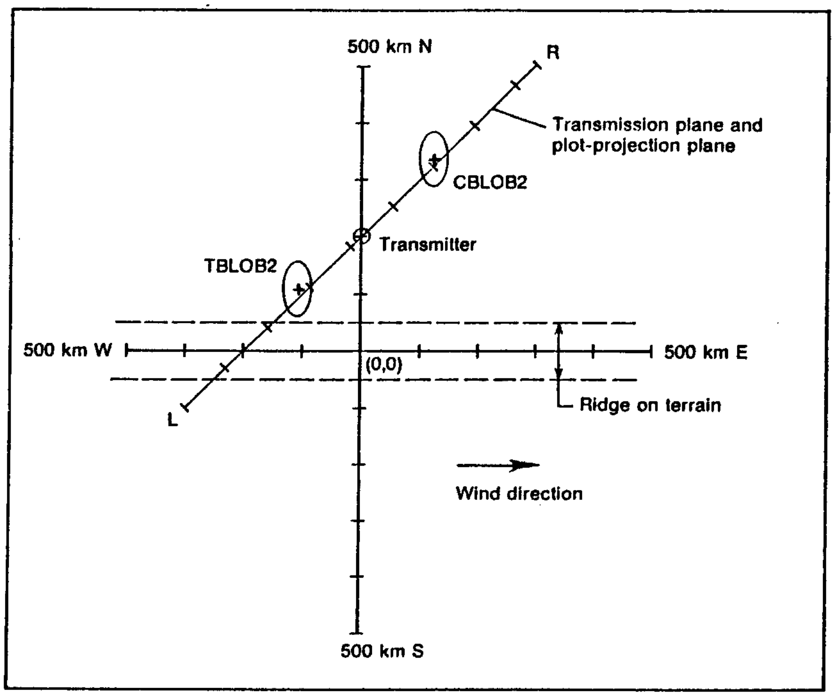


Figure 2.18. Plan view of the major features of the sample-case model, the transmitter location, the transmission direction, and the plot-projection plane.

2.4 Setting Up the Input Data File (W Array) for the Sample Case

Now that we have defined all the parameters needed to specify the atmospheric model, the raypaths desired, as well as the printed, plotted and machine-readable output, we can look at the form in which these input data are communicated to HARPA. To run HARPA, you have to create a file like the one shown for the sample case in Figure 2.19. (Such a file for the sample case comes with the program.) HARPA reads this file into an array named W(n), where n, the array subscript, is the first value on each line (columns 1-3), and W(n) is the second value on each line (columns 4-17).

The values of n corresponding to each input parameter are indicated on each of the input parameter forms we have just looked at. You will notice that not all values of n are listed in the input table for the sample case; those not explicitly defined in the Input Data File assume an initial value that is usually (but not always) zero. The initialization scheme is explained in Section 5.3.1.

Besides the values of n and W(n), the figure contains in columns 18-24 provisions for unit conversion on input, that is, for entering data in various units. For example, the notation AN KM means that angular data has been entered in kilometers and will be converted to radians by the program. For W(3), the height of the transmitter, a T in column 17 converts the height specified to height above the terrain model. Finally, in columns 25-80, descriptive comments identify the data for easy data entry.

There are special values of n (such as zero or negative values) that contain instructions for reading what follows. They will be described in detail in Chapter 5. A negative number in columns 1-3 indicates that tabular or text data follow. A zero in columns 1-3 indicates the end of tabular data or the end of a "run set," the name we give to the input data for a set of ray calculations. For example, the rays for one run set will all appear on a single rayplot. A new run set is necessary to create different plots or to change model parameters. In Figure 2.19, new run sets start with the lines that begin with "S03." For the sample case, each of the two run sets generates a different plot projection of the raypaths through the same atmospheric model. Only the W values that change from the previous run set need be specified; the others remain unchanged.

For now, you need only be aware of this tabular procedure for entering data into HARPA; it would also be instructive to verify that the values entered into the order forms for the sample case correspond to the entries in the W(n) Input Data File.

When you run the program with this input data set, three kinds of output will be produced: a step-by-step printed account of each ray's progress, plots of the raypaths on vertical and horizontal planes, and machine-readable data, including "raysets."

```
col.     col.      col.       col.
1-3      4-17      18-24      25-80
 n       W(n)      UNITS      DESCRIPTION
--------------------------------------------------------------------------------
GEORGES     RB3    X6437
S03-1     SAMPLE CASE FOR HARPA DOCUMENTATION        REV. 2-10-86
  1     6370.               EARTH RADIUS, KM (6370.)
  3       13.              TTRANSMITTER HEIGHT, KM (T=ABOVE TERRAIN)
  4      200.      AN KM    N. TRANSMITTER LATITUDE, KM
  5        0.      AN KM    E. TRANSMITTER LONGITUDE, KM
  7       .05      FQ HZ    INITIAL FREQUENCY, HZ
 11       45.      AN DG    INITIAL AZIMUTH ANGLE, DEG
 15      -20.      AN DG    INITIAL ELEVATION ANGLE, DEG
 16      140.      AN DG    FINAL   ELEVATION ANGLE, DEG
 17        5.      AN DG    STEP IN ELEVATION ANGLE, DEG
 20        5.               RECEIVER HEIGHT, KM
 22        3.               MAXIMUM NUMBER OF HOPS (1.0)
 23     1000.               MAXIMUM NUMBER OF STEPS PER HOP (1000.)
 26      500.               MAXIMUM RAY HEIGHT, KM (500.)
 27       -1.               MINIMUM RAY HEIGHT, KM
 28     1000.               MAXIMUM RANGE, KM
 29  0000100.               DO: EIGRAY/RNG-TIM/RNG-ELV/NEW-PROJ/RAYTRC/CONT/PROF
 33   999.999               MAXIMUM ABSORPTION, DB (999.999)
 42        1.0E-6           SINGLE-STEP INTEGRATION ERROR (1.0E-4)
 44        .1               INITIAL INTEGRATION STEP SIZE, KM (1.0)
 57        2.               PHASE PATH (0=NO; 1=INTEGRATE; 2=INTEGRATE/PRINT)
 58        2.               ABSORPTION (0=NO; 1=INTEGRATE; 2=INTEGRATE/PRINT)
 60        2.               PATH LENGTH (0=NO; 1=INTEGRATE; 2=INTEGRATE/PRINT)
 71       50.               NUMBER OF INTEGRATION STEPS PER PRINT [1.E31]
 72        1.               OUTPUT RAYSETS (1=YES; 0=NO)
 73        0.               DIAGNOSTIC PRINTOUT (1=YES; 0=NO)
 74        0.               PRINT RAY STEPS (0=YES; 1=NO)
 75       .15               FULANN LETTER HEIGHT [0.15 IN]
 76        0.               BINARY RAY OUTPUT (1=YES; 0=NO)
 77       57.               LINES PER PAGE IN PRINTOUT (66.)
 81        1.               RAYPLOT PROJECTION (1=VERT; 2=HORIZ) PLANE
 82        1.               PLOT-EXPANSION FACTOR [1.0]
 83     -100.      AN KM    N. LATITUDE  OF LEFT PLOT EDGE, KM
 84     -300.      AN KM    E. LONGITUDE OF LEFT PLOT EDGE, KM
 85      500.      AN KM    N. LATITUDE  OF RIGHT PLOT EDGE, KM
 86      300.      AN KM    E. LONGITUDE OF RIGHT PLOT EDGE, KM
 87      100.      AN KM    DISTANCE BETWEEN TIC MARKS, KM
 88        0.               HEIGHT ABOVE SEA LEVEL OF BOTTOM OF GRAPH, KM
 89      300.               HEIGHT ABOVE SEA LEVEL OF TOP OF GRAPH, KM
 96      100.               DISTANCE BETWEEN VERTICAL TIC MARKS, KM
100        6.               ULOGZ2 WIND MODEL CHECK NUMBER
102        3.                BACKGROUND WIND DATA SET ID
103        5.      LN  M     REFERENCE WIND SPEED, M/S
104       35                 VON KARMAN'S CONSTANT
105        1.                ROUGHNESS HEIGHT, KM
150        1.               GAMRTDM SOUND SPEED MODEL CHECK NUMBER
175        2.               CBLOB2 MODEL CHECK NUMBER
177        2.                SOUND SPEED PERTURBATION DATA SET ID
178        .5                FRACTIONAL INCREASE OF SQUARED SOUND SPEED
```

Figure 2.19. Input Data File for the sample case.

```
179       125.                 HEIGHT OF MAXIMUM INCREASE, KM
180       335.     AN KM       N. LATITUDE OF MAXIMUM INCREASE, KM
181       125.     AN KM       E. LONGITUDE OF MAXIMUM INCREASE, KM
182        25.                 GAUSSIAN WIDTH IN HEIGHT OF INCREASE, KM
183        50.     AN KM       N-S WIDTH OF THE INCREASE, KM
184        25.     AN KM       E-W WIDTH OF THE INCREASE, KM
200         7.                TTANH5 TEMPERATURE MODEL CHECK NUMBER
202         1.                 BACKGROUND TEMPERATURE DATA SET ID
225         2.                TBLOB2 MODEL CHECK NUMBER
227         2.                 TEMPERATURE PERTURBATION DATA SET ID
228         .5                 FRACTIONAL TEMPERATURE INCREASE
229         0.                 HEIGHT OF MAXIMUM INCREASE, KM
230       105.     AN KM       N. LATITUDE OF MAXIMUM INCREASE, KM
231      -105.     AN KM       E. LONGITUDE OF MAXIMUM INCREASE, KM
232         0.                 GAUSSIAN WIDTH IN HEIGHT OF INCREASE, KM
233        50.     AN KM       N-S WIDTH OF THE INCREASE, KM
234        25.     AN KM       E-W WIDTH OF THE INCREASE, KM
250         1.                MCONST MOLECULAR WEIGHT MODEL CHECK NUMBER
252        29.                 MOLECULAR WEIGHT DATA SET ID
253        29.                 MOLECULAR WEIGHT
275         2.                RTERR RECEIVER MODEL CHECK NUMBER
300         4.                GLORENZ TERRAIN MODEL CHECK NUMBER
302         2.                 TERRAIN MODEL DATA SET ID
303         2.                 HEIGHT OF THE RIDGE, KM
304         0.                 N. LATITUDE OF THE RIDGE CENTER
305        30.     AN KM       HALF-WIDTH OF THE RIDGE, KM
325         0.                NPTERR NO TERRAIN PERTURBATION
500         1.                MUARDC VISC/COND MODEL CHECK NUMBER
502         1.                 VISC/COND MODEL DATA SET ID
503     1.458E-06              VISCOSITY COEFFICIENT BETA
504     110.4                  SUTHERLAND'S CONSTANT, KELVINS
505      .733                  PRANDTL NUMBER
525         0.                NPABS NO VISC/COND PERTURBATION
550         1.                PEXP PRESSURE MODEL CHECK NUMBER
552         1.                 BACKGROUND PRESSURE MODEL DATA SET ID
553   101328.                  PRESSURE AT SEA LEVEL, N/SQ.M.
554        8.5                 PRESSURE SCALE HEIGHT, KM
575         0.                NPPRES   NO PRESSURE PERTURBATION
 -1                           DATA SUBSET FOR BACKGROUND WIND MODEL
  A  LOGARITHMIC EASTWARD WIND PROFILE, U*=.5 M/S, Z0=1 KM
  0      RETURN TO W ARRAY DATA SET
 -2                           DATA SUBSET FOR WIND PERTURBATION MODEL
  A  NO WIND PERTURBATION
  0      RETURN TO W ARRAY DATA SET
 -3                           DATA SUBSET FOR BACKGROUND SOUND-SPEED MODEL
  A  SOUND SPEED IN TERMS OF TEMPERATURE MODEL
  0      RETURN TO W ARRAY DATA SET
 -4                           DATA SUBSET FOR SOUND-SPEED PERTURBATION MODEL
  A  50% INCREASE IN SQ. SOUND SPEED AT 125KM HT, 335KM N, 125KM E
  0      RETURN TO W ARRAY DATA SET
 -5                           DATA SUBSET FOR TEMPERATURE MODEL
  A  U.S. STANDARD ATMOSPHERE 1962 TEMPERATURE PROFILE
  3     999.0
  LN KM                          LN KM
   0.           288.000         0.
```

Figure 2.19. Input Data File (continued).

```
 15.0000        190.500       10.0000
 52.0000        320.000        7.50000
 95.0000        191.000       10.0000
165.0000       1451.000       50.0000
300.0000       1586.000        0.
999.0000
 0      RETURN TO W ARRAY DATA SET
-6                         DATA SUBSET FOR TEMPERATURE PERTURBATION MODEL
 A   50% CYLINDRICAL INCREASE IN TEMPERATURE AT 105KM N., 105 KM W
 0      RETURN TO W ARRAY DATA SET
-7                         DATA SUBSET FOR MOLECULAR WEIGHT MODEL
 A   MOLECULAR WEIGHT = 29
 0      RETURN TO W ARRAY DATA SET
-8                         DATA SUBSET FOR RECEIVER SURFACE MODEL
 A   RECEIVER SURFACE 5 KM ABOVE TERRAIN
 0      RETURN TO W ARRAY DATA SET
-9                         DATA SUBSET FOR TERRAIN MODEL
 A   RIDGE 2-KM HIGH, 30-KM WIDE ALONG EQUATOR
 0      RETURN TO W ARRAY DATA SET
-10                        DATA SUBSET FOR TERRAIN PERTURBATION MODEL
 A   NO TERRAIN PERTURBATION
 0      RETURN TO W ARRAY DATA SET
-17                        DATA SUBSET FOR VISC/COND MODEL
 A   ARDC VISCOSITY AND THERMAL CONDUCTIVITY MODEL
 0      RETURN TO W ARRAY DATA SET
-18                        DATA SUBSET FOR VISC/COND PERTURBATION MODEL
 A   NO VISCOSITY/CONDUCTIVITY PERTURBATION
 0      RETURN TO W ARRAY DATA SET
-19                        DATA SUBSET FOR BACKGROUND PRESSURE MODEL
 A   EXPONENTIAL PRESSURE MODEL, SCALE HEIGHT = 8.5 KM
 0      RETURN TO W ARRAY DATA SET
-20                        DATA SUBSET FOR PRESSURE PERTURBATION MODEL
 A   NO PRESSURE PERTURBATION
 0      RETURN TO W ARRAY DATA SET
 0                         **********  END OF RUN SET NUMBER 1  ***********
S03-2   SAMPLE CASE FOR HARPA DOCUMENTATION          REV. 2-10-86
 71        0.              NUMBER OF INTEGRATION STEPS PER PRINT
 72        0.              OUTPUT RAYSETS  (1=YES; 0=NO)
 73        1.              DIAGNOSTIC PRINTOUT (1=YES; 0=NO)
 81        2.              RAYPLOT PROJECTION (1=VERT; 2=HORIZ) PLANE
 82        3.              PLOT-EXPANSION FACTOR
 0                         **********  END OF RUN SET NUMBER 2  ***********
```

Figure 2.19. Input Data File (continued).

2.5 Printed Output for the Sample Case

Appendix A shows the complete printed output, or "printout," for the sample case. Sections 2.5.1 and 2.5.2 define the terms and quantities used in the printout.

Page 1 of the printout contains the program title block and a list of all the models used for the first run set. The model list includes subroutine names, a number identifying the set of parameters defining that particular model, and comments describing each model.

Pages 2-4 of the printout reproduce the Input Data File for run set number 1.

Page 5 of the printout is a list of n and W(n) for all nonzero W(n). The values of W(n) have been converted to the units used by the program; for example, angles (like latitude and longitude) input in kilometers have been converted to radians. The standard set of units used by the program are: angles in radians, distances in kilometers, and frequency in radians per second.

Pages 6-10 of the printout are in a columnar format that gives a step-by-step account of each ray's progress, with each line showing important raypath quantities at user-specified intervals along the raypath. The meanings of the quantities printed out are explained in Section 2.5.1. The user can specify how often along the ray a line is printed (we specified every 50 integration steps). In addition, a line is printed out every time a ray experiences a "special event," such as a ground reflection, a turnover (apogee) or turnunder (perigee), crosses the receiver height, or a few other events. Section 2.5.2 explains the exact meanings of all the special events.

Each ray calculation terminates when one of the termination conditions is met, such as the maximum number of hops, maximum range, or maximum number of integration steps, whichever occurs first. In the sample case, all of the rays terminate either because they reached the maximum number of hops specified (3) or because the absorption reached the maximum specified. At the end of each ray calculation, the printout shows how much CPU time was used for that ray computation (2.009 s, in this case).

Look at page 1 of the printout in Appendix A. Verify that the models indicated at the bottom of the page coincide with those we specified as input. Verify that the wave frequency and initial azimuth and elevation angles on page 6 are what we want. Look down the first (ERROR) column of the tabular printout (page 6) and verify that none of the numbers in this column exceeds the maximum allowable single-step integration error, W(42), which for the sample case is 10^{-6}. This means that the numerical integration is proceeding correctly.

Look at the first entries in the ELEVATION columns and verify that the ray starts at the correct height above the terrain (13.00 km in this case), as well as the correct height above sea level (slightly more than 13 km in this case, because of the terrain model). The RANGE column should begin with zero range from the transmitter and indicate the range from the transmitter with successive steps.

A general idea of the sequence of events along this ray can be read in the EVENT column. These notations mark the special events along the raypath, which cause printout regardless of the step number. Reading down this column, we see that this ray begins at the transmitter (XMTR), then crosses the receiver (RCVR) height at 5 km altitude above the terrain, then reflects from the ground (GRND REF), passes through the receiver (RCVR) height once again, then executes an APOGEE, or turnover, and stops when it reaches the receiver height (RCVR) a third time. The ray stopped because we specified a maximum of three hops, or intersections with the receiver height (MAX HOPS).

The numbers in the AZIMUTH DEVIATION columns indicate that the ray deviates from the azimuth of transmission because of the wind. The ELEVATION ANGLE columns show changes in the local wave-normal direction and the elevation of the ray point measured from the transmitter.

PULSE TIME gives the time for a pulse or wave packet to reach that point, and PHASE TIME gives the time for a wave phase front to reach that point. The wave phase can be derived from PHASE TIME by multiplying by the wave frequency in appropriate units (cycles per second to get wave phase in cycles, etc.) and removing the integer part. The PATH LENGTH gives the physical length of the ray path.

The second run set shown in the sample-case printout uses the same initial ray conditions, but it changes the rayplot to a horizontal projection, and it adds diagnostic printout that can be useful for studying the details of terrain or receiver-surface intersections.

2.5.1 Definitions of Quantities Listed in Printout (Appendix A)

AZIMUTH ANGLE OF TRANSMISSION -- Azimuth angle (degrees clockwise from north) of the initial ray-launch direction.

ELEVATION ANGLE OF TRANSMISSION -- Elevation angle (degrees upward) between the initial ray-launch direction and local horizontal at the transmitter.

TRANSMITTER LATITUDE -- East latitude of the transmitter in geographic coordinates.

TRANSMITTER LONGITUDE -- North longitude of the transmitter in geographic coordinates.

FREQUENCY -- Acoustic wave frequency in hertz.

SINGLE-STEP ERROR -- Maximum allowable single-step integration error.

ERROR -- Normalized difference between the wave number k computed by numerical integration and k computed from the dispersion relation [Eq. (6.32)].

EVENT -- Nature of special event along the raypath (see Sec. 2.5.2).

HEIGHT -- Height of the ray point above sea level (or above the terrain).

RANGE -- Great-circle distance, measured at sea level between the transmitter and the ray point.

AZIMUTH DEVIATION (XMTR) -- Azimuth angle of the direction of transmission in degrees clockwise from the great circle between the transmitter and the ray point.

AZIMUTH DEVIATION (LOCAL) -- Azimuth angle of the wave normal in degrees clockwise from the great circle between the transmitter and the ray point.

ELEVATION (XMTR) -- Elevation angle (degrees) of the ray point from local horizontal at the transmitter.

ELEVATION (LOCAL) -- Elevation angle (degrees) of the wave normal from the local horizontal at the ray point.

PULSE TIME -- The time (seconds) required for a wave packet (pulse) to travel from the transmitter to the ray point (Sec. 6.1).

PHASE TIME -- The time (seconds) required for a wave front to travel from the transmitter to the ray point (Sec. 6.1.1).

ABSORPTION -- Decrease in acoustic intensity (dB) from the transmitter to the ray point caused by atmospheric dissipation only (Sec. 6.1.2).

PATH LENGTH -- Geometric length of the ray path (kilometers) from the transmitter to the ray point (Sec. 6.1.3).

2.5.2 Meanings of Special Events Along a Raypath

XMTR -- Ray is at the transmitter.

RCVR -- Ray is at the receiver surface.

GRND REF -- Ray has reflected from the terrain surface.

APOGEE -- Ray has passed through a maximum in height.

PERIGEE -- Ray has passed through a minimum in height.

WAVE REV -- Vertical, southward, or eastward component of the wave vector has changed sign.

MAX LAT -- Ray has passed through a maximum (or minimum) in latitude.

MAX LONG -- Ray has passed through a maximum (or minimum) in longitude.

EXTINC -- Absorption has exceeded the maximum allowable.

MAX HOP -- Ray has executed the requested number of hops (receiver-surface crossings).

MAX RANG -- Ray has exceeded the maximum allowable ground range.

MAX HT -- Ray has exceeded the maximum allowable height.

MIN HT -- Ray has gone below the minimum allowable height.

MIN DIST -- Ray has made a closest approach to the receiver surface.

ADDITIONAL EVENTS IN DIAGNOSTIC PRINTOUT

BACK UP0 -- At call to subroutine BACKUP.

BACK UP1 -- Before each numerical integration step in subroutine BACKUP.

GRAZE 0 -- At call to ENTRY point GRAZE in subroutine BACKUP.

GRAZE 1 -- Before each numerical integration step after ENTRY point GRAZE.

BACK UP2 -- After unsuccessfully trying to find a closest approach to the receiver surface.

BACK UP3 -- Before each numerical integration step after BACK UP2.

2.6 Rayplots for the Sample Case

We requested two rayplots, a projection on a vertical plane and a projection on a horizontal plane. These two plots are shown in Figures 2.20 and 2.21. Because we selected the FULANN (full annotation) option in W(75), we have produced a plot with publication-quality lettering. This capability requires the DISSPLA plotting package.

Rayplots are often the most useful output from a ray-tracing program, particularly when the medium is complicated. Depending on the user's plotting and display facilities, rayplots can be produced on paper, microfilm, or video displays.

Because the atmospheric and terrain models used in the sample case cause the acoustic raypaths to behave in complicated ways, a few features of the two rayplot projections call for some explanation. Figure 2.18 shows how the planes of these plots are related to the transmitter location and the features of the atmospheric and terrain models. The letters L and R show the locations of the left and right edges of the two plots; the line connecting the L and R represents the plane of the vertical projection (Figure 2.20) as well as the line across the center of the horizontal projection (Figure 2.21).

The two rayplots show the gross refracting features of this atmospheric model, namely that the temperature gradient bends upgoing rays back toward the ground, and that the eastward wind deflects rays generally downwind. The wind direction is to the right in Figure 2.20 (though not in the plane of the plot) and is down and to the right in Figure 2.21. In the absence of wind, the rays shown in Figure 2.21 would all be coplanar and so would appear as a single straight, horizontal line in that projection, and the plot of Figure 2.20 would exhibit left-right symmetry. By comparing corresponding rays in Figures 2.20 and 2.21, one can get a rough three-dimensional perspective of the raypaths.

The effects of the ridge in the terrain model are not evident in the scales of these plots because the ridge is only 2 km high. The peak of the

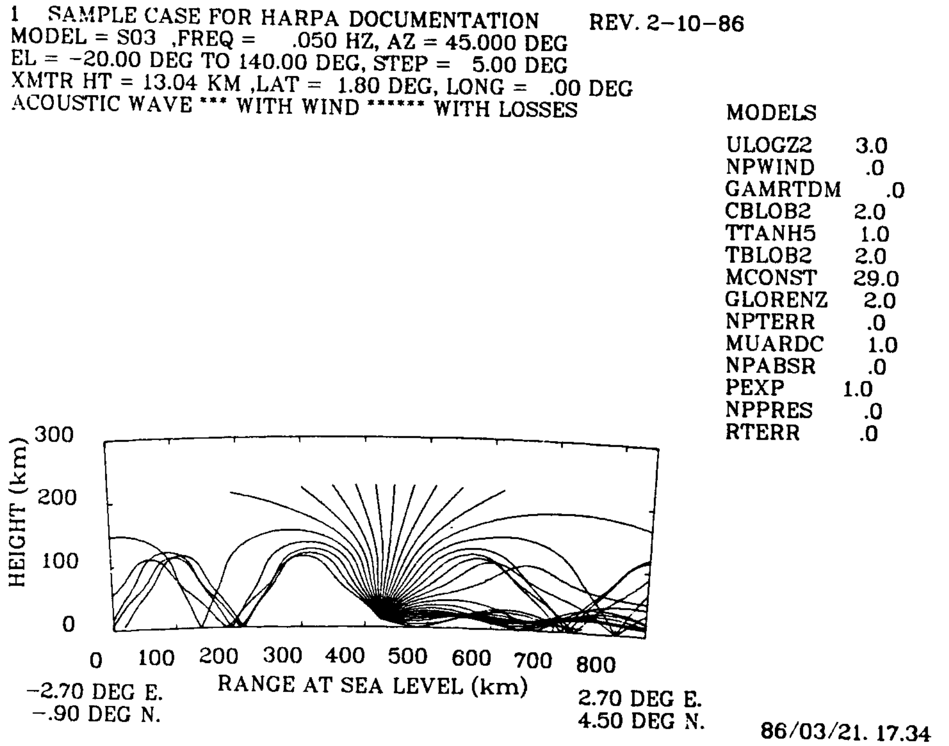


Figure 2.20. Projection of the rays of the sample case onto the vertical plane shown in Figure 2.18.

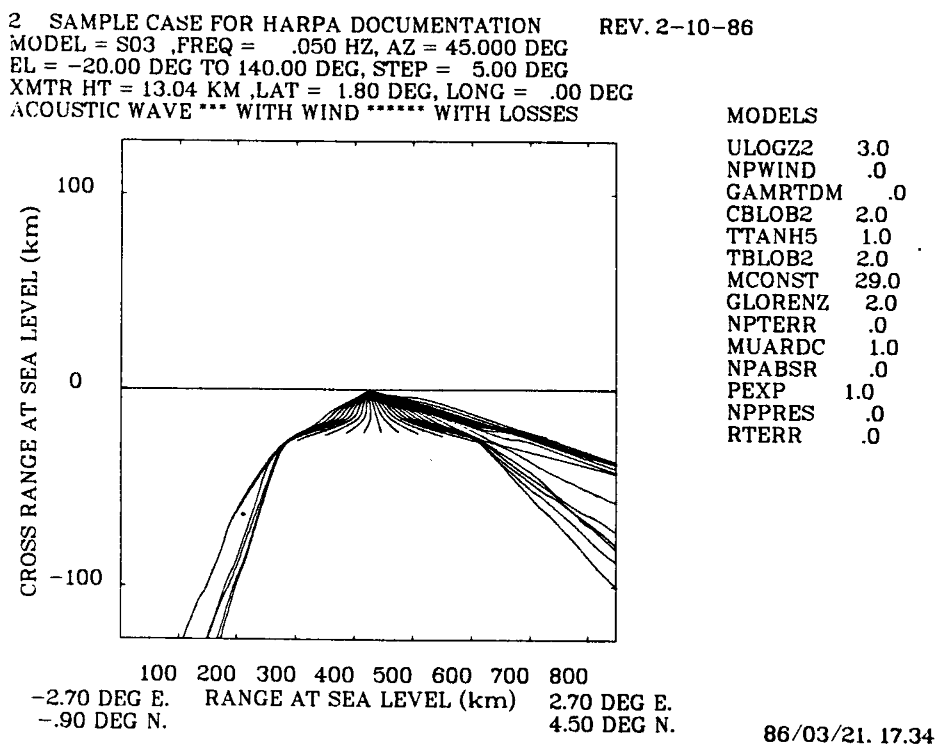


Figure 2.21. Projection of the rays of the sample case onto a horizontal plane whose axis (the horizontal line across the middle of the plot) is shown in Figure 2.18.

ridge intersects the plane of the vertical projection at about 300 km to the left of the transmitter, and rays launched between about 120 and 130 deg elevation reflect from the terrain near the peak of the ridge. They can be observed in detail by launching a dense fan of rays between 120 and 130 deg elevation and magnifying the vertical-projection plot in the vicinity of the ridge. This could be done with just a few modifications of the Input Data File and will be left as an exercise for the user.

2.7 Machine-Readable Output for the Sample Case

HARPA produces two kinds of machine-readable output. One form is called "raysets," which summarize in compressed form some useful ray parameters at each special event (as defined above) along the raypath. The other form of machine-readable output is called "binary raypath data," which permits a complete reconstruction of the raypaths by a supplementary processing program. When stored in machine-readable form (punched cards, magnetic tape, disk files), raysets (as a file named PUNCH) and binary raypath data (as a file named TAPE6) form the input to supplementary processing programs and extend the utility of ray-tracing calculations. Examples are supplementary programs to plot model profiles and contours, to compute amplitude, to plot range versus elevation angle of transmission and range versus travel time, as well as programs that interpolate in elevation angle to estimate eigenrays that reach a specified range. These supplementary capabilities will be documented in another report.

Figure 2.22 shows a portion of the printout of the raysets for the sample case. The complete rayset output for the sample case is given in Appendix A. Each ray begins with a "transmitter rayset," the lines beginning with "S03" in the example. Additional 80-column lines are produced whenever a ray reflects from the terrain surface or crosses the receiver height and at the end of each ray trace. Because of the way hops are counted, two identical raysets are produced each time a ray executes a closest approach to the receiver height.

The compressed rayset format is generally meant to be read by machines, not humans, so it can be rather difficult to inspect for information.

```
S03-1     SAMPLE CASE FOR HARPA DOCUMENTATION        REV.9-23-85
ULOGZ2        3.0 NPWIND          .0    GAMMA RT         .0 CBLOB2        2.0
TTANH5        1.0 TBLOB2         2.0    MCONST         29.0 NO MODL        .0
PEXP          1.0 NPPRES          .0    MUARDC          1.0 NPABSR         .0
GLORENZ       2.0 NPTERR          .0    RTERR            .0 NO MODL        .0
  LOGARITHMIC EASTWARD WIND PROFILE
  50% INCREASE IN SQUARED SOUND SPEED AT 125KM HT, 335KM N, 125KM E.
  U.S. STANDARD ATMOSPHERE 1962 TEMPERATURE PROFILE
  50% CYLINDRICAL INCREASE IN TEMPERATURE AT 105KM N., 105 KM W.
  MOLECULAR WEIGHT = 29
  RIDGE 2-KM HIGH, 30-KM WIDE ALONG EQUATOR
S03      130440   1799      0     50000      3142   4500000  -2000000         0    0 3T
    50367   294887 -3840 -3838-12169   9196124   9196123        0    0     0    0 1R
      313   563108 -2735 -2735 10693  17228767  17228768        0    0     0    0 2G
    50270   831414 -2343 -2340 12148  25260761  25260764        0    0     0    0 2R
    50112  3073288 -4374 -4375-12129  94307633  94307634        0    0     0    0 3R
S03      130440   1799      0     50000      3142   4500000          0       0    0 3T
   130453  2250470 -4938 -4939    -3  68990961  68990962        0    0     0    0 1M
   130453  2250470 -4938 -4939    -3  68990961  68990962        0    0     0    0 2M
   130423  4501421 -4932 -4941    -1 137987009 137987013        0    0     0    0 3M
S03      130440   1799      0     50000      3142   4500000   2000000         0    0 3T
    50170  1946869 -5322 -5324-12129  59828119  59828103        0    0     0    0 1R
      152  2215510 -4860 -4863 10665  67882811  67882797        0    0     0    0 2G
    50137  2484283 -4498 -4500 12121  75937724  75937711        0    0     0    0 2R
    50070  4727741 -4790 -4799-12126 145006034 145006027        0    0     0    0 3R
S03      130440   1799      0     50000      3142   4500000   4000000         0    0 3T
    50103  3741414 -9536 -4137-35775 117807866 117807837        3    0     0    0 1R
      101  3814687 -9488 -4092 34998 120425195 120425167        3    0     0    0 2G
    50099  3887966 -9442 -4046 35772 123042513 123042487        3    0     0    0 2R
    50043  7937415-10564 -5185-35793 249875267 249875381        6    0     0    0 3R
S03      130440   1799      0     50000      3142   4500000   6000000         0    0 3T
    50104  3788921-10259 -3574-59417 113854747 113854834      840    0     0    0 1R
      103  3820800-10249 -3572 58548 115617591 115617680      840    0     0    0 2G
    50102  3852681-10240 -3563 59414 117380431 117380523      840    0     0    0 2R
    50049  7425637-11689 -5024-59423 230295002 230295121     1573    0     0    0 3R
S03      130440   1799      0     50000      3142   4500000   8000000         0    0 3T
  2245590   880081-14390-14394 64191  50090000  50089965999999      0     0    0 1E
S03      130440   1799      0     50000      3142   4500000  10000000         0    0 3T
  2238850   451046208869 28867 65050  49890000  49889964999999      0     0    0 1E
S03      130440   1799      0     50000      3142   4500000  12000000         0    0 3T
    53003  2990950200125   978-57476 115367955 115368106     2511    0     0    0 1R
     2797  3023605200153   553 54429 117198670 117198821     2511    0     0    0 2G
    52607  3056827200187   580 56842 119028620 119028770     2511    0     0    0 2R
    50134  5991692205807  6198-56730 234573129 234573354     3731    0     0    0 3R
S03      130440   1799      0     50000      3142   4500000  14000000         0    0 3T
    58098  2555846202632 -6592-36018 108393614 108393701       11    0     0    0 1R
     6338  2633761202904 -6666 28584 111287389 111287482       11    0     0    0 2G
    55009  2713368203173 -6406 33954 114170553 114170650       11    0     0    0 2R
    50143  5658781212816  3229-33478 231386220 231386499       18    0     0    0 3R
```

Model Identification Header

Transmitter Raysets

Receiver Raysets

Figure 2.22. Rayset output for the first run set of the sample case, using an elevation-angle increment of 20°.

However, since this is occasionally necessary, Figures 2.23 and 2.24 provide the key for reading rayset printouts.

Notice that the last 3 columns preceding the hop identifier in the receiver raysets contain all zeroes for the sample case. The first of these columns is for Doppler shift, which is zero because we did not use a time-varying model atmosphere. The transverse polarization is always zero for pure acoustic waves, but is nonzero for acoustic-gravity waves (Jones et al., 1982, Sec. 4.1).

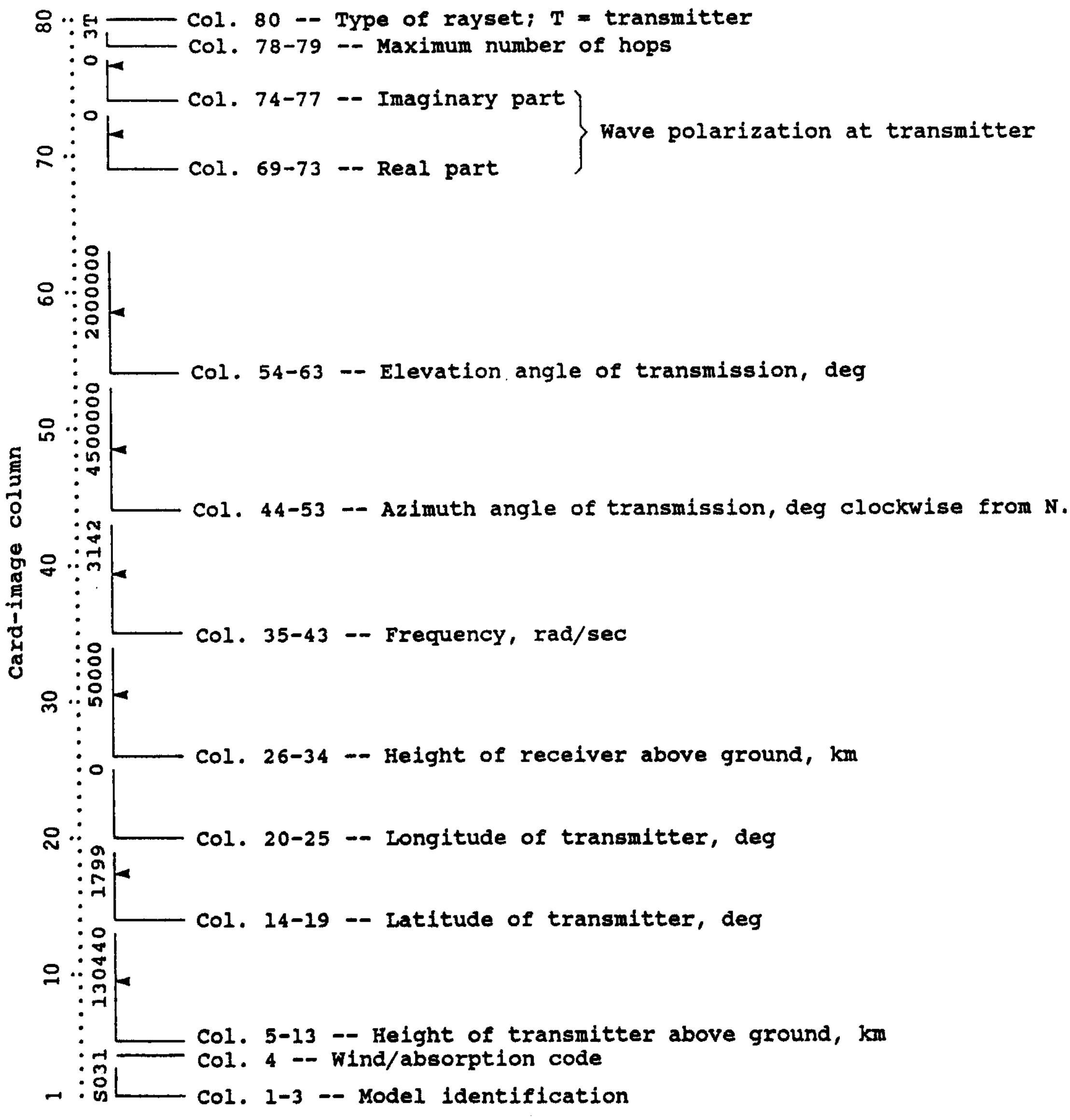


Figure 2.23. Definitions and format for a transmitter rayset. *Wind/absorption code: 0 = no wind, no absorption; 1 = with wind, no absorption; 2 = no wind, with absorption; 3 = with wind, with absorption and ▲ = decimal point.

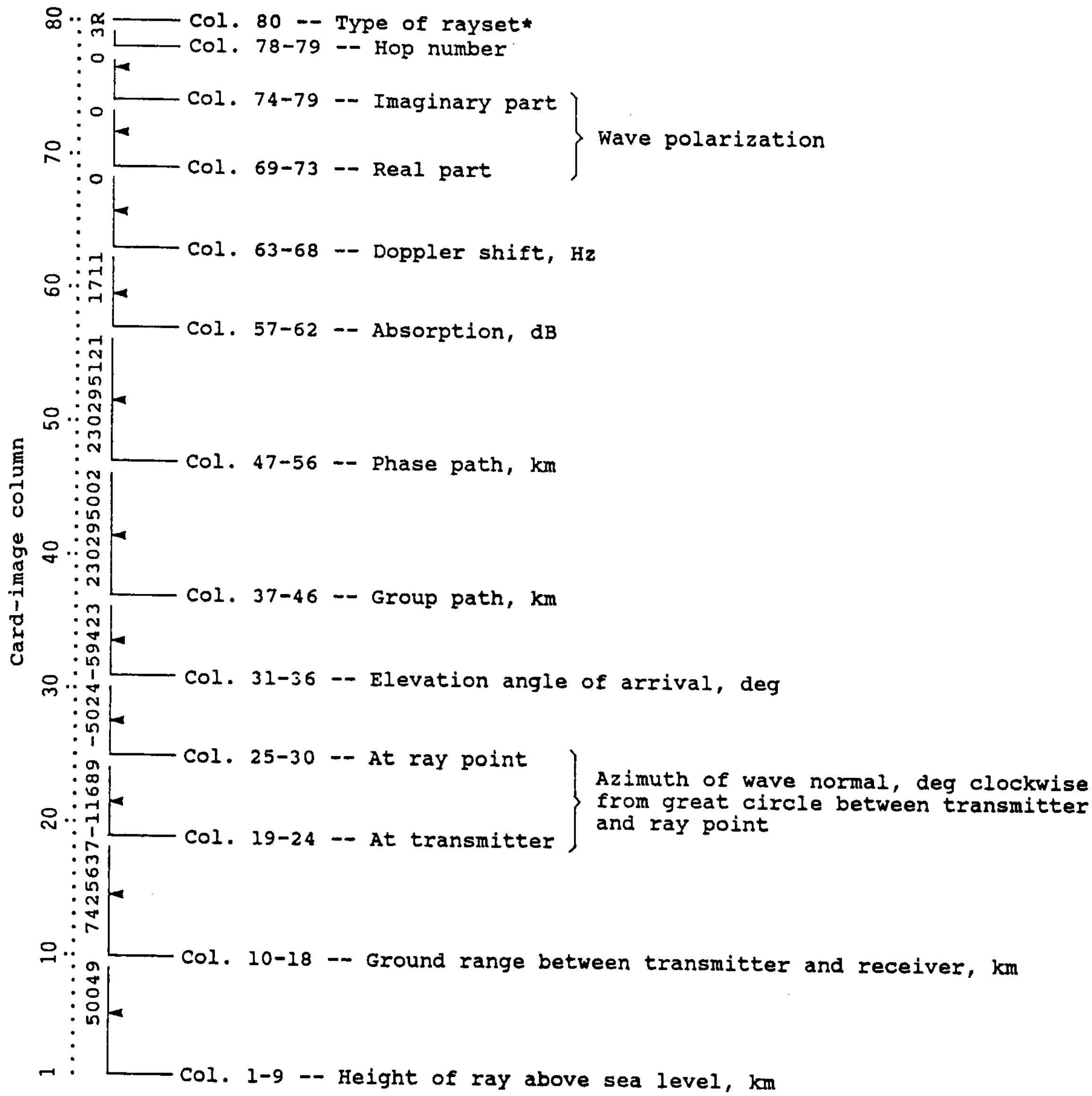


Figure 2.24. Definitions and format for a receiver rayset. * Type of rayset: G = ground reflection; M = closest approach to receiver height; P = penetrated range or height limit; R = at receiver height; S = maximum number of steps; E = extinction; F = exceeded maximum range; U = exceeded maximum height; D = went below minimum height; ▲ = decimal point.

PART II: HOW TO USE THIS PROGRAM

3. How to Get This Program Running on Your Computer

This chapter explains how to get the FORTRAN source code off the distribution tape and onto your computer, and how to get as far as running the sample case. It also deals with the machine-dependent aspects of running HARPA and suggests ways to deal with different computing environments.

3.1 How To Get a Copy of the Program

The FORTRAN source code for the version of HARPA documented in this report and the Input Data File for the sample case are available on magnetic tape. For ordering information, contact the authors at the Wave Propagation Laboratory, Propagation Studies Program Area, 325 Broadway, Boulder, Colorado 80303.

The format of the distribution tape is 0.5 in × 1200 ft, 9 track, 1600 bpi, ASCII character set, block size 1600 bytes, logical record length 80 bytes, no parity.

3.2 ANSI-FORTRAN 77 Compatibility

HARPA was designed to run on a Control Data Corp. (CDC) CYBER 700-800 series with a CDC FORTRAN 77 compiler. It should compile with any FORTRAN compiler that adheres to the ANSI FORTRAN 77 standard, including microcomputer FORTRAN compilers that claim such compatibility.

To ensure portability, we have made many changes in portions of the program that were written before the ANSI standard was established. However, where such changes would have been arduous, and where de facto standards that exist on many systems permit deviations from the ANSI standard, we have retained some non-ANSI code. Following are some exceptions to the ANSI standard:

(1) Some variable and subroutine names have seven characters (six is the ANSI standard).

(2) Some alphanumeric characters are stored eight characters per word in numeric (not character) variables and are output using A8 format.

(3) Some machine-dependent constants are entered in nonstandard format (see the following section).

(4) Sometimes a function is called as though it were a subroutine.

(5) Some real variables are EQUIVALENCEd to integer variables.

(6) In some models and other subroutines, data statements are used to initialize variables contained in labeled-common blocks. For systems that do not permit this, such data statements must be put into separate BLOCK DATA modules. Sequence numbers in the source-code listing identify such statements.

3.2.1 Machine- and System-Dependent Code

We have tried to consolidate any machine- or operating-system-dependent code into two subroutines to make it easier to identify and adapt to new environments.

Before attempting to run HARPA, the user must modify SUBROUTINE DFCNST, which defines machine-dependent constants. The version supplied on the distribution tape is for the CYBER 700-800 series with NOS 2x. Instructions for modifying this routine for several popular machines are included as comments in SUBROUTINE DFCNST (Appendix D).

Another subroutine, DFSYS, contains some operating-system-dependent functions, such as clock and date functions and system-dependent I/O. Users should also check DFSYS and make changes appropriate to their own operating systems.

3.2.2 Word Length

Some problems may arise with machines that have word lengths shorter than the 60-bit word used in our CYBER 840. The numerical integration subroutine (RKAM1) uses double-precision arithmetic to accumulate numerically integrated quantities, but this is almost certainly not necessary with a 60-bit word for ordinary precision requirements. We have not investigated what errors might occur if less precision were used. We recommend testing the precision on a different machine by running the sample case for smaller and smaller values of the single-step integration error, W(42), and verifying that the error value in the first column of the printout maintains at least the accuracy specified by W(42). The level where that accuracy first breaks down is probably the precision limit imposed by the computer's word length.

3.2.3 Execution Speed

For many applications, HARPA runs fast enough on our CYBER 840 to allow virtually interactive (machine load permitting) ray tracing using a graphics terminal for editing program input and for viewing graphical output. Although HARPA may compile and run on smaller machines, its speed may be so slow that interactive ray tracing may no longer be practical. The run times shown on the printout (Appendix A) for the sample case (at the end of each ray) allow you to compare run times between your computer and ours. Tests on a CRAY XMP-48 indicate a factor of 7 speedup over a CYBER 840.

3.2.4 Graphics

The graphics programs included with HARPA were designed to run on the CYBER 700-800 series computer and use the DISSPLA graphics package (by ISSCO, Inc.) and a CDC 250 Microfilm Potting Unit. However, HARPA will run regardless of the plotting facilities you have.

If you have DISSPLA, you can produce the graphic output by running the supplementary program DDSPLA, supplied as File 6 of the distribution tape. This program reads a graphics metafile called TAPE5, which HARPA produces when plots are requested (see Fig. C1).

If you don't have DISSPLA, you can still run HARPA and get the printed and machine-readable outputs, but you won't get any graphics output. Just ignore the TAPE5 file. If you have other graphics devices, you can modify the DD-prefix plotting routines to drive them. PROGRAM DDALT (file 7 of the distribution tape) provides a framework for inserting custom plot-command calls. The functions of these routines and further details about the CDC 250 plot package and the DISSPLA interface are given in Appendix C.

3.3 Unpacking the Program Tape and File Organization

The distribution tape contains the seven files listed in Section 3.3.1. Section 7.1 gives a list of the programs and subroutines on the distribution tape. Normally, you would transfer all tape files to punched cards or permanent disk files, depending on which medium you will use to run the program.

Although HARPA continues to evolve, the source code on the distribution tape will always correspond exactly with the version described in this report. Updates and errata will be documented separately and included in a dated update tape file.

3.3.1 Files on the HARPA Distribution Tape

File #:

1. FORTRAN source code for the Sample Case, including its models, ready to compile, with common and data blocks included.

2. Input Data File for the Sample Case.

3. FORTRAN source code for the "Ray-tracing Core" programs, including plotting (graphics-write) routines.

4. FORTRAN source code for four dispersion-relation routines.

5. FORTRAN source code for all atmospheric-model routines.

6. FORTRAN source code for program DDSPLA for reading the Graphics Output File (TAPE5) for users with DISSPLA.

7. FORTRAN source code for program DDALT, a skeleton routine for reading the Graphics Output File (TAPE5), allowing users to insert plotting modules for their own plotting system.

3.3.2 Setting Up a Run Module

A run module is the subset of programs that you submit to your computer to run a particular application, along with the job-control commands your computer needs to compile and/or run a program. Files 1 and 2 of the distribution tape constitute the run module (minus the job-control cards) for the sample case. It consists of a core of routines (Sec. 7.1.1) that must always be present to trace rays, and a set of selectable routines (Secs. 7.1.2 and 7.1.3) that describe the particular model (those for the sample case in this example). Figure 3.1 shows a representative run module.

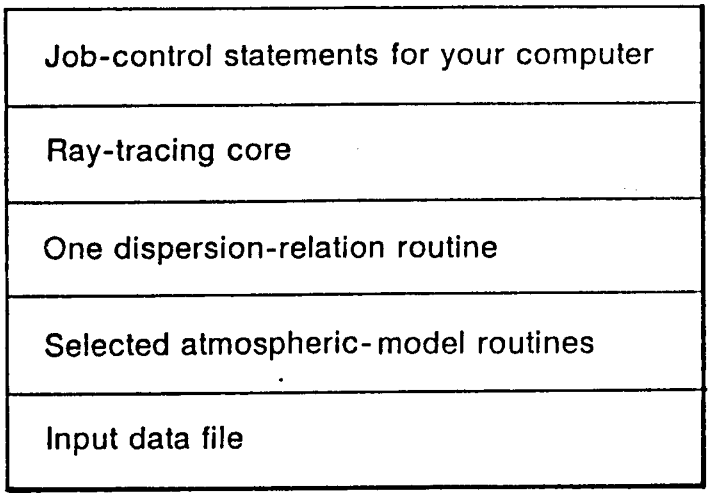


Figure 3.1. Configuration of a run module, assembled from parts consisting either of disk-file modules or punched-card decks.

The selectable part of the run module means that you select only the routines that describe the model you want to use. The run module must contain <u>one and only one</u> (with exceptions noted) of the following kinds of model routines:

- a dispersion-relation routine
- a background sound-speed model
- a background wind model (if you use dispersion models AWWWL or AWWNL)
- a background terrain model
- a perturbation sound-speed model (NPSPEED does nothing)
- a perturbation wind model (NPWIND does nothing) (if you use dispersion models AWWWL or AWWNL)
- a perturbation terrain model (NPTERR does nothing)
- a receiver-surface model.

In addition, you need any other models that are called by any of the above routines, for example, background and temperature-perturbation models called by a sound-speed model (such as GAMRTDM). If you are using a version of the dispersion-relation routine that includes absorption (AWWWL or ANWWL), you will also need models for viscosity/thermal conductivity and pressure, or whatever other parameters it requires. Look at the bottom of the model input parameter forms (Appendix B) to see what other models a given model needs.

3.3.3 If You Are Using Cards To Input Data

If you have no permanent disk storage on your computer, you can load a previously compiled version of HARPA (if you don't change the program itself) into your computer from tape each time you want to run it, and have it read the Input Data File from punched cards. For each run, you have to edit the Input Data File (Deck) by punching new cards for the data you change from a previous run. The Input Data File is arranged in an 80-column format with one input parameter per card so that the deck is easy to edit if you have the contents of the cards interpreted (printed across the card tops).

3.3.4 If You Are Using Permanent Disk Files

It is far more convenient to store both the program and the Input Data File in permanent disk files. A run module (Fig. 3.1) can be constructed by a batch or procedure program that selects the appropriate routines from the HARPA "library." Procedure (or batch) programs also simplify the manipulation of HARPA input and output, but because they depend on the operating system you use, you will generally have to write your own procedures.

4. How to Construct An Atmospheric Model

The easy way to set up an atmospheric model is to select from a general-purpose set of models we have designed (Table 4.1) and choose the model parameters that fit your needs. This requires no programming whatsoever and we encourage that choice whenever possible. Alternatively, you can design your own atmospheric models by writing a FORTRAN subroutine that defines the atmospheric property and its spatial derivatives in a form that is compatible with the rest of the program. This chapter describes both ways.

4.1 Choosing From the Available Atmospheric Models

We have designed some generic atmospheric models that can be adapted to represent common atmospheric structures simply by selecting the appropriate model and its parameters in the Input Data File. They are closed-form expressions for an atmospheric property as a function of geographic latitude, longitude, and height; some accept input parameters in tabular form. There are models for wind, sound-speed and temperature fields, viscosity and thermal conductivity, pressure, molecular weight and terrain surfaces. Though not strictly considered part of the atmospheric model, three models for the receiver surface are also provided. All of these models are described in Appendix B.

Most of the models come in two kinds, "background" and "perturbation." Perturbation models generally superimpose more structure on a background model. Table 4.1 lists the atmospheric models that come with HARPA. To run HARPA, you always have to specify one background model and one perturbation model for each required atmospheric property, even if the perturbation is a do-nothing version.

To put together an atmospheric model from the subroutines we have supplied, first copy the FORM TO SPECIFY AN ATMOSPHERIC MODEL from Appendix B and fill in the name of a model you want to use for each atmospheric parameter, selecting from the choices listed in Table 4.1. Full mathematical descriptions of each model can be found in Appendix B on the order form listed under the model name, and FORTRAN listings for the model subroutines can be found in Appendix D. For now, leave off the numbers of the Data Set ID column until you have selected the models' parameters.

Table 4.1--Available atmospheric models

Model type number	Start of W array parameter block	Model check number	Subroutine name	Description
1.	100			Background wind models:
		1.	WLINEAR	Constant upward and northerly wind, linear easterly wind profile
		6.	ULOGZ2	Logarmithmic atmospheric boundary layer profile
		9.	VVORTX3	Vertical, cylindrical wind vortex
		8.	WGAUSS2	Localized (Gaussian) zonal wind field
		5.	WTIDE	Zonal and meridional profiles that are harmonic in time and height and are in quadrature
2.	125			Wind-perturbation models:
		0.	NPWIND	Do-nothing version
				Sound-speed models:
3.	150	1.	GAMRTDM	$C^2 = \gamma RT/M$
		2.	CSTANH	Profile with linear segments joined by hyperbolic functions
4.	175			Sound-speed perturbation models:
		0.	NPSPEED	Do-nothing version
		2.	CBLOB2	Localized (Gaussian) temperature perturbation
5.	200			Background temperature models:
		0.	NTEMP	Do-nothing temperature model
		1.	TLINEAR	Linear temperature profile
		6.	TTABLE	Tabular temperature profile with cubic interpolation between points
		7.	TTANH5	Profile with linear segments joined by hyperbolic functions
6.	225			Temperature-perturbation models:
		0.	NPTEMP	Do-nothing version
		2.	TBLOB2	Localized (Gaussian) temperature perturbation
7.	250			Molecular weight models:
		1.	MCONST	Constant molecular weight

Table 4.1--Available atmospheric models (continued)

Model type number	Start of W array parameter block	Model check number	Subroutine name	Description
8.	275			Receiver-surface models:
		1.	RHORIZ	A sphere concentric with the earth
		2.	RTERR	A fixed height above the terrain
		3.	RVERT	A vertical surface at a specified fixed range from a specified geographic point
9.	300			Terrain models:
		1.	GHORIZ	A sphere concentric with the earth
		3.	GTANH	A profile of linear segments joined by hyperbolic functions
		4.	GLORENZ	An east-west Lorentzian-shaped ridge
10.	325			Terrain perturbations:
		0.	NPTERR	Do-nothing version
17.	500			Viscosity/thermal conductivity:
		1.	MUARDC	ARDC viscosity model, Prandtl number for thermal conductivity
18.	525			Viscosity/conductivity perturbation:
		0.	NPABS	Do-nothing version
19.	550			Atmospheric pressure:
		1.	PEXP	Exponential profile
20.	575			Pressure perturbation:
		0.	NPPRES	Do-nothing version

Next, select and copy the blank Input Parameter Forms from Appendix B for the models you have selected and fill in the values of the variable parameters that you want. Next transfer the input parameters to a new Input Data File, either constructing one from scratch, according to the format shown in Figure 2.5, or modifying an old one. (If you use an old one, make sure that unused parameters are removed.) Remember to assign an input data-set identification number (F7.3 FORMAT), which uniquely identifies that set of input parameters for each model, and to assign an Atmospheric Identification (ID) for the entire set of models. Put these ID numbers on the FORM TO SPECIFY AN ATMOSPHERIC MODEL and save all these forms as a record of the models you have defined.

Here are a few guidelines for selecting models. If you want a model with no temperature variation, use TLINEAR and set the gradient to zero. If you want a model with no wind, use no wind model and select a dispersion-relation routine with no wind (ANWNL or ANWWL). If you specify any temperature model, you have to use model GAMRTDM, which simply converts temperature to sound speed. If you want to specify a sound-speed field only (such as CSTANH), don't use any temperature model, in which case you don't need GAMRTDM. You can also specify a background model in terms of temperature or sound speed (but not both), and perturbations in terms of temperature or sound speed or both (as in the sample case). If you want no perturbation model for wind speed, temperature, sound speed, or terrain, use the corresponding do-nothing perturbation models NPTEMP, NPWIND, NPSPEED, or NPTERR.

If you are storing HARPA on a permanent disk file, you have to select only the subroutines that define your atmospheric model (and the correct dispersion-relation routine) and assemble them into a separate "run module" (Section 3.3.2). If you are storing the programs on punched cards, you should select and submit only the decks for the model subroutines you want to use. It is convenient to think of HARPA as consisting of a core of ray-tracing routines that are always used, and a set of selectable model-related routines from which you select the ones appropriate to the models you want. The specific routines that fall into each category are listed in Chapter 7.

4.1.1 Model Check Numbers

To guard against accidentally selecting the wrong model subroutine for a run module, each model is assigned a permanent Model Check Number, which is entered on each model input data form (Appendix B). If a model subroutine is selected whose Check Number does not match that specified in the Input Data File, then the program will stop and give an error message.

Another number, called the Input Data Format Code, is not now being used or checked, but may be used in the future.

4.1.2 Tabular Input to Models

Some models, like TTANH5 and TTABLE, can accept so many input parameters that it is inconvenient to specify each one as a separate line in the Input Data File, so a general provision has been made for entering data in tabular form. The use of TTANH5 in the sample case is an example. Tabular data are entered into the Input Data File in a special format illustrated by Figure 2.19 and described fully in Chapter 5.

4.2 Designing Your Own Models

HARPA will accept any atmospheric model specification that provides the desired atmospheric property as a function of the earth-centered spherical-polar coordinates r,θ,ϕ and time t, as well as its spatial and temporal derivatives. There are three important considerations in writing model subroutines: (a) All spatial derivatives must be not only continuous but also analytically consistent with the formulas for the atmospheric property itself; any errors or approximations in those calculations will result in larger-than-desired integration errors, as displayed in the first column of the ray-tracing printout. (b) The input data for the models must come from the part of the W array assigned to that type of model (Table 4.1) and from the tabular-input common blocks assigned to those models (Table 4.2). (c) The output from the atmospheric model must go to the appropriate data-output common block (Table 4.3). Because the model routines are called many times, efficient programming here pays off in execution efficiency.

If you want a new model of wind, temperature, or sound speed that depends only on height, you would normally design a new background model. If you want a three-dimensional model, you could use one of the background models we have supplied and design a new perturbation model. Conceivably, you could design both a new background and a new perturbation model, but the safe way to proceed is to do one at a time.

Those designing a new model should pattern their subroutine after a similar one that comes with HARPA. We will use the model TTANH5 (Fig. 4.1), as an example and discuss its structure in detail to illustrate how to write a model subroutine. In the following paragraphs, general statements will be followed in square brackets by the specific examples from TTANH5.

Table 4.2--Allocation of common blocks for tabular input to the various atmospheric models*

Common block name	Atmospheric model
/B1/	Wind velocity
/B2/	Perturbation to the wind velocity
/B3/	Sound speed
/B4/	Perturbations to the sound speed
/B5/	Temperature
/B6/	Perturbations to the temperature
/B7/	Molecular weight
/B8/	Receiver surface
/B9/	Terrain
/B10/	Terrain perturbation
/B17/	Viscosity/thermal conductivity
/B18/	Viscosity/conductivity perturbation
/B19/	Pressure
/B20/	Pressure perturbation

* In the first 31 elements of each of these common blocks, each atmospheric model indicates the structure of the rest of the common block. Subroutine READW1 stores input data that it reads starting in element 32 of the common block.

Table 4.3--Allocation of common blocks for output from the various atmospheric models

Common block name	Location of description (table number)	Atmospheric model
/UU/	7.10	Wind velocity
/CC/	7.11	Sound speed
/TT/	7.12	Temperature
/MM/	7.13	Molecular weight
/RR/	7.14	Receiver surface
/GG/	7.15	Terrain
/AA/	7.16	Viscosity/thermal conductivity
/PP/	7.17	Atmospheric pressure

There are no restrictions on naming models, but it is useful to assign a name that suggests the model's function. [TTANH5 is the fifth temperature model that used TANH functions to smooth temperature profiles with linear segments. Square brackets enclose examples of parameters for the TTANH5 routine.] Each model subroutine has two entry points whose standard names are given in Table 4.4. These names must be used when designing new subroutines. The first entry point [ENTRY IPTEMP], whose name begins with an "I," is for initialization after new input data have been read in, and that entry point is called the first time the program enters the subroutine. The second entry point [ENTRY TEMP] enters the routine for subsequent computations of the atmospheric parameter [T] and its time and space derivatives [PTT, PTR, PTTH, PTPH] according to the formulas given on the model order from.

The input to each subroutine (geographic coordinates r,θ,ϕ) is through blank common, and output [temperature and its derivatives] is through the labeled common blocks [/TT/], named for each kind of model and listed in Table 4.3. If you need more input parameters than will fit in the assigned block of the Input Data Table [200-224], then you should use a tabular input format, which uses the labeled common blocks [B5] listed in Table 4.2. Tabular

```
      SUBROUTINE TTANH5                                                   TTANH5 9
C     TEMPERATURE PROFILE REPRESENTED BY A SEQUENCE OF LINEAR SEGMENTS   TTANH510
C     SMOOTHLY JOINED BY HYPERBOLIC FUNCTIONS. PARAMETERS ARE INPUT      TTANH511
C     AS TABULAR DATA WITH SLOPES COMPUTED FROM TEMPERATURE DATA.        TTANH512
C     REFERENCE TEMPERATURE T0 IS READ FROM TABULAR DATA.                TTANH513
      DIMENSION C(20), TM(19), Z(19), DL(19)                             TTANH514
C     COMMON DECK "RKAM" INSERTED HERE                                   RKAMCOM2
      REAL KR,KTH,KPH                                                    RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)        RKAMCOM5
C     COMMON DECK "TT" INSERTED HERE                                     CTT    2
      REAL MODT                                                          CTT    4
      COMMON/TT/MODT(4), T,PTT,PTR,PTTH,PTPH                             CTT    5
C     COMMON DECK "WW" INSERTED HERE                                     CWW    2
      PARAMETER (NWARSZ=1000)                                            CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                    CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                 CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),     CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),       CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),           CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),          CWW2   6
     8 (RCVRH,W(20)),                                                    CWW2   7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                       CWW2   9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                       CWW2  10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                      CWW2  11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),        CWW2  12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))         CWW2  13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),         CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))               CWW2  15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                    CWW2  16
      REAL MMODEL,MFORM,MID                                              CWW3   2
C                                                                        CWW3   3
C     WIND          100-124                                              CWW3   4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)            CWW3   5
C                                                                        CWW3   6
C     DELTA WIND          125-149                                        CWW3   7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)         CWW3   8
C                                                                        CWW3   9
C     SOUND SPEED 150-174                                                CWW3  10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)            CWW3  11
      EQUIVALENCE (W(153),REFC)                                          CWW3  12
C                                                                        CWW3  13
C     DELTA SOUND SPEED 175-199                                          CWW3  14
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)         CWW3  15
C                                                                        CWW3  16
C     TEMPERATURE     200-224                                            CWW3  17
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)            CWW3  18
C                                                                        CWW3  19
C     DELTA TEMPERATURE     225-249                                      CWW3  20
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)         CWW3  21
C                                                                        CWW3  22
C     MOLECULAR   250-274                                                CWW3  23
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)            CWW3  24
C                                                                        CWW3  25
C     RECEIVER HEIGHT   275-299                                          CWW3  26
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)            CWW3  27
```

Figure 4.1. Listing for model temperature subroutine TTANH5.

```
C
C     TOPOGRAPHY   300-324
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)
C
C     DELTA TOPOGRAPHY   325-349
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)
C
C     UPPER SURFACE TOPOGRAPHY   350-374
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)
C     PLOT ENHANCEMENTS CONTROL PARAMETERS
C
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)
C     ABSORPTION          500-524
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)
C
C     DELTA ABSORPTION          525-549
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)
C
C     PRESSURE          550-574
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)
C
C     DELTA PRESSURE          575-599
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)
C
C     COMMON DECK "B5" INSERTED HERE
      INTEGER TMX,TNTBL,TITBL,TFRMTBL,IDST(10)
      COMMON/B5/TMX,TNTBL(10),TITBL(10),TFRMTBL(10),TGP(262)
      EQUIVALENCE (TGP,IDST),(ANT,TGP(11))
      EQUIVALENCE             (Z0,TGP(12)),(TM,TGP(33))
      EQUIVALENCE (Z,TGP(13)),(C,TGP(32)),(DL,TGP(53))
C
      DATA RECOGT,N/7.0,0/
      DATA ANT/0.0/
      DATA TMX/2/
      DATA TNTBL/1,11,72,7*0/
      DATA TITBL/1,20,8*0/
      DATA TFRMTBL/1,2,8*0/
C
      COSH (X) = (EXP (X) + 1. / (EXP (X))) / 2.
C
      ENTRY ITEMP
C
      CALL IPTEMP
C
C     IF HAD PREVIOUS CALL BUT NOTHING THIS TIME, EXIT NOW
C     RETAINING PREVIOUS TABULAR DATA COUNT
      IF(N.GT.0 .AND. ANT.EQ.0.0) RETURN
C
      IF(RECOGT .NE. TMODEL)
     1    CALL RERROR('TEMP    ','WRNG MODEL',RECOGT)
C
      MODT(1)=7HTTANH5
      MODT(2)=TID
C
      N=(ANT+1)/3 - 2
C
      IF(N.LE.0)
     1    CALL RERROR('TTANH5','BAD N VALUE',FLOAT(N))
C
```

Figure 4.1. Listing for model temprature subroutine TTANH5 (continued).

```
      ANT=0.0
C
C
C     CONVERT 'T' ARRAY INPUT(OVERLAYS 'C' ARRAY) TO 'C' ARRAY
C
      T0=C(1)
      TIM1=T0
      ZIM1=0.0
      NP1=N+1
      DO 10 I=1,NP1
         TI=TM(I)
         ZI=Z(I)
         C(I)=(TI-TIM1)/(ZI-ZIM1)
         TIM1=TI
10       ZIM1=ZI
C
      RETURN
C
      ENTRY TEMP
      H = R - EARTHR
      SUM = 0.
C
C     LOOP TO SUM OVER ALL COEFFICIENTS
C     USE SPECIAL FUNCTION 'ALCOSH' WHICH ALLOWS FOR LARGE ARGUMENTS.
      DO 1 I = 1, N
    1 SUM = SUM + DL(I) * (C(I + 1) - C(I)) / 2. *(ALCOSH((H - Z
     1(I)) / DL(I)) - ALCOSH((Z(I)-Z0) / DL(I)))
      T = T0 + SUM + (C(1) + C(N + 1)) * (H - Z0) * 0.5
      SUM = 0.
      DO 2 I = 1, N
    2 SUM = SUM + (C(I + 1) - C(I)) / 2. * (1. + TANH ((H - Z(I)) /DL
     1 (I)))
      PTT=0.0
      PTR = C(1) + SUM
      PTTH=0.0
      PTPH=0.0
C
      CALL PTEMP
      RETURN
      END
```

Figure 4.1. Listing for model temperature subroutine TTANH5 (continued).

Table 4.4--Assignment of entry point names and input parameter blocks in the W array for the atmospheric models

Atmospheric model	Entry point names	Input parameter block in the W array
Wind	WINDR, IWINDR	100-124
Wind perturbation	PWINDR, IPWINDR	125-149
Sound-speed	SPEED, ISPEED	150-174
Sound speed perturbation	PSPEED, IPSPEED	175-199
Temperature	TEMP, ITEMP	200-224
Temperature perturbation	PTEMP, IPTEMP	225-249
Molecular weight	MOLWT, IMOLWT	250-274
Receiver surface	RECVR, IRECVR	275-299
Terrain	TOPOG, ITOPOG	300-324
Terrain perturbation	PTOPOG, IPTOPOG	325-349
Viscosity/thermal conductivity	ABSRP, IABSRP	500-524
Viscosity/thermal conductivity perturbation	PABSRP, IPABSRP	525-549
Pressure	PRES, IPRES	550-574
Pressure perturbation	PPRES, IPPRES	575-599

data are read into this common block from the Input Data File according to the format described in Chapter 5 and illustrated near the end of Figure 2.19 for the sample case.

4.2.1 How To Write an Atmospheric Model Subroutine So It Can Receive Tabular Data Read Into Common Blocks by READW1

If you want to write an atmospheric model subroutine that uses tabular input data, then you have to observe some special precautions. In what follows, general statements are exemplified in square brackets for the case of the temperature subroutine TTANH5. You can use this example as a guide in developing new model subroutines that use tabular data.

Tabular [temperature] data are read into a model-related common block [/B5/] by READW1. Table 4.3 gives the names of the common blocks associated with the different model types. The format of the tabular input data can be selected by the user and is determined by a code [3] in columns 1-3 of the Input Data File (see Figure 2.19). [In the case of TTANH5, a three-column format makes sense because there are three input parameters]. READW1 interprets this code according to the formats listed in Table 5.4. The model subroutine [TTANH5] must tell READW1 how it wants the tabular input data stored in the common block [B5] in an array [TGP]. It does so by setting (in DATA statements) the values in variable TMX and in arrays TNTBL, TITBL and TFRMTBL for temperature models (or corresponding names for other model types, in which the first letter is different). The model variables [ZO,TM,C,DL] are EQUIVALENCEd to elements of a GP (for general-purpose) array [called TGP for temperature models]. Table 4.5 defines the structure of common block /B5/, which transmits these variables between READW1 and TTANH5, and it explains how to set the data-block parameters.

4.2.2 Designing Your Own Terrain Models

A terrain model specifies a function $g(r,\theta,\phi)$ such that $g=0$ on the terrain, $g>0$ above the terrain, and $g<0$ below the terrain. To be an allowed model, g must be continuous through second derivatives. A subroutine for a terrain model must calculate g, its three first derivatives, and its six second derivatives for any values of r,θ,ϕ. All of our present terrain models define g to be the height above the terrain, but more general definitions are allowed to handle cliffs, overhangs, and caves. To design a simple model with a few parameters, follow the example of SUBROUTINE GLORENZ. To design a more elaborate model that needs tabular input data, follow the example of SUBROUTINE GTANH. Source-Code listings for these models are in Appendix D.

Table 4.5--Definitions of the parameters in common block /B5/*

Position common	Variable name	Definition
1	TMX	Maximum number of data blocks in /B5/
2-11	TNTBL	An array that contains the beginning location of data blocks within the common block
2	TNTBL(1)	Beginning location (in TGP array) of data block 1
3	TNTBL(2)	Beginning location (in TGP array) of data block 2
4	TNTBL(3)	Beginning location (in TGP array) of data block 3**
12-21	TITBL	An array that contains the iteration (or cycle) length of the data in the data blocks within the common block (if there is more than one array in the data block, then this is the dimension of the arrays)
12	TITBL(1)	Cycle length of data block 1
13	TITBL(2)	Cycle length of data block 2
22-31	TFRMTBL	An array that contains the input format numbers for the data blocks within the common block
22	TFRMTBL(1)	Format type*** for data block 1
23	TFRMTBL(2)	Format type*** for data block 2
32-	TGP	An array containing TMX number of data blocks for tabular input data for atmospheric models

* The values of the first 31 elements in /B5/ define the block structure for the array TGP, and they are defined in the atmospheric models and used in the data read-in routines READW and READW1. The common blocks /B1/, /B2/, /B3/, /B4/, /B6/, and /B7/ have the same structure as /B5/, but have different names for the variables.

** Only two data blocks are now available to use; however, the beginning location of the first data block not used must be specified to indicate the length of the last data block used.

*** Format type 1 implies format number 1 (see Table 5.3).
Format type 2 implies format numbers 2, 3, or 4 (see Table 5.3).

5. How to Specify the Input Data and Set Up an Input Data File

To give HARPA an atmospheric model and to tell it what rays to trace, you have to construct an Input Data File, like the one shown in Figure 5.1 (same as Figure 2.19, reproduced here for the user's convenience) for the sample case. An Input Data File may contain one or more "run sets," each of which can specify a different atmospheric model, different plotting modes, or different initial ray conditions, but which will all be executed as a single computer run. The sample case contains two run sets. After the first run set, only the parameters whose values <u>differ</u> from those specified in the <u>preceding</u> run set need be specified.

After setting up an Input Data File, you run HARPA by combining it with other ray-tracing modules to form a "run module," as explained in Section 3.3.2. All the necessary modules to run the sample case are contained in Files 1 and 2 of the distribution tape (Sec. 3.3.1).

5.1 Editing the Input Data File

Because HARPA contains no built-in way to construct or edit an Input Data File, you have to use an editor of your own to do so. We have designed a specialized editor, called WMOD, for this purpose. It not only permits editing the Input Data File, but it also sets up a "run module" that includes job-submission procedures for our computer. This program, which could run on a local microcomputer, will be documented in another report.

The Input Data File can take the form of either a deck of punched cards or a disk file to be read by HARPA. Some suggestions for those using punched cards are given in Section 3.3.3. Henceforth, we will assume that the Input Data File will be created as a disk file. There are no formal differences between the two methods, however.

Rather than start from scratch, we recommend that you modify an existing Input Data File. After you have run the sample case and have verified that its output agrees with that given in Appendix A, you can modify the Input Data File for the sample case to make the raypath calculations you want.

```
col.    col.       col.       col.
1-3     4-17       18-24      25-80
 n      W(n)       UNITS      DESCRIPTION
____________________________________________________________________________
GEORGES    RB3    X6437
S03-1    SAMPLE CASE FOR HARPA DOCUMENTATION          REV. 2-10-86
  1    6370.              EARTH RADIUS, KM (6370.)
  3      13.             TTRANSMITTER HEIGHT, KM (T=ABOVE TERRAIN)
  4     200.     AN KM    N. TRANSMITTER LATITUDE, KM
  5       0.     AN KM    E. TRANSMITTER LONGITUDE, KM
  7      .05     FQ HZ    INITIAL FREQUENCY, HZ
 11      45.     AN DG    INITIAL AZIMUTH ANGLE, DEG
 15     -20.     AN DG    INITIAL ELEVATION ANGLE, DEG
 16     140.     AN DG    FINAL   ELEVATION ANGLE, DEG
 17       5.     AN DG    STEP IN ELEVATION ANGLE, DEG
 20       5.              RECEIVER HEIGHT, KM
 22       3.              MAXIMUM NUMBER OF HOPS (1.0)
 23    1000.              MAXIMUM NUMBER OF STEPS PER HOP (1000.)
 26     500.              MAXIMUM RAY HEIGHT, KM (500.)
 27      -1.              MINIMUM RAY HEIGHT, KM
 28    1000.              MAXIMUM RANGE, KM
 29 0000100.              DO: EIGRAY/RNG-TIM/RNG-ELV/NEW-PROJ/RAYTRC/CONT/PROF
 33  999.999              MAXIMUM ABSORPTION, DB (999.999)
 42       1.0E-6          SINGLE-STEP INTEGRATION ERROR (1.0E-4)
 44       .1              INITIAL INTEGRATION STEP SIZE, KM (1.0)
 57       2.              PHASE PATH (0=NO; 1=INTEGRATE; 2=INTEGRATE/PRINT)
 58       2.              ABSORPTION (0=NO; 1=INTEGRATE; 2=INTEGRATE/PRINT)
 60       2.              PATH LENGTH (0=NO; 1=INTEGRATE; 2=INTEGRATE/PRINT)
 71      50.              NUMBER OF INTEGRATION STEPS PER PRINT [1.E31]
 72       1.              OUTPUT RAYSETS (1=YES; 0=NO)
 73       0.              DIAGNOSTIC PRINTOUT (1=YES; 0=NO)
 74       0.              PRINT RAY STEPS (0=YES; 1=NO)
 75      .15              FULANN LETTER HEIGHT [0.15 IN]
 76       0.              BINARY RAY OUTPUT (1=YES; 0=NO)
 77      57.              LINES PER PAGE IN PRINTOUT (66.)
 81       1.              RAYPLOT PROJECTION (1=VERT; 2=HORIZ) PLANE
 82       1.              PLOT-EXPANSION FACTOR [1.0]
 83    -100.     AN KM    N. LATITUDE  OF LEFT PLOT EDGE, KM
 84    -300.     AN KM    E. LONGITUDE OF LEFT PLOT EDGE, KM
 85     500.     AN KM    N. LATITUDE  OF RIGHT PLOT EDGE, KM
 86     300.     AN KM    E. LONGITUDE OF RIGHT PLOT EDGE, KM
 87     100.     AN KM    DISTANCE BETWEEN TIC MARKS, KM
 88       0.              HEIGHT ABOVE SEA LEVEL OF BOTTOM OF GRAPH, KM
 89     300.              HEIGHT ABOVE SEA LEVEL OF TOP OF GRAPH, KM
 96     100.              DISTANCE BETWEEN VERTICAL TIC MARKS, KM
100       6.              ULOGZ2 WIND MODEL CHECK NUMBER
102       3.               BACKGROUND WIND DATA SET ID
103       5.     LN  M     REFERENCE WIND SPEED, M/S
104      .35               VON KARMAN'S CONSTANT
105       1.               ROUGHNESS HEIGHT, KM
150       1.              GAMRTDM SOUND SPEED MODEL CHECK NUMBER
175       2.              CBLOB2 MODEL CHECK NUMBER
177       2.               SOUND SPEED PERTURBATION DATA SET ID
178       .5               FRACTIONAL INCREASE OF SQUARED SOUND SPEED
```

Figure 5.1. Input Data File (W array) for the sample case.

```
179      125.              HEIGHT OF MAXIMUM INCREASE, KM
180      335.     AN KM    N. LATITUDE OF MAXIMUM INCREASE, KM
181      125.     AN KM    E. LONGITUDE OF MAXIMUM INCREASE, KM
182       25.              GAUSSIAN WIDTH IN HEIGHT OF INCREASE, KM
183       50.     AN KM    N-S WIDTH OF THE INCREASE, KM
184       25.     AN KM    E-W WIDTH OF THE INCREASE, KM
200        7.             TTANH5 TEMPERATURE MODEL CHECK NUMBER
202        1.              BACKGROUND TEMPERATURE DATA SET ID
225        2.             TBLOB2 MODEL CHECK NUMBER
227        2.              TEMPERATURE PERTURBATION DATA SET ID
228        .5              FRACTIONAL TEMPERATURE INCREASE
229        0.              HEIGHT OF MAXIMUM INCREASE, KM
230      105.     AN KM    N. LATITUDE OF MAXIMUM INCREASE, KM
231     -105.     AN KM    E. LONGITUDE OF MAXIMUM INCREASE, KM
232        0.              GAUSSIAN WIDTH IN HEIGHT OF INCREASE, KM
233       50.     AN KM    N-S WIDTH OF THE INCREASE, KM
234       25.     AN KM    E-W WIDTH OF THE INCREASE, KM
250        1.             MCONST MOLECULAR WEIGHT MODEL CHECK NUMBER
252       29.              MOLECULAR WEIGHT DATA SET ID
253       29.              MOLECULAR WEIGHT
275        2.             RTERR RECEIVER MODEL CHECK NUMBER
300        4.             GLORENZ TERRAIN MODEL CHECK NUMBER
302        2.              TERRAIN MODEL DATA SET ID
303        2.              HEIGHT OF THE RIDGE, KM
304        0.              N. LATITUDE OF THE RIDGE CENTER
305       30.     AN KM    HALF-WIDTH OF THE RIDGE, KM
325        0.             NPTERR NO TERRAIN PERTURBATION
500        1.             MUARDC VISC/COND MODEL CHECK NUMBER
502        1.              VISC/COND MODEL DATA SET ID
503     1.458E-06          VISCOSITY COEFFICIENT BETA
504     110.4              SUTHERLAND'S CONSTANT, KELVINS
505      .733              PRANDTL NUMBER
525        0.             NPABS NO VISC/COND PERTURBATION
550        1.             PEXP PRESSURE MODEL CHECK NUMBER
552        1.              BACKGROUND PRESSURE MODEL DATA SET ID
553   101328.              PRESSURE AT SEA LEVEL, N/SQ.M.
554       8.5              PRESSURE SCALE HEIGHT, KM
575        0.             NPPRES   NO PRESSURE PERTURBATION
 -1                       DATA SUBSET FOR BACKGROUND WIND MODEL
  A  LOGARITHMIC EASTWARD WIND PROFILE, U*=.5 M/S, Z0=1 KM
  0      RETURN TO W ARRAY DATA SET
 -2                       DATA SUBSET FOR WIND PERTURBATION MODEL
  A  NO WIND PERTURBATION
  0      RETURN TO W ARRAY DATA SET
 -3                       DATA SUBSET FOR BACKGROUND SOUND-SPEED MODEL
  A  SOUND SPEED IN TERMS OF TEMPERATURE MODEL
  0      RETURN TO W ARRAY DATA SET
 -4                       DATA SUBSET FOR SOUND-SPEED PERTURBATION MODEL
  A  50% INCREASE IN SQ. SOUND SPEED AT 125KM HT, 335KM N, 125KM E
  0      RETURN TO W ARRAY DATA SET
 -5                       DATA SUBSET FOR TEMPERATURE MODEL
  A  U.S. STANDARD ATMOSPHERE 1962 TEMPERATURE PROFILE
  3     999.0
  LN KM                  LN KM
   0.          288.000      0.
```

Figure 5.1. Input Data File (W array) for the sample case (continued).

```
 15.0000       190.500      10.0000
 52.0000       320.000       7.50000
 95.0000       191.000      10.0000
165.0000      1451.000      50.0000
300.0000      1586.000       0.
999.0000
 0       RETURN TO W ARRAY DATA SET
-6                      DATA SUBSET FOR TEMPERATURE PERTURBATION MODEL
 A   50% CYLINDRICAL INCREASE IN TEMPERATURE AT 105KM N., 105 KM W
 0       RETURN TO W ARRAY DATA SET
-7                      DATA SUBSET FOR MOLECULAR WEIGHT MODEL
 A   MOLECULAR WEIGHT = 29
 0       RETURN TO W ARRAY DATA SET
-8                      DATA SUBSET FOR RECEIVER SURFACE MODEL
 A   RECEIVER SURFACE 5 KM ABOVE TERRAIN
 0       RETURN TO W ARRAY DATA SET
-9                      DATA SUBSET FOR TERRAIN MODEL
 A   RIDGE 2-KM HIGH, 30-KM WIDE ALONG EQUATOR
 0       RETURN TO W ARRAY DATA SET
-10                     DATA SUBSET FOR TERRAIN PERTURBATION MODEL
 A   NO TERRAIN PERTURBATION
 0       RETURN TO W ARRAY DATA SET
-17                     DATA SUBSET FOR VISC/COND MODEL
 A   ARDC VISCOSITY AND THERMAL CONDUCTIVITY MODEL
 0       RETURN TO W ARRAY DATA SET
-18                     DATA SUBSET FOR VISC/COND PERTURBATION MODEL
 A   NO VISCOSITY/CONDUCTIVITY PERTURBATION
 0       RETURN TO W ARRAY DATA SET
-19                     DATA SUBSET FOR BACKGROUND PRESSURE MODEL
 A   EXPONENTIAL PRESSURE MODEL, SCALE HEIGHT = 8.5 KM
 0       RETURN TO W ARRAY DATA SET
-20                     DATA SUBSET FOR PRESSURE PERTURBATION MODEL
 A   NO PRESSURE PERTURBATION
 0       RETURN TO W ARRAY DATA SET
 0                      **********  END OF RUN SET NUMBER 1  ***********
S03-2    SAMPLE CASE FOR HARPA DOCUMENTATION          REV. 2-10-86
 71        0.           NUMBER OF INTEGRATION STEPS PER PRINT
 72        0.           OUTPUT RAYSETS  (1=YES; 0=NO)
 73        1.           DIAGNOSTIC PRINTOUT (1=YES; 0=NO)
 81        2.           RAYPLOT PROJECTION (1=VERT; 2=HORIZ) PLANE
 82        3.           PLOT-EXPANSION FACTOR
 0                      **********  END OF RUN SET NUMBER 2  ***********
```

Figure 5.1. Input Data File (W array) for the sample case (continued).

The best way to be sure you have input all the required data is to fill out the forms for specifying all the model and ray parameters, as discussed for the sample case in Chapter 2. Then translate the data from those forms into the format of the Input Data File. We provide blank forms for all models and procedures in Appendix B.

5.2 Input Data Formats

The Input Data File is read by a FORTRAN program and so must conform to precise format specifications. Originally, the Input Data File consisted of a deck of 80-column punched cards, with one input parameter per card, so the data format is still specified in terms of data fields in card-image columns, even though cards are no longer used. Figure 5.1 for the sample case is an example of the proper format.

Looking at Figure 5.1 you will notice that the first part of the file consists of a series of lines that begin with a positive integer. Each of these lines specifies an element of a Data Input Array, W(n). This format goes as far as the line that begins with 575. At the line beginning with -1, the data format changes to accept tabular input data. First, we will explain how to specify data to be read into W(n); then we will explain how to enter data in the tabular format.

5.3 Specifying the W-Array Input

The W-array input format consists of a single 80-column line with four data fields: n, W(n), unit conversion characters, and a description field.

The first three card-image columns contain the first data field in I3 format and specify the index, n, of the array W(n). The value of n must be between 0 and 999, and if there are fewer than three digits, the entry must be right-justified, or else trailing zeroes will be appended to fill out the three columns. If two or more lines begin with the same value of n, the last one prevails.

The second field, columns 4-17, contains the value of W(n) in E14.7 format. The value can be entered in either E or F format, but if the E format is used, the exponent must be right-justified within the 14 spaces, or else zeroes will be appended.

The third field, columns 18-24, contains characters that tell the program to convert the units of the data, as input, to units used by the program. The present choices available for input in this field are given in Table 5.1; the characters must be input in exactly the columnar format shown.

Table 5.1--Units conversion on input

Units of input value*	Meaning	Conversion needed	Value stored by read-in routine
AN RD	Angle in radians	None	V_i
AN DG	Angle in degrees	degs to radians or deg/s to rad/s	V_i $\pi/180$**
AN KM	Central earth angle in kilometers	km to rad	V_i/r_e***
LN KM	Length in kilometers	None	V_i
LN M	Length in meters	m to km	$V_i/1000$
LN NM	Length in nautical miles	nmi to km	$1.852\ V_i$
LN FT	Length in feet	ft to km	$3.048006096 \times 10^{-4}\ V_i$
FQ HZ	Frequency in hertz	Hz to rad/s	$2\pi V_i$
FQ S	Frequency expressed as a period in seconds	Period in s to frequency in rad/s	$2\pi/V_i$
T****	Transmitter height relative to terrain instead of sea level	Add terrain height to transmitter height	V_i (also, a flag is set)*****

* The five characters listed are to be put in card-image columns 18 through 22 of the W-array input value to be converted, or put above the data-input column of tabular input. For three-column tabular input, for example, the five characters should be in columns 1-13, 14-26 and 27-39. The five characters are automatically put in the appropriate place when using the WMOD editor.

** V_i is the input value.

*** r_e is the radius of the earth. The current value of W(1) in the W array is used for this conversion.

**** Applies only for input to W(3) (transmitter height). The "T" must be put in card-image column 24.

***** At the start of each ray, the status of the flag is checked. If the flag is set, then the terrain height at the longitude and latitude of the transmitter is added to the transmitter height. For general terrain models, the terrain height at a given longitude and latitude can only be estimated. For all of the presently available terrain models, the estimate gives an exact result, however, because $\partial g/\partial r$ is constant.

The fourth field (columns 25-80) contains descriptive comments, which aid the user in setting up the table. These comments are optional and arbitrary as far as HARPA is concerned, but for n $\geq$ 100 and divisible by 25, the first word in the comment field is read when WMOD is used for editing and must be a valid model name. This convention will be described in a report about the supplementary programs. Where practical, the comments should describe the function of all acceptable values of the parameter, not just the present value, so that the comment would not have to be changed when the parameter is changed. We have included nonzero initial values in the comment field, where applicable. To make it easier to see the model groupings, we have adopted the convention of indenting the comments that describe model parameters.

5.3.1 Initialization of the Input Data Parameters

Before reading the Input Data File, the program initializes all of the input parameters, W(n). Most are set to zero, but a few are given nonzero initial values that correspond to common usage. An example is the latitude of north pole of the computational coordinate system, W(24), which usually has a value of $\pi/2$. Section 5.3.2 denotes those nonzero initial values by parentheses. These initial values can be overridden by the Input Data File (including a value of zero), but if no value is specified for a W(n) in the Input Data File, then its initial value prevails.

In addition, some initial values are given "zero-override" priority, which means that W(n) assumes its nonzero initial value if no value is input, but also if a zero is input. This zero override operates when a zero value would produce meaningless results or cause difficulty in program execution. An example is the plot expansion factor, W(82). Section 5.3.2 denotes by square brackets the nonzero initial values that override zero.

To help the user keep track of the unit conversions and initializations, all nonzero W(n) values, in the units actually used by HARPA, are listed at the beginning of the printout (Appendix A). In the examples given above, if W(24) were given a value of zero in the Input Data File, no value would be printed for W(24). On the other hand, if W(82) is given a zero value in the Input Data File, a message is printed indicating "INPUT OVERRIDDEN," and the non-zero override value is printed.

5.3.2 Explanation of the Input Data Parameters

Because HARPA has evolved from ray-tracing programs for other media, some values of the input parameter index, n, are not used in HARPA, but may be used in other versions of the program. As far as possible, n is assigned consistently among the different versions, and blocks of n are assigned to groups of related parameters.

A list is given next of all the parameters used by HARPA, with a description of their meanings and idiosyncrasies. Those with nonzero initial values need not be entered in the Input Data File, if the value is what you want. If no initial value is indicated, a zero will be assigned if you leave it out of the Input Data File. The default units given in parentheses are those which are assumed if no unit conversions are put into columns 18-24. Also included in the table is the FORTRAN name (where one exists) assigned (in EQUIVALENCE statements) to each variable in the program. Those labeled "not used" can be assigned to additional input parameters, but those labeled "used by other programs" or "used internally" should not be used.

W(1) EARTHR (6370.) -- Radius (kilometers) of the earth. Can be set to a very large value for a "flat-earth" approximation.

W(2) RAY -- Used by other programs.

W(3) XMTRH -- Height (kilometers) of the transmitter (source) above sea level. If there is a T in column 24, it is the height above the terrain.

W(4) TLAT -- North geographic latitude (radians) of the transmitter. Can be entered in kilometers (or degrees) by putting AN KM (or AN DG) beginning in column 18.

W(5) TLON -- East geographic longitude (radians) of the transmitter. Can be entered in kilometers (or degrees) by putting AN KM (or AN DG) beginning in column 18.

W(6) OW -- Used internally.

W(7) FBEG -- Initial acoustic wave frequency (rad/s). Can be entered in Hz (or period in seconds) by putting FQ HZ (or FQ S) beginning in column 18.

W(8) FEND -- Final frequency (rad/s). Can be entered in Hz (or period in seconds) by putting FQ HZ (or FQ S) beginning in column 18.

W(9) FSTEP -- Step in frequency (rad/s). Can be entered in Hz (or period in seconds) by putting FQ HZ (or FQ S) beginning in col 18. Set = 0 for no stepping.

W(10) AZ1 -- Used internally.

W(11) AZBEG -- Initial azimuth angle (radians east of north) of transmission. Can be entered in degrees by putting AN DG beginning in column 18.

W(12) AZEND -- Final azimuth angle (radians east of north) of transmission. Can be entered in degrees by putting AN DG beginning in column 18.

W(13) AZSTEP -- Step in azimuth angle (radians east of north) of transmission. Can be entered in degrees by putting AN DG beginning in column 18. Set = 0 for no stepping.

W(14) BETA -- Used internally.

W(15) ELBEG -- Initial elevation angle (radians above horizontal) of transmission. Can be entered in degrees by putting AN DG beginning in column 18.

W(16) ELEND -- Final elevation angle (radians above horizontal) of transmission. Can be entered in degrees by putting AN DG beginning in column 18.

W(17) ELSTEP -- Step in elevation angle (radians) of transmission. Can be entered in degrees by putting AN DG beginning in column 18. Set = 0 for no stepping.

W(18)-W(19) -- Not used.

W(20) RCVRH -- Height (kilometers) above sea level of the receiver surface when model RHORIZ is used; height of the receiver surface above the terrain when model RTERR is used.

W(21) ONLY -- Set = 0 to stop frequency increment when ray goes out of bounds (applies only when elevation and azimuth angles are not stepped).

W(22) HOP -- Maximum number of ray hops (intersections with or closest approaches to the receiver surface); ray calculation stops when reached, printing MAX HOPS. Closest approaches count as two hops.

W(23) MAXSTP (1000.) -- Maximum number of integration steps per hop; ray calculation stops when reached, printing STEP MAX.

W(24) PLAT ($\pi/2$) -- Geographic latitude (radians) of the north pole of the computational coordinate system.

W(25) PLON -- Geographic longitude (radians) of the north pole of the computational coordinate system.

W(26) HMAX (500.) -- Maximum ray height (kilometers) above sea level; ray calculation stops if exceeded, printing MAX HT.

W(27) HMIN -- Minimum ray height (kilometers) above sea level; calculation stops if ray goes below this height, printing MIN HT.

W(28) RGMAX -- Maximum ground range (kilometers at sea level) of the ray from the transmitter; ray calculation stops if exceeded, printing MAX RANG.

W(29) RAYFNC -- A set of seven binary digits to select execution of HARPA and supplementary programs. To run HARPA, use 100. Setting = 0 is the same as all ones and will run HARPA.

W(30)-W(32) Used by other programs.

W(33) EXTINC (999.999) -- Maximum absorption (dB); ray calculation stops if value exceeded, printing EXTINC. Set = 0 for no maximum.

W(34)-W(40) -- Not used.

W(41) INTYP (3.): Integration type:
= 1 for Runge-Kutta integration without error checking;
= 2 for Adams-Moulton integration without error checking;
= 3 for Adams-Moulton integration with relative-error checking;
= 4 for Adams-Moulton integration with absolute-error checking.

W(42) MAXERR (1.E-4) -- Maximum allowable integration error per step. RKAM routine decreases step size to achieve this error.

W(43) ERATIO (50.) [50.] -- Ratio of maximum to minimum single-step integration error; RKAM increases step size when error is smaller that W(42) by this factor.

W(44) STEP1 (1.0) -- Initial integration step size (seconds).

W(45) STPMAX (100.) -- Maximum integration step size (seconds).

W(46) STPMIN (1.E-8) -- Minimum integration step size (seconds).

W(47) FACTR (.5) [0.5] -- Factor multiplying integration step size when decreasing step size.

W(48)-W(56) Not used.

W(57) -- Phase-time integration: 0 to not integrate; 1 to integrate; 2 to integrate and print.

W(58) -- Absorption integration: 0 to not integrate; 1 to integrate; 2 to integrate and print.

W(59) -- Doppler shift integration: 0 to not integrate; 1 to integrate; 2 to integrate and print.

W(60) -- Path-length integration: 0 to not integrate; 1 to integrate; 2 to integrate and print.

W(61)-W(70) -- Assigned to future integration options.

W(71) SKIP -- [1.E31] Number of integration steps between printed lines. Set = 1 to print every step; = 0 to suppress periodic printing.

W(72) RAYSET -- Write machine-readable raysets to file PUNCH -- 1 = yes; 0 = no.

W(73) PCNTRW -- Add diagnostic printout lines: 1 = yes; 0 = no.

W(74) PRTSRP -- Produce normal printout every W(71) steps -- 0 = yes; 1 = no. Also produces printout at special events.

W(75) HITLET [.15] -- Height (inches on our plotter) of lettering on graphs. "FULANN" in description field activates publication-quality lettering on graphs when read by WMOD. Any other comment in description field produces draft-quality lettering.

W(76) BINRAY -- Write binary raypath description to file TAPE6 -- 1 = yes; 0 = no.

W(77) PAGLIN (66.) -- Page length (lines) for printout.

W(78)-W(79) -- Not used.

W(80) APOG, PRIGEE -- 0 for normal rayplots; 1 for apogee plots.

W(81) PLT: Rayplot projection:
1 = vertical plane, polar projection, rectangular expansion;
2 = horizontal plane, lateral expansion;
3 = vertical plane, polar plot, radial expansion;
4 = vertical plane, rectangular plot.
Make negative to superimpose plot on that from previous runset.

W(82) PFACTW [1.] -- Vertical or lateral expansion factor for rayplot.

W(83) LLAT -- North latitude (radians) of left edge of plot. To enter in degrees (kilometers) put AN DG (AN KM) beginning in column 18.

W(84) LLON -- East longitude (radians) of left edge of plot. To enter in degrees (kilometers) put AN DG (AN KM) beginning in column 18.

W(85) RLAT -- North latitude (radians) of right edge of plot. To enter in degrees (kilometers) put AN DG (AN KM) beginning in column 18.

W(86) RLON -- East longitude (radians) of right edge of plot. To enter in degrees (kilometers) put AN DG (AN KM) beginning in column 18.

W(87) TIC -- Distance (radians) between tick marks on horizontal axis of plot. To enter in kilometers, put AN KM beginning in column 18.

W(88) HB -- Height (kilometers) of the bottom of the graph above sea level.

W(89) HT -- Height (kilometers) of the top of the graph above sea level.

W(90)-W(95) -- Used by other programs.

W(96) TICV -- Distance (kilometers) between tick marks on vertical axis of plot. Notice that the default units are kilometers for the vertical ticks and radians for the horizontal ticks.

W(97)-W(99) -- Used by other programs.

W(100) UMODEL -- Check number for background wind model.

W(101) UFORM -- format code for background wind model.

W(102) UID -- Data-set ID for background wind model.

W(103)-W(124) -- Parameters for background wind model.

W(125) DUMODEL -- Check number for perturbation wind model.

W(126) DUFORM -- Format code for perturbation wind model.

W(127) DUID -- Data-set perturbation wind model.

W(128) W(149) -- Parameters for perturbation wind model.

W(150) CMODEL -- Check number for background sound speed model.

W(151) CFORM -- Format code for background sound speed model.

W(152) CID -- Data-set ID for background sound speed model.

W(153) W(174) -- Parameters for background sound speed model.

W(175) DCMODEL -- Check number for perturbation sound speed model.

W(176) DCFORM -- Format code for perturbation sound speed model.

W(177) DCID -- Data-set for perturbation sound speed model.

W(178) W(199) -- Parameters for perturbation sound speed model.

W(200) TMODEL -- Check number for background temperature model.

W(201) TFORM -- Format code for background temperature model.

W(202) TID -- Data-set ID for background temperature model.

W(203) W(224) -- Parameters for background temperature model.

W(225) DTMODEL -- Check number for perturbation temperature model.

W(226) DTFORM -- Format code for perturbation temperature model.

W(227) DTID -- Data-set ID for perturbation temperature model.

W(228) W(249) -- Parameters for perturbation temperature model.

W(250) MMODEL -- Check number for molecular weight model.

W(251) MFORM -- Format code for molecular weight model.

W(252) MID -- Data-set ID for molecular weight model.

W(253) W(274) -- Parameters for molecular weight model.

W(275) RMODEL -- Check number for receiver surface model.

W(276) RFORM -- Format code for receiver surface model.

W(277) RID -- Data-set ID for receiver surface model.

W(278) W(299) -- Parameters for receiver surface model.

W(300) GMODEL -- Check number for background terrain model.

W(301) GFORM -- Format code for background terrain model.

W(302) GID -- Data-set ID for background terrain model.

W(303) W(324) -- Parameters for background terrain model.

W(325) DGMODEL -- Check number for perturbation terrain model.

W(326) DGFORM -- Format code for perturbation terrain model.

W(327) DGID -- Data-set ID for perturbation terrain model.

W(328) W(349) -- Parameters for perturbation terrain model.

W(400) W(500) Parameters for supplementary programs.

W(500) AMODEL -- Check number for viscosity/conductivity model.

W(501) AFORM -- Format code for viscosity/conductivity model.

W(502) AID -- Data-set ID for viscosity/conductivity model.

W(503) W(524) -- Parameters for viscosity/conductivity model.

W(525) DAMODEL -- Check number for perturbation viscosity/conductivity model.

W(526) DAFORM -- Format code for perturbation viscosity/conductivity model.

W(527) DAID -- Data-set ID for perturbation viscosity/conductivity model.

W(528) W(549) -- Parameters for perturbation viscosity/conductivity model.

W(550) PMODEL -- Check number for background pressure model.

W(551) PFORM -- Format code for background pressure model.

W(552) PID -- Data-set ID for background pressure model.

W(553) W(574) -- Parameters for background pressure model.

W(575) DPMODEL -- Check number for perturbation pressure model.

W(576) DPFORM -- Format code for perturbation pressure model.

W(577) DPID -- Data-set ID for perturbation pressure model.

W(578) W(599) -- Parameters for perturbation pressure model.

W(600) W(999) -- Assigned to future atmospheric models

5.4 Specifying Tabular Input

In addition to providing values to the W(n) array, the Input Data File lets you enter tabular data to be used by atmospheric model subroutines. In the following discussion, refer to the Input Data File for the sample case (Fig. 5.1) for examples of tabular data.

5.4.1 Changing to the Tabular Format

When the sign of the W-array index n is read (in columns 1-3), it is checked for a valid negative value, which signals a change in the format of the data to follow. Any valid negative value selects a corresponding "input common block" that has been dedicated to a particular model subroutine. Table 5.2 shows which values select which common blocks. The line in Figure 5.1 that begins with -1 and has the comment "ENTER DATA SUBSET FOR BACKGROUND WIND MODEL" is an example selecting common block /B1/ to receive data. Because this line must conform to the W-array format, comments must begin after column 24.

Table 5.2--Description of the identifying numbers in the first three columns of the Input Data File

Code number	Prefix for common-block variables*	Description
1-999		Input to elements 1-999 of the W array as described in Table 4.4 and Section 5.3.2
-1	U	Signals start of tabular input to common block /B1/ (wind velocity) (see Table 4.2)
-2	DU	Signals start of tabular input to common block /B2/ (perturbation wind velocity)
-3	C	Signals start of tabular input to common block /B3/ (sound speed)
-4	DC	Signals start of tabular input to common block /B4/ (perturbation sound speed)
-5	T	Signals start of tabular input to common block /B5/ (temperature)
-6	DT	Signals start of tabular input to common block /B6/ (perturbation temperature)
-7	M	Signals start of tabular input to common block /B7/ (molecular weight)
-8	R	Signals start of tabular input to common block /B8/ (receiver surface)
-9	G	Signals start of tabular input to common block /B9/ (terrain)
-10	DG	Signals start of tabular input to common block /B10/ (terrain perturbation)
-17	V	Signals start of tabular input to common block /B17/ (viscosity/conductivity)
-18	DV	Signals start of tabular input to common block /B18/ (viscosity/conductivity perturbation)
-19	PR	Signals start of tabular input to common block /B19/ (pressure)
-20	DP	Signals start of tabular input to common block /B20/ (pressure perturbation)
0		Signals end of tabular input to one of the above common blocks, or if that is not appropriate, end of the input data

* See Table 4.5.

5.4.2 The Format Line

The line following the negative value of n begins with an integer that specifies the format of the data to follow. The formats are numbered according to the method shown in Table 5.3.

Format A is alphanumeric and lets you enter descriptive comments on the rest of the line that begins with the format number. That comment is reproduced at the beginning of the program printout. The comment "LOGARITHMIC EASTWARD WIND PROFILE" in Figure 5.1 is an example.

Formats 1, 2, and 3 specify 1, 2, and 3 columns, respectively, of floating-point numbers. If formats 1, 2, or 3 are selected, then columns 4-16 of the format line must contain an end-of-data terminator, such as 999.0 in the sample case. The rest of the format line can contain comments describing the tabular data.

5.4.3 The Units Line

The line following the format line is a line containing unit-conversion specifications for the columnar data to follow. The conversion specifications follow the conventions described in Table 5.1. The 5 characters must be placed in the same 13-character columns as the tabular data to be converted with the extra condition that at least one space must separate adjacent conversion specifications.

5.4.4 The Data Lines

Any number of data lines can follow the format line. They must contain numeric data in the format specified in the format line and must be terminated with the number given in the format line as specifying end-of-data. Data values are read until the terminator is encountered. The data following the "U.S. STANDARD ATMOSPHERE 1962 TEMPERATURE PROFILE" comment in Figure 5.1 is an example of 3-column data entry (format 3) into the common block /B5/, which contains data used by temperature model TTANH5. If the number of data values read in exceeds the maximum allocated in the model subroutine, an error will occur and the program will stop.

Table 5.3--Tabular input data formats available*

Format number	Data block	Format	Data read	Cycle length	Number of columns	Number of arrays in data block
0	1	I3	Indicates end of data read into the common block	NA	NA	NA
A	1	I3,10A8	Format number,** alphanumeric data that describe the atmospheric model	1	NA	1
1	1	I3,G13.6 G13.6	Format number,*** terminator data read consecutively into the data block spaced by the cycle length	Equals dimension of arrays in data block	1	Usually 1
	last	G13.6	Terminator indicates end of data for this data block			
2	1 2-	I3,G13.6 2G13.6	Format number,*** terminator data read consecutively into the data block spaced by the cycle length	Equals dimension of arrays in data block	2	Usually 2
	last	2G13.6	Terminator indicates end of data for this data block			
3	1 2-	I3,G13.6 3G13.6	Format number,*** terminator data read consecutively into the data block spaced by the cycle length	Equals dimension of arrays in data block	3	Usually 3
	last	3G13.6	Terminator indicates end of data for this data block			

* Except for format A, the structure of the data block is a single variable (whose value will be set equal to the number of data values read in to the data block) followed by the data block arrays. This structure must be set up in the subroutines that define the atmospheric model in question.

** Format A implies the data will go into data block 1.

*** Format numbers 1 through 6 imply the data will go into data block 2. The maximum number of columns is now 6.

5.4.5 The Return Line

After the end-of-data number signifies the end of tabular input, you will usually put a line beginning with a 0 in column 3 to return to the W-array data format.

PART III: HOW THE PROGRAM WORKS

6. The Ray-Tracing Equations

6.1 Hamilton's Equations in Spherical Polar Coordinates

HARPA calculates raypaths by numerically integrating Hamilton's equations. Lighthill (1965, 1978) gives Hamilton's equations in four dimensions (three spatial and one temporal) for Cartesian coordinates. Haselgrove (1954) gives Hamilton's equations in three dimensions for spherical polar coordinates, a more useful coordinate system for geophysical media. Combining the two gives Hamilton's equations in four dimensions in which the three spatial coordinates are earth-centered spherical polar (see Table 6.1 for a definition of the symbols):

$$\frac{dr}{d\tau} = \frac{\partial H}{\partial k_r} , \tag{6.1}$$

$$\frac{d\theta}{d\tau} = \frac{1}{r} \frac{\partial H}{\partial k_\theta} , \tag{6.2}$$

$$\frac{d\phi}{d\tau} = \frac{1}{r \sin\theta} \frac{\partial H}{\partial k_\phi} , \tag{6.3}$$

$$\frac{dt}{d\tau} = - \frac{\partial H}{\partial \omega} , \tag{6.4}$$

$$\frac{dk_r}{d\tau} = - \frac{\partial H}{\partial r} + k_\theta \frac{d\theta}{d\tau} + k_\phi \sin\theta \frac{d\phi}{d\tau} , \tag{6.5}$$

$$\frac{dk_\theta}{d\tau} = \frac{1}{r} \left(- \frac{\partial H}{\partial \theta} - k_\theta \frac{dr}{d\tau} + k_\phi\, r \cos\theta \frac{d\phi}{d\tau} \right) , \tag{6.6}$$

$$\frac{dk_\phi}{d\tau} = \frac{1}{r \sin\theta} \left(- \frac{\partial H}{\partial \phi} - k_\phi \sin\theta \frac{dr}{d\tau} - k_\phi\, r \cos\theta \frac{d\theta}{d\tau} \right) , \tag{6.7}$$

$$\frac{d\omega}{d\tau} = \frac{\partial H}{\partial t} . \tag{6.8}$$

The variables r, θ, ϕ are the (Earth-centered) spherical polar coordinates of a point on the raypath; k_r, k_θ, and k_ϕ are the local Cartesian components of the propagation vector (a vector whose magnitude,

Table 6.1--The more important symbols and their definitions

A	In Section 6.1.2, absorption in dB
C	Sound speed
C_{ref}	A reference sound speed (= .344 km/s)
f	Wave frequency (in Hz)
Δf	Frequency shift of a wave due to a time-varying medium
H	Hamiltonian
k_r, k_θ, k_ϕ	Components of the propagation vector in the r, θ, ϕ directions--a vector normal to the wave front having a magnitude $2\pi/\lambda = \omega/v$
k_{disp}	Complex wave number determined by the dispersion relation
M	Mean molecular atmospheric weight (the average is over molecular constituents)
P	Phase path length, phase of the wave divided by the reference wave number ($2\pi/\lambda_o = \omega/C_{ref}$)
P'	Group path length = $C_{ref}t$
p	Atmospheric pressure
R	Gas constant = 8.31436 x 10^{-3} kg (kg mole)$^{-1}$ km^2 s^{-2} K^{-1}
r, θ, ϕ	Spherical polar coordinates of a raypath point
s	Geometric raypath length
T	Atmospheric temperature
t	Time of travel of a wave packet (in some cases, used to express the time dependence of the propagation medium)
V	Wind velocity
V_r, V_θ, V_ϕ	Components of the wind velocity in the r, θ, ϕ direction
v	Wave phase velocity
γ	= 1.4, the ratio of specific heat at constant pressure to that at constant density
θ	Colatitude in spherical polar coordinates
λ	Wavelength
$\lambda_o = (2\pi/\omega)C_{ref}$	Reference wavelength
ρ	Atmospheric density
τ	Independent variable in Hamilton's equations (no physical significance)
ϕ	Longitude in spherical polar coordinates
Ω	$= \omega - \vec{k}\cdot\vec{V}$, the intrinsic wave frequency, the wave frequency as seen by an observer moving with the medium
$\omega = 2\pi f$	Radian wave frequency
$\Delta\omega = 2\pi\Delta f$	Radian frequency shift

$$k = \sqrt{k_r^2 + k_\theta^2 + k_\phi^2} = 2\pi/\lambda \quad , \tag{6.9}$$

is the wave number, and that points in the wave normal direction) in the r, θ, and ϕ directions; t is time--in (6.4) it is the propagation time of a wave packet; in (6.8) it expresses the variation with time of a time-varying medium; τ is a parameter whose value depends on the choice of the Hamiltonian H. Section 6.4 explains how the Hamiltonian is defined.

For actual calculation, HARPA uses group path $P' = C_{ref}\,t$ (where C_{ref} is a standard reference speed) as the independent variable because the derivatives with respect to P' are independent of the choice of Hamiltonian, allowing the program to switch Hamiltonians in the middle of a path. This choice automatically causes the program to take smaller steps in real path length where the calculations are more critical, as when refractive index varies rapidly. These equations are obtained by dividing (6.1) through (6.8) by C_{ref} times (6.4):

$$\frac{dr}{dP'} = -\frac{1}{C_{ref}} \frac{\partial H/\partial k_r}{\partial H/\partial \omega} , \tag{6.10}$$

$$\frac{d\theta}{dP'} = -\frac{1}{rC_{ref}} \frac{\partial H/\partial k_\theta}{\partial H/\partial \omega} , \tag{6.11}$$

$$\frac{d\phi}{dP'} = -\frac{1}{rC_{ref}\sin\theta} \frac{\partial H/\partial k_\phi}{\partial H/\partial \omega} , \tag{6.12}$$

$$\frac{dk_r}{dP'} = \frac{1}{C_{ref}} \frac{\partial H/\partial r}{\partial H/\partial \omega} + k_\theta \frac{d\theta}{dP'} + k_\phi \sin\theta \frac{d\phi}{dP'} , \tag{6.13}$$

$$\frac{dk_\theta}{dP'} = \frac{1}{r} \left(\frac{1}{C_{ref}} \frac{\partial H/\partial \theta}{\partial H/\partial \omega} - k_\theta \frac{dr}{dP'} + k_\phi\, r \cos\theta \frac{d\phi}{dP'}\right) , \tag{6.14}$$

$$\frac{dk_\phi}{dP'} = \frac{1}{r\sin\theta} \left(\frac{1}{C_{ref}} \frac{\partial H/\partial \phi}{\partial H/\partial \omega} - k_\phi \sin\theta \frac{dr}{kP'} - k_\phi\, r\cos\theta \frac{d\theta}{dP'}\right) , \tag{6.15}$$

$$\frac{d(\Delta f)}{dP'} = \frac{1}{2\pi} \frac{d\Delta\omega}{dP'} = \frac{1}{2\pi} \frac{d\omega}{dP'} = -\frac{1}{2\pi} \frac{\partial H/\partial t}{\partial H/\partial \omega} . \tag{6.16}$$

Equation (6.16) for the frequency shift of a wave propagating through a time-varying medium follows directly from Hamilton's equations (6.4) and (6.8). An alternative derivation is given by Bennett (1967). Large frequency shifts should be accumulated along the raypath and the shifted frequency used in calculations at each point on the raypath. Equations (6.1) through (6.8) imply that all eight dependent variables vary along the path, and that at each point on the path the instantaneous value of all parameters (including frequency) is used in further evaluations of the equations. However, the time variation of the atmosphere due to natural causes is so slow that the resulting frequency shifts have negligible effect on the propagation. For this reason, HARPA calculates frequency shift to compare with frequency-shift measurements, but does not adjust the frequency of the wave used in the propagation calculations.

The first six differential equations, (6.10) through (6.15), are always integrated. By setting W(59), the user can choose whether to have the program integrate (6.16) to calculate the frequency shift.

6.1.1 Phase Path

Three other quantities can be calculated by integration along the ray-path. The phase path P (phase divided by the reference wave number $2\pi/\lambda_0 = \omega/C_{ref}$) is calculated by integrating

$$\frac{dP}{dP'} = \frac{C_{ref}}{\omega}\left(k_r \frac{dr}{dP'} + k_\theta\, r \frac{d\theta}{dP'} + k_\phi\, r \sin\theta \frac{d\phi}{dP'}\right)$$

$$= -\frac{1}{\omega}\,\frac{k_r \dfrac{\partial H}{\partial k_r} + k_\theta \dfrac{\partial H}{\partial k_\theta} + k_\phi \dfrac{\partial H}{\partial k_\phi}}{\partial H/\partial \omega} \quad . \qquad (6.17)$$

6.1.2 Absorption

If the absorption per wavelength is small (as it must be for this type of ray tracing to be valid), then an approximate formula can be integrated to give the absorption in decibels:

$$\frac{dA}{dP'} = -\frac{10}{\log_e 10}\,\frac{\omega}{C_{ref}}\,\frac{\mathrm{imag}\,(k^2_{disp})}{k_r^2 + k_\theta^2 + k_\phi^2}\,\frac{dP}{dP'}$$

$$= \frac{10}{\log_e 10}\,\frac{\mathrm{imag}\,(k^2_{disp})}{k_r^2 + k_\theta^2 + k_\phi^2}\,\frac{k_r \dfrac{\partial H}{\partial k_r} + k_\theta \dfrac{\partial H}{\partial k_\theta} + k_\phi \dfrac{\partial H}{\partial k_\phi}}{C_{ref}\ \partial H/\partial \omega} \quad , \qquad (6.18)$$

where k_{disp} is the (complex) wave number determined by the dispersion relation. If you want to include the effects of absorption from some independent formula for absorption, then you can add that as a new equation to HAMLTN. The appropriate equation would be

$$dA/dP' = \omega/C_{ref}\ \alpha/k_{real}\ dP/dP' \ , \qquad (6.19)$$

where α is an absorption in dB/km, and A will be the absorption in dB.

6.1.3 Path Length

The geometrical path length of the ray can be calculated by integrating

$$\frac{ds}{dP'} = \sqrt{\left(\frac{dr}{dP'}\right)^2 + r^2\left(\frac{d\theta}{dP'}\right)^2 + r^2 \sin^2\theta \left(\frac{d\phi}{dP'}\right)^2}$$

$$= \frac{\sqrt{\left(\frac{\partial H}{\partial k_r}\right)^2 + \left(\frac{\partial H}{\partial k_\theta}\right)^2 + \left(\frac{\partial H}{\partial k_\phi}\right)^2}}{c_{ref}\left|\partial H/\partial\omega\right|} \quad . \tag{6.20}$$

The user can choose to integrate and print frequency shift, phase time, absorption, or path length using Equations (6.16), (6.17), (6.18), or (6.20) by setting the appropriate values of W(59), W(57), W(58), W(60), respectively, in the Input Data File (Figure 2.19).

The user can add differential equations to the program by modifying HAMLTN, the subroutine that evaluates Hamilton's equations.

The Hamiltonian and its derivatives are calculated by one of the versions of dispersion-relation subroutine (with entry point DISPER), which also calculates k_{disp}^2.

6.2 Numerical Integration

Subroutines RKAM and RKAM1 integrate the differential equations numerically using an Adams-Moulton predictor-corrector method with a Runge-Kutta starter. RKAM1 was adapted from a program in the CDC CO-OP library called RKAMSUB, written by G. J. Lastman and dated March 1964. The program executes one integration step by one of four methods the user can specify using W(41). The subroutine is called once for each advance of the independent variable. The flow charts of Figures 6.1 through 6.5 show how the integration routine works.

Usually, RKAM1 is run in mode 3, that is, Adams-Moulton integration with relative-error checking. The user can trade execution time for accuracy by varying the single-step integration error, W(42). RKAM1 will increase or decrease the step length to maintain the specified error. Table 6.2 gives an

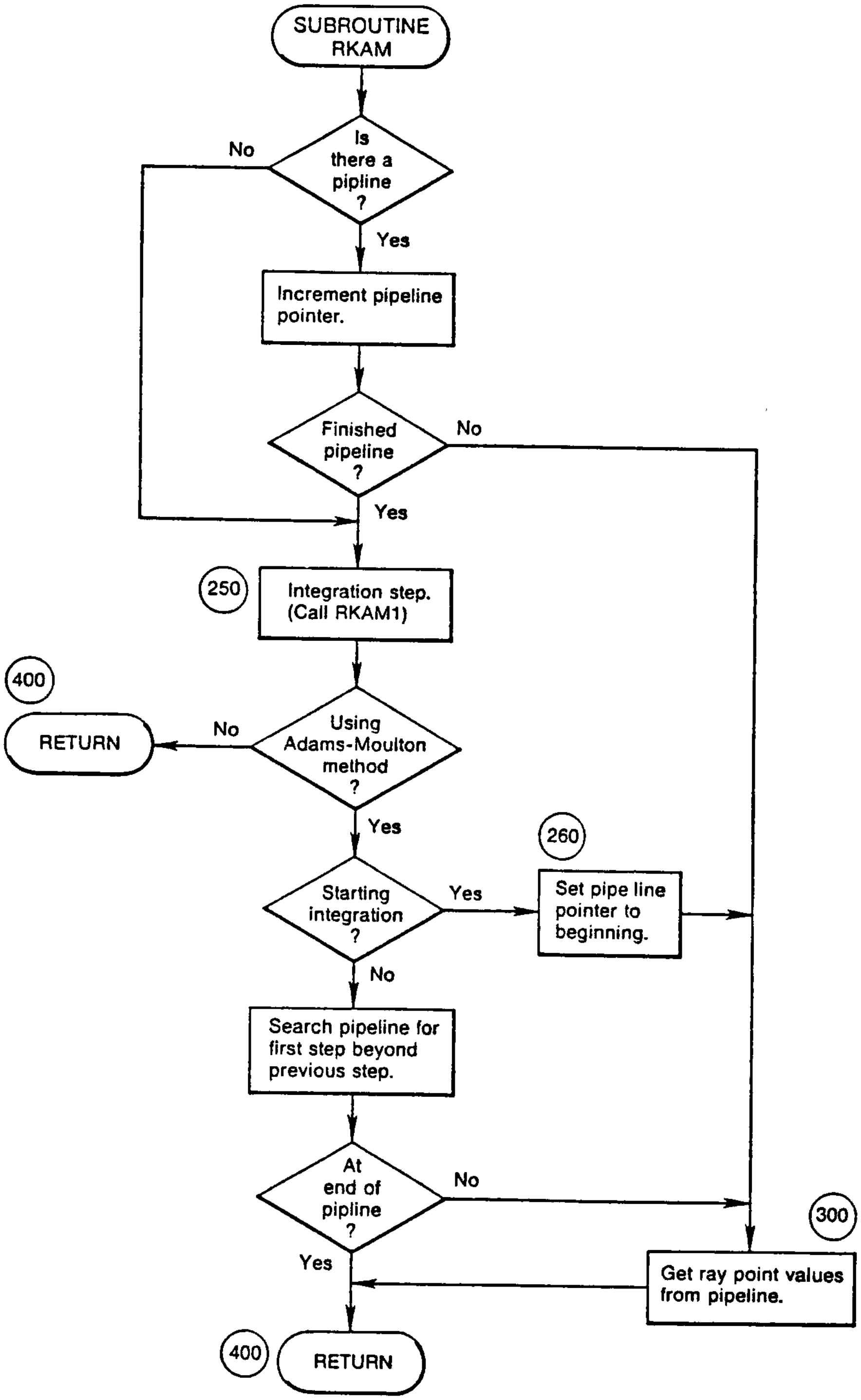


Figure 6.1. Flow chart for subroutine RKAM. The term "pipeline" refers to the sequence of four consecutive integration steps kept by the Adams-Moulton integration method.

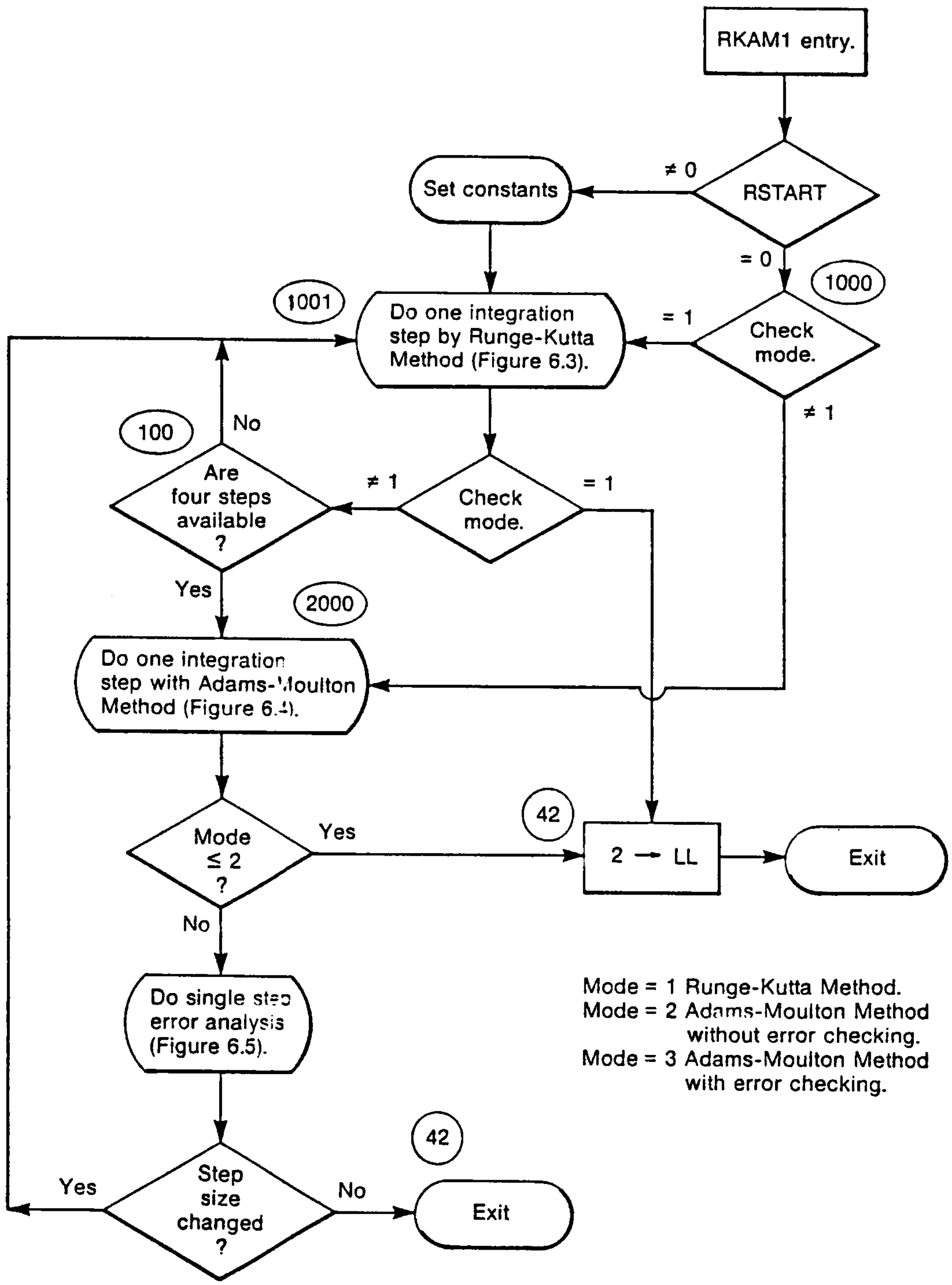


Figure 6.2. Flow chart for subroutine RKAM1.

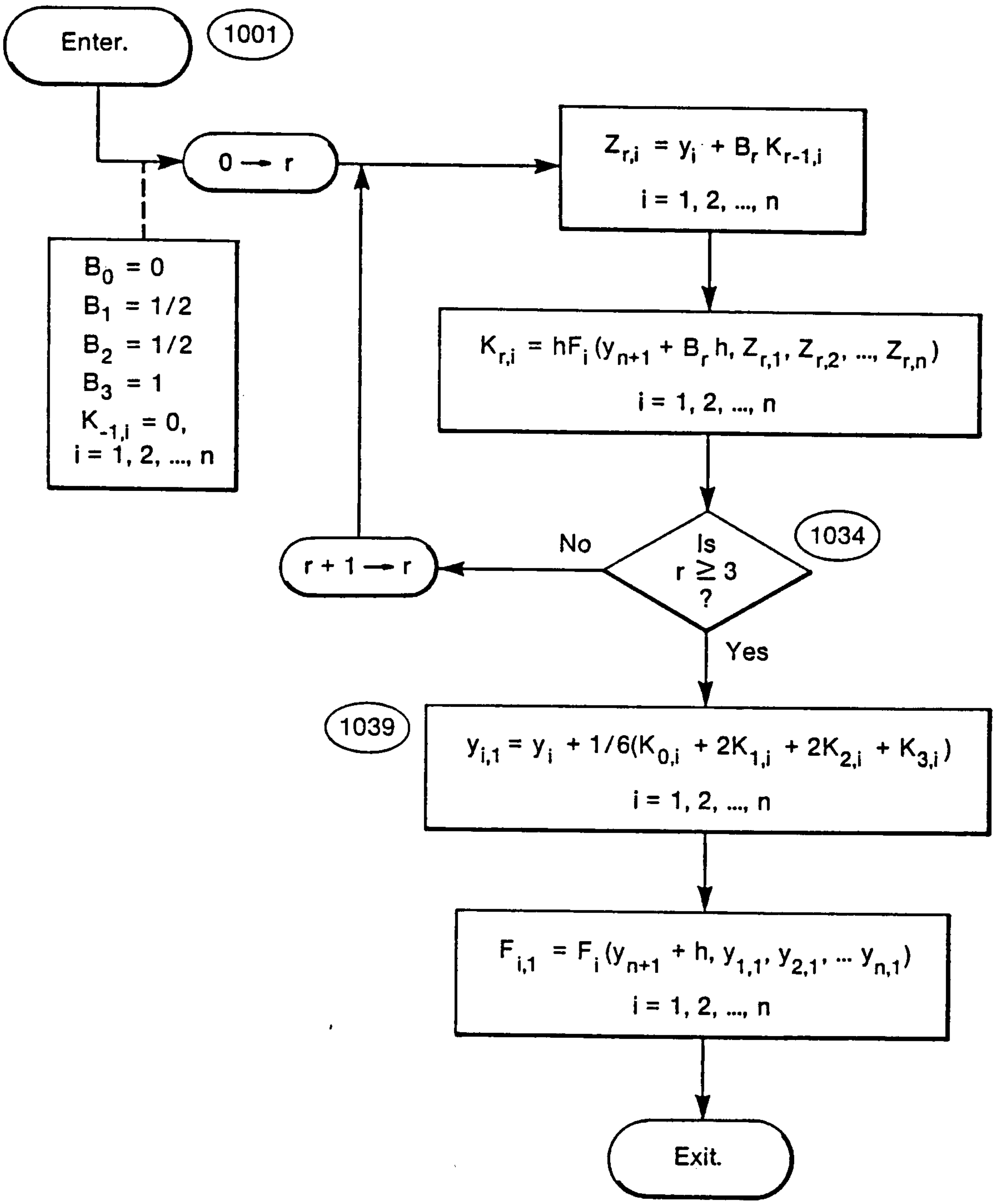


Figure 6.3. Flow chart for the Runge-Kutta procedure. The variables y_1, y_2, ... y_n are the dependent variables, n is the order of the system, and y_{n+1} is the independent variable.

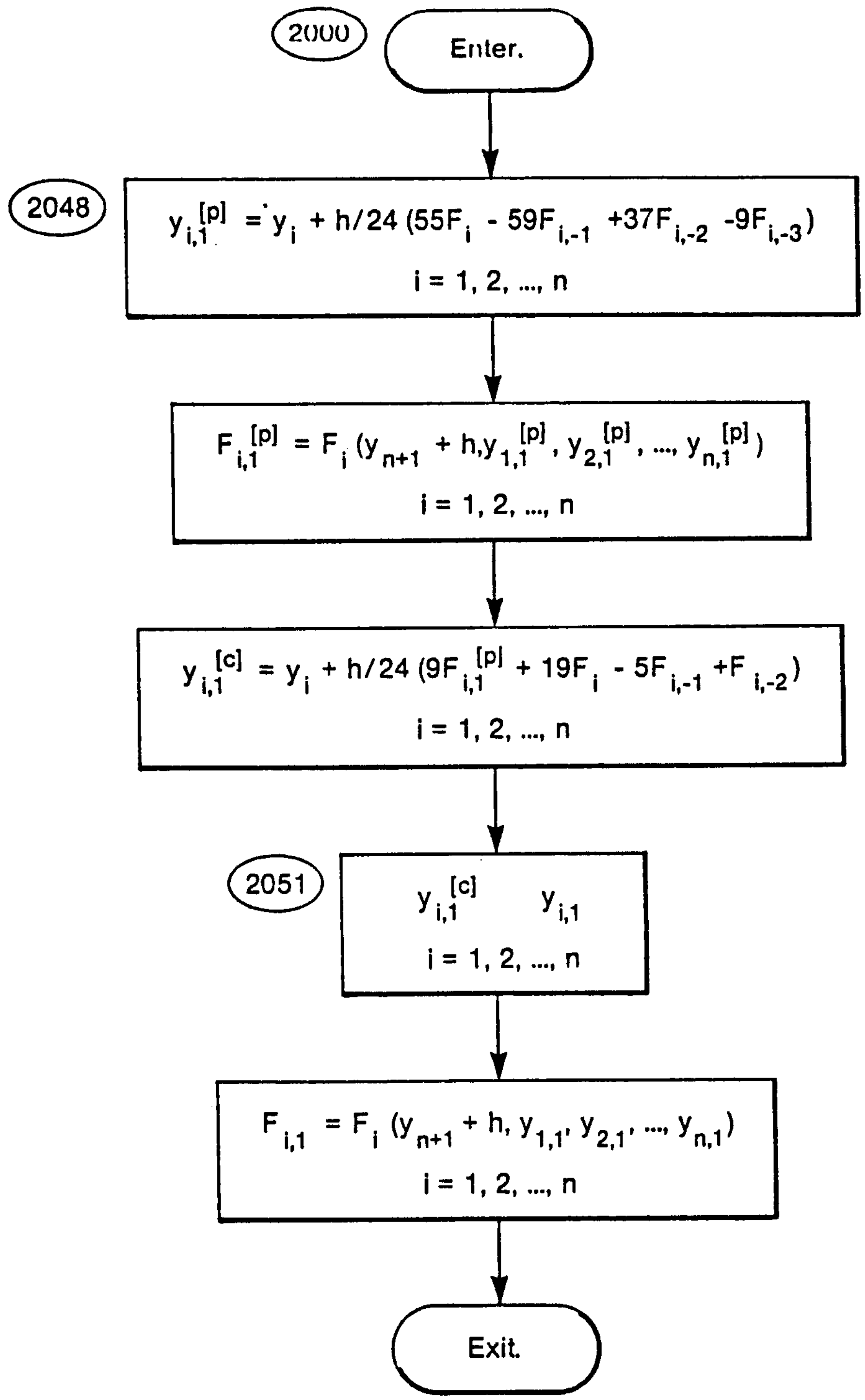


Figure 6.4. Flow chart for the Adams-Moulton integration method. The four starting values needed for this method are supplied by the Runge-Kutta method.

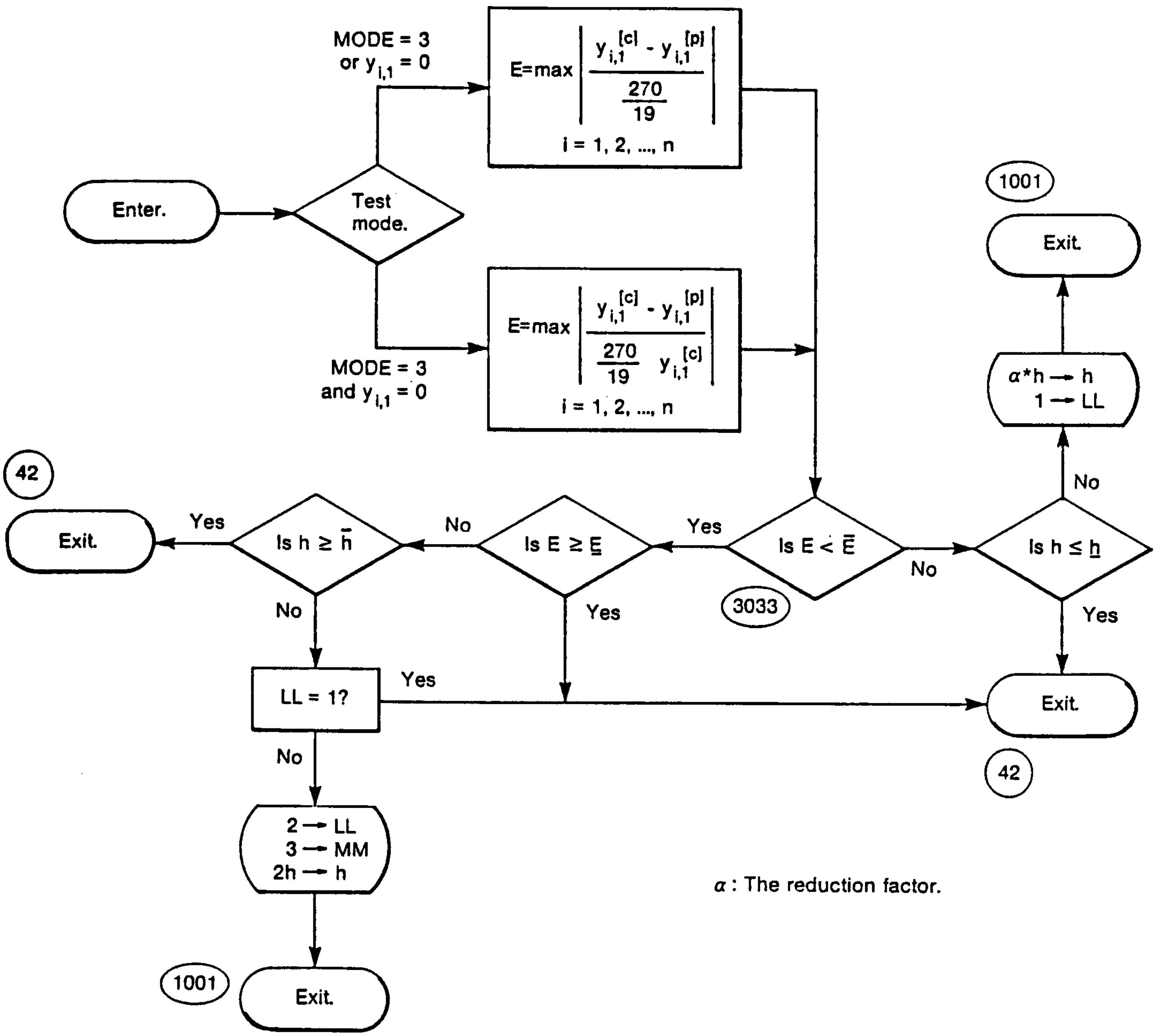


Figure 6.5. Flow chart for the single-step error analysis. The symbol E represents the maximum single-step error, and $\bar{E}$, $\underline{E}$, $\bar{h}$, and $\underline{h}$ represent the maximum and minimum acceptable single-step error and the maximum and minimum mesh size, respectively.

Table 6.2--Run time vs. accuracy

Maximum integration error, W(42)	Run time (s)	Range error
10^{-9}	9.9	$<10^{-7}$
10^{-8}	7.1	$<10^{-7}$
10^{-7}	4.5	1×10^{-7}
10^{-6}	3.3	1×10^{-6}
10^{-5}	2.3	6×10^{-6}
10^{-4}	1.7	1.7×10^{-4}
10^{-3}	1.2	5.7×10^{-3}

Note: Data obtained using model OT2; elevation angle = 4.46°; range = 1000 km; computer: CYBER 750.

idea of how the tradeoff works for a single ray calculation using the ocean version, HARPO and a model of the ocean sound channel described by Georges et al. (1986).

The user can vary other parameters, W(43) - W(47), that control the way RKAM1 adjusts its step length and controls errors (see Sec. 5.3.2), but the initial values assigned in the sample case have been found to work best for most cases met in practice. If the scale of the model differs greatly from the sample case, the initial step length, W(44), should be adjusted.

6.3 The Atmospheric Sound-Speed Model (Subroutine GAMRTDM)

Atmospheric sound-speed models can be specified either directly as $C(r,\theta,\phi,t)$, or they can call models of other atmospheric variables such as temperature $T(r,\theta,\phi,t)$ using a connecting definition of $C(T)$. One choice we provide assumes that the atmosphere is a perfect gas with the equation of state

$$p = \frac{\rho R T}{M} , \tag{6.21}$$

where p is the pressure, ρ is the density,

$$R = 8.31436 \times 10^{-3} \text{ kg (kg mole)}^{-1} \text{ km}^2 \text{ s}^{-2} \text{ K}^{-1} \qquad (6.22)$$

is the universal gas constant, T is the absolute temperature in Kelvins, and

$$M = M(r,\theta,\phi) \qquad (6.23)$$

is the mean molecular weight of the atmosphere.

The square of the sound speed is

$$C^2 = \frac{\gamma RT}{M}, \qquad (6.24)$$

where

$$\gamma = 1.4 \qquad (6.25)$$

is the ratio of the specific heat at constant pressure to the specific heat at constant density. Subroutine GAMRTDM implements Equation (6.24) in HARPA.

The temperature and its gradients are provided as a function of t,r,θ,ϕ by one of the atmospheric temperature-model subroutines. The molecular weight and its gradients are provided as a function of t, r,θ,ϕ by the molecular weight model subroutine. The average molecular weight generally decreases with height because the relative constituency of the atmosphere varies with height. The main effect is from the increase in the fraction of mono-atomic molecules with height. Although the height variation of molecular weight varies with season and time of day as the atmosphere expands and contracts, it can be accurately modeled as a function of the atmospheric pressure (Jones and Georges, 1976). The only molecular weight model included in the present models treats it as a constant, however.

The gradients of the square of the sound speed are as follows.

$$\frac{\partial C^2}{\partial t} = C^2 \left(\frac{1}{T}\frac{\partial T}{\partial t} - \frac{1}{M}\frac{\partial M}{\partial t}\right) \qquad (6.26)$$

$$\frac{\partial C^2}{\partial r} = C^2 \left(\frac{1}{T}\frac{\partial T}{\partial r} - \frac{1}{M}\frac{\partial M}{\partial r}\right) \qquad (6.27)$$

$$\frac{\partial C^2}{\partial \theta} = C^2 \left(\frac{1}{T}\frac{\partial T}{\partial \theta} - \frac{1}{M}\frac{\partial M}{\partial \theta}\right) \qquad (6.28)$$

$$\frac{\partial C^2}{\partial \phi} = C^2 \left(\frac{1}{T}\frac{\partial T}{\partial \phi} - \frac{1}{M}\frac{\partial M}{\partial \phi}\right) . \qquad (6.29)$$

Table 7.11 shows how the output from the sound-speed model subroutine is organized in common block /CC/.

6.4 Acoustic Dispersion Relations

HARPA gains versatility without sacrificing speed by having several versions of some of the subroutines. For example, the four versions of the dispersion relation subroutine allow the user to decide in making up a run module (Section 3.3.2) whether to include or ignore winds and absorption. (If there are no winds (or absorption) in the calculation, it is much cheaper to leave them out of the equations than it is to make the calculations with zero wind, or zero absorption).

The input to the dispersion-relation subroutine is through blank common and common blocks /UU/, /CC/, /TT/, and /MM/. Output is through common block /RIN/. The dispersion-relation subroutine is called through the entry point DISPER. The subroutine names are used only for user identification.

HARPA has four versions of the dispersion-relation subroutine.

(1) ANWNL - Acoustic waves, No Winds, No Losses
(2) AWWNL - Acoustic waves, With Winds, No Losses
(3) ANWWL - Acoustic waves, No Winds, With Losses
(4) AWWWL - Acoustic Waves, With Winds, With Losses

All of these versions calculate a Hamiltonian and its derivatives and the square of the wave number that satisfies the dispersion relation. All of the above variables and some others are in the common block /RIN/ (described in Table 7.9), which has all of the output from the dispersion-relation subroutines.

6.4.1 ANWNL - Acoustic, No Winds, No Losses

The dispersion relation for pure acoustic waves is

$$k^2 = k_r^2 + k_\theta^2 + k_\phi^2 = \frac{\omega^2}{C^2} , \tag{6.30}$$

where $C(t,r,\theta,\phi)$ is the speed of sound (provided by one of the sound-speed model subroutines).

At the beginning of the numerical integration, the magnitude of $\vec{k}$ is automatically set by the program so that the dispersion relation (4.1) is satisfied. During the numerical integration, the components of $\vec{k}$ are allowed to vary according to Hamilton's equations. Because of integration errors, there will be slight differences between k^2 and

$$k_{disp}^2 = \frac{\omega^2}{C^2} , \tag{6.31}$$

the value it would have according to the dispersion relation. As a check on the accuracy of the numerical integration and on the consistency of the equations, the quantity

$$\text{ERROR} \equiv \frac{k^2}{k_{disp}^2} - 1 \tag{6.32}$$

is printed at each step of the raypath calculation. It is possible, however, for ERROR to exceed somewhat the maximum allowable single-step integration error (W42) because k does not vary monotonically along the raypath. ERROR serves mainly as a check that the integration is proceeding correctly and that there are no errors in computing derivatives in the model subroutines. If ERROR is generally smaller than W(42), then integrated quantities that vary monotonically (like travel time) will be computed with the accuracy given by W(42).

We use the following form of the dispersion relation for the Hamiltonian:

$$H(t,r,\theta,\phi,\omega,k_r,k_\theta,k_\phi) = \omega^2 - (k_r^2 + k_\theta^2 + k_\phi^2)\, C^2(t,r,\theta,\phi) . \tag{6.33}$$

The partial derivatives of the Hamiltonian are

$$\frac{\partial H}{\partial t} = - k^2 \frac{\partial C^2}{\partial t} \tag{6.34}$$

$$\frac{\partial H}{\partial r} = - k^2 \frac{\partial C^2}{\partial r} \tag{6.35}$$

$$\frac{\partial}{\partial \theta} = - k^2 \frac{\partial C^2}{\partial \theta} \tag{6.36}$$

$$\frac{\partial H}{\partial \phi} = - k^2 \frac{\partial C^2}{\partial \phi} \tag{6.37}$$

$$\frac{\partial H}{\partial \omega} = 2\omega \tag{6.38}$$

$$\frac{\partial H}{\partial k_r} = - 2 C^2 k_r \tag{6.39}$$

$$\frac{\partial H}{\partial k_\theta} = - 2 C^2 k_\theta \tag{6.40}$$

$$\frac{\partial H}{\partial k_\phi} = - 2 C^2 k_\phi \tag{6.41}$$

$$k \cdot \frac{\partial H}{\partial k} = - 2 C^2 k^2 \quad . \tag{6.42}$$

6.4.2 AWWNL - Acoustic, With Winds, No Losses

The dispersion relation for pure acoustic waves in terms of the sound speed in the presence of winds is

$$\Omega^2 - C^2 k^2 = 0 \ , \tag{6.43}$$

where

$$k^2 = k_r^2 + k_\theta^2 + k_\phi^2 \tag{6.44}$$

and

$$\Omega = \omega - \vec{k} \cdot \vec{V} = \omega - k_r V_r - k_\theta V_\theta - k_\phi V_\phi \tag{6.45}$$

is the intrinsic frequency of the wave (the frequency seen by an observer moving with the wind). $\vec{V}(t,r,\theta,\phi)$ is the wind velocity (provided by a wind-velocity model subroutine).

At the beginning of the numerical integration, the magnitude of k is set by the program so that the dispersion relation (6.43) is satisfied. During the numerical integration, the components of $\vec{k}$ are allowed to vary according to Hamilton's equations. Because of integration errors, there will be slight differences between K^2 and

$$k^2_{disp} = \frac{\omega^2}{\left(c + \frac{\vec{k}\cdot\vec{V}}{k}\right)^2}, \tag{6.46}$$

the value it would have according to the dispersion relation. Notice that

$$\frac{\vec{k}\cdot\vec{V}}{k} = \frac{k_r V_r + k_\theta V_\theta + k_\phi V_\phi}{\sqrt{k_r^2 + k_\theta^2 + k_\phi^2}} \tag{6.47}$$

is independent of the magnitude of $\vec{k}$, as it should be.

As a check on the accuracy of the numerical integration and on the consistency of the equations, the quantity in (6.32) is printed at each step of the raypath calculation.

We use the following form of the dispersion relation for the Hamiltonian:

$$H(t,r,\theta,\phi,\omega,k_r,k_\theta,k_\phi) =$$

$$\Omega^2(t,r,\theta,\phi,\omega,k_r,k_\theta,k_\phi) - (k_r^2+k_\theta^2+k_\phi^2)\ c^2(t,r,\theta,\phi), \tag{6.48}$$

where the intrinsic frequency Ω is given by (6.45)

The partial derivatives of the Hamiltonian are as follows.

$$\frac{\partial H}{\partial t} = 2\Omega\,\frac{\partial \Omega}{\partial t} - k^2\,\frac{\partial c^2}{\partial t} \tag{6.49}$$

$$\frac{\partial H}{\partial r} = 2\Omega\,\frac{\partial \Omega}{\partial r} - k^2\,\frac{\partial c^2}{\partial r} \tag{6.50}$$

$$\frac{\partial H}{\partial \theta} = 2\Omega\,\frac{\partial \Omega}{\partial \theta} - k^2\,\frac{\partial c^2}{\partial \theta} \tag{6.51}$$

$$\frac{\partial H}{\partial \phi} = 2\Omega\,\frac{\partial \Omega}{\partial \phi} - k^2\,\frac{\partial c^2}{\partial \phi} \tag{6.52}$$

$$\frac{\partial H}{\partial \omega} = 2\Omega \tag{6.53}$$

$$\frac{\partial H}{\partial k_r} = -\,2(\Omega V_r + c^2\,k_r) \tag{6.54}$$

$$\frac{\partial H}{\partial k_\theta} = -\,2(\Omega V_\theta + c^2\,k_\theta) \tag{6.55}$$

$$\frac{\partial H}{\partial k_\phi} = -2(\Omega V_\phi + C^2 k_\phi) \tag{6.56}$$

$$k \cdot \frac{\partial H}{\partial k} = k_r \frac{\partial H}{\partial k_r} + k_\theta \frac{\partial H}{\partial k_\phi} + k_\phi \frac{\partial}{\partial k_\phi} = -2(\Omega \vec{k}\cdot\vec{V} + C^2 k^2) \ , \tag{6.57}$$

where

$$\frac{\partial \Omega}{\partial t} = -k_r \frac{\partial V_r}{\partial t} - k_\theta \frac{\partial V_\theta}{\partial t} - k_\phi \frac{\partial V_\phi}{\partial t} \tag{6.58}$$

$$\frac{\partial \Omega}{\partial r} = -k_r \frac{\partial V_r}{\partial r} - k_\theta \frac{\partial V_\theta}{\partial r} - k_\phi \frac{\partial V_\phi}{\partial r} \tag{6.59}$$

$$\frac{\partial \Omega}{\partial \theta} = -k_r \frac{\partial V_r}{\partial \theta} - k_\theta \frac{\partial V_\theta}{\partial \theta} - k_\phi \frac{\partial V_\phi}{\partial \theta} \tag{6.60}$$

$$\frac{\partial \Omega}{\partial \phi} = -k_r \frac{\partial V_r}{\partial \phi} - k_\theta \frac{\partial V_\theta}{\partial \phi} - k_\phi \frac{\partial V_\phi}{\partial \phi} \ . \tag{6.61}$$

6.4.3 ANWWL - Acoustic, No Winds, With Losses and AWWWL - Acoustic, With Winds, With Losses

Propagation losses for acoustic waves in the atmosphere occur when viscosity and thermal conductivity are included in the Navier-Stokes equations that describe fluid dynamics. The acoustic wave dispersion relation that results gives a complex wave number. The imaginary part of that wave number leads to absorption as explained in Section 6.1.2.

If the imaginary part of the wave number is small ($k_{imag}^2 << k_{real}^2$) (as it must be for ordinary ray tracing such as described here to be valid), then we can neglect the effect of viscosity and thermal conductivity on the real part of the wave number so that the raypaths will be unaffected by viscosity and thermal conductivity.

Within this assumption, we can approximate (the square of) the imaginary part of the acoustic wave number for propagation through a perfect gas as (Gossard and Hooke, 1975, pp. 230-233)

$$k_{imag}^2 = -\Omega/(\gamma P)\ [4/3\mu + (\gamma-1)^2\ M/(\gamma R)\ \kappa]\ k_{real}^2 \ , \tag{6.62}$$

where $\gamma = 1.4$ is the ratio of specific heat at constant pressure to that at constant density, P is the atmospheric pressure, μ is the viscosity, M is the atmospheric mean molecular weight, R is the universal gas constant, and κ is the thermal conductivity.

Whenever you use a dispersion relation with losses (ANWWL or AWWWL), you have to supply models for P, μ, κ and M. We provide one model for μ and κ (MUARDC), one for P (PEXP), and one for M (MCONST).

6.5 Reflecting Rays From Irregular Terrain

This section describes the method HARPA uses to find the intersections of a ray with any terrain surface and to compute the correct reflected ray, even in an anisotropic medium (an atmosphere with winds). A more detailed discussion of terrain-reflection algorithms is given in the report by Jones (1982). Essentially the same methods are used to detect receiver-surface crossings.

6.5.1 Detecting Ray Intersections With the Terrain

Detecting that the ray has crossed the terrain surface is straightforward for the simple case shown in Figure 6.6. Figure 6.7 shows a more difficult example in which the ray crosses the terrain surface twice. It is difficult to distinguish such a raypath from the example in Figure 6.8, knowing only the raypath coordinates and ray directions between the integration steps (shown by dots in Figures 6.7 and 6.8). A further difficulty is that iteration of the numerical integration in Figure 6.7 might lead to finding the wrong intersection with the surface of discontinuity, a problem shared by the example in Figure 6.9.

Although infrequent, the difficulties illustrated in Figures 6.7, 6.8 and 6.9 occur often enough that any useful algorithm must be able to handle them. These difficult cases obviously occur more often when large integration steps are used. This section derives algorithms that can handle these difficult cases in addition to the straightforward cases. Figure 6.10 illustrates cases that HARPA's intersection algorithm will not correctly handle unless step length is decreased or the terrain model is made smoother.

HARPA accepts only terrain models whose surface is smooth and whose slope and curvature are continuous. Thus, wedge-shaped surfaces, for example, are

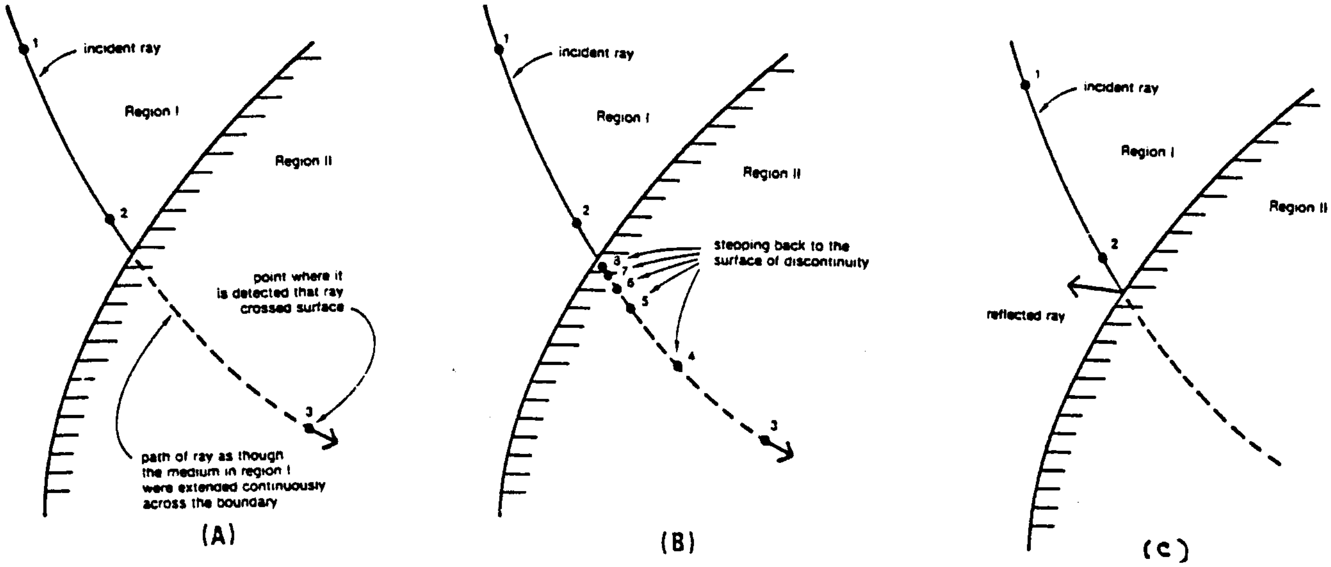


Figure 6.6. Three steps in calculating the intersection and reflection of a ray at a terrain surface: (a) recognition that the ray has crossed the terrain surface (numbers indicate successive positions of the ray after each integration step; the raypath is curved because the atmosphere is inhomogeneous); (b) iteration by numerical integration to find the intersection of the ray with the terrain surface; and (c) computation of the reflected ray ready to start numerical integration in a new direction in the atmosphere. The same algorithm could be used to compute ray refraction at discontinuities of refractive index.

not allowed. Not only are diffraction effects important at such edges, but some of the algorithms developed in this report may not work properly for surfaces with edges. In addition, the algorithms used by HARPA may not always properly handle surfaces that contain caves or tunnels.

At each integration step in the raypath calculation, we assume that the following information is available:

(1) The position of the ray point (r, θ, ϕ) in spherical polar (Earth-centered) coordinates,

(2) The local Cartesian components (k_r, k_θ, k_ϕ) of the wave vector (a vector pointing in the wave-normal direction and normalized so that the magnitude equals the wave number),

(3) The accumulated group time delay, t, of the ray, (which is also the independent variable for the numerical integration), and

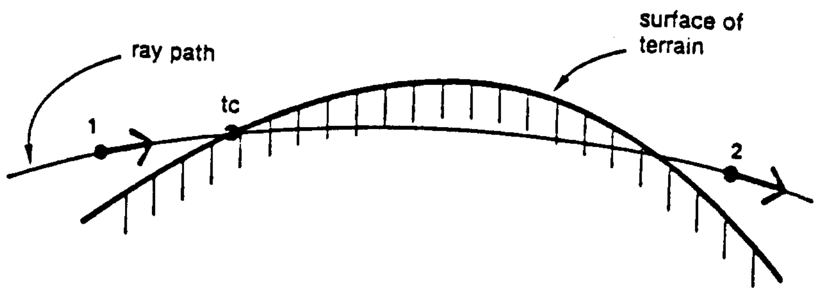


Figure 6.7. A ray crossing a terrain or receiver surface and crossing back again between integration steps. An algorithm that checked only to see if the ray at point 2 was in a different medium than point 1 would miss the intersections.

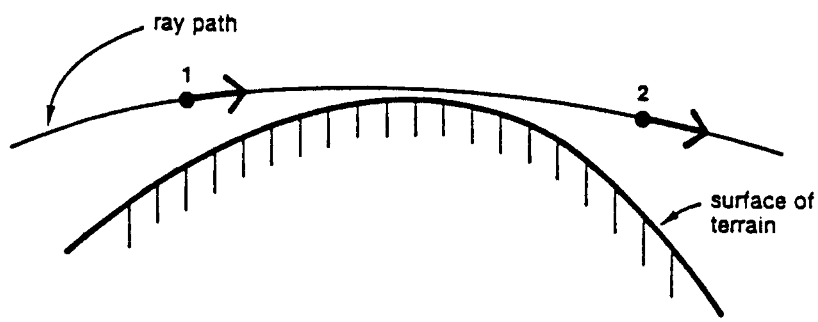


Figure 6.8. A ray that nearly intersects a terrain or receiver surfce. A useful intersection algorithm must be able to distinguish this case from the one depicted in figure 6.7.

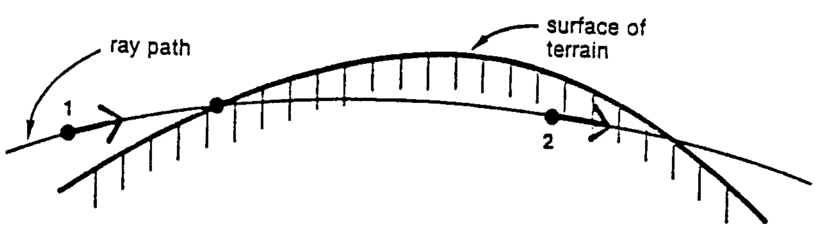


Figure 6.9. A ray crossing a terrain or receiver surface and ending closer to a second intersection. A useful algorithm must step backward and find the first intersection. HARPA will correctly handle the three cases on this page.

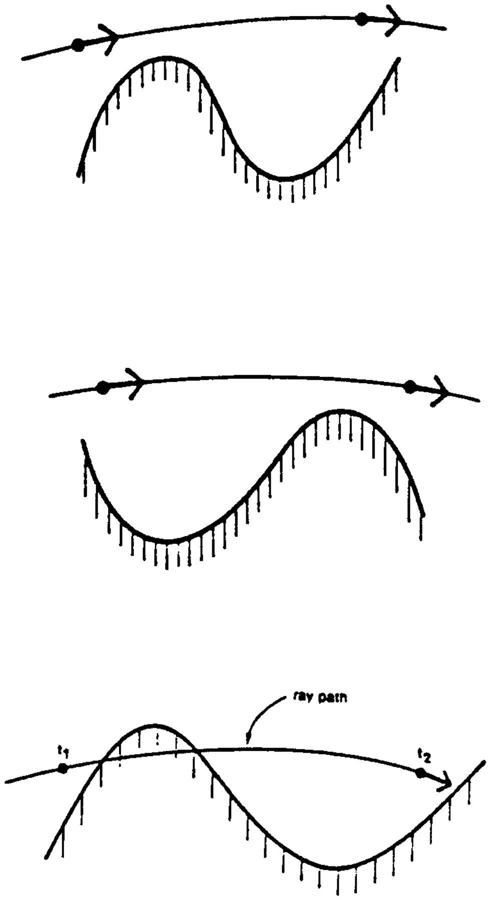


Figure 6.10. Examples of terrain (or receiver-surface) encounters that HARPA's intersection algorithm will not correctly handle. Step length must be decreased, or the terrain model must be smoothed.

(4) The derivatives

$$\dot{r} \equiv dr/dt \tag{6.63}$$

$$\dot{\theta} \equiv d\theta/dt \tag{6.64}$$

$$\dot{\phi} \equiv d\phi/dt \tag{6.65}$$

$$\dot{k}_r \equiv dk_r/dt \tag{6.66}$$

$$\dot{k}_\theta \equiv dk_\theta/dt \tag{6.67}$$

$$\dot{k}_\phi \equiv dk_\phi/dt \; . \tag{6.68}$$

We assume also that the values of any of these variables can be saved from one step to the next for comparison. In particular, we make the following approximations,

$$\ddot{r} \approx (\dot{r}(t_2) - \dot{r}(t_1))/t_2 - t_1) \qquad (6.69)$$

$$\ddot{\theta} \approx (\dot{\theta}(t_2) - \dot{\theta}(t_1))/(t_2 - t_1) \qquad (6.70)$$

$$\ddot{\phi} \approx (\dot{\phi}(t_2) - \dot{\phi}(t_1))/(t_2 - t_1) \ , \qquad (6.71)$$

which allow us to estimate the curvature of ray. In using these algorithms, it would probably be useful to incorporate a test of the validity of the above approximations by evaluating both sides of (6.69) - (6.71), but HARPA does not do that.

The simplest method for predicting whether an extension of the raypath from a point will intersect the terrain surface (or the receiver surface) is to extend the ray in a straight line and see if it meets the terrain when the surface is extended as a plane by using the local slope. However, whenever the curvature of the ray and the terrain are small enough for that approximation to be useful, the ray would probably eventually go below the terrain on one of the integration steps, and the prediction from the simplest algorithm would not be needed.

Because we want an algorithm sophisticated enough to estimate the time of nearest intersection (if one occurs) in cases like those in Figures 6.7 or 6.8, we should include at least the local curvature in any approximations. Also, because it would be difficult to deal with a higher order approximation, a quadratic approximation to both the ray and the terrain seems to be the best compromise.

In addition, when searching for an intersection with the terrain (or receiver) surface, the intersection must be in the correct direction. In the case of the terrain, the intersection will always be into the ground. In the case of the receiver surface, the correct direction of crossing will alternate from one crossing to the next, but for each crossing, the direction will be known. For the purposes of the present development, it is useful to define a parameter S that is equal to +1 if the wanted crossing is upward and -1 if the wanted crossing is downward. Thus, the value of S will always be -1 for a terrain crossing.

6.5.2 The Surface Model

HARPA expresses an arbitrary model of a terrain or receiver surface in the form

$$f(r, \theta, \phi) = 0 \ . \tag{6.72}$$

Because f is zero only on the surface, it always has one sign on one side of the surface and the opposite sign on the other side. Let us call the side of the surface that is underground the inside and the other side the outside. Then we can arbitrarily require that f be positive outside of the surface and negative inside the surface. We can similarly designate an inside and an outside for the receiver surface and make the same requirement on f.

Thus, we have

$$f(r, \theta, \phi) > 0 \tag{6.73}$$

outside the surface and

$$f(r, \theta, \phi) < 0 \tag{6.74}$$

inside the surface. The time derivative of f (which indicates the rate that the ray is moving away from the surface) is

$$\dot{f} = f_r \dot{r} + f_\theta \dot{\theta} + f_\phi \dot{\phi} \tag{6.75}$$

where a subscript indicates a partial derivative with respect to the subscript. The time derivative of (6.75) is

$$\ddot{f} = f_r \ddot{r} + f_\theta \ddot{\theta} + f_\phi \ddot{\phi} + f_{rr} \dot{r}^2 + f_{\theta\theta} \dot{\theta}^2 + f_{\phi\phi}\dot{\phi}^2 + 2f_{r\theta} \dot{r}\dot{\theta} + 2f_{r\phi} \dot{r}\dot{\phi} + 2f_{\theta\phi} \dot{\theta}\dot{\phi}. \tag{6.76}$$

Assuming that (6.76) is nearly constant locally, we have

$$f = f_1 + \dot{f}_1(t - t_1) + \frac{1}{2} \ddot{f}_1(t - t_1)^2 \ , \tag{6.77}$$

where the subscript 1 refers to the values at the time t_1. We want to find the value of t for which the ray intersects the surface, that is, where f = 0. Let t_c be the value for which f = 0. Then

$$0 = f_1 + \dot{f}_1(t_c - t_1) + \frac{1}{2}\ddot{f}_1(t_c - t_1)^2 . \tag{6.78}$$

For simplicity, let us drop the subscript 1 in (6.78), but remember that f and t without subscripts refer to the integration step on the raypath where we are trying to estimate the time t_c where the ray will intersect the surface. Then (6.78) becomes

$$0 = f + \dot{f}(t_c - t) + \frac{1}{2}\ddot{f}(t_c - t)^2 . \tag{6.79}$$

The solution of (6.79), for which the ray is crossing from inside to outside when S = +1 and crossing from outside to inside when S = -1, is

$$t_c - t = \frac{-\dot{f} + S\sqrt{\dot{f}^2 - 2f\ddot{f}}}{\ddot{f}} . \tag{6.80}$$

Within the approximation made here, the ray will intersect the surface if

$$\dot{f}^2 - 2f\ddot{f} > 0 \tag{6.81}$$

but will make a closest approach to the surface if

$$\dot{f}^2 - 2f\ddot{f} < 0 . \tag{6.82}$$

In the latter case, we can estimate the time t_p at which f is a minimum.

$$t_p - t = -\frac{\dot{f}}{\ddot{f}} . \tag{6.83}$$

This is close to the time when the ray makes a closest approach to the surface. If the second derivative in (6.76) is very small, then the formula in (6.80) may be impractical. In that case, the solution

$$t_c - t = \frac{-2f}{\dot{f} + S\sqrt{\dot{f}^2 - 2f\ddot{f}}} \tag{6.84}$$

is more useful. The advantage of (6.84) is that it is uniformly valid as $\ddot{f}$ approaches zero.

In HARPA, different formulas are used in different circumstances: statement 501 in TRACE uses (6.84) to estimate t_c. Subroutine BACKUP uses

$$t_c = t - f/\dot{f} \tag{6.85}$$

when it is stepping to an intersection. Subroutine BACKUP uses (6.83) to step to a closest approach. When BACKUP has tried to step to a closest approach and fails after 10 tries, it tries to find an intersection. In that case, it uses (6.80) to give a first estimate for t_c, then uses (6.85) for iterating. Statement 30 in TRACE uses (6.83) to estimate the time of closest approach. See Section 7.3 for flow charts of these functions.

6.5.3 Reflecting the Ray From the Terrain Surface

Once the intersection with the terrain has been found, the ray must be properly reflected. For an isotropic medium, this is straightforward. The algorithm must first project the wave vector into two components parallel to the surface and the component perpendicular to the surface. It then changes sign on the component perpendicular to the surface.

An anisotropic medium (such as an atmosphere with wind) is more difficult. The two components parallel to the surface remain unchanged, as before, but the component perpendicular to the surface must be changed so that the dispersion relation is satisfied. Although this principle is the same for all media, the solution depends on the dispersion relation. At this point, we must specialize to the particular medium of interest, namely, acoustic waves in the presence of winds.

We need to first separate the wave vector $\vec{k}$ into components perpendicular and parallel to the surface. Let $\vec{n}$ be a unit vector pointing out of the surface. Then the component of $\vec{k}$ normal to the surface is

$$k_\perp = \vec{k} \cdot \vec{n} \tag{6.86}$$

$$\vec{k}_\perp = (\vec{k} \cdot \vec{n}) \, \vec{n} \tag{6.87}$$

and the part parallel to the surface is

$$\vec{k}_\parallel = \vec{k} - (\vec{k} \cdot \vec{n}) \, \vec{n} \ . \tag{6.88}$$

The dispersion relation for acoustic waves in the presence of winds is

$$- \Omega^2 + C^2 \, k^2 = 0 \ , \tag{6.89}$$

where $$\Omega \equiv \omega - \vec{k}\cdot\vec{v} \tag{6.90}$$

is the intrinsic frequency, ω is the wave frequency, and $\vec{v}$ is the wind velocity.

With the help of (6.90), we can separate (6.89) as follows:

$$-(\omega - \vec{k}_\perp\cdot\vec{v} - \vec{k}_\parallel\cdot\vec{v})^2 + c^2\, k_\perp^{\,2} + c^2\, k_\parallel^{\,2} = 0 \ . \tag{6.91}$$

We want to solve (6.91) for $k_\perp$, assuming $\vec{k}_\parallel$ to be known. We can rewrite (6.89) as

$$(c^2 - v_\perp^{\,2})\, k_\perp^{\,2} + 2\, \Omega_\parallel\, v_\perp\, k_\perp - \Omega_\parallel^{\,2} + c^2\, k_\parallel^{\,2} = 0 \ , \tag{6.92}$$

where

$$\Omega_\parallel \equiv \omega - \vec{k}_\parallel\cdot\vec{v} \tag{6.93}$$

and

$$\vec{v}_\perp \equiv (\vec{v}\cdot\vec{n})\ \vec{n} \quad , \quad v_\perp \equiv \vec{v}\cdot\vec{n} \tag{6.94}$$

is the component of wind normal to the surface. The quadratic formula gives the solution to (6.92) as

$$k_\perp = \frac{-\,\Omega_\parallel\, v_\perp \pm c\sqrt{\Omega_\parallel^{\,2} - k_\parallel^{\,2}\,(c^2 - v_\perp^{\,2})}}{c^2 - v_\perp^{\,2}} \ . \tag{6.95}$$

One solution of (6.95) should be the normal component of $\vec{k}$ for the incident wave, the other the normal component of $\vec{k}$ for the reflected wave. To convert from the incident wave to the reflected wave, it is necessary simply to change the sign of

$$k_\perp + \frac{\Omega_\parallel\, v_\perp}{c^2 - v_\perp^{\,2}} \ . \tag{6.96}$$

To do this, we can use (6.87) and (6.88) to write

$$\vec{k} = \vec{k} - (\vec{k}\cdot\vec{n})\vec{n} + (\vec{k}\cdot\vec{n})\vec{n} \ . \tag{6.97}$$

This is equivalent to

$$\vec{k} = \vec{k} - (\vec{k}\cdot\vec{n})\vec{n} - \frac{\Omega_{\parallel}\ v_{\perp}}{c^2 - v_{\perp}^2}\ \vec{n} + (\vec{k}\cdot\vec{n})\vec{n} + \frac{\Omega_{\parallel}\ v_{\perp}\ \vec{n}}{c^2 - v_{\perp}^2}\ . \qquad (6.98)$$

Let us assume that (6.98) applies to the incident wave, that is,

$$\vec{k}_{inc} = \vec{k}_{inc} - (\vec{k}_{inc}\cdot\vec{n})\vec{n} - \frac{\Omega_{\parallel}\ v_{\perp}}{c^2 - v_{\perp}^2}\ \vec{n} + (\vec{k}_{inc}\cdot\vec{n})\vec{n} + \frac{\Omega_{\parallel}\ v_{\perp}\ \vec{n}}{c^2 - v_{\perp}^2}\ , \qquad (6.99)$$

where the subscript inc signifies the incident wave. To get the wave vector for the reflected wave, we need only reverse the sign of the last two terms in (6.99), that is,

$$\vec{k}_{ref} = \vec{k}_{inc} - 2(\vec{k}_{inc}\cdot\vec{n})\vec{n} - 2\ \frac{\Omega_{\parallel}\ v_{\perp}\ \vec{n}}{c^2 - v_{\perp}^2}\ , \qquad (6.100)$$

where the subscript ref signifies the reflected wave.

To be more explicit, we can write (6.100) in terms of components. We assume an earth-centered spherical polar-coordinate system (r, θ, ϕ). We then consider Cartesian components of $\vec{k}$, (k_r, k_θ, k_ϕ) in the r, θ, and ϕ directions. We also consider components of n, (n_r, n_θ, n_ϕ) in the r, θ, and ϕ directions. Then (6.100) is equivalent to

$$k_{r\ ref} = k_{r\ inc} - 2(\vec{k}_{inc}\cdot\vec{n})n_r - \frac{2\Omega_{\parallel}\ v_{\perp}\ n_r}{c^2 - v_{\perp}^2} \qquad (6.101a)$$

$$k_{\theta\ ref} = k_{\theta\ inc} - 2(\vec{k}_{inc}\cdot\vec{n})n_\theta - \frac{2\Omega_{\parallel}\ v_{\perp}}{c^2 - v_{\perp}^2}\ n_\theta \qquad (6.101b)$$

$$k_{\phi\ ref} = k_{\phi\ inc} - 2(\vec{k}_{inc}\cdot\vec{n})n_\phi - \frac{2\Omega_{\parallel}\ v_{\perp}}{c^2 - v_{\perp}^2}\ n_\phi\ . \qquad (6.101c)$$

One might wonder whether it is realistic to allow a component of the wind normal (or even parallel) to the surface of the terrain. Rather than debate this point here, we simply point out that HARPA does not check wind models to see if they satisfy continuity or physical boundary conditions (in fact, many of our models don't). This is the responsibility of those who design wind models, if such conditions are important in their applications. The last term in (6.101) guarantees that the dispersion relation will be satisfied for the reflected wave, even if the wind does not vanish at the surface.

6.5.4 Unit-Normal Directions From the Surface

The previous section requires unit normal directions to the surface. Because f is a constant along the surface, the gradient of f is in the same direction as the unit normal. That is

$$\vec{\nabla} f = f_r \hat{i}_r + \frac{f_\theta}{r} \hat{i}_\theta + \frac{f_\phi}{r \sin\theta} \hat{i}_\theta \propto \vec{n} \; . \tag{6.102}$$

Taking the ratio of components gives

$$n_\theta = \frac{f_\theta}{r f_r} n_r \tag{6.103}$$

and

$$n_\phi = \frac{f_\phi}{f_r r \sin\theta} n_r \; . \tag{6.104}$$

The solution of (6.103), and (6.104), while requiring $n_r^2 + n_\theta^2 + n_\phi^2 = 1$, is

$$n_r = \frac{f_r}{\sqrt{f_r^{\,2} + \dfrac{f_\theta^{\,2}}{r^2} + \dfrac{f_\phi^{\,2}}{r^2 \sin^2\theta}}} \; , \tag{6.105}$$

$$n_\theta = \frac{f_\theta}{r\sqrt{f_r^{\,2} + \dfrac{f_\theta^{\,2}}{r^2} + \dfrac{f_\phi^{\,2}}{r^2 \sin^2\theta}}} \; , \tag{6.106}$$

$$n_\phi = \frac{f_\phi}{r \sin\theta \sqrt{f_r^{\,2} + \dfrac{f_\theta^{\,2}}{r^2} + \dfrac{f_\phi^{\,2}}{r^2 \sin^2\theta}}} \; . \tag{6.107}$$

6.6 Coordinate Systems

HARPA uses two different earth-centered spherical polar-coordinate systems, one geographic and one computational, because it is easier to express some models in a different coordinate system. Input data for the coordinates of the transmitter, W(4) and W(5), and input data for the coordinates of the north pole of the computational coordinate system, W(24) and W(25), are entered in geographic coordinates. Putting W(25) equal to 0° and W(24) equal to 90° would superimpose the two north poles and equate the two coordinate systems.

When the two coordinate systems do not coincide, the atmospheric models calculate wind, sound speed, temperature, and molecular weight in terms of the computational coordinate system. Dudziak (1961) describes the transformations between these coordinate systems.

7. Structure of the Program

This chapter explains how the parts of HARPA work together, including a brief description of each program subroutine and detailed flow charts of the central ray-tracing parts. Also included are hierarchical or organization diagrams that show the calling sequences of the principal ray-tracing operations and details of the common-block structure and usage.

7.1 Description of the Subroutines

Following is a list of all the programs, subroutines, and functions that constitute HARPA, that is, the functions inside the dashed lines of Figure 1.1. They are listed in the order they appear in Files 3, 4, and 5 of the distribution tape and in the source-code listings of Appendix D. The routines are divided into a RAY-TRACING "CORE," the set of programs that is always required to do ray tracing, DISPERSION-RELATION ROUTINES, from which you must select one, and ATMOSPHERIC MODEL SUBROUTINES, from which you select the routines that correspond to the atmospheric models you want to use. A more detailed description of the distribution-tape structure is given in Sec. 3.3.1.

7.1.1 Ray-Tracing Core

PROGRAM RAYTRC -- The main program; sets the initial conditions for each ray and calls TRACE.

SUBROUTINE DFSYS -- Contains system-dependent functions such as date, time (user must modify).

SUBROUTINE DFCNST -- Defines machine-dependent constants (user must modify).

SUBROUTINE READW1 -- Reads variable-length tabular data from input data table.

SUBROUTINE GTUNIT -- Interprets units line in tabular input data.

SUBROUTINE SREAD1 -- Handles unassigned labeled-common blocks.

FUNCTION READW -- Reads input data table into W array, converting units.

SUBROUTINE CLEAR -- Sets n elements of an array to zero.

FUNCTION ND2B -- Converts decimal digits to positionally equivalent binary numbers when reading W(29).

FUNCTION UCON -- Provides keyword units conversion for input data.

SUBROUTINE TRACE -- Calculates a raypath for the requested number of hops.

FUNCTION PCROSS -- Tests whether ray crosses a surface.

SUBROUTINE RCROSS -- Estimates point where ray crosses a surface.

SUBROUTINE HAMLTN -- Calculates Hamilton's equations for ray tracing and other quantities to be integrated.

SUBROUTINE RKAM -- Keeps track of integration steps performed by RKAM1 and makes them available to calling subroutines.

SUBROUTINE RKAM1 -- Numerically integrates Hamilton's equations.

SUBROUTINE BACKUP -- Moves the ray point to the last intersection with receiver or terrain surface.

FUNCTION REFLECT -- Computes normal and parallel components of K vector at ground reflections.

SUBROUTINE FIT -- Computes 3 types of parabolic fit to raypath relative to terrain.

FUNCTION GET -- Gets the value of the terrain function and its derivatives; calls the terrain model subroutine if necessary.

FUNCTION GET1 -- Second version of GET to avoid self-calls.

FUNCTION ITEST -- Passes integer values through for variables typed real.

SUBROUTINE CONBLK -- Data-initialization and file-opening service routine.

FUNCTION WCHANGE -- Determines equivalence of two W arrays for producing raysets.

FUNCTION RENORM -- Normalizes a vector to a specified magnitude.

SUBROUTINE SET2 -- Sets n components of vector to a specified single value.

SUBROUTINE PRINTR -- Prints details of the raypath calculation at specified intervals and produces computer-readable output (raysets).

SUBROUTINE ATMOSHD -- Includes page headings in printout.

SUBROUTINE PUTDES -- Prints model information on printout header.

FUNCTION NUMSTG -- Converts a numeric value to a string.

SUBROUTINE SFILL -- Fills a string with n specified characters.

FUNCTION STRIM -- Determines position of last nonblank character of a string.

FUNCTION RERR -- Computes for subroutine PRINTR the largest relative integration error.

SUBROUTINE RERROR -- Reports error conditions and stops program.

SUBROUTINE STOPIT -- Prints error condition and stops program.

SUBROUTINE PUTKST -- Multiple ENTRY points to produce line-printer output while accounting for line count, new page, etc.

SUBROUTINE OPNREP -- Increases portability among FORTRAN 77 systems for opening files with replacement.

SUBROUTINE OVERRD -- Tests for "zero-override" condition in input data (Sec. 5.3.1).

SUBROUTINE SFILTR -- Filters extraneous characters from plot annotations.

FUNCTION ALCOSH -- Compute log[cosh(x)] and use large-argument approx.

SUBROUTINE GAUSEL -- Calculates coefficients of functions to fit points in TTABLE.

[PLOTTING ROUTINES]

SUBROUTINE RAYPLT -- Main plotting program; initializes, reads input, plots projections of rays on a vertical or horizontal plane.

SUBROUTINE PLOT -- XY plotting routine, called by RAYPLT.

SUBROUTINE LABPLT -- Labels rayplots.

[TICK/ANNOTATION ROUTINES]

SUBROUTINE PLTHLB -- Plots horizontal ticks and annotation for rayplot.

SUBROUTINE PLTANH -- Generic horizontal tick annotation.

SUBROUTINE SETXY -- Plot initialization; sets projection parameters.

SUBROUTINE TIKLINE -- Draws straight line with ticks at intervals.

SUBROUTINE PLTANOT -- Puts general annotations on plots.

SUBROUTINE DRAWTKS -- Draws plot boundary, ticks, and labels for horizontal ray projection.

SUBROUTINE PLTLB -- Puts vertical tick annotations on rayplots.

SUBROUTINE ARCTIC -- Draws curved range axis for rayplot.

BLOCK DATA PLOTBL -- Initializes plot range variables.

[GRAPHICS WRITE ROUTINES]

SUBROUTINE DDINIT -- Initializes plotting process (writes header line to TAPE5).

SUBROUTINE DDBP -- Sets a vector origin (writes IX,IY to TAPE5).

SUBROUTINE DDVC -- Plots a vector (writes IX,IY for vector end point to TAPE5).

SUBROUTINE DDTEXT -- Writes an array (character string) to TAPE 5 in tabular text mode.

SUBROUTINE DDTAB -- Sends instruction to TAPE5 that initializes tabular (text) plotting.

SUBROUTINE DDFR -- Sends instruction to TAPE5 to advance a microfilm frame.

SUBROUTINE DDEND -- Empties plot buffer and releases plotting command file to microfilm plot queue.

SUBROUTINE DASH -- Sets dashed-line mode; that is, all plotted curves will be dashed instead of solid after a call to subroutine DASH.

SUBROUTINE RESET('DASH') -- Sets solid-line mode; that is, all plotted curves will be solid lines after this call.

SUBROUTINE HEIGHT -- If you do not have the DISSPLA* plotting package, load SUBROUTINE SMPANN instead of SUBROUTINE FULANN, and you can ignore this routine.

SUBROUTINE MX1ALF -- If you do not have the DISSPLA* plotting package, load SUBROUTINE SMPANN instad of SUBROUTINE FULANN, and you can ignore this routine.

SUBROUTINE MX2ALF -- If you do not have the DISSPLA* plotting package, load SUBROUTINE SMPANN instead of SUBROUTINE FULANN, and you can ignore this routine.

SUBROUTINE SCMPLX -- If you do not have the DISSPLA* plotting package, load SUBROUTINE SMPANN instead of SUBROUTINE FULANN, and you can ignore this routine.

7.1.2 Dispersion Relation Routines

SUBROUTINE ANWNL -- Acoustic wave, no winds, no losses.

SUBROUTINE AWWNL -- Acoustic wave, with wind, no losses.

SUBROUTINE ANWWL -- Acoustic wave, no wind, with losses.

SUBROUTINE AWWWL -- Acoustic wave, with wind, with losses.

---

* DISSPLA is the proprietary product of ISSCO, Inc.

7.1.3 Atmospheric Model Subroutines

SUBROUTINE WLINEAR -- Constant upward or northward background wind; linear eastward wind profile.

SUBROUTINE WTIDE -- Eastward and northward background wind profiles that are sinusoidal in height and time and are in time quadrature.

SUBROUTINE ULOGZ2 -- Logarithmic atmospheric boundary-layer profile; eastward background wind only.

SUBROUTINE VVORTX3 -- Vertical vortex wind perturbation with viscous core and Gaussian height profile.

SUBROUTINE WGAUSS2 -- Eastward perturbation wind that decays in three dimensions (jet).

SUBROUTINE NPWIND -- Do-nothing wind perturbation model.

SUBROUTINE GAMRTDM -- Sound-speed model: $C^2 = \gamma RT/M$.

SUBROUTINE CSTANH -- Background sound-speed profile with linear segments joined smoothly.

SUBROUTINE NPSPEED -- Do-nothing sound-speed-perturbation model.

SUBROUTINE CBLOB2 -- Sound-speed perturbation with Gaussian decay in three dimensions.

SUBROUTINE TLINEAR -- Background linear temperature profile.

SUBROUTINE TTANH5 -- Background height profile of temperature with linear segments joined smoothly.

SUBROUTINE TTABLE -- Background tabular temperature profiles with cubic interpolation between points.

SUBROUTINE NTEMP -- Do-nothing background temperature model.

SUBROUTINE TBLOB2 -- Temperature perturbation with Gaussian decay in three dimensions.

SUBROUTINE NPTEMP -- Do-nothing temperature perturbation model.

SUBROUTINE MCONST -- Constant molecular weight model.

[TERRAIN MODELS]

SUBROUTINE GHORIZ -- Terrain model at fixed height above sea level.

SUBROUTINE GLORENZ -- Lorentzian-ridge terrain model.

SUBROUTINE GTANH -- 2-D terrain model with a series of linear segments joined smoothly.

[ABSORPTION MODELS]

SUBROUTINE NPTERR -- Do-nothing terrain-perturbation model.

SUBROUTINE MUARDC -- ARDC background absorption formula.

SUBROUTINE NPABSR -- Do-nothing absorption-perturbation model.

SUBROUTINE PEXP -- Exponential background pressure profile.

SUBROUTINE NPPRES -- Do-nothing pressure-perturbation model.

[RECEIVER-SURFACE MODELS]

SUBROUTINE RHORIZ -- Horizontal receiver-surface model.

SUBROUTINE RTERR -- Receiver-surface model at fixed height above terrain.

SUBROUTINE RVERT -- Vertical (conical) receiver surface at a fixed radius from a specified origin.

[ANNOTATION MODELS]

SUBROUTINE SMPANN -- Initializes plot in draft mode (must be used if you don't have DISSPLA).

SUBROUTINE FULANN -- Initializes plot in publication-quality mode (requires DISSPLA).

7.2 HARPA Organization Diagrams

This section contains hierarchical diagrams, Figures 7.1 through 7.4, that show how the principal subroutines are interrelated by calling sequences. These diagrams are not flow charts; they show how control passes among the program modules. Not all subroutines are shown, only the major ones that perform the ray-tracing function.

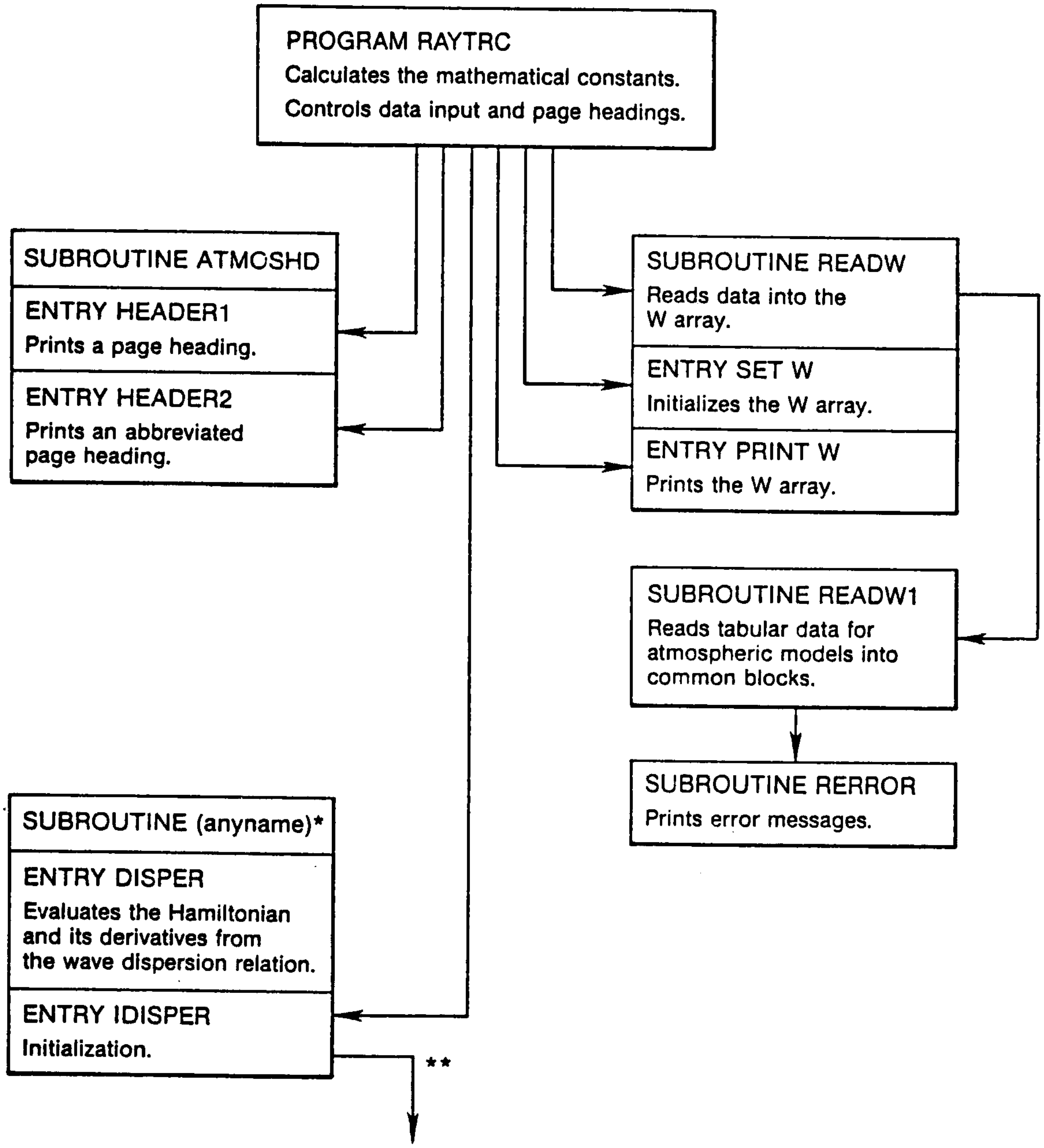


Figure 7.1. Block diagram (not a flow chart) of the ray-tracing program showing the relation (hierarchy, what calls what) of the main program to other subroutines during the initialization stage (immediately after new input data are read in).

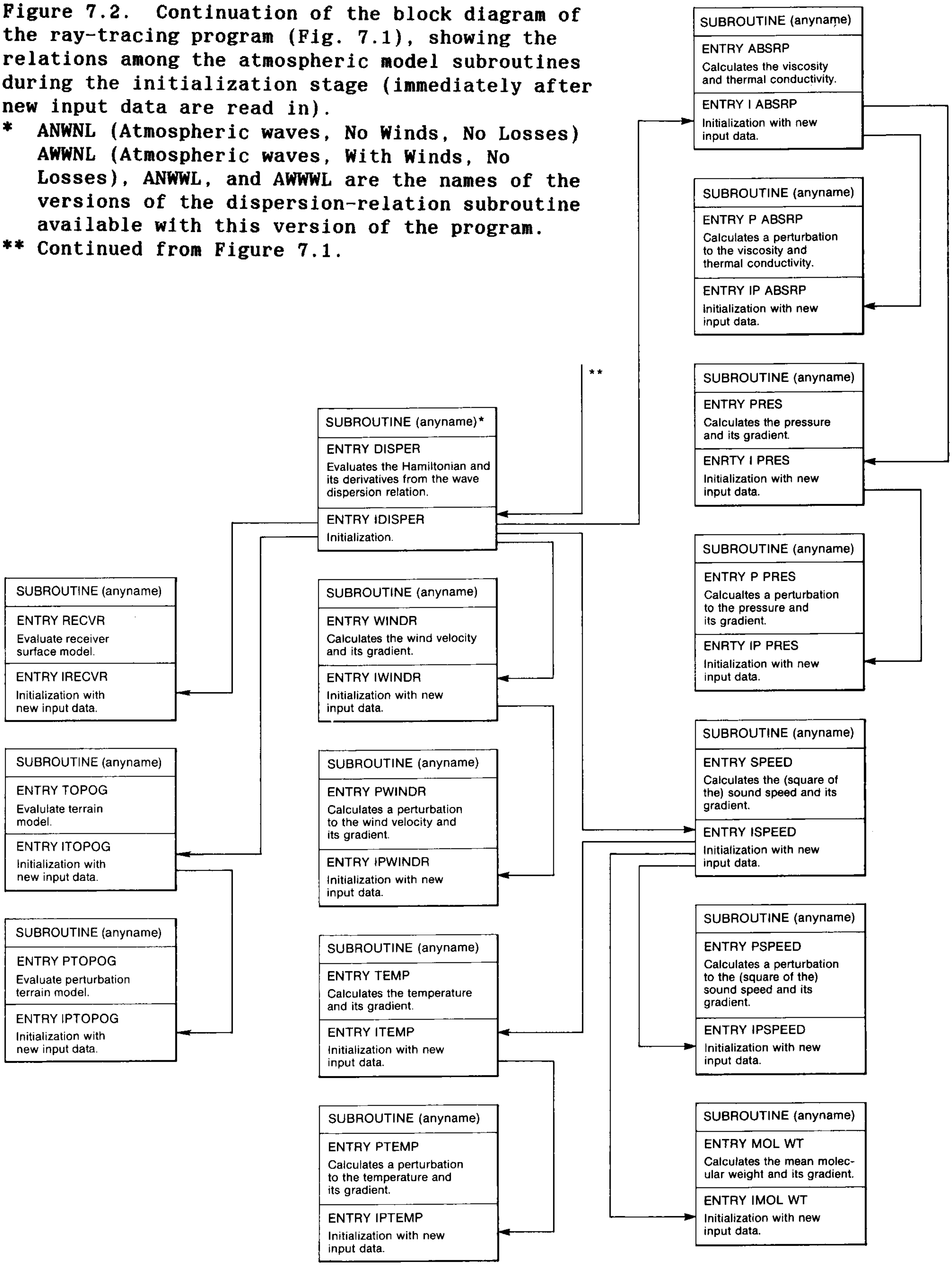
* ANWNL (Acoustic, No Winds, No Losses), AWWNL, With Winds, No Losses), ANWWL, and AWWWL are the names of the versions of the dispersion-relation subroutine available.

** Figure 7.2 shows the continuation of this block diagram.

Figure 7.2. Continuation of the block diagram of the ray-tracing program (Fig. 7.1), showing the relations among the atmospheric model subroutines during the initialization stage (immediately after new input data are read in).

* ANWNL (Atmospheric waves, No Winds, No Losses) AWWNL (Atmospheric waves, With Winds, No Losses), ANWWL, and AWWWL are the names of the versions of the dispersion-relation subroutine available with this version of the program.

** Continued from Figure 7.1.



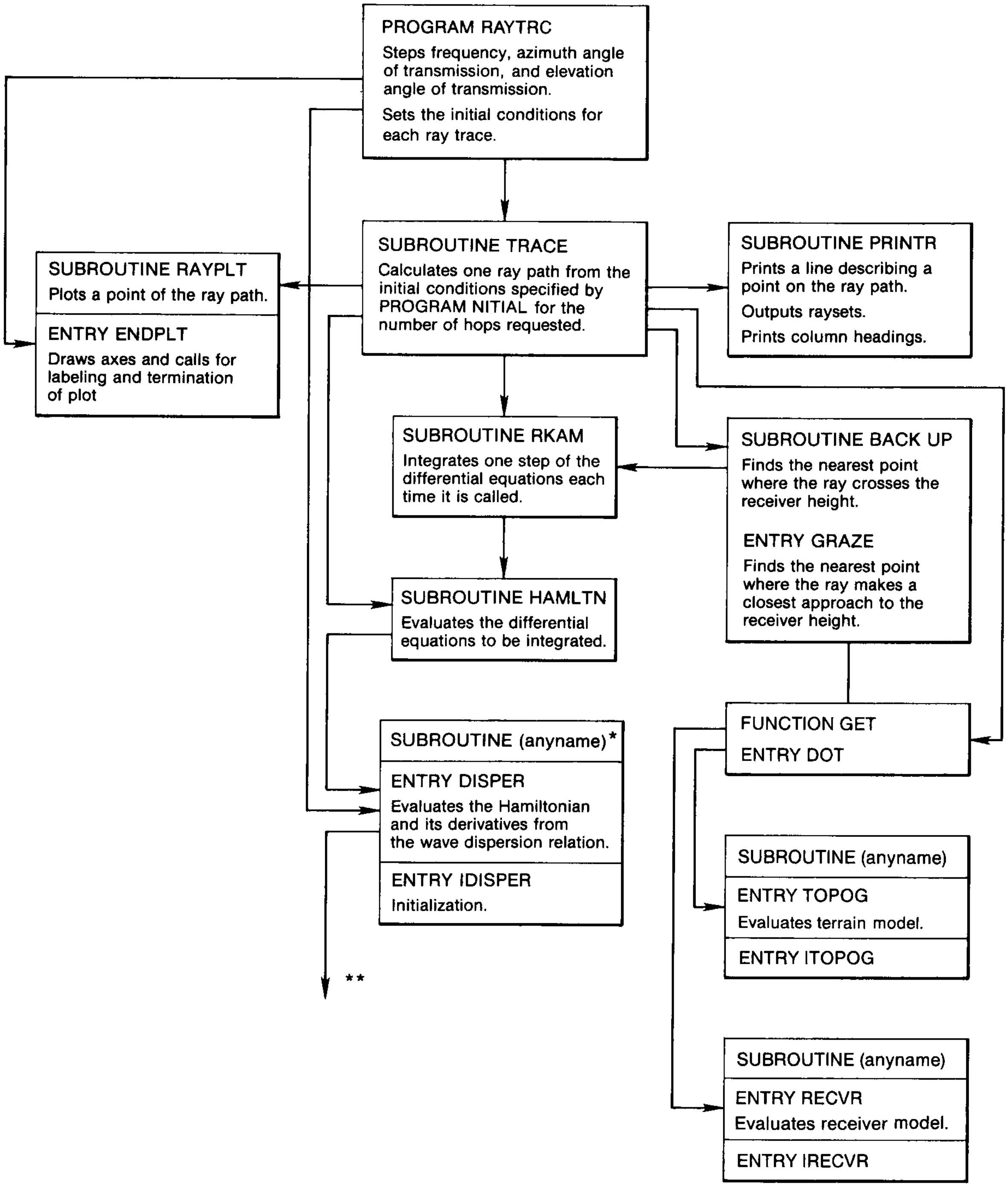


Figure 7.3. Block diagram (not a flow chart) of the ray-tracing program showing the relations (hierarchy, what calls what) among the main program and the subroutines during the raypath calculation stage (after the initialization stage).

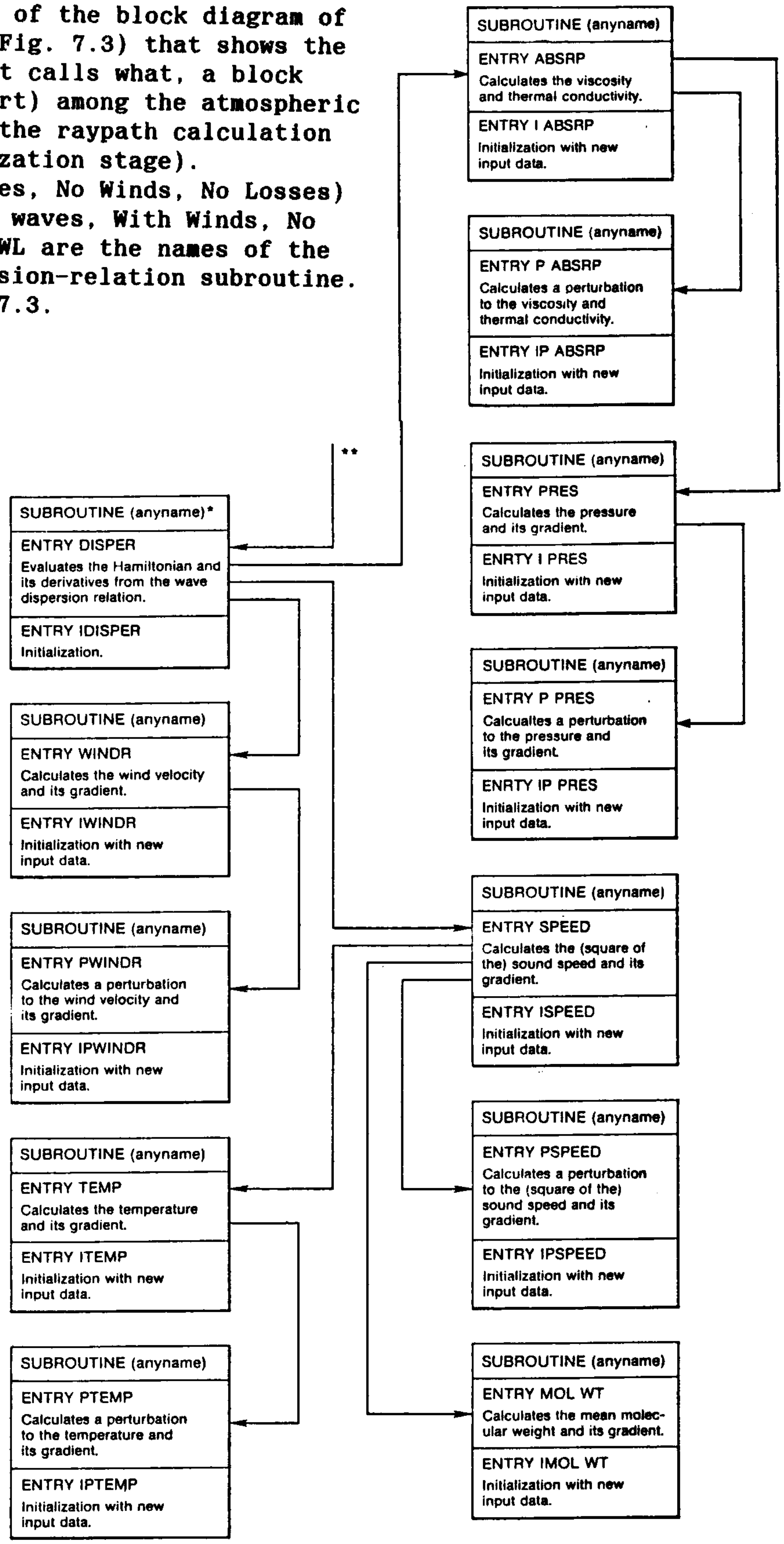
* ANWWL (Acoustic, No Winds, No Losses) and AWWNL (Acoustic, With Winds, No Losses), AWWWL and ANWWL are the names of the versions of the dispersion-relation subroutine.

** Figure 7.4 shows the continuation of the block diagram that shows the relations among the atmospheric model subroutines.

Figure 7.4. Continuation of the block diagram of the ray-tracing program (Fig. 7.3) that shows the relations (hierarchy, what calls what, a block diagram is not a flow chart) among the atmospheric model subroutines during the raypath calculation stage (after the initialization stage).

* ANWNL (Atmospheric waves, No Winds, No Losses) and AWWNL (Atmospheric waves, With Winds, No Losses), AWWWL and ANWWL are the names of the versions of the dispersion-relation subroutine.

** Continued from Figure 7.3.



7.3 Flow Charts for Program RAYTRC and Subroutines TRACE and BACKUP

These three routines contain the central logic of the raypath calculations and so are described in detail in flow-chart form.

This ray-tracing program consists of various subroutines that perform specific tasks in calculating raypaths. The division of labor makes it easier to modify the program to solve specific problems. Often it may be necessary to change only one or two subroutines to convert the program to a different use.

The main program (RAYTRC) sets up the initial conditions (transmitter location, wave frequency, and direction of transmission) for each ray trace. In setting up the initial conditions for each ray trace, the main program (RAYTRC) steps frequency, azimuth angle of transmission, and elevation angle of transmission (see Figs. 7.5 and 7.6). Then subroutine TRACE calculates one raypath for the requested number of crossings of the specified receiver height. Subroutine TRACE is the heart of the ray-tracing program. It is the most complicated subroutine included, but also the most important to understand. The flow charts in Figures 7.6 and 7.7 explain how TRACE works.

Subroutine RKAM integrates the differential equations numerically using an Adams-Moulton predictor-corrector method with a Runge-Kutta starter. Subroutine HAMLTN evaluates the differential equations to be integrated. Subroutine DISPER calculates the Hamiltonian and its derivatives, the wave number from the dispersion relation, and the wave polarization. (Four versions of subroutine DISPER are included.) Subroutines WIND, SPEED, TEMP, MOLWT, RECUR, TOPOG, ABSRP, and PRES calculate the atmospheric wind speed, sound speed, temperature, mean molecular weight, receiver surface, terrain, viscosity/thermal conductivity, and pressure. Several versions of these eight subroutines are included, and it is easy to add more. Subroutine BACKUP finds an intersection of the ray with the receiver surface or with the terrain. The flow charts in Figures 7.8 through 7.10 and Section 6.5 explain how BACKUP works.

Subroutine PRINTR prints information describing the raypath and outputs the results in computer-readable form (raysets). Subroutine RAYPLT plots the raypath. The block diagrams in Figures 7.1 through 7.4 show the relationships among these (and other) subroutines.

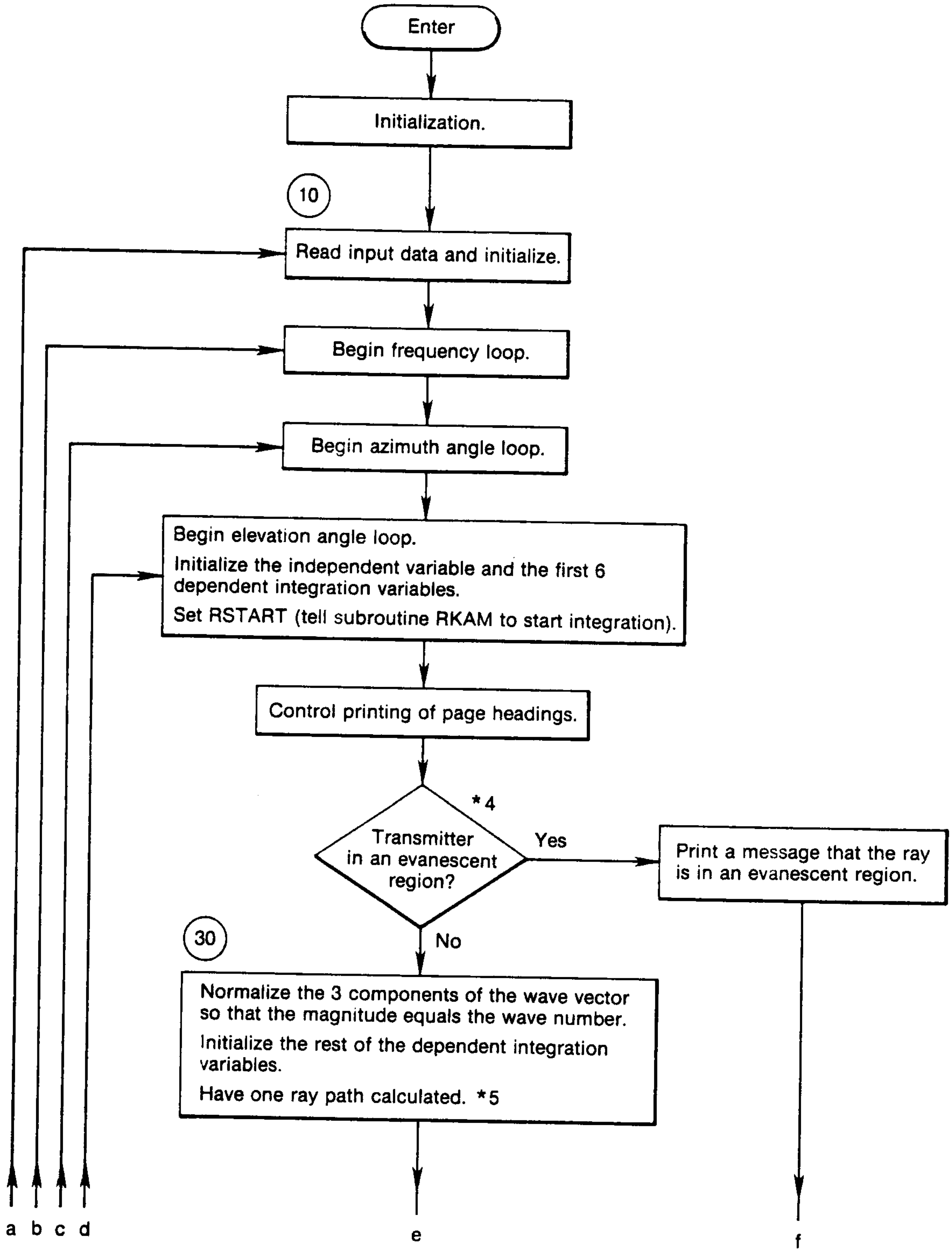


Figure 7.5. Flow chart for program RAYTRC. Circled block numbers correspond to program statement numbers.

*4 There are no evanescent regions for pure acoustic waves (with no cutoff frequency).

*5 Subroutine TRACE calculates one raypath.

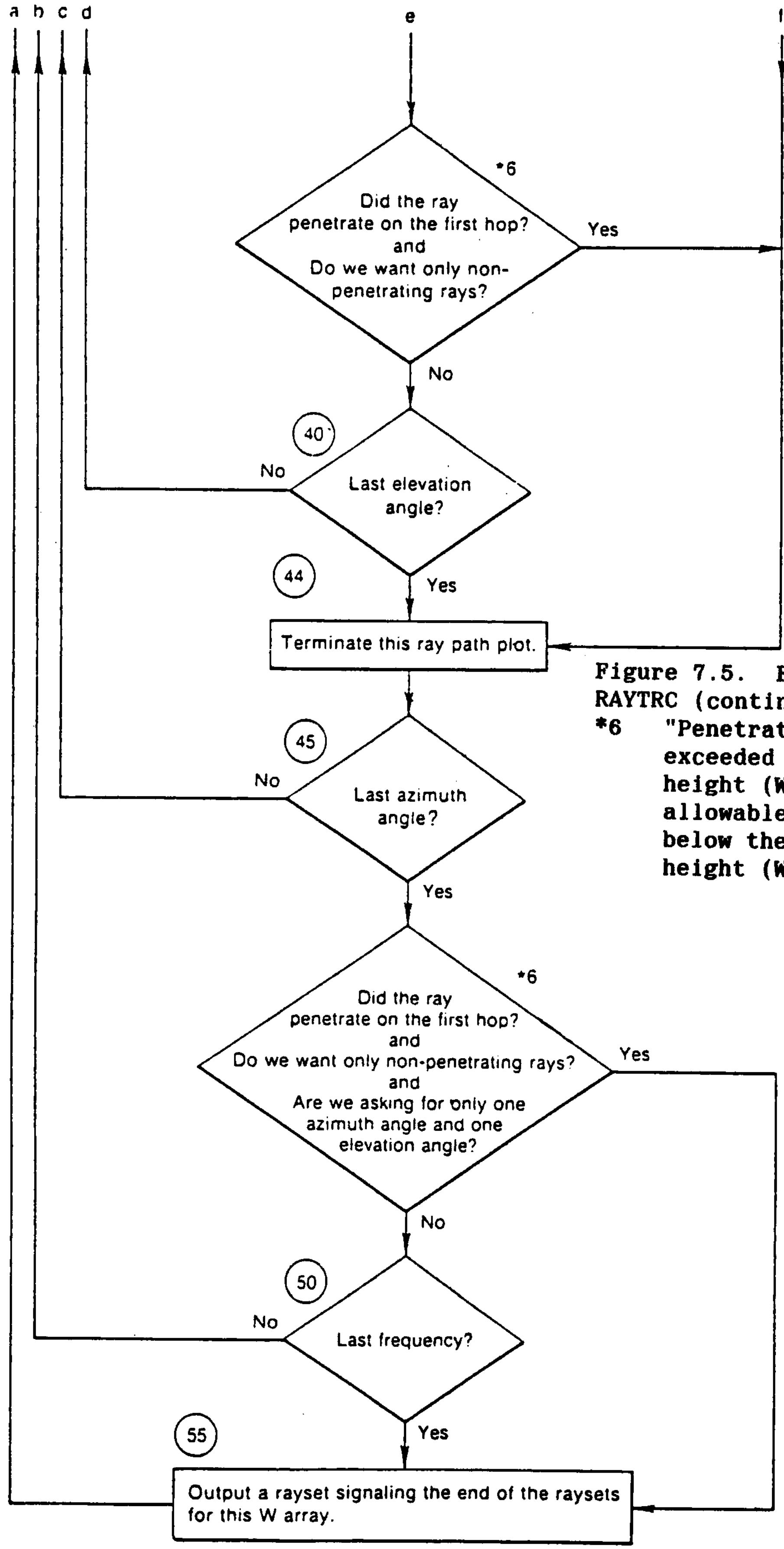


Figure 7.5. Flow chart for program RAYTRC (continued).

***6 "Penetrate" means the ray exceeded the maximum allowable height (W26) or the maximum allowable range (W28) or went below the minimum allowable height (W27).**

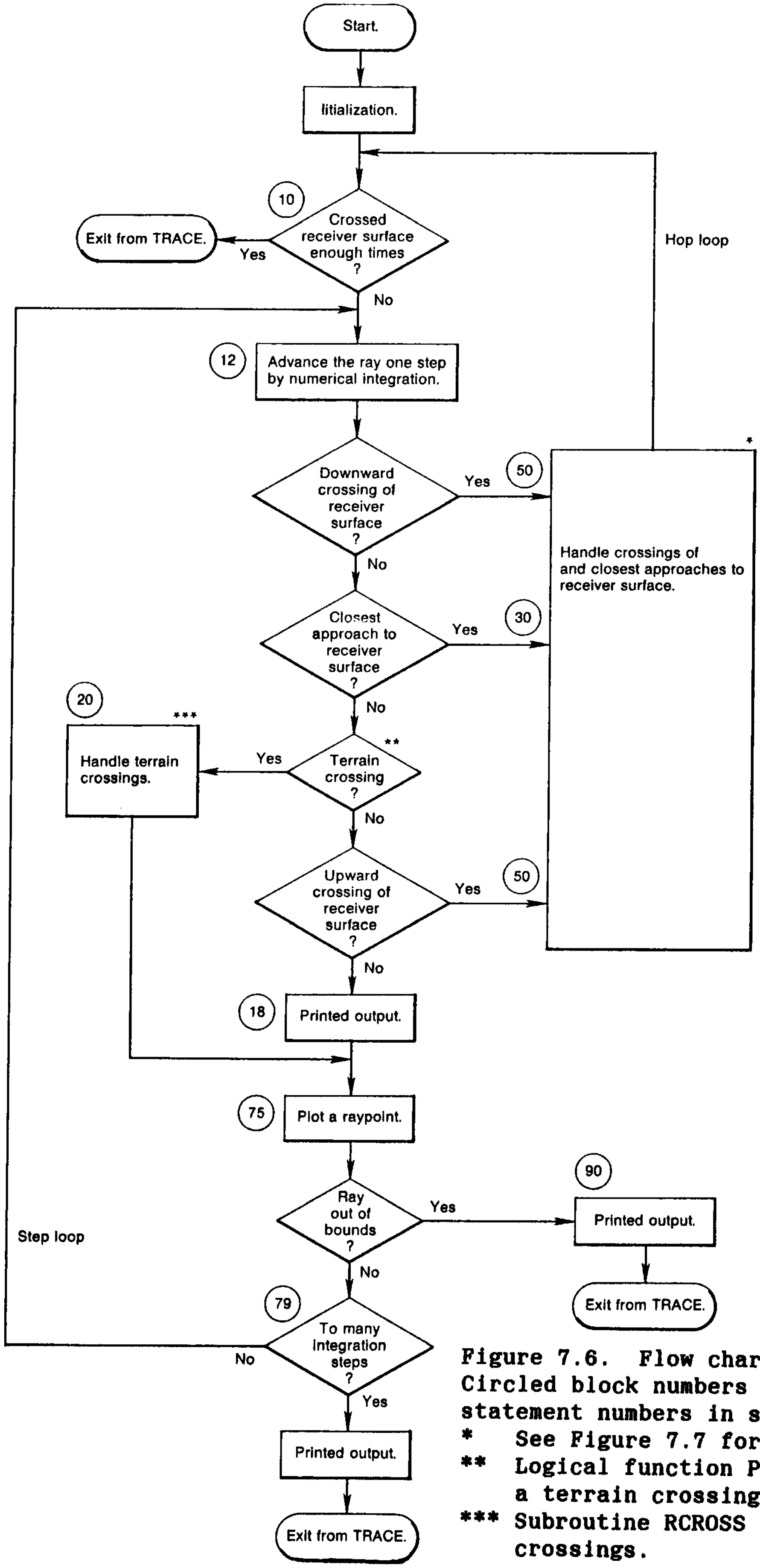


Figure 7.6. Flow chart for subroutine TRACE. Circled block numbers correspond to program statement numbers in subroutine TRACE.

* See Figure 7.7 for details.

** Logical function PCROSS estimates whether a terrain crossing has occurred.

*** Subroutine RCROSS handles terrain crossings.

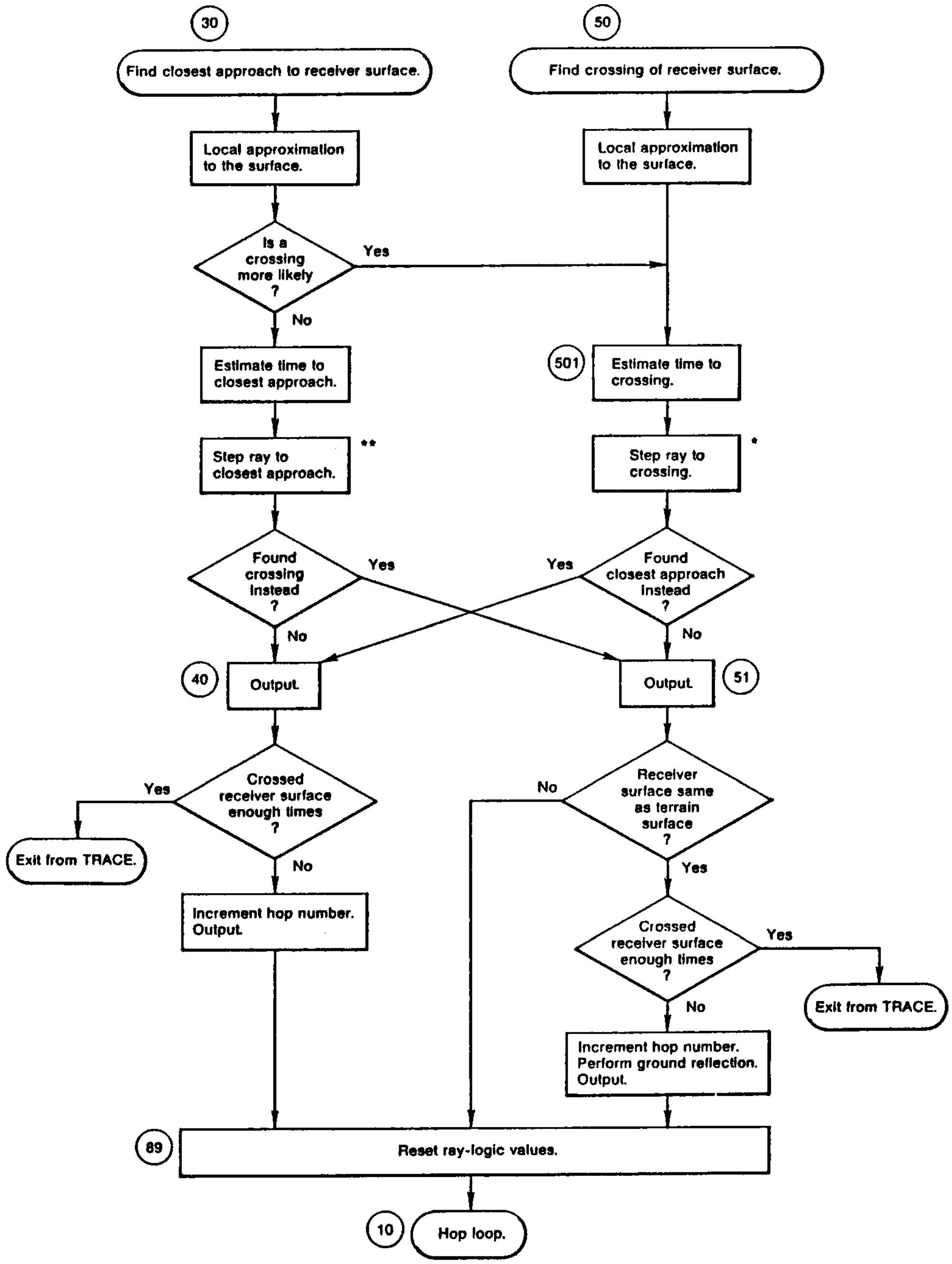


Figure 7.7. Flow chart showing some details of finding crossings of and closest approaches to receiver surface. Circled block numbers correspond to program statement numbers in subroutine TRACE.

* See Figure 7.8 for details (subroutine BACKUP).

** See Figure 7.8 for details (entry point GRAZE in subroutine BACKUP).

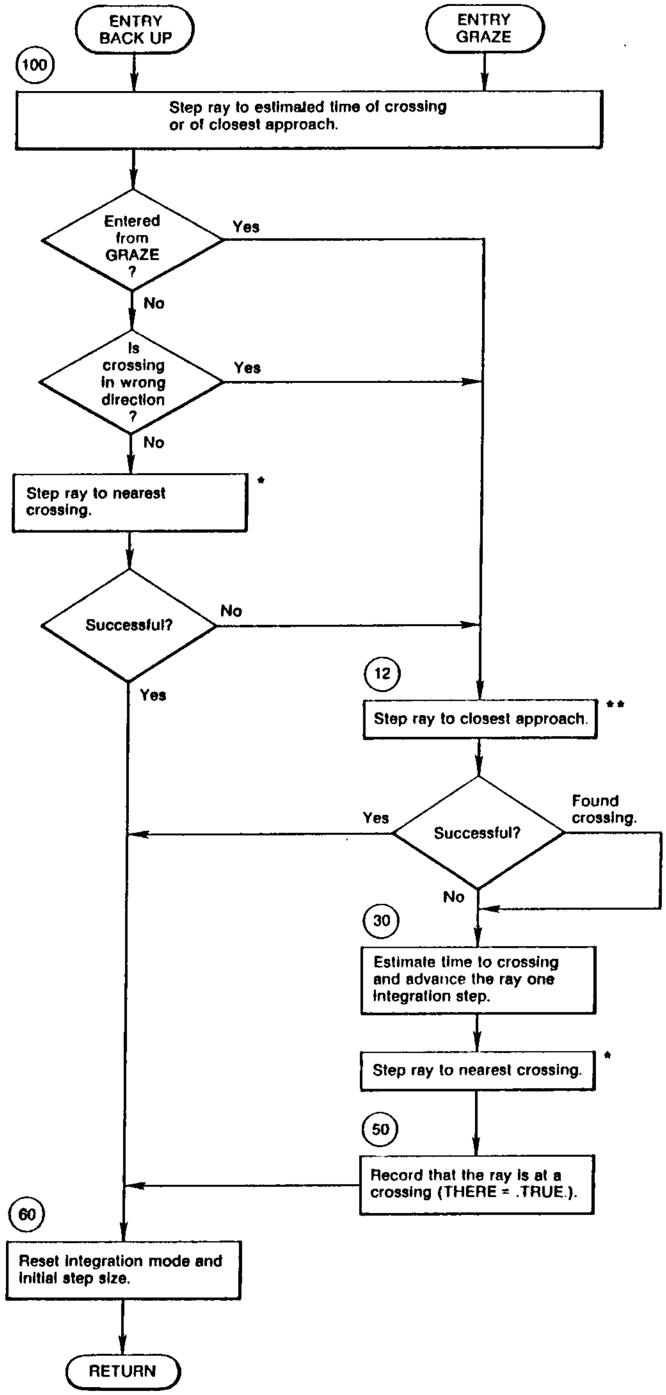


Figure 7.8. Flow chart for subroutine BACKUP. Circled block numbers correspond to program statement numbers. Entry BACKUP steps the ray to a crossing with a specified height. Entry GRAZE steps the ray to a point of closest approach to the specified height. The calling routine (subroutine TRACE) specifies the height with which the ray is to intersect, the direction of crossing (up or down), and estimates the time of crossing (group time delay). Asterisks identify supplementary procedures.

* See Figure 7.9 for details of the algorithm that steps the ray by numerical integration to the nearest crossing of a specified height.

** See Figure 7.10 for details of the algorithm that steps the ray by numerical integration to a closest approach to a specified height.

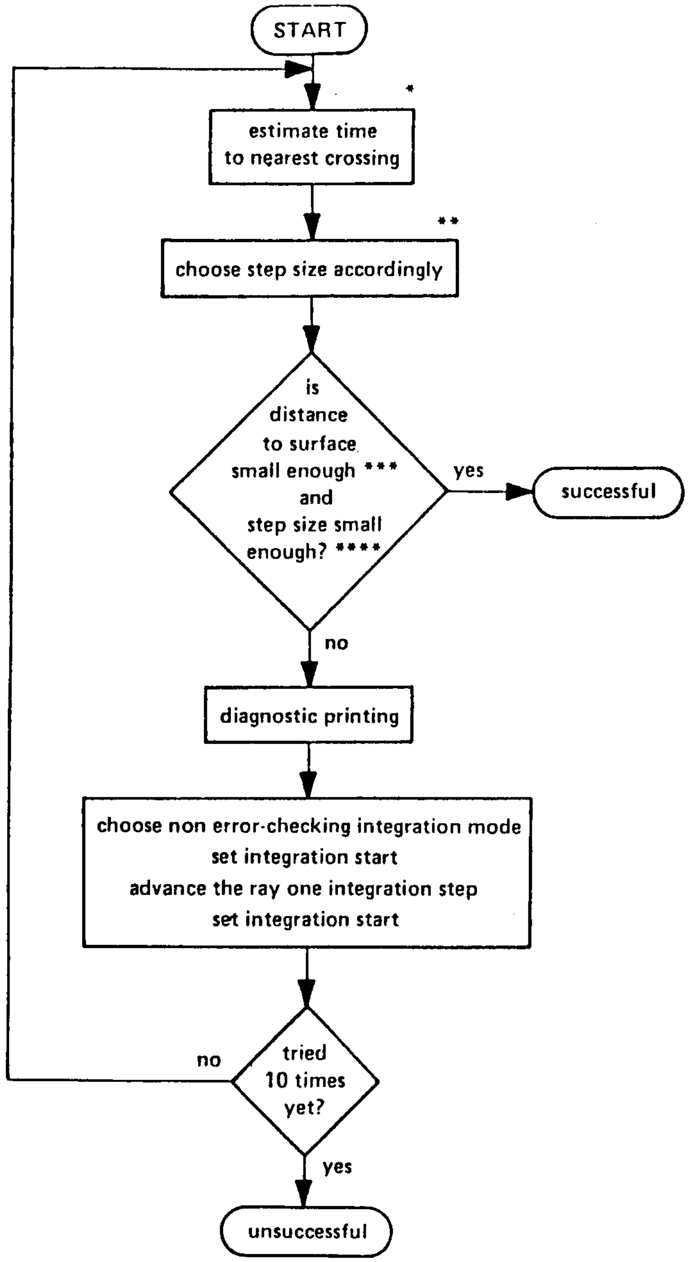


Figure 7.9. Flow chart for the algorithm that steps the ray by numerical integration to a crossing of a specified height. Circled block numbers correspond to program statement numbers.

* See Equation (6.85) to estimate the time of the nearest crossing of the specified height.

** The step size should be no larger than that being used by the numerical integration routine to maintain accuracy in the error-checking mode.

*** 0.5×10^{-4} km.

**** Small enough to ensure the required accuracy and smaller than the smallest allowable step size.

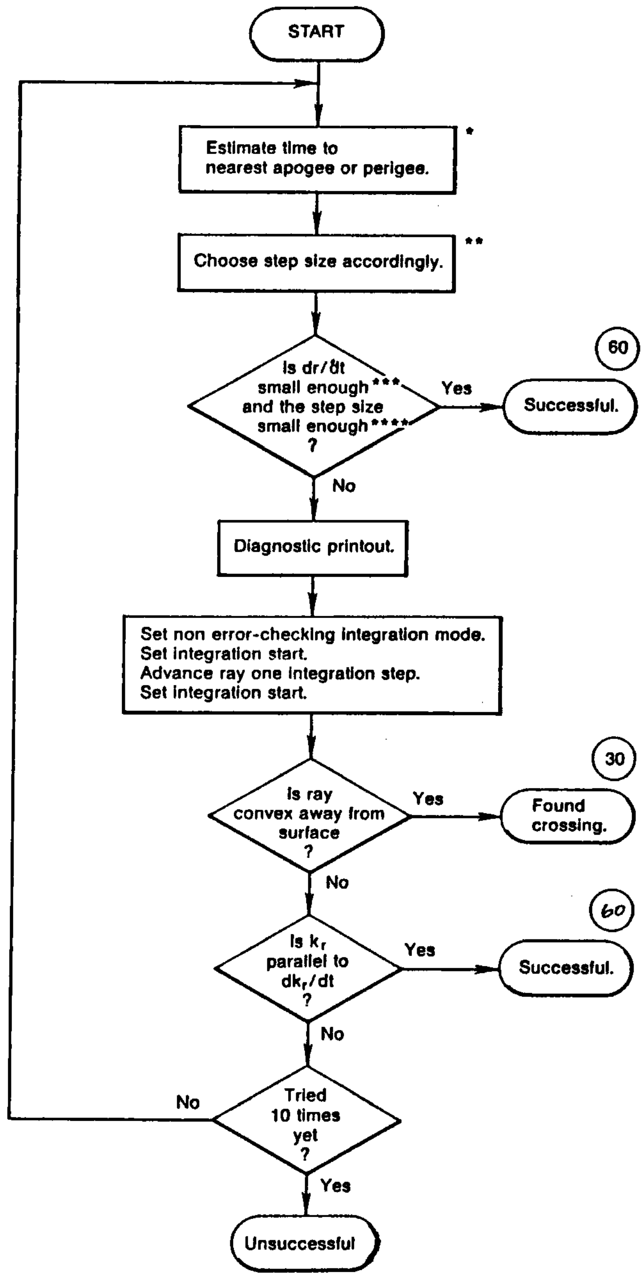


Figure 7.10. Flow chart for the algorithm that steps the ray by numerical integration to a point of closest approach to a specified height. Circled block numbers correspond to program statement numbers.

* See Equation (6.91) to estimate the time of the nearest crossing of the specified height.

** The step size should be no larger than that being used by the numerical integration routine to maintain accuracy in the error-checking mode.

*** 10^{-6} km/km.

**** Small enough to ensure the required accuracy and smaller than the smallest allowable step size.

The listings of most of the subroutines have comments that should help in understanding how they work. In addition, Tables 7.1 through 7.40 define the variables in the common blocks.

Table 7.1--Definitions of the parameters in blank common

Position in common	Variable name	Definition
1-20	R	The dependent variables in the differential equations being integrated--the definitions of the first six are fixed, but the others may be varied by the program user
1	R(1) or R	r
2	R(2) or TH	θ
3	R(3) or PH	ϕ
4	R(4) or KR	k_r
5	R(5) or KTH	k_θ
6	R(6) or KPH	k_ϕ
7-12	R(7)-R(12) or RKVARS(1)-RKVARS(6)	Those variables the user has chosen to integrate, taken in the following order: P -phase path in kilometers A -absorption in decibels Δf -Doppler shift in hertz s -geometrical path length in kilometers
13-20	R(13)-R(20)	Reserved for future expansion
21	TPULSE	Group path length in kilometers (the independent variable in the differential equations)
22	CSTEP	Step length in group path
23-42	DRDT	The derivatives of the dependent variables with respect to the independent variable TPULSE

R and TPULSE are initialized in program RAYTRC and changed in subroutines TRACE, RKAM, and BACK UP.

CSTEP is calculated in subroutine RKAM.

DRDT is calculated in subroutine HAMLTN and used in subroutine RKAM.

7.4 Common-Block Structure and Usage

We use common blocks instead of calling sequences to pass information between subroutines and functions because it is faster. This section describes how those common blocks are organized and which blocks link which program modules, and defines the variables in each block.

Table 7.1 defines the variables in blank common. These are mostly the dependent variables in the numerical integration and their derivatives. Nearly all of the subprograms use this common block.

Table 7.2 describes the common block /MCONST/, which contains mathematical constants. Table 7.3 describes the common block /PCONST/, which contains physical constants.

Many common blocks are used to communicate among the various routines in the program. Table 7.4 lists those common blocks and shows which routines use those common blocks. Blank common, common blocks /MCONST/, /PCONST/, /WW/, and the common blocks listed in Tables 4.2 and 4.3 are not included in Table 7.4. Table 7.5 lists the variable names in these common blocks that are used for input and output by each routine. Tables 7.6 through 7.32 list all of the variables in these common blocks and give the meanings of those variables.

Table 7.17 describes common block /RIN/, which contains parameters output by all of the versions of the dispersion relation subroutines (all of which have the entry point name DISPER). Tables 7.33 through 7.40 describe the common blocks /UU/, /CC/, /TT/, /MM/, /RR/, /GG/, /AA/, and /PP/, which contain the parameters output by the various atmospheric models.

Table 7.2--Definitions of the parameters in common block /MCONST/*

Position in common	Variable name	Definition
1	PI	π
2	PIT2	2π
3	PID2	$\pi/2$
4	DEGS	$180.0/\pi$
5	RAD	$\pi/180.0$
6	ALN10	$\log_e 10$

* These parameters are set in program RAYTRC.

Table 7.3--Definitions of the parameters in common block /PCONST/

Position in common	Variable name	Definition
1	CREF	A reference sound speed (0.344 km/s), the reference sound speed in air at 20°C, Handbook of Chemistry and Physics (1954, page 2312), used to convert group delay time in seconds to an equivalent group path length in kilometers, and to convert a phase time in seconds to an equivalent phase-path length in kilometers
2	RGAS	The gas constant, = 8.31436×10^{-3} kg (kg mole)$^{-1}$ km^2 s^{-2} K^{-1} (this value of the gas constant gives a sound speed in km/s)
3	GAMMA	γ, the ratio of specific heat at constant pressure to that at constant density, = 1.4

Table 7.4.--Common block usage by the core routines

Routine \ Common block	PROCFL	HDR	UCONC	UCONV	RKAMS	CGET	CRMACH	CRKTIME	TRLOCAL	RK	TRAC	FNDER	RIN	ERR	RAYDEV	FLGP	RINPL	FLG	HDRC	FILEC	PLT	RAYCON	ANNCTC	ANNCTL	LABCLT	DD	KNKN	DDLIM
RAYTRC		O								I/O			I	I	I	O		I/O	O	O								
CONBLK	O	O	O	O				O					O		I/O				O			O						
WCHANGE																												
RENORM																												
SET2																												
ITEST																												
PUTDES																												
NUMSTG																												
SFILL																												
STRIM																												
RERR																												
RERROR																												
STOPIT																												
PUTKST																												
OPENREP																												
OVERRD																												
CLEAR																												
ND2B																												
DFSYS																												
DFCNST							O																					
UCON			I	I/O																								
HAMLTN													I															
RKAM					I/O			I/O		I/O																		
RKAM1					O					I/O																		
TRACE								I/O	I/O	I/O	I/O							O										
PCROSS											I																	
RCROSS									O	O	I/O							O										
BACKUP										I/O	I/O																	
REFLECT											I/O	I																
FIT											I/O	I																
GET						O	I	I				O																
GET1						O	I	I																				
READW1															I													
READW															I	I		I										
PRINTR										I			I	O				I/O										
ATMOSHD													I			I	I	I/O	I									
RAYPLT																		I/O		I	O							
PLOT																					I/O					O		
LABPLT													I						I							O		
PLTHLB																												
PLTANH																												
SETXY																						I			I	O		
TIKLINE																									O			
PLTANOT																												
DRAWTKS																							I	I	I	I/O		
PLTLB																												
ARCTIC																									I	I/O		
PLOTBL																												
DDINIT																											O	O
DDBP																												
DDVC																										I	I/O	I/O
DDTEXT																										I	I/O	I/O
DDTAB																										I	I/O	
DDFR																												
DDEND																												
DASH																												
RESET																												
HEIGHT																												
MX1ALF																												
MX2ALF																												
SCMPLX																												

Notes:

1. "I" signifies that the routine uses information from the common block.

2. "O" signifies that the routine puts information into the common block.

3. Blank common, common blocks /MCONST/, /PCONST/, /WW/, and the common blocks listed in Tables 4.2 and 4.3 are not included in this table.

Table 7.5--Input and output variables to routines in common blocks other than blank common and labeled common blocks /MCONST/, /PCONST/, and /WW/ and other than the common blocks listed in Tables 4.2 and 4.3

Routine	Input parameters common block	Input parameters parameter name	Output parameters common block	Output parameters parameter name
RAYTRC	/FLG/	IHOP	/HDRC/	DAT
	/RIN/	KAY2	/HDRC/	INITID
	/FLG/	LINES	/FLG/	LINES
	/RK/	NEQS	/FLG/	NEWWP
	/ERR/	NERG	/FLG/	NEWWR
	/ERR/	NERP	/FILEC/	NPLTDP
	/ERR/	NERR	/FLGP/	NSET
	/ERR/	NERT	/HDR/	SEC
	/RAYDEV/	NRYIND	/HDRC/	TOD
	/RIN/	OMEGMAX	/RK/	RSTART
	/RIN/	OMEGMIN		
	/FLG/	PENET		
	/RIN/	SGN		
DFCNST			/CRMACH/	RMACH
READW1	/RAYDEV/	NDEVTMP		
	/RAYDEV/	NRYIND		
READW	/B1/→/B20/		/B1/→/B20/	
	/FLG/	LINES		
	/RAYDEV/	NDEVTMP		
	/RAYDEV/	NFRMAT		
	/RAYDEV/	NRYIND		
	/FLGP/	NSET		
UCON	/UCONC/	CNVC	/UCONV/	CNVV
	/UCONV/	CNVV		
	/UCONC/	PCV		
TRACE	/TRAC/	D2Z	/TRAC/	DROLD
	/TRAC/	GROUND	/RK/	E1MAX
	/CRKTIME/	IRKTIME	/RK/	E1MIN
	/TRAC/	RAD	/RK/	E2MAX
	/TRAC/	RAD1	/RK/	E2MIN
	/TRAC/	THERE	/RK/	FACT
	/TRAC/	ZDOT	/TRLOCAL/	FDOT
	/TRLOCAL/	RSIGN	/TRLOCAL/	GDOLD
	/TRLOCAL/	HOME	/TRLOCAL/	GDOT
	/TRLOCAL/	FDOT	/TRLOCAL/	GOLD
			/TRAC/	GROUND
			/TRAC/	HOME
			/FLG/	HPUNCH
			/FLG/	IHOP

Table 7.5--(continued)

Routine	Input parameters common block	Input parameters parameter name	Output parameters common block	Output parameters parameter name
TRACE (continued)			/CRKTIME/	IRKTIME
			/RK/	MODE
			/TRAC/	NEWRAY
			/FLG/	NEWTRC
			/FLG/	PENET
			/TRAC/	ROLD
			/TRLOCAL/	RSIGN
			/RK/	RSTART
			/RK/	STEP
			/TRAC/	THERE
			/TRAC/	TOLD
PCROSS	/TRAC/	OSMT		
	/TRAC/	SMT		
RCROSS	/TRAC/	RAD1	/TRLOCAL/	FDOT
	/TRAC/	ZDOT	/TRLOCAL/	HOME
			/FLG/	HPUNCH
			/TRLOCAL/	RSIGN
			/RK/	RSTART
HAMLTN	/RIN/			
RKAM	/RKAMS/	ALPHA	/RK/	E1MAX
	/RK/	E1MAX	/CRKTIME/	IRKTIME
	/RKAMS/	FV	/RKAMS/	MM
	/CRKTIME/	IRKTIME	/RK/	MODE
	/RKAMS/	MM	/RK/	NEQS
	/RK/	MODE	/RKAMS/	XV
	/RK/	NEQS		
	/RK/	RSTART		
	/RK/	STEP		
	/RKAMS/	XV		
	/RKAMS/	YU		
RKAM1	/RK/	E1MAX	/RKAMS/	ALPHA
	/RK/	E1MIN	/RKAMS/	EPM
	/RK/	E2MAX	/RK/	E1MIN
	/RK/	E2MIN	/RK/	FACT
	/RK/	FACT	/RKAMS/	FV
	/RK/	MODE	/RKAMS/	MM
	/RK/	NN	/RK/	RSTART
	/RK/	RSTART	/RKAMS/	XV
	/RK/	SPACE	/RKAMS/	YU

Table 7.5--(continued)

Routine	Input parameters common block	Input parameters parameter name	Output parameters common block	Output parameters parameter name
BACKUP	/TRAC/	DROLD	/RK/	MODE
	/TRAC/	D2Z	/RK/	RSTART
	/RK/	E1MAX	/RK/	STEP
	/RK/	E2MIN	/TRAC/	THERE
	/RK/	FACT	/TRAC/	TOLD
	/RK/	MODE	/TRAC/	ZDOT
	/TRAC/	RAD1		
REFLECT	/FNDER/	NPZPH		
	/FNDER/	NPZR		
	/FNDER/	NPZTH		
FIT	/TRAC/	DROLD	/TRAC/	D2Z
	/FNDER/	NPZPH	/TRAC/	OSMT
	/FNDER/	NPZPHPH	/TRAC/	RAD
	/FNDER/	NPZR	/TRAC/	RAD1
	/FNDER/	NPZRPH	/TRAC/	SMT
	/FNDER/	NPZRR	/TRAC/	ZDOT
	/FNDER/	NPZRTH		
	/FNDER/	NPZTH		
	/FNDER/	NPZTHPH		
	/FNDER/	NPZTHTH		
	/FNDER/	NZ		
	/TRAC/	TOLD		
GET	/CRKTIME/	IRKTIME	/FNDER/	NPZPH
	/CRKTIME/	RKTIME	/FNDER/	NPZPHPH
	/CRMACH/	RMACH	/FNDER/	NPZR
			/FNDER/	NPZRPH
			/FNDER/	NPZRR
			/FNDER/	NPZRTH
			/FNDER/	NPZTH
			/FNDER/	NPZTHPH
			/FNDER/	NPZTHTH
			/FNDER/	NSELECT
			/FNDER/	NTIME
			/FNDER/	NZ
			/CGET/	ZERO
GET1	/CRKTIME/	IRKTIME	/CGET/	ZERO
	/CRKTIME/	RKTIME		
	/CRMACH/	RMACH		
CONBLK	/RAYDEV/	NRYIND	/UCONC/	CNVC
			/UCONV/	CNVV
			/CRKTIME/	IRKTIME
			/RIN/	KVECT

Table 7.5--(continued)

Routine	Input parameters: common block	Input parameters: parameter name	Output parameters: common block	Output parameters: parameter name
CONBLK (continued)			/RIN/	LPOLAR
			/RAYCON/	MCONP
			/RAYDEV/	NDEVTMP
			/RAYDEV/	NFRMAT
			/RAYDEV/	NRYIND
			/RIN/	OMEGMAX
			/RIN/	OMEGMIN
			/UCONC/	PCV
			/RIN/	POLAR
			/RIN/	RAYNAME
			/HDRC/	DAT
			/HDRC/	INITID
			/PROCFL/	LIST
			/PROCFL/	PITBL
			/PROCFL/	PNTBL
			/HDR/	SEC
			/HDRC/	TOD
PRINTR	/FLG/	HPUNCH	/FLG/	LINES
	/FLG/	IHOP	/ERR/	NERG
	/RIN/	KAY2	/ERR/	NERP
	/FLG/	LINES	/ERR/	NERR
	/RK/	NEQS	/ERR/	NERT
	/FLG/	NTYP	/FLG/	NEWWP
	/RIN/	POLAR		
	/RIN/	TYPE		
ATMOSHD	/HDRC/	DAT	/FLG/	LINES
	/RINPL/	DISPM	/FLG/	NTYP
	/FLG/	LINES		
	/RIN/	MODRIN		
	/FLGP/	NSET		
	/RIN/	RAYNAME		
	/HDRC/	TOD		
PUTKST	/FLG/	LINES	/FLG/	LINES
RAYPLT	/FLG/	NEWTRC	/PLT/	ALPHA
	/FLG/	NEWWR	/PLT/	APLT
	/FILEC/	NPLTDP	/FLG/	NEWTRC
			/FLG/	NEWWR
			/PLT/	PRESET
			/PLT/	RMAX
			/PLT/	RMIN
			/PLT/	XL
			/PLT/	XR
			/PLT/	YB
			/PLT/	YT

Table 7.5--(continued)

Routine	Input parameters common block	parameter name	Output parameters common block	parameter name
PLOT	/PLT/	ALPHA	/DD/	IX
	/PLT/	APLT	/DD/	IY
	/PLT/	RESET	/PLT/	RESET
	/PLT/	RMAX		
	/PLT/	RMIN		
	/PLT/	XMAXO		
	/PLT/	XMINO		
	/PLT/	YMAXO		
	/PLT/	YMINO		
LABPLT	/HDRC/	DAT	/DD/	IOR
	/RIN/	MODRIN	/DD/	IS
			/DD/	IT
			/DD/	IX
			/DD/	IY
PLTANH	/RAYCON/	MCONP	/DD/	IOR
	/LABCLT/	PROJCT	/DD/	IX
	/LABCLT/	RMAX	/DD/	IY
	/LABCLT/	RMIN		
	/LABCLT/	THMAX		
	/LABCLT/	THMIN		
SETXY			/LABCLT/	PROJCT
			/LABCLT/	RMAX
			/LABCLT/	RMIN
			/LABCLT/	THMAX
			/LABCLT/	THMIN
PLOTANOT	/ANNCTC/	ANOTES	/DD/	IOR
	/ANNCTC/	HNOTES	/DD/	IX
	/DD/	IX	/DD/	IY
	/DD/	IY		
	/ANNCTL/	LENA		
	/ANNCTL/	LENHA		
	/LABCLT/	THMAX		
	/LABCLT/	THMIN		
PLTLB	/DD/	IX	/DD/	IOR
	/DD/	IY	/DD/	IX
	/LABCLT/	PROJCT		
	/LABCLT/	RMAX		
	/LABCLT/	RMIN		

Table 7.5--(continued)

Routine	Input parameters common block	Input parameters parameter name	Output parameters common block	Output parameters parameter name
PLOTBL			/KNKN/	
			/DDLIM/	
DDBP	/DD/	IX	/KNKN/	KNBP
	/DD/	IY	/DDLIM/	MNIX
	/KNKN/	KNBP	/DDLIM/	MNIY
	/DDLIM/	MNIX	/DDLIM/	MXIX
	/DDLIM/	MNIY	/DDLIM/	MXIY
	/DDLIM/	MXIX		
	/DDLIM/	MXIY		
DDVC	/DD/	IX	/KNKN/	KNVC
	/DD/	IY	/DDLIM/	MNIX
	/KNKN/	KNVC	/DDLIM/	MNIY
	/DDLIM/	MNIX	/DDLIM/	MXIX
	/DDLIM/	MNIY	/DDLIM/	MXIY
	/DDLIM/	MXIX		
	/DDLIM/	MXIY		
DDTEXT	/DD/	IOR	/KNKN/	KNDT
	/DD/	IX		
	/DD/	IY		
	/KNKN/	KNDT		

Table 7.6--Definitions of the parameters in common block /HDR/

Position in common	Variable name	Definition
1	SEC	Total elapsed computer calculation time at end of calculating previous raypath

Table 7.7--Definitions of the parameters in common block /UCONC/

Position in common	Variable name	Definition
1-4	PCV	List of valid unit types for units conversion: blank (no conversion), AN (angle), LN (length), FQ (frequency
5-20	CNVC	An array of lists of valid physical units for each unit type for units conversion: blank (no conversion), M (meters), KM (kilometers), DG (degrees), etc

Table 7.8--Definitions of the parameters in common block /UCONV/

Position in common	Variable name	Definition
1-16	CNVV	An array of units conversion factors corresponding to the array CNVC in common block /UCONC/

Table 7.9--Definitions of the parameters in common block /RKAMS/

Position in common	Variable name	Definition
1-5	XV	Values of independent variable for 5 integration steps
6-85	FV	Values of the derivatives of the 20 dependent variables for 4 integration steps
86-285	YU	Values of the 20 dependent variables for 5 integration steps (in double precision)
286	EPM	The amount by which the independent variable changed during the previous call to SUBROUTINE RKAM1
287	ALPHA	Value of the independent variable at the beginning of the latest integration step
288	MM	Relative integration step number (varies from 1 to 4)

Table 7.10--Definitions of the parameters in common block /CGET/

Position in common	Variable name	Definition
1	ZERO	A great circle distance at sea level corresponding to a central earth angle that is twice the smallest floating point variable that can be stored in one single precision word in the computer being used

Table 7.11--Definitions of the parameters in common block /CRMACH/

Position in common	Variable name	Definition
1	RMACH(1)	Smallest positive magnitude = B**(EMIN-1)
2	RMACH(2)	Largest magnitude = B**EMAX*(1-B**(-T))
3	RMACH(3)	Smallest relative spacing = B**(-T)
4	RMACH(4)	Largest relative spacing = B**(1-T)
5	RMACH(5)	$\mathrm{Log}_{10}B = \mathrm{Log}_{10}2$

Notes: 1. B = the number base used by the computer (= 2 for most computers)
2. T = the number of bits in the mantissa of a floating point number
3. EMIN = the most negative allowable exponent
4. EMAX = the largest allowable positive exponent

Table 7.12--Definitions of the parameters in common block /CRKTIME/

Position in common	Variable name	Definition
1	IRKTIME	The number of times that SUBROUTINE RKAM has been called (used to compare FTIME or GTIME with to know whether F or G need to be updated)
1	RKTIME	Floating point name of IRKTIME

Table 7.13--Definitions of the parameters in common block /TRLOCAL/

Position in common	Variable name	Definition
1	RSIGN	+1 if next receiver-surface crossing is going up; -1 if going down
2	HOME	.TRUE. if ray is going away from receiver surface; .FALSE. otherwise
3	FDOT	Rate of change of distance of ray above the receiver surface
4	GDOT	Rate of change of distance of ray above the terrain
5	GOLD	Value of G at previous integration step (= distance of ray above terrain)
6	GDOLD	Value of GDOT at previous integration step

Table 7.14--Definitions of the parameters in common block /RK/*

Position in common	Variable name	Definition
1	N	The number of equations being integrated
2	STEP	The initial step in group path in kilometers
3	MODE	Defines type of integration used (same as W41), see Section 5.3.2
4	E1MAX	Maximum allowable single step error (same as W42)
5	E1MIN	Minimum allowable single step error (= W42/W43)
6	E2MAX	Maximum step length (same as W45)
7	E2MIN	Minimum step length (same as W46)
8	FACT	Factor to use to decrease step length (same as W47)
9	RSTART	Nonzero to initialize numerical integration, zero to continue integration

* These parameters are calculated in subroutine READW (some are temporarily reset in subroutine BACKUP) and are used in subroutine RKAM.

Table 7.15--Definitions of the parameters in common block /TRAC/*

Position in common	Variable name	Definition
1	GROUND	.TRUE. if the ray is on the surface of the Earth
2	PERIGE	.TRUE. if the ray has just made a perigee
3	THERE	.TRUE. if the ray is at the receiver height
4	MINDIS	.TRUE. if the ray has just made a closest approach to the receiver height
5	NEWRAY	Not used in this version of the program
6	SMT	An estimation of the vertical distance to an apogee or perigee of the ray
7	OSMT	Value of SMT at previous integration step
8-27	ROLD	Value of R(1) (=r in r,θ,ϕ earth-centered spherical polar coordinate system) at previous integration step
28-47	DROLD	Value of dr/dt at previous integration step
48	TOLD	Value of t (= independent variable for numerical integration) at previous integration step
49	ZDOT	dZ/dt (= dF/dt or dG/dt, depending on the situation)
50	D2Z	d^2Z/dt^2 (= d^2F/dt^2 or d^2G/dt^2, depending on the situation)
51	RAD	$(dZ/dt)^2 - 2\ Z\ d^2Z/dt^2$
52	RAD1	$\sqrt{RAD}$

* These parameters are used for communication between subroutines TRACE and BACKUP.

Table 7.16--Definitions of the parameters in common block /FNDER/

Position in common	Variable name	Definition	Value
1	NZ	Relative position of F in common block /RR/ (or G in common block /GG/)	1
2	NPZR	Relative position of PFR in common block /RR/ (or PGR in common block /GG/)	2
3	NPZRR	Relative position of PFRR in common block /RR/ (or PGRR in common block /GG/)	3
4	NPZRTH	Relative position of PFRTH in common block /RR/ (or PGRTH in common block /GG/)	4
5	NPZRPH	Relative position of PFRPH in common block /RR/ (or PGRPH in common block /GG/)	5
6	NPZTH	Relative position of PFTH in common block /RR/ (or PGTH in common block /GG/)	6
7	NPZPH	Relative position of PFPH in common block /RR/ (or PGPH in common block /GG/)	7
8	NPZTHTH	Relative position of PFTHTH in common block /RR/ (or PGTHTH in common block /GG/)	8
9	NPZPHPH	Relative position of PFPHPH in common block /RR/ (or PGPHPH in common block /GG/)	9
10	NPZTHPH	Relative position of PFTHPH in common block /RR/ (or PGTHPH in common block /GG/)	10
11	NSELECT	Relative position of FSELECT in common block /RR/ (or GSELECT in common block /GG/)	11
12	NTIME	Relative position of FTIME in common block /RR/ (or GTIME in common block /GG/)	12

Table 7.17--Definition of the parameters in common block /RIN/*

Position in common	Variable name	Definition
1-8	MODRIN	Description of version of DISPER in BCD
9-14	RAYNAME	Hollerith names of the characteristic rays in a birefringent medium (= blank for this version of the program)
15-17	TYPE	= Hollerith 1 or 3 if this version of DISPER includes wind, = Hollerith 2 or 3 if this version of DISPER includes losses
18	SPACE	TRUE, if the ray is in a homogeneous non-dissipative medium (Unconditionally set to FALSE in this version of the program)
19	OMEGMIN	Minimum frequency for nonevanescent propagation (= 0 for this version of the program)
20	OMEGMAX	Maximum frequency for nonevanescent propagation (not applicable for this version of the program, set to 0)
21,22	KAY2	k^2, square of the complex wave number
23,24	H	Hamiltonian (complex)
25,26	PHPT	$\partial H/\partial t$ (complex)
27,28	PHPR	$\partial H/\partial r$ (complex)
29,30	PHPTH	$\partial H/\partial \theta$ (complex)
31,32	PHPPH	$\partial H/\partial \phi$ (complex)
33,34	PHPOM	$\partial H/\partial \omega$ (complex)
35,36	PHPKR	$\partial H/\partial k_r$ (complex)
37,38	PHPKTH	$\partial H/\partial k_\theta$ (complex)
39,40	PHPKPH	$\partial H/\partial k_\phi$ (complex)
41,42	KPHPK	$\vec{k} \cdot \partial H/\partial \vec{k}$ (complex) $= k_r \partial H/\partial k_r + k_\theta\ \partial H/\partial k_\theta + k_\phi\ \partial H/\partial k_\phi$
43,44	POLAR	Characteristic transverse polarization of the wave (complex) (= 0 for this version of the program)
45,46	LPOLAR	Characteristic longitudinal polarization of the wave (complex) (= 1 for this version of the program)
47	SGN	= +1 or -1; used for ray tracing in complex space

* These parameters are calculated in subroutine DISPER and used in subroutine HAMLTN.

Note: In some subroutines, the real and imaginary parts of the complex variables have separate names.

Table 7.18--Definitions of the parameters in common block /ERR/

Position in common	Variable name	Definition
1	NERG	Index number for the dependent variable for the integration that gives G
2	NERR	Index number for the dependent variable for the integration that gives $\partial G/\partial r$
3	NERT	Index number for the dependent variable for the integration that gives $\partial G/\partial \theta$
4	NERP	Index number for the dependent variable for the integration that gives $\partial G/\partial \phi$

Table 7.19--Definitions of the parameters in common block /RAYDEV/

Position in common	Variable name	Definition
1	NRYIND	Device unit number for input data
2	NDEVTMP	Device unit number for temporary output and input
3	NFRMAT	Device unit number for secondary input file (not used by ray tracing program)
4		Device unit number for graphics output
5		Device unit number for binary raypath coordinate output

Table 7.20--Definitions of the parameters in common block /FLGP/

Position in common	Variable name	Definition
1	NSET	Runset number

Table 7.21--Definitions of the parameters in common block /RINPL/

Position in common	Variable name	Definition
1	DISPM	Character string identifier for the dispersion relation model

Table 7.22--Definitions of the parameters in common block /FLG/*

Position in common	Variable	Definition
1	NTYP	Wave polarization indicator (not used in this version of program)
2	NEWWR	Set equal to .TRUE. to tell subroutine RAYPLT there is a new W array
3	NEWWP	Set equal to .TRUE. to tell subroutine PRINTR there is a new W array
4	PENET	Set equal to .TRUE. if the ray left the allowed region of the atmosphere
5	LINES	Number of lines printed on the current page
6	IHOP	Hop number (at the beginning of each ray, subroutine TRACE sets this parameter to zero so that subroutine RAYPLT will begin a new line in plotting the raypath, and subroutine PRINTR will print column headings and punch a transmitter rayset)
7	HPUNCH	The height to be output on the raysets

* The parameters are used to communicate between various subroutines.

Table 7.23--Definitions of the parameters in common block /RINPL/

Position in common	Variable name	Definition
1	DISPM	Character string identifier for the dispersion relation model

Table 7.24--Definitions of the parameters in common block /FILEC/

Position in common	Variable name	Definition
1	NPLTDP	Set equal to the device unit number for binary raypath coordinate output (variable in position 5 of common block /RAYDEV/) if binary raypath coordinate output has been requested, set to zero otherwise

Table 7.25--Definitions of the parameters in common block /PLT/*

Position in common	Variable name	Definition
1	XMINO,XL	The x-coordinate of the left side of the plotting area in kilometers
2	XMAXO,XR	The x-coordinate of the right side of the plotting area in kilometers
3	XMINO,YB	The y-coordinate of the bottom of the plotting area in kilometers
4	XMAXO,YT	The y-coordinate of the top of the plotting area in kilometers
5	RESET	Set equal to one whenever the plotting area is changed

* These parameters are used for communication between subroutine RAYPLT and subroutine PLOT.

Table 7.26--Definitions of the parameters in common block /RAYCON/

Position in common	Variable name	Definition
1	MCONP	Set to zero for the raytracing program to indicate that the abscissa in raypath plots is a central-earth angle in radians, set non-zero for the contouring program to indicate that the abscissa in contour plots is a great-circle distance in kilometers

Table 7.27--Definitions of the parameters in common block /ANNCTC/

Position in common	Variable name	Definition
1-8	ANOTES	Character strings to label the ordinate of raypath plots
9-20	HNOTES	Character strings to label the abscissa of raypath plots

Table 7.28--Definitions of the parameters in common block /ANNCTL/

Position in common	Variable name	Definition
1-4	LENA	Lengths of the character strings that label the ordinate of raypath plots
5-7	LENHA	Lengths of the character strings that label the abscissa of raypath plots

Table 7.29--Definitions of the parameters in common block /LABCLT/

Position in common	Variable name	Definition
1	PROJCT	Number that indicates which type of projection is being used for raypath plots
2	THMIN	θ_{min}, minimum value of the abscissa of a raypath plot
3	THMAX	θ_{max}, maximum value of the abscissa of a raypath plot
4	RMIN	r_{min}, minimum value of the ordinate of a raypath plot
5	RMAX	r_{max}, maximum value of the ordinate of a raypath plot

Table 7.30--Definitions of the parameters in common block /DD/

Position in common	Variable name	Definition
1	IN	Intensity IN = 0 specifies normal intensity IN = 1 specifies high intensity
2	IOR	Orientation IOR = 0 specifies upright orientation IOR = 1 specifies rotated orientation (90° counterclockwise)
3	IT	Italics (Font) IT= 0 specifies non-italic (Roman) symbols IT = 1 specifies italic symbols
4	IS	Symbol size IS = 0 specifies miniature size IS = 1 specifies small size IS = 2 specifies medium size IS = 3 specifies large size
5	IC	Symbol case IC = 0 specifies uppercase IC = 1 specifies lowercase
6	ICC	Character code, 0-63 (R1 format) ICC and IC together specify the symbol plotted
7	IX	X-coordinate, 0-1023
8	IY	Y-coordinate, 0-1023

Table 7.31--Definitions of the parameters in common block /KNKN/

Position in common	Variable name	Definition
1	KNBP	Number of times SUBROUTINE DDBP was called
2	KNVC	Number of times SUBROUTINE DDVC was called
3	KNDT	Number of times SUBROUTINE DDTEXT was called

Table 7.32--Definitions of the parameters in common block /DDLIM/

Position in common	Variable name	Definition
1	MXIX	Maximum value of IX
2	MXIY	Maximum value of IY
3	MNIX	Minimum value of IX
4	MNIY	Minimum value of IY

Table 7.33--Definitions of the parameters in common block /UU/

Position in common	Variable name	Definition
1-4	MODU	Wind-velocity model and parameter identification
1	MODU(1)	Name of wind-velocity subroutine
2	MODU(2)	Wind-velocity parameter identification number
3	MODU(3)	Name of wind-velocity perturbation subroutine
4	MODU(4)	Wind-velocity perturbation parameter identification number
5	V	$\|V\|$, wind speed in km/s
6	PVT	$\partial\|V\|/\partial t$
7	PVR	$\partial\|V\|/\partial r$
8	PVTH	$\partial\|V\|/\partial\theta$
9	PVPH	$\partial\|V\|/\partial\phi$
10	VR	V_r, upward component of wind velocity
11	PVRT	$\partial V_r/\partial t$
12	PVRR	$\partial V_r/\partial r$
13	PVRRTH	$\partial V_r/\partial\theta$
14	PVRPH	$\partial V_r/\partial\phi$
15	VTH	V_θ, southward component of wind velocity
16	PVTHT	$\partial V_\theta/\partial t$
17	PVTHR	$\partial V_\theta/\partial r$
18	PVTHTH	$\partial V_\theta/\partial\theta$
19	PVTHPH	$\partial V_\theta/\partial\phi$
20	VPH	V_ϕ, eastward component of wind velocity
21	PVPHT	$\partial V_\phi/\partial t$
22	PVPHR	$\partial V_\phi/\partial r$
23	PVPHTH	$\partial V_\phi/\partial\theta$
24	PVPHPH	$\partial V_\phi/\partial\phi$

Table 7.34--Definitions of the parameters in common block /CC/

Position in	Variable	Definition common name
1-4	MODC	Sound-speed model and parameter identification
1	MODC(1)	Name of sound-speed subroutine
2	MODC(2)	Sound-speed parameter identification number
3	MODC(3)	Name of sound-speed perturbation subroutine
4	MODC(4)	Sound-speed perturbation parameter identification number
5	CS	C^2, square of sound speed in km^2/s^2
6	PCST	$\partial C^2/\partial t$
7	PCSR	$\partial C^2/\partial r$
8	PCSTH	$\partial C^2/\partial \theta$
9	PCSPH	$\partial C^2/\partial \phi$

Table 7.35--Definitions of the parameters in common block /TT/

Position in common	Variable name	Definition
1-4	MODT	Temperature model and parameter identification
1	MODT(1)	Name of temperature subroutine
2	MODT(2)	Temperature parameter identification number
3	MODT(3)	Name of temperature-perturbation subroutine
4	MODT(4)	Temperature-perturbation-parameter identification number
5	T	T, temperature in kelvins
6	PTT	$\partial T/\partial t$
7	PTR	$\partial T/\partial r$
8	PTTH	$\partial T/\partial \theta$
9	PTPH	$\partial T/\partial \phi$

Table 7.36--Definitions of the parameters in common block /MM/

Position in common	Variable name	Definition
1-4	MODM	Molecular weight model and parameter identification
	MODM(1)	Name of molecular weight subroutine
2	MODM(2)	Parameter identification number for molecular weight model
3	MODM(3)	Unused now
4	MODM(4)	Unused now
5	M	M, mean molecular weight
6	PMT	$\partial M/\partial t$
7	PMR	$\partial M/\partial r$
8	PMTH	$\partial M/\partial \theta$
9	PMPH	$\partial M/\partial \phi$

Table 7.37--Definitions of the parameters in common block /RR/

Position in common	Variable name	Definition
1-4	MODREC	Receiver-surface model and parameter identification
1	MODREC(1)	Name of receiver-surface subroutine
2	MODREC(2)	Parameter identification number for receiver-surface model
3	MODREC(3)	Unused now
4	MODREC(4)	Unused now
5	F	$f(r,\theta,\phi)$ defined in (6.72)-(6.74)
6	PFR	$\partial f/\partial r$
7	PFRR	$\partial^2 f/\partial r^2$
8	PFRTH	$\partial^2 f/\partial r\partial\theta$
9	PFRPH	$\partial^2 f/\partial r\partial\phi$
10	PFTH	$\partial f/\partial\theta$
11	PFPH	$\partial f/\partial\phi$
12	PFTHTH	$\partial^2 f/\partial\theta^2$
13	PFPHPH	$\partial^2 f/\partial\phi^2$
14	PFTHPH	$\partial^2 f/\partial\theta\partial\phi$
15	FSELECT	= "RECEIVER"
16	FTIME	An integer that is initialized to equal -1 at the beginning of each raypath calculation and is incremented by 1 at each integration step so that it is possible to determine whether the variables in this common block are current

Table 7.38--Definitions of the parameters in common block /GG/

Position in common	Variable name	Definition
1-4	MODG	Terrain model and parameter identification
1	MODG(1)	Name of terrain subroutine
2	MODG(2)	Terrain-parameter identification number
3	MODG(3)	Name of terrain-perturbation subroutine
4	MODG(4)	Terrain-perturbation parameter identification number
5	G	$g(r,\theta,\phi)$ defined in the same way as $f(r,\theta,\phi)$ in (6.72)-(6.74)
6	PGR	$\partial g/\partial r$
7	PGRR	$\partial^2 g/\partial r^2$
8	PGRTH	$\partial^2 g/\partial r\partial\theta$
9	PGRKPH	$\partial^2 g/\partial r\partial\phi$
10	PGTH	$\partial g/\partial\theta$
11	PGPH	$\partial g/\partial\theta$
12	PGTHTH	$\partial^2 g/\partial\theta^2$
13	PGPHPH	$\partial^2 g/\partial\phi^2$
14	PGTHPH	$\partial^2 g/\partial\theta\partial\phi$
15	GSELECT	= "TERRAIN"
16	GTIME	An integer that is initialized to equal -1 at the beginning of each raypath calculation and is incremented by 1 at each integration step so that it is possible to determine whether the variables in this common block are current

Table 7.39--Definitions of the parameters in common block /AA/

Position in common	Variable name	Definition
1-4	MODA	Viscosity/thermal conductivity model and parameter identification
1	MODA(1)	Name of viscosity/thermal conductivity subroutine
2	MODA(2)	Viscosity/thermal conductivity parameter identification number
3	MODA(3)	Name of viscosity/thermal conductivity perturbation subroutine
4	MODA(4)	Viscosity/thermal conductivity perturbation parameter identification number
5	MU	μ, viscosity
6	MUPT	$\partial\mu/\partial t$
7	MUPR	$\partial\mu/\partial r$
8	MUPTH	$\partial\mu/\partial\theta$
9	MUPPH	$\partial\mu/\partial\phi$
10	KAPPA	κ, thermal conductivity
11	KAPPT	$\partial\kappa/\partial t$
12	KAPPR	$\partial\kappa/\partial r$
13	KAPPTH	$\partial\kappa/\partial\theta$
14	KAPPPH	$\partial\kappa/\partial\phi$

Table 7.40--Definitions of the parameters in common block /PP/

Position in common	Variable name	Definition
1-4	MODP	Pressure model and parameter identification
1	MODP(1)	Name of pressure subroutine
2	MODP(2)	Pressure-parameter identification number
3	MODP(3)	Name of pressure-perturbation subroutine
4	MODP(4)	Pressure-perturbation parameter identification number
5	P	P, pressure
6	PPT	$\partial p/\partial t$
7	PPR	$\partial p/\partial r$
8	PPTH	$\partial p/\partial \theta$
9	PPPH	$\partial p/\partial \phi$

REFERENCES

Bennett, J. A. (1967): The calculation of Doppler shifts due to a changing ionosphere. J. Atmos. Terr. Phys., 29, 887-891.

Budden, K. G. (1961): Radio Waves in the Ionosphere. University Press, Cambridge, England.

Budden, K. G. (1972): The theory of coupling of characteristic radio waves in the ionosphere. J. Atmos. Terr. Phys., 34, 1909-1921.

Croft, T. A., and L. Gregory (1963): A fast, versatile ray-tracing program for IBM 7090 digital computers. Rept. SEL-63-107 (TR 82), Contract No. 225(64), Stanford Electronics Laboratories, Stanford, Calif.

Cornyn, J. J. (1973): GRASS: A digital computer ray-tracing and transmission-loss-prediction system. Vol. 1, NRL Report 7621, Naval Research Laboratory, Washington, D.C.

Courant, R., and D. Hilbert (1962): Methods of Mathematical Physics, vol. II, Partial Differential Equations. Interscience Publishers, John Wiley and Sons, New York.

Dudziak, W. F. (1961): Three-dimensional ray trace computer program for electromagnetic wave propagation studies. RM 61 TMP-32, DASA 1232, G. E. TEMPO, Santa Barbara, Calif.

Felsen, L. B., and N. Marcuvitz (1973): Radiation and Scattering of Waves. Prentice-Hall, Englewood Cliffs, N.J.

Feynman, R. P., and A. R. Hibbs (1965): Quantum Mechanics and Path Integrals. McGraw-Hill, New York.

Garabedian, P. R. (1964): Partial Differential Equations. John Wiley and Sons, New York.

Georges, T. M. (1967): Ionospheric effects of atmospheric waves. ESSA Tech. Rept. IER 57-ITSA-54, ESSA Institute for Telecommunication Sciences, Boulder, Colo.

Georges, T. M., and J. J. Stephenson (1968): HF radar signatures of traveling ionospheric irregularities: 3D ray-tracing simulation. Radio Sci., 4, 679-696.

Georges, T. M. (1970): Amplification of ionospheric heating and triggering of 'spread-F' by natural irregularities. J. Geophys. Res., 75, 6436-6438.

Georges, T. M. (1971): A program for calculating three-dimensional acoustic-gravity ray paths in the atmosphere. NOAA Tech. Rep. ERL 212-WPL 16, NOAA Environmental Research Laboratories, Boulder, Colo.

Georges, T. M. (1972): Acoustic ray paths through a model vortex with a viscous core. J. Acoust. Soc. Am., 51, 206-209.

Georges, T. M., and W. H. Beasley (1977): Refraction of infrasound by upper atmospheric winds. J. Acoust. Soc. Am., 61, 28-34.

Georges, T. M., R. M. Jones, and J. P. Riley (1986): Simulating ocean acoustic tomography measurements with Hamiltonian ray tracing. IEEE J. Oceanic Engr., OE11, 58-71.

Gossard, E. E., and W. H. Hooke (1975): Waves in the Atmosphere. Elsevier, New York.

Haselgrove, J. (1954): Ray theory and a new method for ray tracing. Report of Conference on the Physics of the Ionosphere (London Physical Society), 355-364.

Jones, R. M. (1966): A three-dimensional ray tracing computer program. ESSA Tech. Rep., IER 17-ITSA 17, ESSA Institute for Telecommunication Sciences, Boulder, Colo.

Jones, R. M. (1968): Modifications to the three-dimensional ray tracing program described in IER 17-ITSA 17. ESSA Tech. Memo. ERLTM-ITS 134, ESSA Institute for Telecommunication Sciences, Boulder, Colo.

Jones, R. M. (1970): Ray theory for lossy media. Radio Sci., 6, 793-801.

Jones, R. M., and J. J. Stephenson (1975): A versatile three-dimensional ray tracing computer program for radio waves in the ionosphere. OT Rep.

75-76, U.S. Department of Commerce/Office of Telecommunications, Boulder, Colo.

Jones, R. M., and T. M. Georges (1976): Infrasound from convective storms. III. Propagation to the ionosphere. J. Acoust. Soc. Am., 59, 765-779.

Jones, R. M. (1981a): Bending of a ray in a random inhomogeneous medium. NOAA Tech. Memo. ERL WPL-77, NOAA Environmental Research Laboratories, Boulder, Colo.

Jones, R. M. (1981b): The frequency shift of a pulse by a time-independent dispersive, lossy medium. NOAA Tech. Memo. ERL WPL-80, NOAA Environmental Research Laboratories, Boulder, Colo.

Jones, R. M. (1982): Algorithms for reflecting rays from general topographic surfaces in a ray-tracing program. NOAA Tech. Memo. ERL WPL-98, NOAA Environmental Research Laboratories, Boulder, Colo., 58 pp.

Jones, R. M., J. P. Riley, and T. M. Georges (1982): A versatile three-dimensional Hamiltonian ray-tracing computer program for acoustic waves in the atmosphere. NOAA Tech. Memo. ERL WPL-103, NOAA Environmental Research Laboratories, Boulder, Colo., 242 pp.

Jones, R. M. (1983): A survey of underwater-acoustic ray-tracing techniques. NOAA Tech. Memo. ERL WPL-111, NOAA Environmental Research Laboratories, Boulder, Colo.

Jones, R. M., T. M. Georges, and J. P. Riley (1984): Modeling acoustic remote sensing in the Florida Straits with ray tracing. IEEE Trans. Geosci. Remote Sens., GE-22, 633-640.

Jones, R. M., J. P. Riley, and T. M. Georges (1986): HARPO--A versatile three-dimensional Hamiltonian ray-tracing program for acoustic waves in an ocean with arbitrary bottom. NOAA Special Rep., NOAA Environmental Research Laboratories, Boulder, Colo.

Keller, J. B. (1962): Geometrical theory of diffraction. J. Opt. Soc. Am., 52, 116-130.

Lighthill, M. J. (1965): Group velocity. J. Inst. Math. Appl., 1, 1-28.

Lighthill, M. J. (1978): Waves in Fluids. Cambridge Univ. Press, Cambridge, England, Sec. 4.5.

Ludwig, D. (1966): Uniform asymptotic expansions at a caustic. Commun. Pure Appl. Math., 19, 215-250.

NOAA, NASA, and USAF, "U.S. Standard Atmosphere, 1976". U.S. Government Printing Office, Washington, D.C. October 1976, p. 19.

Pedersen, M. A. (1961): Acoustic intensity anomalies introduced by constant velocity gradients. J. Acoust. Soc. Am., 33, 465-474.

Pierce, A. D. (1965): Extension of the method of normal modes to sound propagation in an almost-stratified medium. J. Acoust. Soc. Am., 37, 19-27.

Raspet, R., S. W. Lee, E. Kuester, D. C. Chang, W. F. Richards, R. Gilbert, and N. Bong (1985): Fast-field program for sound propagation in a layered atmosphere above an impedance ground. J. Acoust. Soc. Am., 77, 345-352.

Roberts, B. G., Jr. (1974): Horizontal-gradient acoustical ray-trace program TRIMAIN. NRL Report 7827, Naval Research Laboratory, Washington, D.C.

Sears, F. W. (1956): An Introduction to Thermodynamics, the Kinetic Theory of Gases, and Statistical Mechanics. Second Edition, Addison-Wesley, Reading, Massachusetts.

Stephenson, J. J., and T. M. Georges (1969): Computer routines for synthesizing ground-backscatter from three-dimensional raysets. ESSA Tech. Rep. ERL 120-ITS84, NOAA Environmental Research Laboratories, Boulder, Colo.

Tappert, F. D. (1977): The parabolic approximation method. In Wave Propagation in Underwater Acoustics, edited by J. B. Keller and J. S. Papadakis, Springer-Verlag, New York.

Tolstoy, I., and C. S. Clay (1966): Ocean Acoustics, Theory and Experiment in Underwater Sound. McGraw-Hill, New York.

Valley, S. L. (1965): Handbook of Geophysics and Space Environments. McGraw-Hill, New York.

White, D., and M. Pedersen (1981): Evaluation of shadow-zone fields by uniform asymptotics and complex rays. J. Acoust. Soc. Am., 69, 1029-1059.

APPENDIX A: PRINTOUT AND RAYSET LISTING FOR THE SAMPLE CASE

This appendix contains an abbreviated printout and rayset listing for the sample case. To save space, we have listed the printout for an elevation-angle increment of 20° instead of 5°, the increment used to produce the ray plots and raysets. Users should compare their sample-case output with this printout to be sure they are identical. The meanings of the printed quantities are explained in Sections 2.5.1 and 2.5.2, and the meanings of rayset quantities are listed in Figures 2.9 and 2.10.

***** H A R P A *****

HAMILTONIAN ACOUSTIC RAY-TRACING PROGRAM FOR THE ATMOSPHERE

BY

R. M. JONES, J. P. RILEY AND T. M. GEORGES
WAVE PROPAGATION LABORATORY
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
BOULDER, COLORADO 80303

RUN SET NUMBER 1

ATMOSPHERIC MODEL ID — S03

ATMOSPHERIC MODEL DESCRIPTION — SAMPLE CASE FOR HARPA DOCUMENTATION REV. 2-10-86

MODEL TYPE	SUBROUTINE NAME	DATA SET ID	DESCRIPTION
DISPERSION RELATION	AWWWL		ACOUSTIC WAVE *** WITH WIND ****** WITH LOSSES
BACKGROUND WIND VELOCITY	ULOGZ2	3.00	LOGARITHMIC EASTWARD WIND PROFILE, U*=.5 M/S, Z0=1 KM
WIND VELOCITY PERTURBATION	NPWIND	.00	NO WIND PERTURBATION
BACKGROUND SOUND SPEED	GAMRTDM	.00	SOUND SPEED IN TERMS OF TEMPERATURE MODEL
SOUND SPEED PERTURBATION	CBLOB2	2.00	50% INCREASE IN SQ. SOUND SPEED AT 125KM HT, 335KM N, 125KM E
BACKGROUND TEMPERATURE	TTANH5	1.00	U.S. STANDARD ATMOSPHERE 1962 TEMPERATURE PROFILE
TEMPERATURE PERTURBATION	TBLOB2	2.00	50% CYLINDRICAL INCREASE IN TEMPERATURE AT 105KM N., 105 KM W
MOLECULAR WEIGHT	MCONST	29.00	MOLECULAR WEIGHT = 29
BACKGROUND TERRAIN	GLORENZ	2.00	RIDGE 2-KM HIGH, 30-KM WIDE ALONG EQUATOR
TERRAIN PERTURBATION	NPTERR	.00	NO TERRAIN PERTURBATION
BACKGROUND VISCOSITY/ THERMAL CONDUCTIVITY	MUARDC	1.00	ARDC VISCOSITY AND THERMAL CONDUCTIVITY MODEL
VISCOSITY/THERMAL CONDUCTIVITY PERTURBATION	NPABSR	.00	NO VISCOSITY/CONDUCTIVITY PERTURBATION
BACKGROUND PRESSURE	PEXP	1.00	EXPONENTIAL PRESSURE MODEL, SCALE HEIGHT = 8.5 KM
PRESSURE PERTURBATION	NPPRES	.00	NO PRESSURE PERTURBATION
RECEIVER SURFACE	RTERR	.00	RECEIVER SURFACE 5 KM ABOVE TERRAIN

INPUT DATA FILE FOR RUN SET NUMBER   1

```
S03-1   SAMPLE CASE FOR HARPA DOCUMENTATION        REV. 2-10-86
  1    6370.           EARTH RADIUS, KM (6370.)
  3      13.          TTRANSMITTER HEIGHT, KM (T=ABOVE TERRAIN)
  4     200.   AN KM  N. TRANSMITTER LATITUDE, KM
  5       0.   AN KM  E. TRANSMITTER LONGITUDE, KM
  7      .05   FQ HZ  INITIAL FREQUENCY, HZ
 11      45.   AN DG  INITIAL AZIMUTH ANGLE, DEG
 15     -20.   AN DG  INITIAL ELEVATION ANGLE, DEG
 16     140.   AN DG  FINAL   ELEVATION ANGLE, DEG
 17      20.   AN DG  STEP IN ELEVATION ANGLE, DEG
 20       5.          RECEIVER HEIGHT, KM
 22       3.          MAXIMUM NUMBER OF HOPS (1.0)
 23    1000.          MAXIMUM NUMBER OF STEPS PER HOP (1000.)
 26     500.          MAXIMUM RAY HEIGHT, KM (500.)
 27      -1.          MINIMUM RAY HEIGHT, KM
 28    1000.          MAXIMUM RANGE, KM
 29  0000100.         DO: EIGRAY/RNG-TIM/RNG-ELV/NEW-PROJ/RAYTRC/CONT/PROF
 33   999.999         MAXIMUM ABSORPTION, DB (999.999)
 42       1.0E-6      SINGLE-STEP INTEGRATION ERROR (1.0E-4)
 44       .1          INITIAL INTEGRATION STEP SIZE, KM (1.0)
 57       2.          PHASE PATH (0=NO; 1=INTEGRATE; 2=INTEGRATE/PRINT)
 58       2.          ABSORPTION (0=NO; 1=INTEGRATE; 2=INTEGRATE/PRINT)
 60       2.          PATH LENGTH (0=NO; 1=INTEGRATE; 2=INTEGRATE/PRINT)
 71      50.          NUMBER OF INTEGRATION STEPS PER PRINT [1.E31]
 72       1.          OUTPUT RAYSETS (1=YES; 0=NO)
 73       0.          DIAGNOSTIC PRINTOUT (1=YES; 0=NO)
 74       0.          PRINT RAY STEPS (0=YES; 1=NO)
 75      .15          FULANN LETTER HEIGHT [0.15 IN]
 76       0.          BINARY RAY OUTPUT (1=YES; 0=NO)
 77      57.          LINES PER PAGE IN PRINTOUT (57.)
 81       1.          RAYPLOT PROJECTION (1=VERT; 2=HORIZ) PLANE
 82       1.          PLOT-EXPANSION FACTOR [1.0]
 83    -100.   AN KM  N. LATITUDE  OF LEFT PLOT EDGE, KM
 84    -300.   AN KM  E. LONGITUDE OF LEFT PLOT EDGE, KM
 85     500.   AN KM  N. LATITUDE  OF RIGHT PLOT EDGE, KM
 86     300.   AN KM  E. LONGITUDE OF RIGHT PLOT EDGE, KM
 87     100.   AN KM  DISTANCE BETWEEN TIC MARKS, KM
 88       0.          HEIGHT ABOVE SEA LEVEL OF BOTTOM OF GRAPH, KM
 89     300.          HEIGHT ABOVE SEA LEVEL OF TOP OF GRAPH, KM
 96     100.          DISTANCE BETWEEN VERTICAL TIC MARKS, KM
100       6.          ULOGZ2 WIND MODEL CHECK NUMBER
102       3.           BACKGROUND WIND DATA SET ID
103       5.   LN  M   REFERENCE WIND SPEED, M/S
104      .35           VON KARMAN'S CONSTANT
105       1.           ROUGHNESS HEIGHT, KM
150       1.          GAMRTDM SOUND SPEED MODEL CHECK NUMBER
175       2.          CBLOB2 MODEL CHECK NUMBER
177       2.           SOUND SPEED PERTURBATION DATA SET ID
178       .5           FRACTIONAL INCREASE OF SQUARED SOUND SPEED
179     125.           HEIGHT OF MAXIMUM INCREASE, KM
```

```
180      335.    AN KM   N. LATITUDE OF MAXIMUM INCREASE, KM
181      125.    AN KM   E. LONGITUDE OF MAXIMUM INCREASE, KM
182       25.            GAUSSIAN WIDTH IN HEIGHT OF INCREASE, KM
183       50.    AN KM   N-S WIDTH OF THE INCREASE, KM
184       25.    AN KM   E-W WIDTH OF THE INCREASE, KM
200        7.           TTANH5 TEMPERATURE MODEL CHECK NUMBER
202        1.            BACKGROUND TEMPERATURE DATA SET ID
225        2.           TBLOB2 MODEL CHECK NUMBER
227        2.            TEMPERATURE PERTURBATION DATA SET ID
228         .5           FRACTIONAL TEMPERATURE INCREASE
229        0.            HEIGHT OF MAXIMUM INCREASE, KM
230      105.    AN KM   N. LATITUDE OF MAXIMUM INCREASE, KM
231     -105.    AN KM   E. LONGITUDE OF MAXIMUM INCREASE, KM
232        0.            GAUSSIAN WIDTH IN HEIGHT OF INCREASE, KM
233       50.    AN KM   N-S WIDTH OF THE INCREASE, KM
234       25.    AN KM   E-W WIDTH OF THE INCREASE, KM
250        1.           MCONST MOLECULAR WEIGHT MODEL CHECK NUMBER
252       29.            MOLECULAR WEIGHT DATA SET ID
253       29.            MOLECULAR WEIGHT
275        2.           RTERR RECEIVER MODEL CHECK NUMBER
300        4.           GLORENZ TERRAIN MODEL CHECK NUMBER
302        2.            TERRAIN MODEL DATA SET ID
303        2.            HEIGHT OF THE RIDGE, KM
304        0.            N. LATITUDE OF THE RIDGE CENTER
305       30.    AN KM   HALF-WIDTH OF THE RIDGE, KM
325        0.           NPTERR NO TERRAIN PERTURBATION
500        1.           MUARDC VISC/COND MODEL CHECK NUMBER
502        1.            VISC/COND MODEL DATA SET ID
503     1.458E-06        VISCOSITY COEFFICIENT BETA
504     110.4            SUTHERLAND'S CONSTANT, KELVINS
505       1.91           PRANDTL NUMBER
525        0.           NPABS NO VISC/COND PERTURBATION
550        1.           PEXP PRESSURE MODEL CHECK NUMBER
552        1.            BACKGROUND PRESSURE MODEL DATA SET ID
553   101328.            PRESSURE AT SEA LEVEL, N/SQ.M.
554       8.5            PRESSURE SCALE HEIGHT, KM
575        0.           NPPRES   NO PRESSURE PERTURBATION
 -1                     DATA SUBSET FOR BACKGROUND WIND MODEL
  A  LOGARITHMIC EASTWARD WIND PROFILE, U*=.5 M/S, Z0=1 KM
  0      RETURN TO W ARRAY DATA SET
 -2                     DATA SUBSET FOR WIND PERTURBATION MODEL
  A  NO WIND PERTURBATION
  0      RETURN TO W ARRAY DATA SET
 -3                     DATA SUBSET FOR BACKGROUND SOUND-SPEED MODEL
  A  SOUND SPEED IN TERMS OF TEMPERATURE MODEL
  0      RETURN TO W ARRAY DATA SET
 -4                     DATA SUBSET FOR SOUND-SPEED PERTURBATION MODEL
  A  50% INCREASE IN SQ. SOUND SPEED AT 125KM HT, 335KM N, 125KM E
  0      RETURN TO W ARRAY DATA SET
 -5                     DATA SUBSET FOR TEMPERATURE MODEL
  A  U.S. STANDARD ATMOSPHERE 1962 TEMPERATURE PROFILE
  3     999.0           TEMPERATURE PROFILE FOR 1962 STD ATMOSPHERE
   0.          288.000       0.
 15.0000       190.500      10.0000
 52.0000       320.000       7.50000
```

```
 95.0000      191.000       10.0000
165.0000     1451.000       50.0000
300.0000     1586.000        0.
999.0000
 0       RETURN TO W ARRAY DATA SET
-6                            DATA SUBSET FOR TEMPERATURE PERTURBATION MODEL
 A   50% CYLINDRICAL INCREASE IN TEMPERATURE AT 105KM N., 105 KM W
 0       RETURN TO W ARRAY DATA SET
-7                            DATA SUBSET FOR MOLECULAR WEIGHT MODEL
 A   MOLECULAR WEIGHT = 29
 0       RETURN TO W ARRAY DATA SET
-8                            DATA SUBSET FOR RECEIVER SURFACE MODEL
 A   RECEIVER SURFACE 5 KM ABOVE TERRAIN
 0       RETURN TO W ARRAY DATA SET
-9                            DATA SUBSET FOR TERRAIN MODEL
 A   RIDGE 2-KM HIGH, 30-KM WIDE ALONG EQUATOR
 0       RETURN TO W ARRAY DATA SET
-10                           DATA SUBSET FOR TERRAIN PERTURBATION MODEL
 A   NO TERRAIN PERTURBATION
 0       RETURN TO W ARRAY DATA SET
-17                           DATA SUBSET FOR VISC/COND MODEL
 A   ARDC VISCOSITY AND THERMAL CONDUCTIVITY MODEL
 0       RETURN TO W ARRAY DATA SET
-18                           DATA SUBSET FOR VISC/COND PERTURBATION MODEL
 A   NO VISCOSITY/CONDUCTIVITY PERTURBATION
 0       RETURN TO W ARRAY DATA SET
-19                           DATA SUBSET FOR BACKGROUND PRESSURE MODEL
 A   EXPONENTIAL PRESSURE MODEL, SCALE HEIGHT = 8.5 KM
 0       RETURN TO W ARRAY DATA SET
-20                           DATA SUBSET FOR PRESSURE PERTURBATION MODEL
 A   NO PRESSURE PERTURBATION
 0       RETURN TO W ARRAY DATA SET
 0                            **********  END OF RUN SET NUMBER 1  ***********
```

INITIAL VALUES FOR THE W ARRAY
ONLY NONZERO VALUES PRINTED — ALL ANGLES IN RADIANS

N	W(N)
1	.637000000000000E+04
3	.130440097799511E+02
4	.313971742543171E-01
7	.314159265358979E+00
11	.785398163397446E+00
15	-.349065850398864E+00
16	.244346095279205E+01
17	.349065850398864E+00
20	.500000000000000E+01
22	.300000000000000E+01
23	.100000000000000E+04
24	.157079632679490E+01
26	.500000000000000E+03
27	-.100000000000000E+01
28	.100000000000000E+04
29	.100000000000000E+03
33	.999999000000000E+03
41	.300000000000000E+01
42	.999999999999997E-06
43	.500000000000000E+02
44	.100000000000000E+00
45	.100000000000000E+03
46	.100000000000000E-07
47	.500000000000000E+00
57	.200000000000000E+01
58	.200000000000000E+01
60	.200000000000000E+01
71	.500000000000000E+02
72	.100000000000000E+01
75	.150000000000000E+00
77	.570000000000000E+02
81	.100000000000000E+01
82	.100000000000000E+01
83	-.156985871271585E-01
84	-.470957613814755E-01
85	.784929356357926E-01
86	.470957613814755E-01
87	.156985871271585E-01
89	.300000000000000E+03
96	.100000000000000E+03
100	.600000000000000E+01
102	.300000000000000E+01
103	.500000000000000E-02
104	.350000000000000E+00
105	.100000000000000E+01
150	.100000000000000E+01

N	W(N)
175	.200000000000000E+01
177	.200000000000000E+01
178	.500000000000000E+00
179	.125000000000000E+03
180	.525902668759810E-01
181	.196232339089482E-01
182	.250000000000000E+02
183	.784929356357927E-02
184	.392464678178964E-02
200	.700000000000000E+01
202	.100000000000000E+01
225	.200000000000000E+01
227	.200000000000000E+01
228	.500000000000000E+00
230	.164835164835164E-01
231	-.164835164835164E-01
233	.784929356357927E-02
234	.392464678178964E-02
250	.100000000000000E+01
252	.290000000000000E+02
253	.290000000000000E+02
275	.200000000000000E+01
300	.400000000000000E+01
302	.200000000000000E+01
303	.200000000000000E+01
305	.470957613814754E-02
500	.100000000000000E+01
502	.100000000000000E+01
503	.145800000000000E-05
504	.110400000000000E+03
505	.191000000000000E+01
550	.100000000000000E+01
552	.100000000000000E+01
553	.101328000000000E+06
554	.850000000000000E+01

AZIMUTH ANGLE OF TRANSMISSION = 45.000000 DEG TRANSMITTER LATITUDE = 1.798926 DEG FREQUENCY = .050000 HZ
ELEVATION ANGLE OF TRANSMISSION = -20.000000 DEG TRANSMITTER LONGITUDE = .000000 DEG SINGLE STEP ERROR = 1.000000E-06

ERROR	EVENT	ELEVATION ABOVE SEA LEVEL KM	ELEVATION ABOVE TERRAIN KM	RANGE KM	AZIMUTH DEVIATION XMTR DEG	AZIMUTH DEVIATION LOCAL DEG	ELEVATION ANGLE XMTR DEG	ELEVATION ANGLE LOCAL DEG	PULSE TIME SEC	PHASE TIME SEC	ABSORPTION DB	PATH LENGTH KM
-.1E-13	XMTR	13.0440	13.0000	.0000				-20.000	.0000	.0000	.0000	.0000
.2E-06	RCVR	5.0367	5.0000	29.4887	-3.840	-3.838	-15.304	-12.169	91.9612	91.9612	.0000	30.6167
.2E-06	GRND REF	.0313	.0000	56.3108	-2.735	-2.735	-13.252	10.682	172.2877	172.2877	.0000	57.9268
.8E-07	RCVR	5.0270	5.0000	83.1414	-2.343	-2.340	-5.874	12.148	252.6076	252.6076	.0000	85.2361
-.1E-06		30.1362	30.1186	179.8168	-3.794	-3.792	4.603	5.546	553.4076	553.4076	.0000	185.6886
-.1E-06	APOGEE	30.8025	30.7860	196.4513	-3.954	-3.953	4.264	-.461	601.4076	601.4076	.0000	202.4308
-.1E-06	WAVE REV	30.8025	30.7860	196.4513	-3.954	-3.953	4.264	-.461	601.4076	601.4076	.0000	202.4308
.1E-06		11.0830	11.0709	283.7756	-4.435	-4.436	-1.671	-18.721	870.2076	870.2076	.0000	292.7749
.3E-06	RCVR	5.0112	5.0000	307.3288	-4.374	-4.375	-2.877	-12.129	943.0763	943.0763	.0000	317.1400
.3E-06	MAX HOPS	5.0112	5.0000	307.3288	-4.374	-4.375	-2.877	-12.129	943.0763	943.0763	.0000	317.1400

THIS RAY CALCULATION TOOK 2.430 SEC

ELEVATION ANGLE OF TRANSMISSION = .0000 DEG

-.7E-14	XMTR	13.0440	13.0000	.0000				.000	.0000	.0000	.0000	.0000
.1E-07	APOGEE	19.8306	19.8069	114.8628	-4.945	-4.945	2.856	-.376	352.1000	352.1000	.0000	115.4148
.1E-07	WAVE REV	19.8306	19.8069	114.8628	-4.945	-4.945	2.856	-.376	352.1000	352.1000	.0000	115.4148
.1E-07		19.1628	19.1412	135.8244	-4.991	-4.991	1.962	-3.544	416.1000	416.1000	.0000	136.4540
.4E-07	MIN DIST	13.0453	13.0303	225.0470	-4.938	-4.939	-1.012	-.003	689.9096	689.9096	.0000	226.1364
.4E-07	MIN DIST	13.0453	13.0303	225.0470	-4.938	-4.939	-1.012	-.003	689.9096	689.9096	.0000	226.1364
.3E-07	WAVE REV	13.0683	13.0535	229.2405	-4.930	-4.932	-1.025	.687	702.7096	702.7096	.0000	230.3388
.5E-07	APOGEE	19.8297	19.8195	339.8931	-4.937	-4.942	-.388	-.372	1041.9096	1041.9096	.0000	341.5344
.5E-07	WAVE REV	19.8297	19.8195	339.8931	-4.937	-4.942	-.388	-.372	1041.9096	1041.9096	.0000	341.5344
.1E-06		17.9660	17.9568	377.5546	-4.964	-4.970	-.953	-5.329	1157.1096	1157.1096	.0000	379.3655
.7E-07	MIN DIST	13.0423	13.0348	450.1421	-4.932	-4.941	-2.025	-.001	1379.8701	1379.8701	.0000	452.3210
.7E-07	MAX HOPS	13.0423	13.0348	450.1421	-4.932	-4.941	-2.025	-.001	1379.8701	1379.8701	.0000	452.3210

THIS RAY CALCULATION TOOK .903 SEC

ELEVATION ANGLE OF TRANSMISSION = 20.0000 DEG

-.7E-14	XMTR	13.0440	13.0000	.0000				20.000	.0000	.0000	.0000	.0000
-.4E-06	APOGEE	30.8011	30.7734	83.6250	-5.591	-5.592	11.572	-.390	256.1000	256.0999	.0000	86.3124
-.4E-06	WAVE REV	30.8011	30.7734	83.6250	-5.591	-5.592	11.572	-.390	256.1000	256.0999	.0000	86.3124
-.4E-06		30.2160	30.1900	96.9373	-5.605	-5.605	9.576	-5.195	294.5000	294.4999	.0000	99.7046
.1E-06	RCVR	5.0170	5.0000	194.6869	-5.322	-5.324	-3.233	-12.129	598.2812	598.2810	.0000	201.1925
.2E-06	GRND REF	.0152	.0000	221.5510	-4.860	-4.863	-4.358	10.662	678.8281	678.8280	.0000	228.5831
.1E-06	RCVR	5.0137	5.0000	248.4283	-4.498	-4.500	-2.966	12.121	759.3772	759.3771	.0000	255.9751
.3E-08		30.1249	30.1149	345.1581	-4.646	-4.651	1.271	5.571	1060.1772	1060.1771	.0000	356.4343
.2E-08	APOGEE	30.7976	30.7881	361.7989	-4.694	-4.699	1.172	-.437	1108.1772	1108.1771	.0000	373.1773
.2E-08	WAVE REV	30.7976	30.7881	361.7989	-4.694	-4.699	1.172	-.437	1108.1772	1108.1771	.0000	373.1773
.3E-06		11.0982	11.0907	449.1534	-4.851	-4.860	-2.268	-18.731	1376.9772	1376.9772	.0000	463.5327
.5E-06	RCVR	5.0070	5.0000	472.7741	-4.790	-4.799	-3.098	-12.126	1450.0603	1450.0603	.0000	487.9710
.5E-06	MAX HOPS	5.0070	5.0000	472.7741	-4.790	-4.799	-3.098	-12.126	1450.0603	1450.0603	.0000	487.9710

THIS RAY CALCULATION TOOK 1.968 SEC

AZIMUTH ANGLE OF TRANSMISSION = 45.000000 DEG TRANSMITTER LATITUDE = 1.798926 DEG FREQUENCY = .050000 HZ
ELEVATION ANGLE OF TRANSMISSION = 40.000000 DEG TRANSMITTER LONGITUDE = .000000 DEG SINGLE STEP ERROR = 1.000000E-06

ERROR	EVENT	ELEVATION ABOVE SEA LEVEL KM	ELEVATION ABOVE TERRAIN KM	RANGE KM	AZIMUTH DEVIATION XMTR DEG	AZIMUTH DEVIATION LOCAL DEG	ELEVATION ANGLE XMTR DEG	ELEVATION ANGLE LOCAL DEG	PULSE TIME SEC	PHASE TIME SEC	ABSORPTION DB	PATH LENGTH KM
-.7E-14	XMTR	13.0440	13.0000	.0000				40.000	.0000	.0000	.0000	.0000
.6E-07		43.9246	43.8912	48.7564	-6.654	-6.654	32.014	28.195	169.7000	169.7000	.0000	58.0370
-.6E-06		88.4749	88.4526	134.7274	-7.168	-7.222	28.443	39.168	438.5000	438.4999	.0001	155.8842
-.7E-06	APOGEE	104.2225	104.2036	175.8021	-7.340	-5.047	26.407	-1.194	555.3000	555.2998	.0016	201.2805
-.7E-06	WAVE REV	104.2225	104.2036	175.8021	-7.340	-5.047	26.407	-1.194	555.3000	555.2998	.0016	201.2805
-.8E-06		101.0975	101.0797	192.8408	-7.530	-3.080	23.481	-21.450	596.9000	596.8998	.0024	218.9577
.6E-06		62.3998	62.3857	263.9313	-8.907	-3.505	9.343	-26.431	827.3000	827.2997	.0031	301.0331
-.1E-06		15.0673	15.0566	359.8298	-9.527	-4.128	-1.297	-39.337	1124.9000	1124.8997	.0031	409.0699
.5E-06	RCVR	5.0103	5.0000	374.1414	-9.536	-4.137	-2.911	-35.775	1178.0787	1178.0784	.0031	426.5820
.4E-06	GRND REF	.0101	.0000	381.4687	-9.488	-4.092	-3.670	34.996	1204.2519	1204.2517	.0031	435.4618
.4E-06	RCVR	5.0099	5.0000	388.7966	-9.442	-4.046	-2.930	35.772	1230.4251	1230.4249	.0031	444.3416
.2E-05		48.1493	48.1410	462.4781	-9.681	-4.287	2.238	23.933	1478.4251	1478.4251	.0031	530.3500
.7E-06		93.4113	93.4046	554.2751	-10.088	-4.698	5.685	35.429	1763.2251	1763.2256	.0034	633.9752
.5E-06	APOGEE	104.7397	104.7335	592.3256	-10.201	-4.812	6.050	-1.123	1867.2251	1867.2256	.0050	674.8220
.5E-06	WAVE REV	104.7397	104.7335	592.3256	-10.201	-4.812	6.050	-1.123	1867.2251	1867.2256	.0050	674.8220
.4E-06		98.6801	98.6742	618.2918	-10.256	-4.869	5.031	-27.897	1934.4251	1934.4256	.0063	702.0891
.2E-05		57.6827	57.6776	694.9466	-10.509	-5.125	.526	-24.084	2180.8251	2180.8258	.0068	790.0545
-.2E-06		8.8419	8.8375	788.1424	-10.575	-5.196	-3.849	-37.480	2478.4251	2478.4263	.0068	896.3113
.2E-05	RCVR	5.0043	5.0000	793.7415	-10.564	-5.185	-4.148	-35.793	2498.7527	2498.7538	.0068	903.1062
.2E-05	MAX HOPS	5.0043	5.0000	793.7415	-10.564	-5.185	-4.148	-35.793	2498.7527	2498.7538	.0068	903.1062

THIS RAY CALCULATION TOOK 2.999 SEC

ELEVATION ANGLE OF TRANSMISSION = 60.0000 DEG

ERROR	EVENT	ELEVATION ABOVE SEA LEVEL KM	ELEVATION ABOVE TERRAIN KM	RANGE KM	AZIMUTH DEVIATION XMTR DEG	AZIMUTH DEVIATION LOCAL DEG	ELEVATION ANGLE XMTR DEG	ELEVATION ANGLE LOCAL DEG	PULSE TIME SEC	PHASE TIME SEC	ABSORPTION DB	PATH LENGTH KM
-.7E-14	XMTR	13.0440	13.0000	.0000				60.000	.0000	.0000	.0000	.0000
-.2E-06		46.3009	46.2626	25.0438	-9.587	-9.587	52.778	54.898	124.9000	124.9000	.0000	41.7284
-.4E-08		89.3597	89.3273	60.1377	-10.437	-10.438	51.268	60.151	284.9000	284.8999	.0001	97.5354
-.5E-06		132.3433	132.3186	116.6785	-9.041	-9.240	44.786	21.235	444.9000	444.8998	.0357	170.6495
-.4E-06	APOGEE	144.8719	144.8545	199.2131	-7.589	-1.829	32.272	-.221	572.1000	572.0997	.5014	256.4120
-.4E-06	WAVE REV	144.8719	144.8545	199.2131	-7.589	-1.829	32.272	-.221	572.1000	572.0997	.5014	256.4120
-.4E-06		144.7473	144.7304	205.5626	-7.656	-1.487	31.401	-2.205	581.7000	581.6997	.5557	262.9118
-.4E-06		98.5307	98.5184	308.1062	-8.967	-2.280	13.990	-57.611	792.9000	792.8996	.9118	380.9940
.9E-05		42.8401	42.8290	352.0479	-9.918	-3.232	3.232	-56.821	997.7000	997.7001	.9121	452.4670
.3E-05	RCVR	5.0104	5.0000	378.8921	-10.259	-3.574	-2.917	-59.417	1138.5475	1138.5483	.9121	498.9910
.9E-06	GRND REF	.0103	.0000	382.0800	-10.249	-3.572	-3.669	58.546	1156.1759	1156.1768	.9121	504.9226
.1E-05	RCVR	5.0102	5.0000	385.2681	-10.240	-3.563	-2.925	59.414	1173.8043	1173.8052	.9121	510.8543
.5E-05		55.5496	55.5403	422.6504	-10.644	-3.967	3.809	55.842	1359.4043	1359.4055	.9121	573.9536
-.2E-05		111.6528	111.6443	468.6814	-11.217	-4.542	9.657	46.144	1564.2043	1564.2058	.9138	647.1544
-.2E-05	APOGEE	144.7403	144.7336	567.1420	-11.193	-4.522	10.360	-1.201	1743.4043	1743.4054	1.3398	755.3699
-.2E-05	WAVE REV	144.7403	144.7336	567.1420	-11.193	-4.522	10.360	-1.201	1743.4043	1743.4054	1.3398	755.3699
-.2E-05		129.1384	129.1325	632.9859	-11.142	-4.473	7.426	-28.754	1852.2043	1852.2051	1.6910	824.9860
.2E-05		71.4679	71.4625	692.1644	-11.446	-4.779	1.676	-58.729	2060.2043	2060.2046	1.7111	909.4548
.6E-06		10.1142	10.1093	739.1827	-11.686	-5.021	-3.551	-60.623	2284.2043	2284.2055	1.7111	987.0659
.4E-05	RCVR	5.0049	5.0000	742.5637	-11.689	-5.024	-3.958	-59.423	2302.9500	2302.9512	1.7111	993.1950
.4E-05	MAX HOPS	5.0049	5.0000	742.5637	-11.689	-5.024	-3.958	-59.423	2302.9500	2302.9512	1.7111	993.1950

THIS RAY CALCULATION TOOK 3.219 SEC

AZIMUTH ANGLE OF TRANSMISSION = 45.000000 DEG   TRANSMITTER LATITUDE = 1.798926 DEG   FREQUENCY = .050000 HZ
ELEVATION ANGLE OF TRANSMISSION = 80.000000 DEG   TRANSMITTER LONGITUDE = .000000 DEG   SINGLE STEP ERROR = 1.000000E-06

ERROR	EVENT	ELEVATION ABOVE SEA LEVEL KM	ELEVATION ABOVE TERRAIN KM	RANGE KM	AZIMUTH DEVIATION XMTR DEG	AZIMUTH DEVIATION LOCAL DEG	ELEVATION ANGLE XMTR DEG	ELEVATION ANGLE LOCAL DEG	PULSE TIME SEC	PHASE TIME SEC	ABSORPTION DB	PATH LENGTH KM
.0E+00	XMTR	13.0440	13.0000	.0000				80.000	.0000	.0000	.0000	.0000
-.7E-07		35.1397	35.0968	6.6778	-20.387	-20.387	73.094	79.301	73.7000	73.7000	.0000	23.0919
-.9E-06		77.2842	77.2436	21.0030	-20.997	-20.997	71.681	79.854	203.3000	203.2999	.0000	67.6482
-.1E-05		108.2005	108.1614	32.1000	-21.928	-21.929	71.049	76.820	302.5000	302.4998	.0009	100.5561
-.1E-05		147.3742	147.3380	48.3230	-19.107	-19.110	69.768	69.185	382.5000	382.4998	.1114	143.1468
-.1E-05		198.2845	198.2527	73.8397	-15.606	-15.609	67.609	64.942	462.5000	462.4997	48.2402	200.5408
-.1E-05	EXTINC	224.5590	224.5294	88.0081	-14.390	-14.394	66.634	64.191	500.9000	500.8997	1076.3153	230.6701

THIS RAY CALCULATION TOOK .964 SEC

ELEVATION ANGLE OF TRANSMISSION = 100.0000 DEG

ERROR	EVENT	ELEVATION ABOVE SEA LEVEL KM	ELEVATION ABOVE TERRAIN KM	RANGE KM	AZIMUTH DEVIATION XMTR DEG	AZIMUTH DEVIATION LOCAL DEG	ELEVATION ANGLE XMTR DEG	ELEVATION ANGLE LOCAL DEG	PULSE TIME SEC	PHASE TIME SEC	ABSORPTION DB	PATH LENGTH KM
-.7E-14	XMTR	13.0440	13.0000	.0000				80.000	.0000	.0000	.0000	.0000
.2E-12	MAX LONG	13.3109	13.2669	.0332	-135.240	44.760	82.899	80.016	.9000	.9000	.0000	.2690
-.8E-07		32.0143	31.9693	2.3953	-124.524	55.476	82.768	79.618	63.7000	63.7000	.0000	19.1228
-.5E-06		70.8034	70.7559	7.8116	-122.131	57.869	82.213	79.622	182.1000	182.1000	.0000	58.2976
-.1E-05		103.9528	103.9032	12.5474	-114.124	65.875	82.014	78.188	290.9000	290.8999	.0005	91.8275
-.1E-05	MAX LONG	110.6394	110.5893	13.5646	-115.247	64.752	81.950	76.537	308.5000	308.4998	.0012	98.5992
-.1E-05		140.9497	140.8962	19.2197	-127.481	52.517	81.266	70.665	370.9000	370.8998	.0511	129.6767
-.1E-05		190.9721	190.9095	33.5431	-144.932	35.066	79.006	66.040	450.9000	450.8997	20.1684	182.4295
-.1E-05	EXTINC	224.9996	224.9286	45.5105	-151.291	28.707	77.457	65.030	500.5000	500.4996	1126.0582	218.9079

THIS RAY CALCULATION TOOK .977 SEC

ELEVATION ANGLE OF TRANSMISSION = 120.0000 DEG

ERROR	EVENT	ELEVATION ABOVE SEA LEVEL KM	ELEVATION ABOVE TERRAIN KM	RANGE KM	AZIMUTH DEVIATION XMTR DEG	AZIMUTH DEVIATION LOCAL DEG	ELEVATION ANGLE XMTR DEG	ELEVATION ANGLE LOCAL DEG	PULSE TIME SEC	PHASE TIME SEC	ABSORPTION DB	PATH LENGTH KM
-.7E-14	XMTR	13.0440	13.0000	.0000				60.000	.0000	.0000	.0000	.0000
-.1E-06		46.1915	46.1409	16.1398	-165.235	14.765	63.861	56.548	123.3000	123.3000	.0000	36.9103
.4E-06		90.1163	90.0541	36.7256	-162.825	17.175	64.177	62.055	283.3000	283.2999	.0001	85.5976
.4E-06		119.2382	119.1634	54.4176	-163.537	16.462	62.382	41.494	382.5000	382.4999	.0047	120.1510
.6E-06		152.4306	152.2250	135.0070	-169.420	5.792	44.937	3.174	548.9000	548.9000	.9310	211.3774
.2E-06	APOGEE	152.7178	152.4727	145.0531	-169.378	2.943	42.894	-.188	564.9000	564.9000	1.1915	221.6716
.2E-06	WAVE REV	152.7178	152.4727	145.0531	-169.378	2.943	42.894	-.188	564.9000	564.9000	1.1915	221.6716
-.1E-05		111.3236	109.6405	243.7566	-164.096	-3.210	20.668	-48.079	766.5000	766.5000	2.3500	335.7976
.6E-06	MAX LONG	93.6766	92.4536	253.9614	-163.165	-2.285	16.333	-61.003	827.3000	827.2999	2.3514	356.7367
-.2E-05	MAX LONG	79.4894	78.5126	259.9170	-162.436	-1.561	13.070	-61.038	881.7000	881.6999	2.3515	372.5074
.8E-05		54.7302	54.1010	272.6007	-161.359	-.490	7.422	-56.142	971.3000	971.3000	2.3516	400.8575
-.6E-07	RCVR	5.3003	5.0000	299.0948	-159.875	.978	-2.826	-57.476	1153.6795	1153.6806	2.3516	457.5233
-.7E-06	GRND REF	.2797	.0000	302.3604	-159.847	.553	-3.774	54.094	1171.9866	1171.9878	2.3516	463.5177
-.5E-06	RCVR	5.2607	5.0000	305.6826	-159.813	.580	-2.831	56.842	1190.2861	1190.2873	2.3516	469.5109
.7E-05		56.6194	56.4724	334.2104	-158.498	1.889	5.884	55.667	1380.6861	1380.6876	2.3516	528.8274
-.2E-05	MAX LONG	81.0713	80.9539	347.1532	-157.769	2.617	9.444	60.760	1470.2861	1470.2882	2.3516	556.9184
-.4E-06	MAX LONG	94.3487	94.2425	353.1353	-157.340	3.046	11.270	60.149	1521.4861	1521.4881	2.3518	571.7558
-.3E-05		112.9029	112.8138	365.1172	-156.847	3.538	13.507	45.584	1585.4861	1585.4879	2.3536	594.2539
-.3E-05	APOGEE	147.4966	147.4612	453.9279	-156.085	4.301	14.257	-.783	1771.0861	1771.0874	2.9599	694.3670
-.3E-05	WAVE REV	147.4966	147.4612	453.9279	-156.085	4.301	14.257	-.783	1771.0861	1771.0874	2.9599	694.3670
-.2E-05		134.5112	134.4879	509.7279	-155.833	4.554	10.956	-24.695	1873.4861	1873.4872	3.4692	753.3817
-.9E-06	MAX LONG	93.3241	93.3063	551.2271	-155.276	5.113	5.734	-60.529	2017.4861	2017.4867	3.5120	813.9165
-.3E-05	MAX LONG	80.0572	80.0402	557.3032	-155.031	5.358	4.297	-60.571	2068.6861	2068.6867	3.5121	828.7383
-.9E-06		76.6838	76.6669	558.9313	-154.973	5.416	3.933	-59.875	2081.4861	2081.4866	3.5121	832.5357
.1E-05		13.1140	13.1003	594.5164	-154.227	6.163	-2.667	-59.566	2315.0861	2315.0875	3.5121	905.9410
-.2E-05	RCVR	5.0134	5.0000	599.1689	-154.193	6.198	-3.461	-56.730	2345.7311	2345.7326	3.5121	915.2955
-.2E-05	MAX HOPS	5.0134	5.0000	599.1689	-154.193	6.198	-3.461	-56.730	2345.7311	2345.7326	3.5121	915.2955

THIS RAY CALCULATION TOOK 3.209 SEC

AZIMUTH ANGLE OF TRANSMISSION = 45.000000 DEG TRANSMITTER LATITUDE = 1.798926 DEG FREQUENCY = .050000 HZ
ELEVATION ANGLE OF TRANSMISSION = 140.000000 DEG TRANSMITTER LONGITUDE = .000000 DEG SINGLE STEP ERROR = 1.000000E-06

ERROR	EVENT	ELEVATION ABOVE SEA LEVEL KM	ELEVATION ABOVE TERRAIN KM	RANGE KM	AZIMUTH DEVIATION XMTR DEG	AZIMUTH DEVIATION LOCAL DEG	ELEVATION ANGLE XMTR DEG	ELEVATION ANGLE LOCAL DEG	PULSE TIME SEC	PHASE TIME SEC	ABSORPTION DB	PATH LENGTH KM
-.7E-14	XMTR	13.0440	13.0000	.0000				40.000	.0000	.0000	.0000	.0000
-.2E-06		41.9304	41.8733	30.7729	-170.963	9.037	42.927	34.744	147.3000	147.3000	.0000	42.3367
-.2E-06		72.7148	72.6319	68.5015	-170.511	9.475	40.560	40.574	307.3000	307.2999	.0000	91.3556
-.5E-06		105.4267	105.3047	99.5121	-169.250	10.185	42.160	27.018	467.3000	467.2999	.0010	137.0069
-.2E-06	APOGEE	111.7581	111.5888	123.0044	-169.527	7.625	37.922	-.231	534.5000	534.4999	.0048	162.0291
-.2E-06	WAVE REV	111.7581	111.5888	123.0044	-169.527	7.625	37.922	-.231	534.5000	534.4999	.0048	162.0291
-.7E-06		99.0190	98.7170	154.3398	-168.405	-1.721	28.212	-39.256	627.3000	627.2999	.0091	197.3682
.4E-05		50.6802	49.2224	203.6148	-161.833	-10.394	9.505	-32.617	854.5000	854.5001	.0095	270.1590
.2E-05	RCVR	5.8098	5.0000	255.5846	-157.368	-6.592	-2.768	-36.018	1083.9361	1083.9370	.0095	341.4045
.2E-05	GRND REF	.6338	.0000	263.3761	-157.096	-6.666	-3.879	27.499	1112.8739	1112.8748	.0095	350.8503
.1E-05	RCVR	5.5009	5.0000	271.3368	-156.827	-6.406	-2.810	33.954	1141.7055	1141.7065	.0095	360.2767
.1E-05		51.4803	51.3472	329.9398	-153.800	-3.387	5.126	29.986	1388.1055	1388.1070	.0095	436.8222
.2E-05	MAX LONG	77.5366	77.4586	365.2769	-152.219	-1.808	8.298	40.528	1528.9055	1528.9074	.0096	482.1444
.5E-06	MAX LONG	96.7313	96.6704	384.5933	-151.260	-.849	10.441	38.317	1624.9055	1624.9075	.0098	510.2986
.2E-07		100.4016	100.3440	389.2695	-151.074	-.663	10.786	32.940	1644.1055	1644.1075	.0100	516.4371
.2E-06	APOGEE	109.2373	109.1943	417.3020	-150.307	.103	10.980	-.377	1727.3055	1727.3075	.0129	547.2455
.2E-06	WAVE REV	109.2373	109.1943	417.3020	-150.307	.103	10.980	-.377	1727.3055	1727.3075	.0129	547.2455
.8E-06		98.8690	98.8362	446.8125	-149.588	.822	8.767	-35.440	1816.9055	1816.9075	.0158	580.0213
.4E-06	MAX LONG	95.7525	95.7208	450.5518	-149.468	.942	8.285	-39.400	1832.9055	1832.9075	.0159	585.0253
.9E-06	MAX LONG	76.5533	76.5264	470.0849	-148.799	1.612	5.524	-40.094	1928.9055	1928.9075	.0161	613.1416
.1E-05		52.8328	52.8115	502.7766	-148.075	2.336	2.238	-30.029	2056.9055	2056.9078	.0162	654.3369
.1E-05	RCVR	5.0143	5.0000	565.8781	-147.184	3.229	-3.356	-33.478	2313.8622	2313.8650	.0162	734.3758
.1E-05	MAX HOPS	5.0143	5.0000	565.8781	-147.184	3.229	-3.356	-33.478	2313.8622	2313.8650	.0162	734.3758

THIS RAY CALCULATION TOOK 2.778 SEC

***** H A R P A *****
HAMILTONIAN ACOUSTIC RAY-TRACING PROGRAM FOR THE ATMOSPHERE

BY
R. M. JONES, J. P. RILEY AND T. M. GEORGES
WAVE PROPAGATION LABORATORY
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
BOULDER, COLORADO 80303

RUN SET NUMBER 2

ATMOSPHERIC MODEL ID — S03

ATMOSPHERIC MODEL DESCRIPTION — SAMPLE CASE FOR HARPA DOCUMENTATION REV. 2-10-86

MODEL TYPE	SUBROUTINE NAME	DATA SET ID	DESCRIPTION
DISPERSION RELATION	AWWWL		ACOUSTIC WAVE *** WITH WIND ****** WITH LOSSES
BACKGROUND WIND VELOCITY	ULOGZ2	3.00	LOGARITHMIC EASTWARD WIND PROFILE, U*=.5 M/S, Z0=1 KM
WIND VELOCITY PERTURBATION	NPWIND	.00	NO WIND PERTURBATION
BACKGROUND SOUND SPEED	GAMRTDM	.00	SOUND SPEED IN TERMS OF TEMPERATURE MODEL
SOUND SPEED PERTURBATION	CBLOB2	2.00	50% INCREASE IN SQ. SOUND SPEED AT 125KM HT, 335KM N, 125KM E
BACKGROUND TEMPERATURE	TTANH5	1.00	U.S. STANDARD ATMOSPHERE 1962 TEMPERATURE PROFILE
TEMPERATURE PERTURBATION	TBLOB2	2.00	50% CYLINDRICAL INCREASE IN TEMPERATURE AT 105KM N., 105 KM W
MOLECULAR WEIGHT	MCONST	29.00	MOLECULAR WEIGHT = 29
BACKGROUND TERRAIN	GLORENZ	2.00	RIDGE 2-KM HIGH, 30-KM WIDE ALONG EQUATOR
TERRAIN PERTURBATION	NPTERR	.00	NO TERRAIN PERTURBATION
BACKGROUND VISCOSITY/ THERMAL CONDUCTIVITY	MUARDC	1.00	ARDC VISCOSITY AND THERMAL CONDUCTIVITY MODEL
VISCOSITY/THERMAL CONDUCTIVITY PERTURBATION	NPABSR	.00	NO VISCOSITY/CONDUCTIVITY PERTURBATION
BACKGROUND PRESSURE	PEXP	1.00	EXPONENTIAL PRESSURE MODEL, SCALE HEIGHT = 8.5 KM
PRESSURE PERTURBATION	NPPRES	.00	NO PRESSURE PERTURBATION
RECEIVER SURFACE	RTERR	.00	RECEIVER SURFACE 5 KM ABOVE TERRAIN

INPUT DATA FILE FOR RUN SET NUMBER 2

```
S03-2    SAMPLE CASE FOR HARPA DOCUMENTATION        REV. 2-10-86
 71        0.            NUMBER OF INTEGRATION STEPS PER PRINT
 72        0.            OUTPUT RAYSETS  (1=YES; 0=NO)
 73        1.            DIAGNOSTIC PRINTOUT (1=YES; 0=NO)
 81        2.            RAYPLOT PROJECTION (1=VERT; 2=HORIZ) PLANE
 82        3.            PLOT-EXPANSION FACTOR
  0                      **********  END OF RUN SET NUMBER 2  ***********
```

INITIAL VALUES FOR THE W ARRAY
ONLY NONZERO VALUES PRINTED — ALL ANGLES IN RADIANS

N	W(N)	
1	.637000000000000E+04	
3	.130440097799511E+02	
4	.313971742543171E-01	
7	.314159265358979E+00	
11	.785398163397446E+00	
15	-.349065850398864E+00	
16	.244346095279205E+01	
17	.349065850398864E+00	
20	.500000000000000E+01	
22	.300000000000000E+01	
23	.100000000000000E+04	
24	.157079632679490E+01	
26	.500000000000000E+03	
27	-.100000000000000E+01	
28	.100000000000000E+04	
29	.100000000000000E+03	
33	.999999000000000E+03	
41	.300000000000000E+01	
42	.999999999999997E-06	
43	.500000000000000E+02	
44	.100000000000000E+00	
45	.100000000000000E+03	
46	.100000000000000E-07	
47	.500000000000000E+00	
57	.200000000000000E+01	
58	.200000000000000E+01	
60	.200000000000000E+01	
71	.999999999999999E+30	INPUT OVERRIDDEN
73	.100000000000000E+01	
75	.150000000000000E+00	
77	.570000000000000E+02	
81	.200000000000000E+01	
82	.300000000000000E+01	
83	-.156985871271585E-01	
84	-.470957613814755E-01	
85	.784929356357926E-01	
86	.470957613814755E-01	
87	.156985871271585E-01	
89	.300000000000000E+03	
96	.100000000000000E+03	
100	.600000000000000E+01	
102	.300000000000000E+01	
103	.500000000000000E-02	
104	.350000000000000E+00	
105	.100000000000000E+01	
150	.100000000000000E+01	

N	W(N)
175	.200000000000000E+01
177	.200000000000000E+01
178	.500000000000000E+00
179	.125000000000000E+03
180	.525902668759810E-01
181	.196232339089482E-01
182	.250000000000000E+02
183	.784929356357927E-02
184	.392464678178964E-02
200	.700000000000000E+01
202	.100000000000000E+01
225	.200000000000000E+01
227	.200000000000000E+01
228	.500000000000000E+00
230	.164835164835164E-01
231	-.164835164835164E-01
233	.784929356357927E-02
234	.392464678178964E-02
250	.100000000000000E+01
252	.290000000000000E+02
253	.290000000000000E+02
275	.200000000000000E+01
300	.400000000000000E+01
302	.200000000000000E+01
303	.200000000000000E+01
305	.470957613814754E-02
500	.100000000000000E+01
502	.100000000000000E+01
503	.145800000000000E-05
504	.110400000000000E+03
505	.191000000000000E+01
550	.100000000000000E+01
552	.100000000000000E+01
553	.101328000000000E+06
554	.850000000000000E+01

AZIMUTH ANGLE OF TRANSMISSION = 45.000000 DEG TRANSMITTER LATITUDE = 1.798926 DEG FREQUENCY = .050000 HZ
ELEVATION ANGLE OF TRANSMISSION = -20.000000 DEG TRANSMITTER LONGITUDE = .000000 DEG SINGLE STEP ERROR = 1.000000E-06

ERROR	EVENT	ELEVATION ABOVE SEA LEVEL KM	ELEVATION ABOVE TERRAIN KM	RANGE KM	AZIMUTH DEVIATION XMTR DEG	AZIMUTH DEVIATION LOCAL DEG	ELEVATION ANGLE XMTR DEG	ELEVATION ANGLE LOCAL DEG	PULSE TIME SEC	PHASE TIME SEC	ABSORPTION DB	PATH LENGTH KM
-.1E-13	XMTR	13.0440	13.0000	.0000				-20.000	.0000	.0000	.0000	.0000
.2E-06	BACK UP0	4.7586	4.7222	30.8625	-3.791	-3.790	-15.146	-11.856	96.1000	96.1000	.0000	32.0196
.2E-06	BACK UP1	5.0367	5.0000	29.4888	-3.840	-3.838	-15.304	-12.169	91.9614	91.9614	.0000	30.6168
.2E-06	RCVR	5.0367	5.0000	29.4887	-3.840	-3.838	-15.304	-12.169	91.9612	91.9612	.0000	30.6167
.2E-06	BACK UP0	-.4292	-.4600	58.7368	-2.616	-2.612	-13.171	-10.714	179.5612	179.5613	.0000	60.3988
.2E-06	BACK UP1	.0313	.0000	56.3108	-2.735	-2.731	-13.252	-10.703	172.2877	172.2877	.0000	57.9268
.2E-06	GRND REF	.0313	.0000	56.3108	-2.735	-2.735	-13.252	10.682	172.2877	172.2877	.0000	57.9268
.8E-07	BACK UP0	5.0597	5.0326	83.3006	-2.344	-2.341	-5.842	12.186	253.0877	253.0877	.0000	85.3987
.8E-07	BACK UP1	5.0270	5.0000	83.1414	-2.343	-2.340	-5.874	12.148	252.6076	252.6076	.0000	85.2361
.8E-07	RCVR	5.0270	5.0000	83.1414	-2.343	-2.340	-5.874	12.148	252.6076	252.6076	.0000	85.2361
-.1E-06	APOGEE	30.8025	30.7860	196.4513	-3.954	-3.953	4.264	-.461	601.4076	601.4076	.0000	202.4308
-.1E-06	WAVE REV	30.8025	30.7860	196.4513	-3.954	-3.953	4.264	-.461	601.4076	601.4076	.0000	202.4308
.3E-06	BACK UP0	4.7476	4.7365	308.6340	-4.367	-4.368	-2.925	-11.832	947.0076	947.0076	.0000	318.4730
.3E-06	BACK UP1	5.0112	5.0000	307.3289	-4.374	-4.375	-2.877	-12.129	943.0764	943.0765	.0000	317.1400
.3E-06	RCVR	5.0112	5.0000	307.3288	-4.374	-4.375	-2.877	-12.129	943.0763	943.0763	.0000	317.1400
.3E-06	MAX HOPS	5.0112	5.0000	307.3288	-4.374	-4.375	-2.877	-12.129	943.0763	943.0763	.0000	317.1400

THIS RAY CALCULATION TOOK 2.009 SEC

ELEVATION ANGLE OF TRANSMISSION = .0000 DEG

ERROR	EVENT	ELEVATION ABOVE SEA LEVEL KM	ELEVATION ABOVE TERRAIN KM	RANGE KM	AZIMUTH DEVIATION XMTR DEG	AZIMUTH DEVIATION LOCAL DEG	ELEVATION ANGLE XMTR DEG	ELEVATION ANGLE LOCAL DEG	PULSE TIME SEC	PHASE TIME SEC	ABSORPTION DB	PATH LENGTH KM
-.7E-14	XMTR	13.0440	13.0000	.0000				.000	.0000	.0000	.0000	.0000
.1E-07	APOGEE	19.8306	19.8069	114.8628	-4.945	-4.945	2.856	-.376	352.1000	352.1000	.0000	115.4148
.1E-07	WAVE REV	19.8306	19.8069	114.8628	-4.945	-4.945	2.856	-.376	352.1000	352.1000	.0000	115.4148
.4E-07	GRAZE 1	13.0538	13.0389	227.5993	-4.933	-4.935	-1.021	.417	697.7000	697.7000	.0000	228.6941
.4E-07	GRAZE 1	13.0453	13.0303	225.0474	-4.938	-4.939	-1.012	-.003	689.9110	689.9110	.0000	226.1369
.4E-07	GRAZE 1	13.0453	13.0303	225.0470	-4.938	-4.939	-1.012	-.003	689.9096	689.9096	.0000	226.1364
.4E-07	MIN DIST	13.0453	13.0303	225.0470	-4.938	-4.939	-1.012	-.003	689.9096	689.9096	.0000	226.1364
.4E-07	MIN DIST	13.0453	13.0303	225.0470	-4.938	-4.939	-1.012	-.003	689.9096	689.9096	.0000	226.1364
.3E-07	WAVE REV	13.0683	13.0535	229.2405	-4.930	-4.932	-1.025	.687	702.7096	702.7096	.0000	230.3388
.5E-07	APOGEE	19.8297	19.8195	339.8931	-4.937	-4.942	-.388	-.372	1041.9096	1041.9096	.0000	341.5344
.5E-07	WAVE REV	19.8297	19.8195	339.8931	-4.937	-4.942	-.388	-.372	1041.9096	1041.9096	.0000	341.5344
.7E-07	GRAZE 1	13.0505	13.0431	452.6454	-4.930	-4.939	-2.035	.411	1387.5096	1387.5097	.0000	454.8295
.7E-07	GRAZE 1	13.0423	13.0348	450.1425	-4.932	-4.941	-2.025	-.001	1379.8714	1379.8714	.0000	452.3214
.7E-07	GRAZE 1	13.0423	13.0348	450.1421	-4.932	-4.941	-2.025	-.001	1379.8701	1379.8701	.0000	452.3210
.7E-07	MIN DIST	13.0423	13.0348	450.1421	-4.932	-4.941	-2.025	-.001	1379.8701	1379.8701	.0000	452.3210
.7E-07	MAX HOPS	13.0423	13.0348	450.1421	-4.932	-4.941	-2.025	-.001	1379.8701	1379.8701	.0000	452.3210

THIS RAY CALCULATION TOOK .908 SEC

S03-2 SAMPLE CASE FOR HARPA DOCUMENTATION REV. 2-10-86 86/03/21. 10.53.06. PAGE 15

AZIMUTH ANGLE OF TRANSMISSION = 45.000000 DEG TRANSMITTER LATITUDE = 1.798926 DEG FREQUENCY = .050000 HZ
ELEVATION ANGLE OF TRANSMISSION = 20.000000 DEG TRANSMITTER LONGITUDE = .000000 DEG SINGLE STEP ERROR = 1.000000E-06

ERROR	EVENT	ELEVATION ABOVE SEA LEVEL KM	ELEVATION ABOVE TERRAIN KM	RANGE KM	AZIMUTH DEVIATION XMTR DEG	AZIMUTH DEVIATION LOCAL DEG	ELEVATION ANGLE XMTR DEG	ELEVATION ANGLE LOCAL DEG	PULSE TIME SEC	PHASE TIME SEC	ABSORPTION DB	PATH LENGTH KM
-.7E-14	XMTR	13.0440	13.0000	.0000				20.000	.0000	.0000	.0000	.0000
-.4E-06	APOGEE	30.8011	30.7734	83.6250	-5.591	-5.592	11.572	-.390	256.1000	256.0999	.0000	86.3124
-.4E-06	WAVE REV	30.8011	30.7734	83.6250	-5.591	-5.592	11.572	-.390	256.1000	256.0999	.0000	86.3124
.1E-06	BACK UP0	4.7875	4.7706	195.8209	-5.307	-5.309	-3.291	-11.869	601.7000	601.6998	.0000	202.3515
.1E-06	BACK UP1	5.0170	5.0000	194.6869	-5.322	-5.324	-3.233	-12.129	598.2812	598.2811	.0000	201.1925
.1E-06	RCVR	5.0170	5.0000	194.6869	-5.322	-5.324	-3.233	-12.129	598.2812	598.2810	.0000	201.1925
.2E-06	BACK UP0	-.4426	-.4577	223.9645	-4.806	-4.807	-4.450	-10.681	686.0812	686.0810	.0000	231.0486
.2E-06	BACK UP1	.0152	.0000	221.5510	-4.860	-4.862	-4.358	-10.669	678.8281	678.8280	.0000	228.5831
.2E-06	GRND REF	.0152	.0000	221.5510	-4.860	-4.863	-4.358	10.662	678.8281	678.8280	.0000	228.5831
.1E-06	BACK UP0	5.0443	5.0306	248.5778	-4.497	-4.499	-2.958	12.156	759.8281	759.8280	.0000	256.1279
.1E-06	BACK UP1	5.0137	5.0000	248.4283	-4.498	-4.500	-2.966	12.121	759.3772	759.3771	.0000	255.9751
.1E-06	RCVR	5.0137	5.0000	248.4283	-4.498	-4.500	-2.966	12.121	759.3772	759.3771	.0000	255.9751
.2E-08	APOGEE	30.7976	30.7881	361.7989	-4.694	-4.699	1.172	-.437	1108.1772	1108.1771	.0000	373.1773
.2E-08	WAVE REV	30.7976	30.7881	361.7989	-4.694	-4.699	1.172	-.437	1108.1772	1108.1771	.0000	373.1773
.5E-06	BACK UP0	4.7578	4.7507	474.0078	-4.784	-4.794	-3.131	-11.845	1453.7772	1453.7772	.0000	489.2314
.5E-06	BACK UP1	5.0070	5.0000	472.7741	-4.790	-4.799	-3.098	-12.126	1450.0604	1450.0604	.0000	487.9710
.5E-06	RCVR	5.0070	5.0000	472.7741	-4.790	-4.799	-3.098	-12.126	1450.0603	1450.0603	.0000	487.9710
.5E-06	MAX HOPS	5.0070	5.0000	472.7741	-4.790	-4.799	-3.098	-12.126	1450.0603	1450.0603	.0000	487.9710

THIS RAY CALCULATION TOOK 1.995 SEC

ELEVATION ANGLE OF TRANSMISSION = 40.0000 DEG

ERROR	EVENT	ELEVATION ABOVE SEA LEVEL KM	ELEVATION ABOVE TERRAIN KM	RANGE KM	AZIMUTH DEVIATION XMTR DEG	AZIMUTH DEVIATION LOCAL DEG	ELEVATION ANGLE XMTR DEG	ELEVATION ANGLE LOCAL DEG	PULSE TIME SEC	PHASE TIME SEC	ABSORPTION DB	PATH LENGTH KM
-.7E-14	XMTR	13.0440	13.0000	.0000				40.000	.0000	.0000	.0000	.0000
-.7E-06	APOGEE	104.2225	104.2036	175.8021	-7.340	-5.047	26.407	-1.194	555.3000	555.2998	.0016	201.2805
-.7E-06	WAVE REV	104.2225	104.2036	175.8021	-7.340	-5.047	26.407	-1.194	555.3000	555.2998	.0016	201.2805
.5E-06	BACK UP0	4.7800	4.7697	374.4830	-9.535	-4.136	-2.946	-35.687	1179.3000	1179.2997	.0031	426.9943
.5E-06	BACK UP1	5.0103	5.0000	374.1414	-9.536	-4.137	-2.911	-35.775	1178.0786	1178.0784	.0031	426.5820
.5E-06	RCVR	5.0103	5.0000	374.1414	-9.536	-4.137	-2.911	-35.775	1178.0787	1178.0784	.0031	426.5820
.4E-06	BACK UP0	-1.0136	-1.0236	382.9177	-9.471	-4.073	-3.822	-34.925	1209.4787	1209.4784	.0031	437.2394
.4E-06	BACK UP1	.0101	.0000	381.4687	-9.488	-4.089	-3.670	-34.999	1204.2520	1204.2517	.0031	435.4618
.4E-06	GRND REF	.0101	.0000	381.4687	-9.488	-4.092	-3.670	34.996	1204.2519	1204.2517	.0031	435.4618
.4E-06	BACK UP0	5.1658	5.1559	389.0277	-9.442	-4.046	-2.908	35.833	1231.2519	1231.2517	.0031	444.6205
.4E-06	BACK UP1	5.0099	5.0000	388.7966	-9.442	-4.046	-2.930	35.772	1230.4251	1230.4249	.0031	444.3416
.4E-06	RCVR	5.0099	5.0000	388.7966	-9.442	-4.046	-2.930	35.772	1230.4251	1230.4249	.0031	444.3416
.5E-06	APOGEE	104.7397	104.7335	592.3256	-10.201	-4.812	6.050	-1.123	1867.2251	1867.2256	.0050	674.8220
.5E-06	WAVE REV	104.7397	104.7335	592.3256	-10.201	-4.812	6.050	-1.123	1867.2251	1867.2256	.0050	674.8220
.2E-05	BACK UP0	4.6132	4.6089	794.3212	-10.562	-5.183	-4.179	-35.647	2500.8251	2500.8263	.0068	903.8062
.2E-05	BACK UP1	5.0043	5.0000	793.7415	-10.564	-5.185	-4.148	-35.793	2498.7527	2498.7538	.0068	903.1062
.2E-05	RCVR	5.0043	5.0000	793.7415	-10.564	-5.185	-4.148	-35.793	2498.7527	2498.7538	.0068	903.1062
.2E-05	MAX HOPS	5.0043	5.0000	793.7415	-10.564	-5.185	-4.148	-35.793	2498.7527	2498.7538	.0068	903.1062

THIS RAY CALCULATION TOOK 2.989 SEC

AZIMUTH ANGLE OF TRANSMISSION = 45.000000 DEG TRANSMITTER LATITUDE = 1.798926 DEG FREQUENCY = .050000 HZ
ELEVATION ANGLE OF TRANSMISSION = 60.000000 DEG TRANSMITTER LONGITUDE = .000000 DEG SINGLE STEP ERROR = 1.000000E-06

ERROR	EVENT	ELEVATION ABOVE SEA LEVEL KM	ELEVATION ABOVE TERRAIN KM	RANGE KM	AZIMUTH DEVIATION XMTR DEG	AZIMUTH DEVIATION LOCAL DEG	ELEVATION ANGLE XMTR DEG	ELEVATION ANGLE LOCAL DEG	PULSE TIME SEC	PHASE TIME SEC	ABSORPTION DB	PATH LENGTH KM
-.7E-14	XMTR	13.0440	13.0000	.0000				60.000	.0000	.0000	.0000	.0000
-.4E-06	APOGEE	144.8719	144.8545	199.2131	-7.589	-1.829	32.272	-.221	572.1000	572.0997	.5014	256.4120
-.4E-06	WAVE REV	144.8719	144.8545	199.2131	-7.589	-1.829	32.272	-.221	572.1000	572.0997	.5014	256.4120
.3E-05	BACK UP0	4.5785	4.5681	379.1757	-10.259	-3.574	-2.982	-59.317	1140.1000	1140.1009	.9121	499.5078
.3E-05	BACK UP1	5.0104	5.0000	378.8921	-10.259	-3.574	-2.917	-59.417	1138.5475	1138.5483	.9121	498.9910
.3E-05	RCVR	5.0104	5.0000	378.8921	-10.259	-3.574	-2.917	-59.417	1138.5475	1138.5483	.9121	498.9910
.9E-06	BACK UP0	-1.0303	-1.0406	382.7092	-10.243	-3.558	-3.825	-58.392	1159.7475	1159.7484	.9121	506.1395
.9E-06	BACK UP1	.0103	.0000	382.0800	-10.249	-3.565	-3.669	-58.549	1156.1759	1156.1768	.9121	504.9226
.9E-06	GRND REF	.0103	.0000	382.0800	-10.249	-3.572	-3.669	58.546	1156.1759	1156.1768	.9121	504.9226
.1E-05	BACK UP0	5.2243	5.2141	385.4089	-10.241	-3.563	-2.894	59.464	1174.5759	1174.5768	.9121	511.1106
.1E-05	BACK UP1	5.0102	5.0000	385.2681	-10.240	-3.563	-2.925	59.414	1173.8043	1173.8052	.9121	510.8543
.1E-05	RCVR	5.0102	5.0000	385.2681	-10.240	-3.563	-2.925	59.414	1173.8043	1173.8052	.9121	510.8543
-.2E-05	APOGEE	144.7403	144.7336	567.1420	-11.193	-4.522	10.360	-1.201	1743.4043	1743.4054	1.3398	755.3699
-.2E-05	WAVE REV	144.7403	144.7336	567.1420	-11.193	-4.522	10.360	-1.201	1743.4043	1743.4054	1.3398	755.3699
.4E-05	BACK UP0	4.8787	4.8738	742.6466	-11.689	-5.024	-3.968	-59.394	2303.4043	2303.4055	1.7111	993.3461
.4E-05	BACK UP1	5.0049	5.0000	742.5637	-11.689	-5.024	-3.958	-59.423	2302.9500	2302.9512	1.7111	993.1950
.4E-05	RCVR	5.0049	5.0000	742.5637	-11.689	-5.024	-3.958	-59.423	2302.9500	2302.9512	1.7111	993.1950
.4E-05	MAX HOPS	5.0049	5.0000	742.5637	-11.689	-5.024	-3.958	-59.423	2302.9500	2302.9512	1.7111	993.1950

THIS RAY CALCULATION TOOK 3.260 SEC

ELEVATION ANGLE OF TRANSMISSION = 80.0000 DEG

.0E+00	XMTR	13.0440	13.0000	.0000				80.000	.0000	.0000	.0000	.0000
-.1E-05	EXTINC	224.5590	224.5294	88.0081	-14.390	-14.394	66.634	64.191	500.9000	500.8997	1076.3153	230.6701

THIS RAY CALCULATION TOOK .944 SEC

ELEVATION ANGLE OF TRANSMISSION = 100.0000 DEG

-.7E-14	XMTR	13.0440	13.0000	.0000				80.000	.0000	.0000	.0000	.0000
.2E-12	MAX LONG	13.3109	13.2669	.0332	-135.240	44.760	82.899	80.016	.9000	.9000	.0000	.2690
-.1E-05	MAX LONG	110.6394	110.5893	13.5646	-115.247	64.752	81.950	76.537	308.5000	308.4998	.0012	98.5992
-.1E-05	EXTINC	224.9996	224.9286	45.5105	-151.291	28.707	77.457	65.030	500.5000	500.4996	1126.0582	218.9079

THIS RAY CALCULATION TOOK .967 SEC

AZIMUTH ANGLE OF TRANSMISSION = 45.000000 DEG TRANSMITTER LATITUDE = 1.798926 DEG FREQUENCY = .050000 HZ
ELEVATION ANGLE OF TRANSMISSION = 120.000000 DEG TRANSMITTER LONGITUDE = .000000 DEG SINGLE STEP ERROR = 1.000000E-06

ERROR	EVENT	ELEVATION ABOVE SEA LEVEL KM	ELEVATION ABOVE TERRAIN KM	RANGE KM	AZIMUTH DEVIATION XMTR DEG	AZIMUTH DEVIATION LOCAL DEG	ELEVATION ANGLE XMTR DEG	ELEVATION ANGLE LOCAL DEG	PULSE TIME SEC	PHASE TIME SEC	ABSORPTION DB	PATH LENGTH KM
-.7E-14	XMTR	13.0440	13.0000	.0000				60.000	.0000	.0000	.0000	.0000
.2E-06	APOGEE	152.7178	152.4727	145.0531	-169.378	2.943	42.894	-.188	564.9000	564.9000	1.1915	221.6716
.2E-06	WAVE REV	152.7178	152.4727	145.0531	-169.378	2.943	42.894	-.188	564.9000	564.9000	1.1915	221.6716
.6E-06	MAX LONG	93.6766	92.4536	253.9614	-163.165	-2.285	16.333	-61.003	827.3000	827.2999	2.3514	356.7367
-.2E-05	MAX LONG	79.4894	78.5126	259.9170	-162.436	-1.561	13.070	-61.038	881.7000	881.6999	2.3515	372.5074
-.6E-07	BACK UP0	5.2948	4.9945	299.0982	-159.875	.978	-2.827	-57.473	1153.7000	1153.7012	2.3516	457.5298
-.6E-07	BACK UP1	5.3003	5.0000	299.0948	-159.875	.978	-2.826	-57.476	1153.6795	1153.6806	2.3516	457.5233
-.6E-07	RCVR	5.3003	5.0000	299.0948	-159.875	.978	-2.826	-57.476	1153.6795	1153.6806	2.3516	457.5233
-.7E-06	BACK UP0	-.5223	-.7987	302.9344	-159.849	.997	-3.924	-54.285	1174.8795	1174.8806	2.3516	464.5041
-.7E-06	BACK UP1	.2797	.0000	302.3604	-159.847	1.001	-3.774	-54.763	1171.9866	1171.9878	2.3516	463.5177
-.7E-06	GRND REF	.2797	.0000	302.3604	-159.847	.553	-3.774	54.094	1171.9866	1171.9878	2.3516	463.5177
-.5E-06	BACK UP0	5.6107	5.3512	305.8975	-159.809	.585	-2.765	57.003	1191.5866	1191.5878	2.3516	469.9225
-.5E-06	BACK UP1	5.2607	5.0000	305.6826	-159.813	.580	-2.831	56.842	1190.2861	1190.2873	2.3516	469.5109
-.5E-06	RCVR	5.2607	5.0000	305.6826	-159.813	.580	-2.831	56.842	1190.2861	1190.2873	2.3516	469.5109
-.2E-05	MAX LONG	81.0713	80.9539	347.1532	-157.769	2.617	9.444	60.760	1470.2861	1470.2882	2.3516	556.9184
-.4E-06	MAX LONG	94.3487	94.2425	353.1353	-157.340	3.046	11.270	60.149	1521.4861	1521.4881	2.3518	571.7558
-.3E-05	APOGEE	147.4966	147.4612	453.9279	-156.085	4.301	14.257	-.783	1771.0861	1771.0874	2.9599	694.3670
-.3E-05	WAVE REV	147.4966	147.4612	453.9279	-156.085	4.301	14.257	-.783	1771.0861	1771.0874	2.9599	694.3670
-.9E-06	MAX LONG	93.3241	93.3063	551.2271	-155.276	5.113	5.734	-60.529	2017.4861	2017.4867	3.5120	813.9165
-.3E-05	MAX LONG	80.0572	80.0402	557.3032	-155.031	5.358	4.297	-60.571	2068.6861	2068.6867	3.5121	828.7383
-.2E-05	BACK UP0	4.6476	4.6343	599.3981	-154.193	6.198	-3.497	-56.556	2347.0861	2347.0875	3.5121	915.7272
-.2E-05	BACK UP1	5.0134	5.0000	599.1689	-154.193	6.198	-3.461	-56.730	2345.7311	2345.7326	3.5121	915.2955
-.2E-05	RCVR	5.0134	5.0000	599.1689	-154.193	6.198	-3.461	-56.730	2345.7311	2345.7326	3.5121	915.2955
-.2E-05	MAX HOPS	5.0134	5.0000	599.1689	-154.193	6.198	-3.461	-56.730	2345.7311	2345.7326	3.5121	915.2955

THIS RAY CALCULATION TOOK 3.174 SEC

ELEVATION ANGLE OF TRANSMISSION = 140.0000 DEG

ERROR	EVENT	ELEVATION ABOVE SEA LEVEL KM	ELEVATION ABOVE TERRAIN KM	RANGE KM	AZIMUTH DEVIATION XMTR DEG	AZIMUTH DEVIATION LOCAL DEG	ELEVATION ANGLE XMTR DEG	ELEVATION ANGLE LOCAL DEG	PULSE TIME SEC	PHASE TIME SEC	ABSORPTION DB	PATH LENGTH KM
-.7E-14	XMTR	13.0440	13.0000	.0000				40.000	.0000	.0000	.0000	.0000
-.2E-06	APOGEE	111.7581	111.5888	123.0044	-169.527	7.625	37.922	-.231	534.5000	534.4999	.0048	162.0291
-.2E-06	WAVE REV	111.7581	111.5888	123.0044	-169.527	7.625	37.922	-.231	534.5000	534.4999	.0048	162.0291
.2E-05	BACK UP0	5.6287	4.8252	255.8244	-157.357	-6.582	-2.808	-35.841	1084.9000	1084.9009	.0095	341.7092
.2E-05	BACK UP1	5.8098	5.0000	255.5846	-157.368	-6.592	-2.768	-36.018	1083.9362	1083.9370	.0095	341.4045
.2E-05	RCVR	5.8098	5.0000	255.5846	-157.368	-6.592	-2.768	-36.018	1083.9361	1083.9370	.0095	341.4045
.2E-05	BACK UP0	.1264	-.4906	264.2764	-157.076	-6.316	-3.984	-28.922	1115.9361	1115.9371	.0095	351.8883
.2E-05	BACK UP1	.6338	.0000	263.3761	-157.096	-6.336	-3.879	-29.668	1112.8739	1112.8748	.0095	350.8503
.2E-05	GRND REF	.6338	.0000	263.3761	-157.096	-6.666	-3.879	27.499	1112.8739	1112.8748	.0095	350.8503
.1E-05	BACK UP0	6.1439	5.6559	272.2417	-156.788	-6.368	-2.674	34.607	1145.2739	1145.2749	.0095	361.4027
.1E-05	BACK UP1	5.5670	5.0674	271.4310	-156.823	-6.402	-2.796	34.022	1142.0739	1142.0749	.0095	360.3933
.1E-05	BACK UP1	5.5009	5.0000	271.3369	-156.827	-6.406	-2.810	33.954	1141.7057	1141.7067	.0095	360.2768
.1E-05	RCVR	5.5009	5.0000	271.3368	-156.827	-6.406	-2.810	33.954	1141.7055	1141.7065	.0095	360.2767
.2E-05	MAX LONG	77.5366	77.4586	365.2769	-152.219	-1.808	8.298	40.528	1528.9055	1528.9074	.0096	482.1444
.5E-06	MAX LONG	96.7313	96.6704	384.5933	-151.260	-.849	10.441	38.317	1624.9055	1624.9075	.0098	510.2986
.2E-06	APOGEE	109.2373	109.1943	417.3020	-150.307	.103	10.980	-.377	1727.3055	1727.3075	.0129	547.2455
.2E-06	WAVE REV	109.2373	109.1943	417.3020	-150.307	.103	10.980	-.377	1727.3055	1727.3075	.0129	547.2455
.4E-06	MAX LONG	95.7525	95.7208	450.5518	-149.468	.942	8.285	-39.400	1832.9055	1832.9075	.0159	585.0253
.9E-06	MAX LONG	76.5533	76.5264	470.0849	-148.799	1.612	5.524	-40.094	1928.9055	1928.9075	.0161	613.1416
.1E-05	BACK UP0	4.6166	4.6024	566.4724	-147.183	3.230	-3.398	-33.043	2316.1055	2316.1083	.0162	735.0914
.1E-05	BACK UP1	5.0143	5.0000	565.8781	-147.184	3.229	-3.356	-33.478	2313.8622	2313.8650	.0162	734.3758
.1E-05	RCVR	5.0143	5.0000	565.8781	-147.184	3.229	-3.356	-33.478	2313.8622	2313.8650	.0162	734.3758
.1E-05	MAX HOPS	5.0143	5.0000	565.8781	-147.184	3.229	-3.356	-33.478	2313.8622	2313.8650	.0162	734.3758

END OF INPUT DATA

THIS RAY CALCULATION TOOK 2.725 SEC

```
S03-1    SAMPLE CASE FOR HARPA DOCUMENTATION        REV. 2-10-86
ULOGZ2       3.0 NPWIND         .0   GAMRTDM        .0 CBLOB2        2.0
TTANH5       1.0 TBLOB2        2.0   MCONST       29.0 NO MODL        .0
PEXP         1.0 NPPRES         .0   MUARDC        1.0 NPABSR         .0
GLORENZ      2.0 NPTERR         .0   RTERR          .0 NO MODL        .0
LOGARITHMIC EASTWARD WIND PROFILE, U*=.5 M/S, Z0=1 KM
NO WIND PERTURBATION
SOUND SPEED IN TERMS OF TEMPERATURE MODEL
50% INCREASE IN SQ. SOUND SPEED AT 125KM HT, 335KM N, 125KM E
U.S. STANDARD ATMOSPHERE 1962 TEMPERATURE PROFILE
50% CYLINDRICAL INCREASE IN TEMPERATURE AT 105KM N., 105 KM W
MOLECULAR WEIGHT = 29
RIDGE 2-KM HIGH, 30-KM WIDE ALONG EQUATOR
S033    130440   1799      0     50000      3142    4500000  -2000000            0     0 3T
    50367    294887 -3840 -3838-12169    9196124    9196123         0      0     0     0 1R
      313    563108 -2735 -2735 10693   17228767   17228768         0      0     0     0 2G
    50270    831414 -2343 -2340 12148   25260761   25260764         0      0     0     0 2R
    50112   3073288 -4374 -4375-12129   94307633   94307634         0      0     0     0 3R
S033    130440   1799      0     50000      3142    4500000  -1500000            0     0 3T
    59174    560059 -3651 -3649   -11   16962526   16962525         0      0     0     0 1M
    59174    560059 -3651 -3649   -11   16962526   16962525         0      0     0     0 2M
    59185   3248490 -4500 -4501    -2   99099938   99099939         0      0     0     0 3M
S033    130440   1799      0     50000      3142    4500000  -1000000            0     0 3T
    95112    406604 -4161 -4160   -12   12374135   12374135         0      0     0     0 1M
    95112    406604 -4161 -4160   -12   12374135   12374135         0      0     0     0 2M
    95115   2807459 -4767 -4769    -3   85924969   85924971         0      0     0     0 3M
S033    130440   1799      0     50000      3142    4500000   -500000            0     0 3T
   119700    256779 -4424 -4424   -14    7831383    7831383         0      0     0     0 1M
   119700    256779 -4424 -4424   -14    7831383    7831383         0      0     0     0 2M
   119705   2542307 -4873 -4875    -3   77890316   77890317         0      0     0     0 3M
S033    130440   1799      0     50000      3142    4500000         0            0     0 3T
   130453   2250470 -4938 -4939    -3   68990961   68990962         0      0     0     0 1M
   130453   2250470 -4938 -4939    -3   68990961   68990962         0      0     0     0 2M
   130423   4501421 -4932 -4941    -1  137987009  137987013         0      0     0     0 3M
S033    130440   1799      0     50000      3142    4500000    500000            0     0 3T
   119724   2027348 -4988 -4989    -4   62186567   62186568         0      0     0     0 1M
   119724   2027348 -4988 -4989    -4   62186567   62186568         0      0     0     0 2M
   119693   4312567 -4949 -4957    -1  132228323  132228325         0      0     0     0 3M
S033    130440   1799      0     50000      3142    4500000   1000000            0     0 3T
    95149   1993238 -5015 -5017    -4   61147865   61147865         0      0     0     0 1M
    95149   1993238 -5015 -5017    -4   61147865   61147865         0      0     0     0 2M
    95110   4393846 -4931 -4939    -1  134683577  134683579         0      0     0     0 3M
S033    130440   1799      0     50000      3142    4500000   1500000            0     0 3T
    59270   2127216 -4948 -4950    -4   65140567   65140562         0      0     0     0 1M
    59270   2127216 -4948 -4950    -4   65140567   65140562         0      0     0     0 2M
    59192   4814803 -4791 -4801    -1  147243522  147243515         0      0     0     0 3M
S033    130440   1799      0     50000      3142    4500000   2000000            0     0 3T
    50170   1946869 -5322 -5324-12129   59828119   59828103         0      0     0     0 1R
      152   2215510 -4860 -4863 10665   67882811   67882797         0      0     0     0 2G
    50137   2484283 -4498 -4500 12121   75937724   75937711         0      0     0     0 2R
    50070   4727741 -4790 -4799-12126  145006034  145006027         0      0     0     0 3R
S033    130440   1799      0     50000      3142    4500000   2500000            0     0 3T
    50164   2044020 -5581 -5584-19279   62541213   62541173         0      0     0     0 1R
      154   2198518 -5299 -5304 18131   67332802   67332761         0      0     0     0 2G
    50145   2353069 -5054 -5059 19271   72124354   72124310         0      0     0     0 2R
```

```
    50073  4608314 -5232 -5245-19276 141475510 141475369      0       0      0    0 3R
 S033   130440  1799      0    50000      3142   4500000  3000000              0    0 3T
    50145   2387553 -5803 -5808-25431  71841724  71841676      0       0      0    0 1R
      139   2499430 -5617 -5624 24351  75461318  75461270      0       0      0    0 2G
    50133   2611336 -5446 -5453 25425  79080884  79080835      0       0      0    0 2R
    50065   5163074 -5562 -5579-25432 156441221 156441073      0       0      0    0 3R
 S033   130440  1799      0    50000      3142   4500000  3500000              0    0 3T
    50063   5513279 -7508 -5984-29065 162974268 162974290      2       0      0    0 1R
       61   5608414 -7434 -5912 28128 166149079 166149107      2       0      0    0 2G
    50060   5703560 -7363 -5841 29063 169323893 169323927      2       0      0    0 2R
   615060 10006822 -7658 -6166-14096 295883893 295883991      5       0      0    0 3F
 S033   130440  1799      0    50000      3142   4500000  4000000              0    0 3T
    50103   3741414 -9536 -4137-35775 117807866 117807837      3       0      0    0 1R
      101   3814687 -9488 -4092 34998 120425195 120425167      3       0      0    0 2G
    50099   3887966 -9442 -4046 35772 123042513 123042487      3       0      0    0 2R
    50043   7937415-10564 -5185-35793 249875267 249875381      7       0      0    0 3R
 S033   130440  1799      0    50000      3142   4500000  4500000              0    0 3T
    50125   3091011 -9743 -3791-44213 102900551 102900586      6       0      0    0 1R
      123   3145487 -9709 -3762 43355 105086946 105086986      6       0      0    0 2G
    50121   3199967 -9676 -3729 44209 107273331 107273377      6       0      0    0 2R
    50058   6455868-11160 -5225-44221 215828181 215828361     16       0      0    0 3R
 S033   130440  1799      0    50000      3142   4500000  5000000              0    0 3T
    50135   2885700-10288 -2785-49919  98659438  98659471     15       0      0    0 1R
      133   2930535-10275 -2780 49061 100647939 100647981     15       0      0    0 2G
    50131   2975374-10263 -2768 49914 102636428 102636484     15       0      0    0 2R
    50065   6062118-12249 -4762-49923 207558128 207558752     40       0      0    0 3R
 S033   130440  1799      0    50000      3142   4500000  5500000              0    0 3T
    50133   3020692-11403  -647-53317 101084059 101084117     56       0      0    0 1R
      131   3060560-11420  -673 52507 102978779 102978846     56       0      0    0 2G
    50130   3100432-11437  -689 53313 104873491 104873569     56       0      0    0 2R
    50067   6218924-14191 -3448-53321 210124746 210125585    115       0      0    0 3R
 S033   130440  1799      0    50000      3142   4500000  6000000              0    0 3T
    50104   3788921-10259 -3574-59417 113854747 113854834    912       0      0    0 1R
      103   3820800-10249 -3572 58548 115617591 115617680    912       0      0    0 2G
    50102   3852681-10240 -3563 59414 117380431 117380523    912       0      0    0 2R
    50049   7425637-11689 -5024-59423 230295002 230295121   1711       0      0    0 3R
 S033   130440  1799      0    50000      3142   4500000  6500000              0    0 3T
    50057   5821218 -6448 -6098-64257 142399710 142400008183625       0      0    0 1R
       56   5847356 -6435 -6089 63372 144082823 144083135183625       0      0    0 2G
    50056   5873495 -6422 -6077 64256 145765935 145766261183625       0      0    0 2R
  1684896 10011309 -6189 -5873-18166 234725935 234726873365871       0      0    0 3F
 S033   130440  1799      0    50000      3142   4500000  7000000              0    0 3T
  2195558   1936549 -7763 -7799 30028  59050000  59049975999999       0      0    0 1E
 S033   130440  1799      0    50000      3142   4500000  7500000              0    0 3T
  2225112   1285167-10383-10389 49233  52810000  52809965999999       0      0    0 1E
 S033   130440  1799      0    50000      3142   4500000  8000000              0    0 3T
  2245590    880081-14390-14394 64191  50090000  50089965999999       0      0    0 1E
 S033   130440  1799      0    50000      3142   4500000  8500000              0    0 3T
  2247730    555666-22417-22421 77477  48650000  48649969999999       0      0    0 1E
 S033   130440  1799      0    50000      3142   4500000  9000000              0    0 3T
  2251046    297077-44996-12280 90000  48250000  48249969999999       0      0    0 1E
 S033   130440  1799      0    50000      3142   4500000  9500000              0    0 3T
  2248568    228757247892 67890 77673  48650000  48649969999999       0      0    0 1E
 S033   130440  1799      0    50000      3142   4500000 10000000              0    0 3T
  2249996    455105208709 28707 65030  50050000  50049964999999       0      0    0 1E
```

```
S033   130440  1799      0     50000      3142   4500000  10500000              0    0 3T
  2233740    789299196959 16939 51415  52490000  52489964999999       0       0    0 1E
S033   130440  1799      0     50000      3142   4500000  11000000              0    0 3T
  2208428  1259606191906  8518 36751  57130000  57129972999999       0       0    0 1E
S033   130440  1799      0     50000      3142   4500000  11500000              0    0 3T
  2124423  2422518195818 -4466  9647  71370000  71370035999999       0       0    0 1E
S033   130440  1799      0     50000      3142   4500000  12000000              0    0 3T
    53003  2990948200125   978-57476 115367948 115368064  2352       0       0    0 1R
     2797  3023604200153   553 54429 117198663 117198779  2352       0       0    0 2G
    52607  3056826200187   580 56842 119028614 119028727  2352       0       0    0 2R
    50134  5991689205807  6198-56730 234573114 234573255  3512       0       0    0 3R
S033   130440  1799      0     50000      3142   4500000  12500000              0    0 3T
    60949  2479241201442 -6590-49243 105715988 105716092   185       0       0    0 1R
     9420  2525083201621 -7410 43874 107849212 107849323   185       0       0    0 2G
    58035  2573712201820 -7226 45970 109976807 109976925   185       0       0    0 2R
    50205  5082746211925  2861-45468 216607576 216608152   214       0       0    0 3R
S033   130440  1799      0     50000      3142   4500000  13000000              0    0 3T
    67320  2291697202541-11202-46567 101774969 101775077    44       0       0    0 1R
    15155  2342259202856-11710 40332 104050187 104050302    44       0       0    0 2G
    62782  2396807203195-11392 41940 106320535 106320655    44       0       0    0 2R
    50204  5018625215998  1379-41034 214865783 214866002    58       0       0    0 3R
S033   130440  1799      0     50000      3142   4500000  13500000              0    0 3T
    66041  2326700202456 -9909-42477 102595360 102595461    18       0       0    0 1R
    13438  2386724202782-10342 35491 105091010 105091118    18       0       0    0 2G
    60892  2450630203127-10017 38011 107580043 107580156    18       0       0    0 2R
    50180  5222047215158  1990-37119 219900993 219901351    26       0       0    0 3R
S033   130440  1799      0     50000      3142   4500000  14000000              0    0 3T
    58098  2555846202632 -6592-36018 108393614 108393701    10       0       0    0 1R
     6338  2633761202904 -6666 28584 111287389 111287482    10       0       0    0 2G
    55009  2713368203173 -6406 33954 114170553 114170650    10       0       0    0 2R
    50143  5658781212816  3229-33478 231386220 231386499    16       0       0    0 3R
```

APPENDIX B: BLANK INPUT-PARAMETER FORMS

This appendix provides blank Input Parameter Forms for all of the atmospheric models (including terrain and receiver-surface models) that we have developed for HARPA. The forms describe each model mathematically and list the variable input parameters you have to specify. We recommend reproducing these forms and filling them out when setting up atmospheric models for HARPA. The filled-out forms should then be saved as a record of the models you have defined.

The FORTRAN source codes for the corresponding model subroutines are listed in Appendix D under the model name. No forms are given for the do-nothing versions NTEMP, NPTEMP, NPSPEED, NPTERR, NPPRES, and NPWIND. The forms are arranged as follows:

FORM TO SPECIFY INPUT DATA FOR A
THREE-DIMENSIONAL RAYPATH CALCULATION

Name________________
Date________________

Atmospheric ID (3 characters)_____

Title (77 characters)____

Transmitter: Height	________	km, nm, ft (W3)
	________	above terrain
	________	above sea level
Latitude	________	rad, deg, km (W4)
Longitude	________	rad, deg, km (W5)
Frequency, initial	________	rad/s, Hz, s (W7)
final	________	(W8)
step	________	(W9)
Azimuth angle, initial	________	rad, deg clockwise of north (W11)
final	________	(W12)
step	________	(W13)
Elevation angle, initial	________	rad, deg (W15)
final	________	(W16)
step	________	(W17)
Receiver: Height	________	km, nm, ft (W20)
	________	above sea level (rcvr model RHORIZ)
	________	above terrain (rcvr model RTERR)
Distance from origin	________	rad, deg, km (W30) (rcvr model RVERT)
Latitude of origin	________	rad, deg, km (W31) (rcvr model RVERT)
Longitude of origin	________	rad, deg, km (W32) (rcvr model RVERT)
Stop frequency stepping when ray goes out of bounds	________	(W21 = 1.)
Maximum height	________	km (W26)
Minimum height	________	km (W27)
Maximum range	________	km (W28)
Maximum number of hops	________	(W22)
Maximum number of steps per hop	________	(W23)
Maximum allowable error per step	________	(W42)
Additional calculations:		= 1. to integrate = 2. to integrate and print
Phase path	________	(W57)
Absorption	________	(W58)
Doppler shift	________	(W59)
Path length	________	(W60)
Printout:	Every ________	steps of the ray trace (W71)
Computer readable output (raysets):	________	(W72 = 1.)
Diagnostic printing:	________	(W73 = 1.)
Suppress all printout	________	(W74 = 1.)

FORM TO SPECIFY INPUT PARAMETERS FOR PLOTTING A
PROJECTION OF THE RAYPATH

Model ID:__________

Plot directly during raypath calculations______________, or

plot from precomputed raypaths______________

in disk file______________

Normal or apogee plots:

Normal__________(W80=0.0)

Plot apogees only__________(W80=1.0)

Projection:

Vertical plane, polar plot, rectangular expansion__________(W81=1.0)

Horizontal plane, lateral expansion __________(W81=2.0)

Vertical plane, polar plot, radial expansion __________(W81=3.0)

Vertical plane, rectangular plot __________(W81=4.0)

Superimpose these raypath plots on the graph of the previous runset:

Yes__________(W81 negative.)

No __________(W81 positive.)

Vertical or lateral expansion factor __________(W82)

Coordinates of the left edge of the graph:

Latitude = __________(rad, deg, km) north (W83)

Longitude = __________(rad, deg, km) east (W84)

Coordinates of the right edge of the graph:

Latitude = __________(rad, deg, km) north (W85)

Longitude = __________(rad, deg, km) east (W86)

Distance between horizontal tick marks = __________rad, deg, km (W87)

Height above sea level of bottom of graph = __________km (W88)

Height above sea level of top of graph = __________km (W89)

Distance between vertical tick marks = __________km (W96)

FORM TO SPECIFY AN ATMOSPHERIC MODEL
(including terrain model)

Name________________________________ Date___________________

Atmospheric ID (3 characters)__________

Coordinates of the north pole of the computational coordinate system:

North geographic latitude: __________rad, km, deg (W24)

East geographic longitude: __________rad, km, deg (W25)

Models:

	Subroutine Name	Data set ID
Dispersion relation	__________	
Wind velocity	__________	__________(W102)
Wind-velocity perturbation	__________	__________(W127)
Sound speed	__________	__________(W152)
Sound-speed perturbation	__________	__________(W177)
Temperature	__________	__________(W202)
Temperature perturbation	__________	__________(W227)
Molecular weight	__________	__________(W252)
Terrain	__________	__________(W302)
Terrain perturbation	__________	__________(W327)
Viscosity/conductivity	__________	__________(W502)
Viscosity/conductivity perturbation	__________	__________(W527)
Pressure	__________	__________(W552)
Pressure perturbation	__________	__________(W557)
Receiver surface*	__________	
Plot-annotation model*	__________	

*The receiver-surface model and plot-annotation model are not considered part of the atmospheric ID

FORM TO SPECIFY INPUT DATA FOR
WIND-VELOCITY MODEL WLINEAR

This subroutine specifies constant radial (upward), zonal (eastward) and meridional (southward) winds, allowing a linear height gradient of the zonal component.

$$U_\theta = U_{\theta o}$$

$$U_\phi = U_{\phi o} + \frac{du_\phi}{dz} z$$

$$u_r = U_{ro}$$

$z = r - r_e$, where r_e is the Earth radius, and r is the radial coordinate of ray point.

Specify--

the model check for WLINEAR = ___1.0___(W100)

the input data-format code = __________(W101)

an input data-set identification number = __________(W102)

an 80-character description of the wind-velocity profile:

____

the constant upward wind, U_{ro} = __________km/s, m/s (W103)

the constant southward wind, $U_{\theta o}$ = __________km/s, m/s (W104)

the ground value of the eastward wind, $U_{\phi o}$ = __________km/s, m/s (W105)

the height gradient of u_ϕ, du_ϕ/dz = __________km/s/km, m/s/km (W106)

(This subroutine can be used with its input parameters zero when no wind field is desired.)

OTHER MODELS REQUIRED: Any wind-perturbation model. Use NPWIND if no perturbation is desired.

FORM TO SPECIFY INPUT DATA FOR WIND-VELOCITY MODEL WTIDE

This subroutine represents the wind field of the atmospheric tides by zonal and meridional height profiles that are sinusoidal and in phase quadrature. The profiles progress downward with time, giving a corkscrew effect:

$$u_\theta = U_{\theta o} \sin \left\{2\pi \left(\frac{z}{\lambda_z} + \frac{t}{\tau}\right)\right\}$$

$$u_\phi = U_{\phi o} \cos \left\{2\pi \left(\frac{z}{\lambda_z} + \frac{t}{\tau}\right)\right\}$$

$z = r - r_e$, where r_e is the Earth radius, and r is the radial coordinate of the ray point.

Specify--

the model check for WTIDE = 5.0 (W100)

the input data-format code = __________(W101)

an input data-set identification number = __________(W102)

an 80-character description of the model, including description of parameter values:

____

the amplitude of the meridional component, $U_{\theta o}$ = __________km/s, m/s (W104)

the amplitude of the zonal component, $U_{\phi o}$ = __________km/s, m/s (W103)

the vertical wavelength, λ_z = __________km (W105)

the time in wave periods, t/τ = __________(W106)

the wave period, τ = __________sec (W107)

(The Earth's poles should be avoided in ray calculations because discontinuities appear there.)

OTHER MODELS REQUIRED: Any wind-perturbation model. Use NPWIND if no perturbation is desired.

FORM TO SPECIFY INPUT DATA
FOR WIND-VELOCITY MODEL ULOGZ2

A logarithmic wind profile of the atmospheric boundary layer neglecting Coriolis forces. The eastward wind is given by

$$u_\phi = \frac{u_*}{k} \ln \frac{z}{z_0} \qquad \text{for} \qquad z > z_0 e$$

$$u_\phi = \frac{u_*}{k} \frac{z}{z_0 e} \qquad \text{for} \qquad z \leq z_0 e \quad ,$$

where $z = G(r,\theta,\phi)$ is determined by the terrain model and is the height above or some kind of distance from the terrain, depending on the terrain model, and r is the radial coordinate of the ray point.

Specify--

the model check for ULOGZ2 = ____6.0____(W100)

the input data-format code = __________(W101)

an input data-set identification number = __________(W102)

an 80-character description of the wind velocity profile:

____

the reference wind speed, u_* = __________km/s, m/s (W103)

von Kármán's constant, k = __________(W104) (.35 recommended)

the roughness height, z_0 = __________km (W105)

OTHER MODELS REQUIRED: Any wind-perturbation model. Use NPWIND if no perturbation is desired.

($\overline{uw} = -u_*^2$ is the surface stress at the ground.)

FORM TO SPECIFY INPUT DATA FOR
WIND-VELOCITY MODEL VVORTX3

This subroutine models a vortex with a viscous core and a Gaussian intensity profile in the vertical. The axis of the vortex is vertical and may be positioned above any geographic latitude and longitude. The vortex rotates anticlockwise looking down. The core (inside r_0) is essentially a solid-rotating fluid, while outside r_0, $|u|$ falls off as the inverse radius.

$$u_\theta = -\frac{1.397\ R_e U_o r_o}{r^2}\left(1 - e^{-1.26\ r^2/r_o^2}\right)(\phi - \phi_o)e^{-\left[\frac{h - h_{max}}{w_H}\right]^2}$$

$$u_\phi = \frac{1.397\ R_e U_o r_o}{r^2}\left(1 - e^{-1.26\ r^2/r_o^2}\right)(\theta - \theta_o)e^{-\left[\frac{h - h_{max}}{w_H}\right]^2},$$

where $\theta_0 = \pi/2 - \lambda_0$ and r is the radial distance from the vortex center. The numerical constants normalize the function so that $|U| = U_o$ at $r = r_o$. R_e is the radius of the Earth, θ is the colatitude, ϕ is the longitude, and h is the height above sea level.

Specify--

the model check for VVORTX3 = 9.0 (W100)

the input data-format code = __________ (W101)

an input data-set identification number = __________ (W102)

an 80-character description of the model, including description of parameter values:

---

the maximum tangential wind, U_0 = __________ km/s, m/s (W103)

the radius of the vortex core (to $u = U_0$), r_0 = __________ km (W104)

the latitude of the vortex center, λ_0 = __________ rad, deg, km N (W105)

the longitude of the vortex center, ϕ_0 = __________ rad, deg, km E (W106)

the Gaussian width in height of the vortex, W_H = __________ km, m (W107)

the height of the vortex, h_{max} = __________ km, m (W108)

OTHER MODELS REQUIRED: Any wind-perturbation model. Use NPWIND if no perturbation is desired.

FORM TO SPECIFY INPUT DATA FOR
WIND-VELOCITY MODEL WGAUSS2

This subroutine specifies a zonal (eastward) wind field whose intensity decays in a Gaussian manner in all three space dimensions.

$$u_\phi = U_{\phi o} \exp\left\{-\left(\frac{z-z_o}{W_z}\right)^2 - \left(\frac{\theta-\theta_o}{W_\theta}\right)^2 - \left(\frac{\phi-\phi_o}{W_\phi}\right)^2\right\}$$

$z = r - r_e$, where r_e is the Earth radius, $\theta_o = \pi/2 - \lambda_o$, and r is the radial coordinate of the ray point. θ is the colatitude. ϕ is the longitude. Notice that this wind field does not satisfy continuity if $W_\phi \neq 0$.

Specify--

the model check for WGAUSS2 = ___8.0___(W100)

the input data-format code = __________(W101)

an input data-set identification number = __________(W102)

an 80-character description of the model, including description of parameter values:

____

the maximum value of u_ϕ, $U_{\phi o}$ = __________km/s, m/s (W103)

the height where u_ϕ, maximizes, z_o = __________km (W107)

the Gaussian width in height of u_ϕ, W_z = __________km (W104)*

the latitude where u_ϕ maximizes, λ_o = __________rad, deg, km N (W108)

the meridional width of u_ϕ, W_θ = __________rad, deg, km (W105)*

the longitude where u_ϕ, maximizes, ϕ_o = __________rad, deg, km E (W109)

the zonal width of u_ϕ, W_ϕ = __________rad, deg, km (W106)*

OTHER MODELS REQUIRED: Any wind-perturbation model. Use NPWIND if no perturbation is desired.

*Setting W_z, W_θ or W_ϕ = 0 results in no space variation in that direction.

FORM TO SPECIFY
SOUND-SPEED MODEL GAMRTDM

This model specifies sound speed in terms of a background temperature model using

$$C^2 = \frac{\gamma RT}{M} ,$$

where $\gamma = 1.4$, R is the universal gas constant, T is the absolute temperature in Kelvins, and $M(r,\theta,\phi)$ is a model of the mean molecular weight of the atmosphere. See Sec. 6.3 for further description of this model.

Specify --

The model check for GAMRTDM = 1.0 (W150)

OTHER MODELS REQUIRED: Any background temperature model; any molecular weight model.

FORM TO SPECIFY INPUT DATA FOR
SOUND-SPEED MODEL CSTANH

This model represents the sound speed (squared) profile by a sequence of linear segments that are smoothly joined by hyperbolic functions:

$$C^2 = C_o^2 + \frac{b_1}{2}(z-z_o) + \sum_{i=1}^{n} \delta_i \left(\frac{b_{i+1} - b_i}{2}\right) \ln \left\{ \frac{\cosh\left(\frac{z-z_i}{\delta_i}\right)}{\cosh\left(\frac{z_i-z_o}{\delta_i}\right)} \right\} + \frac{b_{n+1}}{2}(z-z_o)$$

$$\frac{dC^2}{dz} = b_1 + \sum_{i=1}^{n} \left(\frac{b_{i+1} - b_i}{2}\right) \left\{\tanh\left(\frac{z-z_i}{\delta_i}\right) + 1\right\}$$

$$b_i = (C_i^2 - C_{i-1}^2)/(z_i - z_{i-1}) \quad .$$

$z = r - r_e$, where r_e is the Earth radius, and r is the radial coordinate of the ray point. Thus, δ_i is the half-thickness of a region centered at approximately z_i km, in which dC^2/dz changes from b_i to b_{i+1}. Start by drawing a profile with linear segments, and get C_i^2 and z_i from the corners. Then select δ_i to round the corners. The final profile will not go through (C_i^2, z_i).

Specify--

the model check for CSTANH = ____2.0____ (W150)

the input data-format code = ____________ (W151)

an input data-set identification number = ____________ (W152)

an 80-character description of the model with parameters:

____

and the profile values:

the number of points in the profile -2 = n = ____________

the profile: i	z_i (km,m)	C_i (km/s, m/s)	δ_i (km,m)

OTHER MODELS REQUIRED: Any sound-speed-perturbation model. Use NPSPEED if no perturbation is desired. FUNCTION ALCOSH.

FORM TO SPECIFY INPUT DATA FOR SOUND-SPEED
PERTURBATION MODEL CBLOB2

An increase (or decrease) in sound speed in a localized region that decays in a Gaussian manner in all three spatial directions.

$$C^2(r,\theta,\phi) = C_o^2(r,\theta,\phi)\left(1 + \Delta \exp\left\{-\left(\frac{z-z_o}{W_z}\right)^2 - \left(\frac{\theta-\theta_o}{W_\theta}\right)^2 - \left(\frac{\phi-\phi_o}{W_\phi}\right)^2\right\}\right)$$

$C_o^2(r,\theta,\phi)$ is the square of the sound speed specified by a sound-speed model. (r,θ,ϕ) are the coordinates of the ray point in an Earth-centered spherical polar-coordinate system. $\theta_o = \pi/2 - \lambda_o$ and $z = r - r_e$, where r_e is the Earth radius.

Specify--

the model check for subroutine CBLOB2 = 2.0 (W175)

the input data-format code = ____________ (W176)

an input data-set identification number = ____________ (W177)

an 80-character description for the sound-speed perturbation model, including description of parameter values:

____

the strength of the fractional increase (or decrease), Δ = ______ (W178)

the height of maximum effect, z_o = ____________ km (W179)

the latitude of maximum effect, λ_o = ____________ rad, deg, km N (W180)

the longitude of maximum effect, ϕ_o = ____________ rad, deg, km E (W181)

the Gaussian width in height of the effect, W_z = ____________ km (W182)*

the meridional width of the effect, W_θ = __________ rad, deg, km (W183)*

the zonal width of the effect, W_ϕ = ____________ rad, deg, km (W184)*

OTHER MODELS REQUIRED: none.

---

* Setting W_z, W_θ, or W_ϕ = zero results in no space variation in that direction.

FORM TO SPECIFY INPUT DATA FOR
ATMOSPHERIC TEMPERATURE MODEL TLINEAR

This subroutine specifies an atmospheric temperature that increases linearly with height.

$$T = T_o + \left(\frac{dT}{dz}\right) z$$

$z = r - r_e$, where r_e is the Earth radius and r is the radial coordinate of the ray point.

Specify--

the model check for TLINEAR = ___1.0___ (W200)

the input data-format code = ____________ (W201)

an input data-set identification number = ____________ (W202)

an 80-character description of the model, including description of parameter values:

____

the ground temperature, T_o = ____________ °K (W203)

the temperature gradient, dT/dz = ____________ °K/km (W204)

(set = 0 for isothermal atmosphere)

OTHER MODELS REQUIRED: Any temperature-perturbation model. Use NPTEMP if no perturbations are desired.

FORM TO SPECIFY INPUT DATA FOR TEMPERATURE MODEL TTANH5

This model represents the temperature profile by a sequence of linear segments that are smoothly joined by hyperbolic functions:

$$T = T_o + \frac{c_1}{2}(z-z_o) + \sum_{i=1}^{n} \delta_i \left(\frac{c_{i+1} - c_i}{2}\right) \ln \left\{ \frac{\cosh\left(\frac{z-z_i}{\delta_i}\right)}{\cosh\left(\frac{z_i-z_o}{\delta_i}\right)} \right\} + \frac{c_{n+1}}{2}(z-z_o)$$

$$\frac{dT}{dz} = c_1 + \sum_{i=1}^{n} \left(\frac{c_{i+1} - c_i}{2}\right) \left\{\tanh\left(\frac{z-z_i}{\delta_i}\right) + 1\right\}$$

$$c_i = (T_i - T_{i-1})/(z_i - z_{i-1}) \quad .$$

$z = r - r_e$, where r_e is the Earth radius, and r is the radial coordinate of the ray point. Thus, δ_i is the half-thickness of a region centered at approximately z_i km, in which dT/dz changes from c_i to c_{i+1}. Start by drawing a profile using linear segments and get T_i and z_i from the corners. Then select δ_i to round the corners. The final profile will not go through (T_i, z_i).

Specify--

the model check for TTANH5 = 7.0 (W200)

the input data-format code = ______________ (W201)

an input data-set identification number = ______________ (W202)

an 80-character description of the model with parameters:

____

and the profile values:

the number of points in the profile -2 = n = ______________

the profile: i	z_i (km,m)	T_i (°K)	δ_i (km,m)

OTHER MODELS REQUIRED: Any temperature-perturbation model. Use NPTEMP if no perturbations are desired. FUNCTION ALCOSH.

FORM TO SPECIFY INPUT DATA FOR
ATMOSPHERIC TEMPERATURE MODEL TTABLE

This model represents the temperature profile by a sequence of cubic segments such that the temperature gradient is continuous through each profile point. This is not a cubic spline; the coefficients of the cubic fit in each segment depend on only the four nearest profile points.

The coefficients of the cubic are calculated as follows: each set of three successive points in the profile is first fit with a quadratic. The slope of that quadratic at the middle profile point is then assigned to that profile point. This procedure assigns a slope to every profile point except the first and last. A slope of zero is assigned to the first and last point. Between each pair of profile points the coefficients of the cubic are chosen so that the curve goes through the two points and matches the assigned slope at the two points. Those four conditions determine the four coefficients. Both the temperature and its gradient are continuous throughout the profile, even at the profile points.

Specify--

the model check number for TTABLE = ____6.0____(W200)

the input data-format code = ____2.0____(W201)

an input data-set identification number = ____________(W202)

an 80-character description of the profile:

____

and the profile values:

the number of points in the profile, n = ____________

the profile: height (km) Temperature (K)

OTHER MODELS REQUIRED: Subroutine GAUSEL and any temperature-perturbation model. Use NPTEMP if no perturbations are desired.

FORM TO SPECIFY INPUT DATA FOR
ATMOSPHERIC TEMPERATURE-PERTURBATION MODEL TBLOB2

An increase (or decrease) in temperature in a localized region that decays in a Gaussian manner in all three spatial directions.

$$T\,(r,\theta,\phi) = T_o\,(r,\theta,\phi)\;\{1 + \Delta\;\exp\,[-\,(\frac{z-z_o}{W_z})^2 - (\frac{\theta-\theta_o}{W_\theta})^2 - (\frac{\phi-\phi_o}{W_\phi})^2]\}$$

$T_o(r,\theta,\phi)$ is the temperature specified by a temperature model. (r,θ,ϕ) are the coordinates of the ray point in an Earth-centered spherical polar coordinate system. $\theta_o = \pi/2 - \lambda_o$ and $z = r - r_e$, where r_e is the Earth radius. Specify--

the model check for subroutine TBLOB2 = 2.0 (W225)

the input data-format code = ____________ (W226)

an input data-set identification number = ____________ (W227)

an 80-character description for the temperature-perturbation model, including description of parameter values:

____

the strength of the increase (or decrease), Δ = ____________ (W228)

the height of maximum effect, z_o = ____________ km (W229)

the latitude of maximum effect, λ_o = ____________ rad, deg, km N (W230)

the longitude of maximum effect, ϕ_o = ____________ rad, deg, km E (W231)

the Gaussian width in height of the effect, W_z = ____________ km (W232)*

the meridional width of the effect, W_θ = ____________ rad, deg, km (W233)*

the zonal width of the effect, W_ϕ = ____________ rad, deg, km (W234)*

OTHER MODELS REQUIRED: none.

---

* Setting W_z, W_θ, or W_ϕ = zero results in no space variation in that direction.

FORM TO SPECIFY INPUT DATA FOR
ATMOSPHERIC MOLECULAR-WEIGHT MODEL MCONST

A constant molecular weight (independent of height, longitude, latitude, and time)

Specify--

the model check for MCONST = ____1.0____(W250)

the input data-format code = ____________(W251)

an input data-set identification number = ____________(W252)

an 80-character description of the molecular weight:

____

the value of the constant molecular weight, M = ____________(W253)

OTHER MODELS REQUIRED: none.

FORM TO SPECIFY INPUT DATA FOR
VISCOSITY/CONDUCTIVITY MODEL MUARDC

This subroutine calculates the atmospheric molecular viscosity using the ARDC formula for viscosity and calculates atmospheric thermal conductivity from the value of viscosity using a Prandtl number specified by the user. This model is used only to calculate acoustic absorption when either AWWWL or ANWWL is used.

The ARDC formula for viscosity is (U.S. Standard Atmosphere, 1976, p. 19, NOAA, NASA, USAF, U.S. Government Printing Office, Washington, D.C., October 1976)

$$\mu = \beta\ T^{3/2}/(S+T)\ ,$$

where T is the atmospheric temperature in Kelvins.

The atmospheric thermal conductivity using the Prandtl approximation (e.g., Francis Weston Sears, Thermodynamics, Addison-Wesley, 1956, pp. 267-9) is

$$\kappa = \gamma\ R\mu/((\gamma-1)M\ Pr),$$

where γ is the ratio specific heats = 1.4,
R is the universal gas constant,
and M is the mean atmospheric molecular weight.

Specify --

the model check for subroutine MUARDC = ___1.0___(W500)

the input data-format code = __________(W501)

an input data-set identification number = __________(W502)

an 80-character description for the absorption model, including description of parameter values:

____

the viscosity constant, β = __________kg s^{-1} m^{-1} $K^{-1/2}$ (W503)
(1.458×10^{-6} kg s^{-1} m^{-1} $K^{-1/2}$ suggested)

Sutherland's constant, S = __________Kelvins (W504)

(110.4 Kelvins suggested)

Prandtl number, Pr = __________(W505) (0.733 suggested)

OTHER MODELS REQUIRED: Any atmospheric temperature model and any atmospheric molecular weight model.

FORM TO SPECIFY INPUT DATA FOR
PRESSURE MODEL PEXP

This model is used only to calculate absorption when either AWWWL or ANWWL is used. The pressure is given by

$$P = P_0 \exp(-z/H),$$

where z is the height above sea level.

Specify --

the model check for subroutine PEXP = 1.0 (W550)

the input data-format code = __________(W551)

an input data-set identification number = __________(W552)

an 80-character description for the pressure model, including description of parameter values:

____

the pressure at sea level, P_0 = __________Newtons/m^2 (W553)

(1.01328×10^5 Newtons/m^2 suggested)

the pressure scale height, H = __________km, m (W554)

OTHER MODELS REQUIRED: Any pressure-perturbation model. Use NPPRES if no perturbation is desired.

FORM TO SPECIFY INPUT DATA FOR
TERRAIN MODEL GHORIZ

A constant-height terrain model, i.e., a sphere concentric with the Earth.

$$g(r,\theta,\phi) = h - z_o ,$$

where $h = r - r_e$,

$$\frac{\partial g}{\partial r} = 1 , \frac{\partial g}{\partial \theta} = 0 , \frac{\partial g}{\partial \phi} = 0 ,$$

$$\frac{\partial^2 g}{\partial r^2} = \frac{\partial^2 g}{\partial r \partial \theta} = \frac{\partial^2 g}{\partial \theta \partial r} = \frac{\partial^2 g}{\partial r \partial \phi} = \frac{\partial^2 g}{\partial \phi \partial r} = \frac{\partial^2 g}{2\theta 2} = \frac{\partial^2 g}{\partial \theta \partial \phi} = \frac{\partial^2 g}{\partial \phi \partial \theta} = \frac{\partial^2 g}{\partial \phi^2} = 0 ,$$

and r_e is the radius of the Earth.

Specify --

The model check number for GHORIZ = ____1.0____ (W300)

The input data-format code number = ____________ (W301)

The input data-set identification number = ____________ (W302)

an 80-character description of the model including parameters:

____

The constant terrain height, z_o = ____________ km, (W303)

OTHER MODELS REQUIRED: Any terrain-perturbation model. Use NPTERR if no perturbation is desired.

FORM TO SPECIFY INPUT DATA FOR TERRAIN MODEL GLORENZ

An east-west Lorentzian-shaped ridge.

$$g(r,\theta,\phi) = h - z,$$

where $h = r - r_e$,

$$z = z_0/(1 + ((\theta-\theta_0)/\Delta\theta)^2) + z_B ,$$

$$\theta_0 = \pi/2 - \lambda_0 ,$$

and r_e is the radius of the Earth.

Specify--

the model check number for GLORENZ = 4.0 (W300)

the input data-format code number = ____________(W301)

the input data-set identification number = ____________(W302)

an 80-character description of the model including parameters:

____

the height of the ridge, z_0 = ____________km, m (W303)

the latitude of the ridge center, λ_0 = ____________rad, deg, km (W304)

the half-width of the ridge, $\Delta\theta$ = ____________rad, deg, km (W305)

base of the ridge (negative if below sea level) z_B = ________m,km (W306)

OTHER MODELS REQUIRED: Any terrain-perturbation model. Use NPTERR if no perturbation is desired.

FORM TO SPECIFY INPUT DATA FOR TERRAIN MODEL GTANH

This model represents the terrain by a sequence of linear segments that are smoothly joined by hyperbolic functions:

$$g(r,\theta,\phi) = h - z(\theta), \text{ where}$$

$$z(\theta) = z_o + \frac{c_1}{2}(\theta-\theta_o) - \sum_{i=1}^{n} \delta_i \left(\frac{c_{i+1} - c_i}{2}\right) \ln \left\{ \frac{\cosh\left(\frac{\theta-\theta_i}{\delta_i}\right)}{\cosh\left(\frac{(\theta_i-\theta_o)}{\delta_i}\right)} \right\} + \frac{c_{n+1}}{2}(\theta-\theta_o)$$

$$\frac{dz}{d\theta} = c_1 + \sum_{i=1}^{n} \left(\frac{c_{i+1} - c_i}{2}\right) \left\{- \tanh\left(\frac{\theta-\theta_i}{\delta_i}\right) + 1\right\}$$

$$c_i = (z_i - z_{i-1})/(\theta_i - \theta_{i-1})$$

$h = r - r_e$, where r_e is the Earth radius, and r is the radial coordinate of the ray point. $\theta_i = \pi/2 - \lambda_i$. Thus, δ_i is the half-thickness of a region centered at approximately θ_i, in which $dz/d\theta$ changes from c_i to c_{i+1}. Start by drawing a profile using linear segments, and θ_i and z_i from the corners. Then select δ_i to round the corners. The final profile will not go through (θ_i, z_i).

Specify--

the model check for GTANH = 3.0 (W300)

the input data-format code = ______________ (W301)

an input data-set identification number = ______________ (W302)

an 80-character description of the model with parameters:

____

and the profile values:

the number of points in the profile -2 = n = ______________

the profile: i	λ_i (rad,deg)	z_i (km,m)	δ_i (rad,deg)

OTHER MODELS REQUIRED: Any terrain-perturbation model. Use NPTERR if no perturbation is desired. FUNCTION ALCOSH.

FORM TO SPECIFY INPUT DATA
FOR RECEIVER-SURFACE MODEL RHORIZ

A receiver-surface model that is a horizontal surface (i.e., a sphere concentric with the Earth).

$$f(r,\theta,\phi) = h - z_R ,$$

where

$$h = r - r_e$$

and

r_e is the Earth radius

$$\frac{\partial f}{\partial t} = \frac{\partial f}{\partial \theta} = \frac{\partial f}{\partial \phi} = 0$$

$$\frac{\partial f}{\partial r} = 1.0 \quad .$$

Specify--

the model check number for subroutine RHORIZ = ____1.0____(W275)

the input data-format code number = ____________(W276)

an 80-character description of the model including parameters:

____

the receiver surface height, z_R = ____________km (W20)

OTHER MODELS REQUIRED: none.

FORM TO SPECIFY INPUT DATA
FOR RECEIVER-SURFACE MODEL RTERR

A receiver-surface model in which the receiver surface is a fixed height above the terrain surface.

$$f(r,\theta,\phi) = g(r,\theta,\phi) + z_R$$

$$\frac{\partial f}{\partial r} = \frac{\partial g}{\partial r}, \quad \frac{\partial f}{\partial \theta} = \frac{\partial g}{\partial \phi}, \quad \frac{\partial f}{\partial \phi} = \frac{\partial g}{\partial \phi},$$

where $g(r,\theta,\phi)$ and its derivatives are specified in common block /GG/ by the terrain model.

Specify--

the model check number for subroutine RTERR = ~~3.0~~ 2.0 (W275)

the input data-format code number = ____________(W276)

an 80-character description of the model including parameters:

____

the height of the receiver surface above the terrain, z_R = _____km (W20)

OTHER MODELS REQUIRED: Any terrain model.

FORM TO SPECIFY INPUT DATA FOR
RECEIVER-SURFACE MODEL RVERT

A receiver surface that is a vertical (conical) surface a constant distance from a given origin on the Earth's surface

$$f(r,\theta,\phi) = \sin\lambda_o \cos\theta + \cos\lambda_o \sin\theta \cos(\phi-\phi_o) - \cos\alpha_o$$

$$\frac{\partial f}{\partial t} = \frac{\partial f}{\partial r} = 0$$

$$\frac{\partial f}{\partial \theta} = - \sin\lambda_o \sin\theta + \cos\lambda_o \cos\theta \cos(\phi-\phi_o)$$

$$\frac{\partial f}{\partial \phi} = - \cos\lambda_o \sin\theta \sin(\phi-\phi_o) \ .$$

Specify--

the model check number for subroutine RVERT = ~~2.0~~ 3.0 (W275)

the input data-format code number = __________(W276)

an 80-character description of the model including parameters:

____

the distance of the surface from the origin,

α_o = __________rad, deg, km (278)

the latitude of the origin, λ_o = __________rad, deg, km N (W279)

the longitude of the origin, ϕ_o = __________rad, deg, km E (W280)

OTHER MODELS REQUIRED: none.

APPENDIX C: CDC 250 PLOT PACKAGE AND DISSPLA INTERFACE

This appendix describes the plotting commands used by DDPLOT, our local microfilm plotting system, and also an interface called DDSPLA to the DISSPLA* plot package in common use. Figure C1 shows the steps necessary to obtain graphical output from HARPA, if you have DISSPLA. If you do not have DISSPLA and want graphical output on your own plotting system, you will have to insert the equivalent instructions used by your system into a skeleton plotting routine DDALT. This information was taken with permission from "User's Guide to Cathode-Ray Plotter Subroutines" by L. David Lewis, ESSA Technical Memorandum ERL TM-ORSS5, January 1970. The routines used in this version of HARPA assume DISSPLA version 9.0 and are listed in Appendix D.

The CDC-250 Microfilm Recorder, under control of the NOAA Boulder CDC-CYBER 750 computer, plots data on the face of a high-resolution cathode ray tube, which is photographed onto standard size, perforated, 35-mm film.

The plotting area, called a frame, is a square. Plotting positions are described in rectangular coordinates. Coordinate values are integers in the range 0 - 1023; (0,0) is the "lower left-hand corner."

Plotting specifications are transmitted to the DDPLOT routines via the following COMMON.

```
COMMON/DD/IN,IOR,IT,IS,IC,ICC,IX,IY
```

The usage of each of the eight variables is listed below, followed by an explanation of the subroutine calls.

IN	Intensity. IN=0 specifies normal intensity. IN=1 specifies high intensity.
IOR	Orientation IOR=0 specifies upright orientation. IOR=1 specifies rotated orientation (90° counter-clockwise).

*DISSPLA is the proprietary product of ISSCO, Inc.

IT	Italics (Font). IT=0 specifies non-italic (Roman) symbols. IT=1 specifies italic symbols.
IS	Symbol size. IS=0 specifies miniature size. IS=1 specifies small size. IS=2 specifies medium size. IS=3 specifies large size.
IC	Symbol case. IC=0 specifies uppercase. IC=1 specifies lowercase.
ICC	Character code, 0-63 (R1 format). ICC and IC together specify the symbol plotted.
IX	X-coordinate, 0-1023.
IY	Y-coordinate, 0-1023.

CALL DDINIT (N,ID) is required to initialize the plotting process.

ID is a string of characters to identify the person getting the plot and giving the telephone extension and the place to deliver the microfilm plot.

N is the number of characters in the string "ID."

CALL DDBP — defines a vector origin in position IX, IY.

CALL DDVC — plots a vector (straight line), with intensity IN, from the vector origin defined by the previous DDBP or DDVC call, to the vector end position at IX, IY. A single call to DDBP followed by successive calls to DDVC (with changing IX and IY) plots connected vectors.

CALL DDTAB — initializes tabular plotting.

CALL DDTEXT (N,NT) plots a given array in a tabular mode after initiating tabular plotting by using DDTAB, as described above. NT is an array of length N, containing "text" for tabular plotting. Text consists of character codes, packed eight per word (A8 Format). Text characters are plotted as tabular symbols until the command character ≠ (octal code 14, card code 4,8, or the alphabetic shift counterpart of the = on the keypunch) occurs. The command character is not plotted. DDTEXT interprets the next character as a command; after the command is processed, tabular plotting resumes until ≠ is again encountered. ≠ means end of text: DDTEXT returns to the calling routine.

CALL DDFR — causes a frame advance operation. Plotting on the current frame is completed, and the film advances to the next frame.

CALL DDEND — empties the plot buffer and releases the plotting command file to the microfilm plot queue.

DISSPLA CALLS

HARPA calls the following DISSPLA routines directly, rather than through the DDPLOT package. Therefore, if you want to do plotting with a plotting package other than DISSPLA, you will have to convert the following DISSPLA calls to corresponding calls in your plotting package.

CALL DASH — sets dashed-line mode. That is, all plotted curves will be dashed instead of solid.

CALL RESET('DASH') — sets solid-line mode. That is, all plotted curves will be solid lines after this call.

HARPA calls a routine named SETANN. SETANN is an entry point in SUBROUTINE FULANN and also in SUBROUTINE SMPANN. When running HARPA, you must make a choice whether to load FULANN or SMPANN. If you do not have the DISSPLA plotting package, then load SMPANN because it makes no special characater-generating calls to DISSPLA routines. If you do have the DISSPLA plotting package, then load FULANN. SUBROUTINE FULANN calls the following DISSPLA subroutines directly. If you have the DISSPLA plotting package, then the manual will explain the meaning of these routines. If you do not have the DISSPLA plotting package, then load SMPANN, and ignore these routines.

HEIGHT

MX1ALF

MX2ALF

SCMPLX

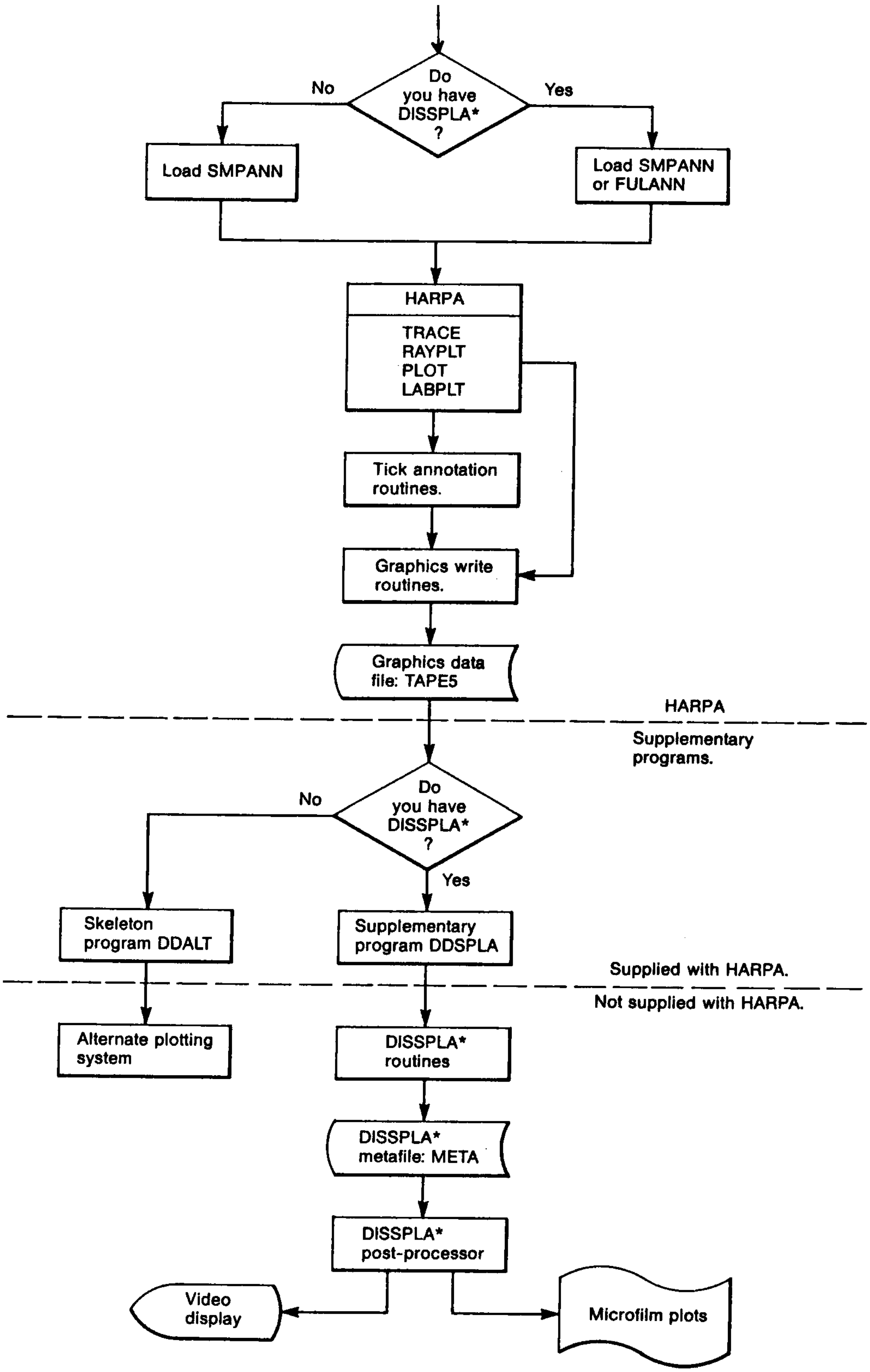


Figure C1. An organization chart that shows how graphical output is produced by a series of programs, some of which are a part of HARPA, and others of which are either supplied along with HARPA or are commercial packages.

APPENDIX D: FORTRAN SOURCE CODE LISTING

This appendix contains the FORTRAN source-code listing for HARPA, including all of its subroutines and atmospheric models. Their order is the same as the order of the programs in Files 3 through 7 of the distribution tape and the list in Section 7.1. Table D1 lists the routines in alphabetical order, the page where the source code can be found, and the approximate size of each module in bytes.

Where DATA statements are used to initialize variables contained in labeled-common blocks, the sequence number (col. 73-80) contains the code "BL." To adhere strictly to the FORTRAN 77 standard, these statements must be put into separate BLOCK DATA modules. Most FORTRAN environments permit our syntax, however.

****

WARNING: THIS LISTING IS PROVIDED FOR INFORMATION PURPOSES ONLY. THOSE WHO WANT TO USE THE PROGRAM MUST OBTAIN THE SOURCE CODE ON MAGNETIC TAPE FROM THE AUTHORS (SEE SECTION 3.1). COPYING THE CODE LISTED HERE WILL NOT PRODUCE A USABLE PROGRAM.

****

Table D1--Alphabetical list of source-code modules

Module Name	Source Bytes	Page	Module Name	Source Bytes	Page
ALCOSH	384	295	NUMSTG	512	291
ANWNL	4736	321	OPNREP	896	294
ANWWL	5376	326	OVERRD	1024	294
ARCTIC	1792	316	PCROSS	768	257
ATMOSHD	8192	286	PEXP	3712	385
AWWNL	5120	323	PLOT	3200	302
AWWWL	5760	329	PLOTBL	128	317
BACKUP	3968	264	PLTANH	2816	309
CBLOB2	4480	354	PLTANOT	2432	311
CLEAR	256	250	PLTHLB	128	309
CONBLK	7424	273	PLTLB	1792	314
CSTANH	4992	349	PRINTR	14720	278
DASH	128	319	PUTDES	512	291
DDALT	6016	405	PUTKST	2176	293
DDBP	512	317	RAYPLT	8320	297
DDEND	256	319	RAYTRC	8576	229
DDFR	128	319	RCROSS	2688	257
DDINIT	256	317	READW	8960	243
DDSPLA	15488	399	READW1	6656	237
DDTAB	128	318	REFLECT	2944	266
DDTEXT	384	318	RENORM	512	278
DDVC	512	318	RERR	512	292
DFCNST	1536	234	RERROR	256	293
DFSYS	1152	233	RESET	128	319
DRAWTKS	1280	313	RHORIZ	3584	390
FIT	2432	268	RKAM	1408	260
FULANN	1792	397	RKAM1	3072	262
GAMRTDM	3968	347	RTERR	3584	391
GAUSEL	2688	295	RVERT	8832	394
GET	3456	271	SCMPLX	128	320
GET1	2944	269	SET2	128	278
GHORIZ	3328	373	SETXY	384	307
GLORENZ	4480	375	SFILL	512	292
GTANH	5376	378	SFILTR	128	295
GTUNIT	128	240	SMPANN	1536	396
HAMLTN	2944	258	SREAD1	512	242
HEIGHT	128	320	STOPIT	256	293
ITEST	256	271	STRIM	512	292
LABPLT	6400	305	TBLOB2	4480	367
MCONST	3456	371	TIKLINE	640	311
MUARDC	4096	381	TLINEAR	3840	356
MX1ALF	128	320	TRACE	9344	252
MX2ALF	128	320	TTABLE	6016	361
ND2B	384	250	TTANH5	5120	359
NPABSR	3200	384	UCON	2432	250
NPPRES	3072	388	ULOGZ2	4480	337
NPSPEED	3072	352	VVORTX3	4992	340
NPTEMP	3072	369	WCHANGE	512	277
NPTERR	384	381	WGAUSS2	4480	343
NPWIND	3456	345	WLINEAR	4096	333
NTEMP	3328	365	WTIDE	4352	335

RAY-TRACING CORE (Tape File 3)

```
      PROGRAM RAYTRC                                                    RAYTRC 2
C     MAIN PROGRAM FOR THE RAY TRACING PACKAGE.                         RAYTRC 3
C          SETS THE INITIAL CONDITIONS FOR EACH RAY AND CALLS TRACE     RAYTRC 4
C                                                                       RAYTRC 5
C     COMMON DECK "RAYDEV" INSERTED HERE                                CRAYDEV2
C     DEVICE ASSIGNED TO RAYTRC INPUT FILE                              CRAYDEV4
      COMMON/RAYDEV/NRYIND,NDEVTMP,NFRMAT,NDEVGRP,NDEVBIN               CRAYDEV5
      LOGICAL WCHANGE,FIRST                                             RAYTRC 7
      REAL WS(400)                                                      RAYTRC 8
C     COMMON DECK "FILEC" INSERTED HERE                                 CFILEC 2
      COMMON /FILEC/NPLTDP                                              CFILEC 4
C     COMMON DECK "CERR" INSERTED HERE                                  CERR   2
      COMMON/ERR/NERG,NERR,NERT,NERP                                    CERR   3
C     COMMON DECK "GG" INSERTED HERE                                    CGG    2
      REAL MODG                                                         CGG    4
      COMMON/GG/MODG(4)                                                 CGG    5
      COMMON/GG/G,PGR,PGRR,PGRTH,PGRPH                                  CGG    6
      COMMON/GG/PGTH,PGPH,PGTHTH,PGPHPH,PGTHPH,GSELECT,GTIME            CGG    7
C     COMMON DECK "HDR" INSERTED HERE                                   CHDR   2
      CHARACTER*10 INITID*80,DAT,TOD                                    CHDR   4
      COMMON/HDR/SEC                                                    CHDR   5
      COMMON/HDRC/INITID,DAT,TOD                                        CHDR   6
C     COMMON DECK "CONST" INSERTED HERE                                 CCONST 2
      COMMON/PCONST/CREF,RGAS,GAMMA                                     CCONST 4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                         CCONST 5
C     COMMON DECK "FLAG" INSERTED HERE                                  CFLAG  2
      LOGICAL NEWWR,NEWWP,NEWTRC,PENET                                  CFLAG  4
      COMMON /FLG/ NTYP,NEWWR,NEWWP,NEWTRC,PENET,LINES,IHOP,HPUNCH      CFLAG  5
      COMMON/FLGP/NSET                                                  CFLAG  6
C     COMMON DECK "RINPLEX" INSERTED HERE                               CRINPLE2
      REAL KAY2,KAY2I                                                   CRINPLE4
      COMPLEX PNP,POLAR,LPOLAR                                          CRINPLE5
      LOGICAL SPACE                                                     CRINPLE6
      CHARACTER DISPM*6                                                 CRINPLE7
      COMMON/RINPL/DISPM                                                CRINPLE8
      COMMON /RIN/ MODRIN(8),RAYNAME(2,3),TYPE(3),SPACE                 CRINPLE9
      COMMON/RIN/OMEGMIN,OMEGMAX,KAY2,KAY2I                             CRINPL10
      COMMON/RIN/PNP(10),POLAR,LPOLAR,SGN                               CRINPL11
C     COMMON DECK "RK" INSERTED HERE                                    CRK    2
C     DEFINE SIZE REQUIRED FOR RAY STATE SAVE ARRAY                     CRK    4
      PARAMETER (LRKAMS=87+2*100,NXRKMS=12+LRKAMS,MXEQPT=21)            CRK    5
      PARAMETER (NRKSAV=NXRKMS+MXEQPT-1)                                CRK    6
      COMMON /RK/ NEQS,STEP,MODE,E1MAX,E1MIN,E2MAX,E2MIN,FACT,RSTART    CRK    7
C     COMMON DECK "RKAM" INSERTED HERE                                  RKAMCOM2
      REAL KR,KTH,KPH                                                   RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)       RKAMCOM5
C     COMMON DECK "WWR" INSERTED HERE                                   CWWR   2
      PARAMETER (NWARSZ=1000)                                           CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
     8 (RCVRH,W(20)),                                                   CWW2   7
```

```
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),
     6 (HMIN,W(27)),(RGMAX,W(28)),
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))
C
      CHARACTER TMP80*80
C
      DATA FIRST/.TRUE./
C
C     INITIALIZE FILES , SET CONTANTS AND MAKE SYSTEM CALLS
      CALL STDINI
C     INITIALIZE LINE COUNTS FOR CURRENT AND LAST HEADER
      LINES=0
      LHDRPG=0
C
      REWIND NRYIND
C  INITIALIZE RAYSET FILE
5     READ(NRYIND,'(A)',END=8) TMP80
      WRITE(9,'(A)') TMP80
      GO TO 5
C
8     WRITE(9,'(A)') '****SOFT EOF***+++++'
      REWIND NRYIND
C     READ IN USERS NAME AND TELEPHONE EXTENSION FOR IDENTIFYING
C     MICROFILM PLOTS.
C
C     REPOSITION DINP AFTER IDENTIFIER
      READ(NRYIND,'(A)',END=6000) TMP80
C
C********* READ W ARRAY AND PRINT NON-ZERO VALUES
      IRUN=0
      NSET=0
C
10    CALL READW
      NSET=NSET+1
      ICODE=ND2B(INT(RAYFNC))
C PROCESS RAYPATH CALCULATIONS ONLY IF W(29) IS BEING USED
      IF((RAYFNC.NE.0.).AND.(AND(ICODE,4).EQ.0)) GO TO 10
C
C********* LET ROUTINES PRINTR AND RAYPLT KNOW THERE IS A NEW W ARRAY
      NEWWP=.TRUE.
      NEWWR=.TRUE.
      IRUN=IRUN+1
C
C     SET BINARY RAY FILE UNIT TO ZERO IF NO OUTPUT WANTED
      NPLTDP=0
      IF(BINRAY.NE.0.0) NPLTDP=NDEVBIN
C
C******** INITIALIZE THE MODELS VIA 'DISPER'
      CALL IDISPER
```

```
C
      IF(FIRST.AND.PLT.NE.0.) CALL DDINIT(8,INITID)
      IF(PLT.NE.0.) FIRST=.FALSE.
C
C     PREVENT RERUNNING SAME MODEL CASES
      IF(IRUN.EQ.1) GO TO 12
      IF(.NOT.WCHANGE(WS(1),W(1))) GO TO 10
      PRINT *,'PROCESSING FOR RUNSET #',NSET
C
12    CALL RMOVE(WS,W,400)
C
      OW=0.
      BETA=0.
      AZ1=0.
C
      CALL HEADER1
C
C********* PRINT OUT THE CONTENTS OF THE 'W' ARRAY
C********* DETERMINE TRANSMITTER LOCATION IN COMPUTATIONAL COORDINATE
C********* SYSTEM (GEOMAGNETIC COORDINATES IF DIPOLE FIELD IS USED)
      SP=SIN (PLAT)
      CP=SIN (PID2-PLAT)
      SDPH=SIN (TLON-PLON)
      CDPH=SIN (PID2-(TLON-PLON))
      SL=SIN (TLAT)
      CL=SIN (PID2-TLAT)
      ALPHA=ATAN2(-SDPH*CP,-CDPH*CP*SL+SP*CL)
      TH0=ACOS (CDPH*CP*CL+SP*SL)
      PH0=ATAN2(SDPH*CL,CDPH*SP*CL-CP*SL)
C
      R=EARTHR
      TH=TH0
      PH=PH0
      CALL TOPOG
C
C     OBTAIN ABSOLUTE HEIGHT OF THE TRANSMITTER.
C     IF IT WAS SPECIFIED AS RELATIVE TO THE TERRAIN, THEN REMOVE
C     THE FLAG VALUE 10**40 WHICH WAS ADDED AT INPUT.
C
      TMP=XMTRH
      IF(XMTRH .EQ. 1.E-40) XMTRH=0.0
      IF( ABS(XMTRH) .GE. 1.E20 ) XMTRH=XMTRH*1.E-40
      IF(TMP.NE.XMTRH) XMTRH=XMTRH-G/PGR
C
C     CHECK THAT TRANSMITTER IS ABOVE TERRAIN.
C
      IF(-G/PGR .LE. XMTRH) GO TO 655
C
      PRINT 640, IRUN
      WRITE(3,640) IRUN
640   FORMAT('0****** TRANSMITTER BELOW TERRAIN. RUN ',I3
     1 ,' TERMINATED.'/'SEE W-ARRAY PRINTOUT. INPUT CONTINUES.'//)
      CALL SETOVR
      CALL PRINTW
      GO TO 10
```

```
C
655   CALL SETOVR
      CALL PRINTW
      CALL IPRINTR
C******** INITIALIZE PRINT CONTROL PARAMTERS
      LINSPP=PAGLN
      LNPHPG=LINSPP/2
      IF(LNPHPG.LT.40) LNPHPG=LINSPP
C
C********* LOOP ON FREQUENCY, AZIMUTH ANGLE, AND ELEVATION ANGLE
      NFREQ=1
      IF (FSTEP.NE.0.) NFREQ=(FEND-FBEG)/FSTEP+1.5
      NAZ=1
      IF (AZSTEP.NE.0.) NAZ=(AZEND-AZBEG)/AZSTEP+1.5
      NBETA=1
      IF (ELSTEP.NE.0.) NBETA=(ELEND-ELBEG)/ELSTEP+1.5
      DO 50 NF=1,NFREQ
      OW=FBEG+(NF-1)*FSTEP
      DO 45 J=1,NAZ
      AZ1=AZBEG+(J-1)*AZSTEP
      AZA=AZ1*DEGS
      GAMMA1=PI-AZ1+ALPHA
      SGAMMA=SIN (GAMMA1)
      CGAMMA=SIN (PID2-GAMMA1)
      DO 40 I=1,NBETA
      BETA=ELBEG+(I-1)*ELSTEP
      EL=BETA*DEGS
      CBETA=SIN (PID2-BETA)
      R=EARTHR+XMTRH
      TH=TH0
      PH=PH0
      KR=SIN (BETA)
      KTH=CBETA*CGAMMA
      KPH=CBETA*SGAMMA
      TPULSE=0.
      RSTART=1.
C********* THE FOLLOWING LINE NEEDED FOR RAY TRACING IN COMPLEX SPACE
      SGN=1.0
C********* CALL MODELS
      CALL DISPER
C
      LINPG=LINES-LHDRPG
      IF(I.EQ.1  .OR.  LINPG.GE.LINSPP-20) THEN
C
C     PUT OUT SUBHEADERS FROM MEDIA AND PRINTR ROUTINES
         CALL HEADER2
         CALL PRNHD1(' ')
C
C     COMPUTE LINE COUNT OF THIS HEADER
         LHDRPG=LINES/LINSPP*LINSPP
      ELSEIF(LINPG.GE.LNPHPG-10 .AND. LINPG.LE.LNPHPG) THEN
C     PUT OUT PAGE FEED WITH SUBHEADER IF AT HALF PAGE
         CALL PRNHD2('1')
      ELSE
C     PUT OUT SUBHEADER
```

```
         CALL PRNHD2(' ')
      ENDIF
C
      IF (KAY2.GT.0.) GO TO 30
      WRITE(3,2900) OMEGMIN,OMEGMAX
 2900 FORMAT (58H0TRANSMITTER IN EVANESCENT REGION, TRANSMISSION IMPOSSI
     1BLE/20H0MINIMUM FREQUENCY =,E17.10,20H MAXIMUM FREQUENCY =,E17.10)
      GO TO 44
30    CALL RENORM(KR,KAY2,3)
      CALL CLEAR(RKVARS,NEQS-6)
      CALL TOPOG
      IF(NERG.GT.0) RKVARS(NERG)=G
      IF(NERR.GT.0) RKVARS(NERR)=PGR
      IF(NERT.GT.0) RKVARS(NERT)=PGTH
      IF(NERP.GT.0) RKVARS(NERP)=PGPH
C
C     CALCULATE ONE RAY PATH
      CALL TRACE
      OSEC=SEC
      CALL SYSSEC(SEC)
      DIFF=SEC-OSEC
C
C     ADD TO LINES COUNT FOR ELAPSED TIME REPORT
      LINES=LINES+2
      WRITE(3,3500) DIFF
3500  FORMAT (/T93,'THIS RAY CALCULATION TOOK ',F8.3,' SEC')
C
      IF (PENET.AND.ONLY.NE.0..AND.IHOP.EQ.1) GO TO 44
   40 CONTINUE
44    IF(PLT.GT.0.AND.(NAZ.LE.1.OR.NBETA.GT.1)) CALL ENDPLT
   45 CONTINUE
      IF(PLT.GT.0.AND.NAZ.GT.1.AND.NBETA.LE.1)CALL ENDPLT
      IF(PENET.AND.ONLY.NE.0..AND.IHOP.EQ.1.AND.NAZ.EQ.1.AND.NBETA.EQ.1)
     1 GO TO 55
   50 CONTINUE
   55 IF (RAYSET.NE.0.) WRITE(9,5000)
 5000 FORMAT (78X,1H-)
      GO TO 10
6000  PRINT *,'DINP EMPTY OR NOT FOUND'
      STOP
      END



      SUBROUTINE DFSYS
      CHARACTER DAT*(*),DATX*10
      CHARACTER TIM*(*),TIMX*10
C
C     ALL ROUTINES WHICH RELY ON SYSTEM BASED FUNCTIONS, SUCH AS TIME
C     OF DAY OR MAY BENEFIT FROM SYSTEM AVAILABLE ROUTINES FOR BETTER
C     PERFORMANCE, SUCH AS 'RMOVE' HAVE BEEN COLLECTED IN THIS ROUTINE.
C
      ENTRY RMOVE(X,Y,N)
```

```
C     MOVE 'N' COMPONENTS OF ARRAY 'Y' TO ARRAY 'X'.
C     THIS LOGIC WILL HANDLE OVERLAP CASES ONLY FOR X(M)=X(M+N1)
C     WITH N1>0, M=1,N.
      REAL X(N),Y(N)
C
      IF(N.LE.0) RETURN
C     USE CYBER BUILT-IN ROUTINE
C      CALL MOVLEV(Y,X,N)
C
C     THE EQUIVALENT FORTRAN CODING FOR THIS ROUTINE ARE THE FOLLOWING
C     TWO LINES.
      DO 10 I=1,N
10    X(I)=Y(I)
C
      RETURN
      ENTRY SYSSEC(SECS)
C
C     OBTAIN CURENT TIME OF DAY IN SECONDS FROM SYSTEM ROUTINE
      SECS=0
C     SYSTEM CALL NEEDED FOR THE CYBER NOS OP SYS
C      CALL SECOND(SECS)
C     SYSTEM CALL NEEDED FOR CRAY COMPUTER UNDER CTSS
C      CALL TIMEUSED(ICPU,IO,ISYS,MEM)
C      SECS=ICPU/1.E6
      RETURN
C
      ENTRY SYSTIM(TIM)
C
C     OBTAIN CURENT TIME OF DAY AS A CHARACTER STRING FROM SYSTEM ROUTI
      TIM='TIME'
C     SYSTEM CALL NEEDED FOR THE CYBER NOS OP SYS
C      CALL TIME(TIM)
C     SYSTEM CALL NEEDED FOR THE CRAY CTSS OP SYS
C     CALL TIMEDATE(TIM,DATX,MACH)
      RETURN
C
      ENTRY SYSDAT(DAT)
C
C     OBTAIN CURENT DATE AS A CHARACTER STRING FROM SYSTEM ROUTINE
      DAT='DATE'
C     SYSTEM CALL NEEDED FOR THE CYBER NOS OP SYS
C      CALL DATE(DAT)
C     SYSTEM CALL NEEDED FOR THE CRAY CTSS OP SYS
C     CALL TIMEDATE(TIMX,DAT,MACH)
      RETURN
      ENTRY MORTEM
      STOP 'POST MORTEM'
      END


      SUBROUTINE DFCNST
C
```

```
C     THIS MODULE PROVIDES CONSTANTS WHICH ARE EXPLICITLY DEPENDENT     DFCNST 7
C     ON MACHINE ARCHITECTURE AND SHOULD BE EXAMINED WHEN MAKING ANY    DFCNST 8
C     MIGRATION OF THE RAY TRACING ROUTINES.                            DFCNST 9
C                                                                       DFCNST10
C     COMMON DECK "RMACH" INSERTED HERE                                 CRMACH 2
      COMMON/CRMACH/RMACH(5)                                            CRMACH 4
C                                                                       DFCNST12
C     SINGLE PRECISION MACHINE CONSTANTS                                DFCNST13
C      (SEE DIGITAL SIGNAL PROCESSING, IEEE PRESS 1979 P S-7)           DFCNST14
C                                                                       DFCNST15
C     THE CONSTANTS ARE DERIVED FROM THE FOLLOWING                      DFCNST16
C     FORM FOR FLOATING POINT NUMBERS:                                  DFCNST17
C                                                                       DFCNST18
C        SIGN (B**E)*( (X(1)/B) + ... + (X(T)/B**T) )                   DFCNST19
C     FOR MOST MACHINES THE BASE B=2.                                   DFCNST20
C     FOR THE CYBER 60-BIT MACHINES                                     DFCNST21
C        T=48, EMIN=-974 AND EMAX=1070.                                 DFCNST22
C     FOR THE PDP-11 AND VAX-11 MACHINES                                DFCNST23
C        T=24, EMIN=-127 AND EMAX=127                                   DFCNST24
C                                                                       DFCNST25
C     RMACH(1)  SMALLEST POSITIVE MAGNITUDE = B**(EMIN-1)               DFCNST26
C     RMACH(2)  LARGEST MAGNITUDE           = B**EMAX*(1-B**(-T))       DFCNST27
C     RMACH(3)  SMALLEST RELATIVE SPACING   = B**(-T)                   DFCNST28
C     RMACH(4)  LARGEST RELATIVE SPACING    = B**(1-T)                  DFCNST29
C     RMACH(5)                              = LOG10(B=2)                DFCNST30
C                                                                       DFCNST31
C     CONSTANTS HERE ARE FOR CDC 6000/7000/CYBER SERIES COMPUTERS       DFCNST32
C     PLEASE SUBSTITUTE FOR OTHER MACHINES                              DFCNST33
C                                                                       DFCNST34
C                                                                       DFCNST35
C      INTEGER SMALL(2)                                                 DFCNST36
C      INTEGER LARGE(2)                                                 DFCNST37
C      INTEGER RIGHT(2)                                                 DFCNST38
C      INTEGER DIVER(2)                                                 DFCNST39
C      INTEGER LOG10(2)                                                 DFCNST40
CC                                                                      DFCNST41
C      EQUIVALENCE (RMACH(1),SMALL(1))                                  DFCNST42
C      EQUIVALENCE (RMACH(2),LARGE(1))                                  DFCNST43
C      EQUIVALENCE (RMACH(3),RIGHT(1))                                  DFCNST44
C      EQUIVALENCE (RMACH(4),DIVER(1))                                  DFCNST45
C      EQUIVALENCE (RMACH(5),LOG10(1))                                  DFCNST46
C                                                                       DFCNST47
C     MACHINE CONSTANTS FOR CYBER 170/180 COMPUTERS                     DFCNST48
C                                                                       DFCNST49
      DATA RMACH(1) / O"00014000000000000000" /                         DFCNST50
      DATA RMACH(2) / O"37767777777777777777" /                         DFCNST51
      DATA RMACH(3) / O"16404000000000000000" /                         DFCNST52
      DATA RMACH(4) / O"16414000000000000000" /                         DFCNST53
      DATA RMACH(5) / O"17164642023241175720" /                         DFCNST54
C                                                                       DFCNST55
C***REFERENCES  FOX, P.A., HALL, A.D., SCHRYER, N.L, *FRAMEWORK FOR     DFCNST56
C                 A PORTABLE LIBRARY*, ACM TRANSACTIONS ON MATHE-       DFCNST57
C                 MATICAL SOFTWARE, VOL. 4, NO. 2, JUNE 1978,           DFCNST58
C                 PP. 177-188.                                          DFCNST59
C***ROUTINES CALLED  XERROR                                             DFCNST60
```

```
C***END PROLOGUE  R1MACH                                                DFCNST61
C     MACHINE CONSTANTS FOR VAX 11/780                                  DFCNST62
C     (EXPRESSED IN INTEGER AND HEXADECIMAL)                            DFCNST63
C    ***THE HEX FORMAT BELOW MAY NOT BE SUITABLE FOR UNIX SYSYEMS***    DFCNST64
C    *** THE INTEGER FORMAT SHOULD BE OK FOR UNIX SYSTEMS***            DFCNST65
C                                                                       DFCNST66
C     DATA SMALL(1), SMALL(2) /       128,           0 /                DFCNST67
C     DATA LARGE(1), LARGE(2) /    -32769,          -1 /                DFCNST68
C     DATA RIGHT(1), RIGHT(2) /      9344,           0 /                DFCNST69
C     DATA DIVER(1), DIVER(2) /      9472,           0 /                DFCNST70
C     DATA LOG10(1), LOG10(2) /  546979738,  -805796613 /               DFCNST71
C                                                                       DFCNST72
C     DATA SMALL(1), SMALL(2) / Z00000080, Z00000000 /                  DFCNST73
C     DATA LARGE(1), LARGE(2) / ZFFFF7FFF, ZFFFFFFFF /                  DFCNST74
C     DATA RIGHT(1), RIGHT(2) / Z00002480, Z00000000 /                  DFCNST75
C     DATA DIVER(1), DIVER(2) / Z00002500, Z00000000 /                  DFCNST76
C     DATA LOG10(1), LOG10(2) / Z209A3F9A, ZCFF884FB /                  DFCNST77
C                                                                       DFCNST78
C   MACHINE CONSTANTS FOR VAX 11/780 (G-FLOATING)                       DFCNST79
C     (EXPRESSED IN INTEGER AND HEXADECIMAL)                            DFCNST80
C    ***THE HEX FORMAT BELOW MAY NOT BE SUITABLE FOR UNIX SYSYEMS***    DFCNST81
C    *** THE INTEGER FORMAT SHOULD BE OK FOR UNIX SYSTEMS***            DFCNST82
C                                                                       DFCNST83
C     DATA SMALL(1), SMALL(2) /        16,           0 /                DFCNST84
C     DATA LARGE(1), LARGE(2) /    -32769,          -1 /                DFCNST85
C     DATA RIGHT(1), RIGHT(2) /     15552,           0 /                DFCNST86
C     DATA DIVER(1), DIVER(2) /     15568,           0 /                DFCNST87
C     DATA LOG10(1), LOG10(2) /  1142112243, 2046775455 /               DFCNST88
C                                                                       DFCNST89
C     DATA SMALL(1), SMALL(2) / Z00000010, Z00000000 /                  DFCNST90
C     DATA LARGE(1), LARGE(2) / ZFFFF7FFF, ZFFFFFFFF /                  DFCNST91
C     DATA RIGHT(1), RIGHT(2) / Z00003CC0, Z00000000 /                  DFCNST92
C     DATA DIVER(1), DIVER(2) / Z00003CD0, Z00000000 /                  DFCNST93
C     DATA LOG10(1), LOG10(2) / Z44133FF3, Z79FF509F /                  DFCNST94
C     MACHINE CONSTANTS FOR THE CRAY 1                                  DFCNST95
C                                                                       DFCNST96
C     DATA SMALL(1) / 201354000000000000000B /                          DFCNST97
C     DATA SMALL(2) / 000000000000000000000B /                          DFCNST98
C                                                                       DFCNST99
C     DATA LARGE(1) / 577767777777777777777B /                          DFCNS100
C     DATA LARGE(2) / 000007777777777777774B /                          DFCNS101
C                                                                       DFCNS102
C     DATA RIGHT(1) / 376434000000000000000B /                          DFCNS103
C     DATA RIGHT(2) / 000000000000000000000B /                          DFCNS104
C                                                                       DFCNS105
C     DATA DIVER(1) / 376444000000000000000B /                          DFCNS106
C     DATA DIVER(2) / 000000000000000000000B /                          DFCNS107
C                                                                       DFCNS108
C     DATA LOG10(1) / 377774642023241175717B /                          DFCNS109
C     DATA LOG10(2) / 000007571421742254654B /                          DFCNS110
C                                                                       DFCNS111
C                                                                       DFCNS112
      END                                                               DFCNS113
```

```
      SUBROUTINE READW1(AB,NWOK,MSET,MX,NTBL,ITBL,FRMTBL,GP)            READW1 2
C     HANDLES INPUT OF TABULAR DATA REQUIRED BY SOME MODELS.            READW1 3
C     COMMON DECK "RAYDEV" INSERTED HERE                                CRAYDEV2
C     DEVICE ASSIGNED TO RAYTRC INPUT FILE                              CRAYDEV4
      COMMON/RAYDEV/NRYIND,NDEVTMP,NFRMAT,NDEVGRP,NDEVBIN               CRAYDEV5
C     COMMON DECK "CUCON" INSERTED HERE                                 CUCON  2
      COMMON/UCONV/CNVV(4,4)                                            CUCON  4
      CHARACTER PCV*3,CNVC*2                                            CUCON  5
      COMMON/UCONC/PCV(4),CNVC(4,4)                                     CUCON  6
C     COMMON DECK "CONST" INSERTED HERE                                 CCONST 2
      COMMON/PCONST/CREF,RGAS,GAMMA                                     CCONST 4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                         CCONST 5
C     COMMON DECK "WWR" INSERTED HERE                                   CWWR   2
      PARAMETER (NWARSZ=1000)                                           CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
     8 (RCVRH,W(20)),                                                   CWW2   7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2   9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2  10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2  11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2  12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2  13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2  15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2  16
      LOGICAL NWOK,AB                                                   READW1 8
C                                                                       READW1 9
      CHARACTER LINE*80,ALPHA*3,FBUF*3                                  READW110
      PARAMETER (NVAR=6,ALPHA='  A',NACDE=-1,NFIELD=13)                 READW111
      INTEGER IPV(NVAR),NTBL(10),ITBL(10),FRMTBL(10)                    READW112
      CHARACTER*8 PU(NVAR)                                              READW113
      REAL GP(*),V(NVAR),CONV(NVAR)                                     READW114
C                                                                       READW115
C     THE DATA STRUCTURE EXPECTED BY THIS PROGRAM FOR A DATA BLOCK IS   READW116
C     SPECIFIED FOR A GIVEN MODEL BY THE THREE ARRAYS NTBL,ITBL,FRMTBL. READW117
C     THE TYPE OF DATA EXPECTED FOR A GIVEN FORMAT GROUP IS TAKEN FROM  READW118
C     ARRAY FRMTBL. IT MUST HAVE AN ENTRY FOR A GIVEN TYPE OF FORMAT    READW119
C     WHICH IS EITHER ZERO, IF THAT FORMAT IS NOT ALLOWED OR AN ENTRY   READW120
C     WHOSE VALUE IS EQUAL TO THAT FORMAT TYPE. CURRENTLY ONLY TWO      READW121
C     FORMAT TYPES ARE ALLOWED, 1 FOR ALPHA AND 2 FOR NUMERIC.          READW122
C     TO ALLOW MORE FLEXIBLITY IN THE INPUT FILE THE NUMERIC FORMAT     READW123
C     TYPE IN TURN ALLOWS FOR 3 DIFFERENT DATA FORMATS, SEE BELOW.      READW124
C                                                                       READW125
C     OFFSETS INTO THE GENERAL ARRAY 'GP' FOR EACH FORMAT ARE GIVEN     READW126
C     IN THE ARRAY 'NTBL'. THE SPACING BETWEEN DATA VALUES IN EACH LINE READW127
C     READ IS GIVEN IN ARRAY 'ITBL'. THIS SCHEME DOES NOT ALLOW FOR X,Y READW128
C     ,Z CYCLES BUT RATHER SEPARATE ARRAYS OF X VALUES, THEN Y VALUES,  READW129
```

```
C      AND THEN Z VALUES(ETC. UP TO 'NVAR' VARIABLES).
C
       NWOK=.TRUE.
C
C      INITIALIZE ANY ALPHANUMERIC ARRAYS TO BLANKS
       IF(FRMTBL(1).EQ.1) CALL SET2(GP(NTBL(1)),1H ,NTBL(2)-NTBL(1))
C
C      BEGIN MULTI-FORMAT LOOP
5      READ(NRYIND,'(A,A)',END=200) FBUF,LINE
       IF(NDEVTMP.GT.0) WRITE(NDEVTMP,'(A,A)') FBUF,LINE
       IF(FBUF.EQ.ALPHA) THEN
          NFRM=NACDE
       ELSE
          READ(FBUF,10) NFRM
       ENDIF
10     FORMAT(BZ,I3,A)
       IF(NFRM .EQ. 0) RETURN
C
C      FORMAT #1 IS ASSIGNED TO ALPHANUMERIC INPUT AND LARGER NUMBERS
C      FOR NUMERIC INPUT.
C
C      CHECK NOW FOR NUMERICAL FORMATS.
C      THESE ARE PROVIDED FOR EASE OF INPUT OF TABULATED DATA
C      AND ARE PARTLY 'TRANSPARENT' TO THE MODEL INVOLVED. I.E.
C      THE USER HAS THE OPTION OF SPECIFING THE NUMBER OF DATA COLUMNS.
C      INPUT FORMAT NUMBER SPECIFIES THE NUMBER OF COLUMNS OF INPUT DATA
C      I.E. FOR VALUE 1 A SINGLE DATA COLUMN IS EXPECTED, FOR A VALUE
C      2 TWO DATA COLUMNS ARE EXPECTED, ETC.
C
C      DETERMINE FORMAT NUMBER
       IF(NFRM.EQ.NACDE) THEN
          IT=1
C      ALPHANUMERIC FORMATS MUST BE FIRST IN LIST
C      SINCE NO OFFSET SPECIFIER IS ALLOWED IN A FULL ALPHA LINE
          IF(FRMTBL(1).EQ.1) GO TO 100
       ELSE
          READ(LINE,'(I2)') IT
          IT=MAX0(IT,2)
          IF(IT.LE.MX) GO TO 100
       ENDIF
C
15     AB=.TRUE.
       PRINT 20, MSET,NFRM,MX
20     FORMAT(' FOR SET',I5,' FORMAT NUMBER',I5,' EXCEEDS LIMIT OF',I5)
C       ERROR IN INPUT DATA , STOP HERE
       STOP
C
C      THE 'NTBL' TABLE PROVIDES A LIST OF OFFSETS TO VARIABLE GROUPS IN
C      MODEL ARRAYS, 'ITBL' GIVES ELEMENT SEPARATION TO ALLOW FOR 2-
C      DIMENSIONAL ARRAYS AND 'FRMTBL' SPECIFIES THE INPUT FORMAT.
C
100    N1=NTBL(IT)
       N3=ITBL(IT)
       N2=NTBL(IT+1)-N3
C
```

```
      IF(NFRM.NE.NACDE) GO TO 110
C
      READ(LINE,1010)((GP(I+J-1),I=N1,N2,N3),J=1,N3)
1010  FORMAT(10A8)
      GO TO 5
C
110   IF(NFRM.LT.1 .OR. NFRM.GT.NVAR) GO TO 200
C     HANDLE NUMERIC FORMATS
C
C     N1 IS STARTING ELEMENT WHICH WILL BE AFTER THE DATA COUNT
C     VALUE(NEED NELS*N3+1 TOTAL ELEMENTS)
      N1=N1+1
      N=0
C
C     # ELEMENTS IS EQUAL TO FORMAT NUMBER
      NELS=NFRM
C     READ UNITS CONVERSION SPECIFICATION LINE
      READ(LINE,1018) (PU(I),I=1,3)
      IF(PU(2).NE.' ') READ(LINE,2024) DELIM
      IF(PU(2).EQ.' ') READ(LINE,1020) DELIM,(PU(I),I=1,NELS)
C
C  ALLOW FOR BLANK CONVERSION LINE PRODUCED BY EIGENRAY ROUTINE
      IF(DELIM.EQ.0) READ(LINE,2024) DELIM
C
C     READ NEXT LINE AND EXTRACT UNIT CONVERSION SPECS IF PRESENT
C     IN FLOATING FORMAT
      READ(NRYIND,'(A)') LINE
      CALL GTUNIT(LINE,PU,NFIELD,NELS, NCARY)
C     CARRY COUNT IS ZERO IF LINE CONTAINED UNITS SPECIFIERS
      IF(NCARY.EQ.0.AND.NDEVTMP.GT.0) WRITE(NDEVTMP,'(A)') LINE
C
1018  FORMAT(10(A,5X))
1020  FORMAT(BZ,3X,G10.3,8X,10A)
2024  FORMAT(T43,G10.3)
1025  FORMAT(BZ,10G13.6)
C
C     LOOK UP CONVERSION CONSTANTS
      DO 2030 I=1,NELS
2030  CALL UCON(IPV(I),PU(I),CONV(I))
C
C     DELIMITOR TESTING REQUIRES THE INNER LOOP BE EXECUTED
C     AT LEAST ONCE
      NX=MAX0(1,N3)
      N2=MAX0(N2,N1)
C     BEGIN SINGLE DATA FORMAT LOOP
      DO 150 J=0,10000
         IF(NELS.GT.1) THEN
            IF(NCARY.EQ.0) READ(NRYIND,'(A)') LINE
            NCARY=0
            IF(NDEVTMP.GT.0) WRITE(NDEVTMP,'(A)') LINE
            READ(LINE,1025) (V(I),I=1,NELS)
         ENDIF
         K=0
         DO 145 I=N1+J,N2+J,NX
               K=K+1
```

```
            IF(NELS.EQ.1) THEN
               IF(NCARY.EQ.0) READ(NRYIND,'(A)',END=10000) LINE
               NCARY=0
               IF(NDEVTMP.GT.0) WRITE(NDEVTMP,'(A)') LINE
C     FOR SINGLE COLUMN CASE READ NOW
               READ(LINE,1020) V(K)
            ENDIF
            IF(V(K) .EQ. DELIM) GO TO 160
C
C          APPLY ANY SCALING CONVERSIONS
            V(K)=V(K)*CONV(K)
            IF(J.LT.N3) GP(I)=V(K)
145         N=N+1
150   CONTINUE
C
10000 IF(V(1).NE.DELIM) CALL RERROR('READW1','NO DELIM',FLOAT(MSET))
C
C     SAVE NUMBER OF VALUES AND ELEMENTS(AS FRACTIONAL PART)
160   IF(N1.GT.1) GP(N1-1)=N+NELS/10.
C
C     CONTINUE MULTI-FORMAT LOOP
      GO TO 5
C
200   CALL RERROR('READW1','BAD FORMAT',FLOAT(MSET))
C
      END



      SUBROUTINE GTUNIT(LINE,PU,NFIELD,NVAR, LN)
C     GTUNIT IS USED TO EXTRACT UNIT CONVERSION SPECIFIERS FROM TABULAR
C     DATA INPUT.
C     CONVERSION SPECIFICATIONS ARE TAKEN FROM 'LINE' INPUT STRING.
C     A FLOATING FORMAT FALLING WITHIN SUCCESSIVE FIELDS OF
C     LENGTH 'NFIELD' IS PROVIDED FOR.
C     OUTPUT IS TO UNIT SPECIFIER ARRAY 'PU' FOR A MAXIMUM OF 'NVAR'
C     FIELDS. FOR A SUCCESS/FAIL RETURN VALUE, 'LN' GIVES THE NUMBER
C     OF NON-BLANK CHARACTERS USEFUL FOR UNIT DECODING, ZERO IMPLIES
C     NO USEFUL UNIT INFORMATION WAS FOUND.
C
      PARAMETER (MAXLN=130)
      CHARACTER*(*) LINE,PU(NVAR),TMP*(MAXLN),TMP1*(MAXLN)
      INTEGER STRIM
C
      LN=STRIM(LINE)
      IF(LN.LT.1) THEN
         DO 5 I=1,NVAR
5           PU(I)=' '
         RETURN
      ENDIF
C
      TMP=' '
      CALL SFILTR(LINE,TMP,' 0123456789+-.E')
```

```
      IF(STRIM(TMP).LT.1) RETURN
C     SIGNIFY WE CONSUMED THIS LINE
      LN=0
C
      TMP1=LINE
C     PLACE INTER-SPECIFIER DELIMITORS
      DO 10 I=NFIELD,NFIELD*NVAR,NFIELD
         IF(TMP1(I:I).NE.' ') THEN
            IF(TMP1(I+1:I+1).NE.' ') THEN
C     IF NO ROOM FOR A BLANK, ITS AN ERROR
               CALL RERROR('GTUNIT','FORMAT ER1',FLOAT(I))
            ELSE
               TMP1(I+1:I+1)='#'
            ENDIF
         ELSE
            TMP1(I:I)='#'
         ENDIF
10    CONTINUE
C
C     NOW SQUEEZE OUT THE BLANKS
      CALL SFILTR(TMP1,TMP,' ')
C     HOPEFULLY SOMETHING LEFT
C
      NL=STRIM(TMP)
      ILIN=1
      DO 20 I=1,NVAR
C     CHECK FOR EMPTY CONVERSIONS
         IF(TMP(ILIN:ILIN).EQ.'#') THEN
            ILIN=ILIN+1
            PU(I)=' '
            GO TO 20
         ENDIF
C     GET NEXT UNIT TYPE
         PU(I)(1:2)=TMP(ILIN:ILIN+1)
C     GET NEXT PHYSICAL UNIT(BLANKS HAVE BEEN REMOVED)
C     SHORT PHRASE IF A BLANK WAS REMOVED
         IF(TMP(ILIN+3:ILIN+3).NE.'#') THEN
            IF(TMP(ILIN+4:ILIN+4).NE.'#') THEN
C     IF INCORRECT GROUPING OF LETTERS, ITS AN ERROR
               CALL RERROR('GTUNIT','FORMAT ER2',FLOAT(ILIN))
            ELSE
C     ADD TWO LETTER UNIT
               PU(I)(3:5)=' '//TMP(ILIN+2:ILIN+3)
               ILIN=ILIN+5
            ENDIF
         ELSE
C     ADD ONE LETTER UNIT
            PU(I)(3:5)='  '//TMP(ILIN+2:ILIN+2)
            ILIN=ILIN+4
         ENDIF
20    CONTINUE
C
C     SHOULD HAVE USED UP ALL THE TEXT BY NOW
      IF(ILIN.LE.NL) CALL RERROR('GTUNIT','FORMAT ER3',FLOAT(ILIN))
      END
```

```
      SUBROUTINE SREAD1(AB,NWOK,NW)                                     SREAD1 9
C     THIS IS THE FALLTHROUGH FEATURE FOR READW WHEN ENCOUNTERING       SREAD110
C     A NON STANDARD LABELED COMMON BLOCK. SREAD1 WILL ADD TABULAR      SREAD111
C     DATA TO THE DEFAULT GP ARRAY PGP AND INCLUDE INDEX OFFSET         SREAD112
C     VALUES TO A LOOKUP ARRAY 'LIST'.                                  SREAD113
C     COMMON DECK "CPROCFL" INSERTED HERE                               CPROCFL2
      INTEGER PMX,PNTBL,PITBL,PFRMTBL,IDSP(10)                          CPROCFL4
C     PARAMETER DECK "PGROUPS"                                          PGROUPS2
      PARAMETER (NCHPG1=11,NWPV=250,NSPGP=NCHPG1+2*NWPV+1)              PGROUPS3
      PARAMETER (MNGRP=9,MXGRP=69,MXLIST=MXGRP-MNGRP+2)                 PGROUPS4
      COMMON/PROCFL/LIST(MXLIST)                                        CPROCFL6
      COMMON/PROCFL/PMX,PNTBL(10),PITBL(10),PFRMTBL(10),PGP(NSPGP)      CPROCFL7
      EQUIVALENCE (PGP,IDSP)                                            CPROCFL8
C                                                                       SREAD115
      CHARACTER ITOC*7                                                  SREAD116
C                                                                       SREAD117
      DATA LIST/MXLIST*0/                                               SAD1BL 2
      DATA PMX/2/                                                       SAD1BL 3
      DATA PNTBL/1,NCHPG1,NSPGP,7*0/                                    SAD1BL 4
      DATA PITBL/1,NWPV,8*0/                                            SAD1BL 5
      DATA PFRMTBL/1,2,8*0/                                             SAD1BL 6
C                                                                       SREAD120
      NWA=IABS(NW)-(MNGRP-2)                                            SREAD121
      IF(NWA.LT.2 .OR. NWA.GT.MXLIST) CALL STOPIT                       SREAD122
     1  ( 'SREAD1 NW='//ITOC(NW)//',LIST(1)='//ITOC(LIST(1)) )          SREAD123
C                                                                       SREAD124
      CALL READW1(AB,NWOK,NW,PMX,PNTBL,PITBL,PFRMTBL,PGP)               SREAD125
C                                                                       SREAD126
C     UPDATE THE LIST, LIST(1)=MAXIMUM USED ELEMENT                     SREAD127
      LIST(1)=MAX0(LIST(1),NWA)                                         SREAD128
      LIST(NWA)=PNTBL(1)                                                SREAD129
C                                                                       SREAD130
C     GET NEXT NUMERICAL INDEX FROM THE POINTER ARRAY                   SREAD131
      NLN=PNTBL(2)                                                      SREAD132
C                                                                       SREAD133
C     NUMBER OF VALUES IS INTEGRAL PART OF FIRST ELEMENT                SREAD134
      NPTS=PGP(NLN)                                                     SREAD135
C     NUMBER OF ROWS IS KEPT AS FRACTIONAL PART                         SREAD136
      NELS=(PGP(NLN)-NPTS)*10.+.5                                       SREAD137
      N2=NPTS/NELS                                                      SREAD138
      NMX=PITBL(2)                                                      SREAD139
C     MAKE ADJUSTMENTS TO LENGTHS AND OFFSETS                           SREAD140
      PITBL(2)=NMX-N2-(NCHPG1-1)                                        SREAD141
      NUP=NPTS+NCHPG1                                                   SREAD142
      PNTBL(1)=PNTBL(1)+NUP                                             SREAD143
      PNTBL(2)=PNTBL(2)+NUP                                             SREAD144
      NLN=NLN+1                                                         SREAD145
C                                                                       SREAD146
C     LOOP TO MOVE NELS-1 ROWS OF PGP(NMX,NELS) TO PGP(N2,NELS)         SREAD147
C     THUS MAKING PGP A CONTIGUOUS ARRAY                                SREAD148
```

```
      NTO=NLN+N2                                                        SREAD149
      NFRM=NLN+NMX                                                      SREAD150
      DO 100 I=2,NELS                                                   SREAD151
      IF(NTO.GT.0 .AND. NFRM+N2-1.LE.NSPGP)                             SREAD152
     1        CALL RMOVE(PGP(NTO),PGP(NFRM),N2)                         SREAD153
      NTO=NTO+N2                                                        SREAD154
  100 NFRM=NFRM+NMX                                                     SREAD155
      END                                                               SREAD156
```

```
      FUNCTION READW(XX)                                                READW   2
C     READS W ARRAY MAKING ANY NECESSARY CONVERSION OF UNITS.           READW   3
C     VARIABLE LENGTH 'TABULAR' DATA IS READ BY SUBROUTINE 'READW1'     READW   4
C     WHICH IS CALLED FOR NEGATIVE W-ARRAY INDICES.                     READW   5
C                                                                       READW   6
C     COMMON DECK "CONST" INSERTED HERE                                 CCONST 2
      COMMON/PCONST/CREF,RGAS,GAMMA                                     CCONST 4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                         CCONST 5
C     COMMON DECK "WWR" INSERTED HERE                                   CWWR    2
      PARAMETER (NWARSZ=1000)                                           CWW1    3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1    4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2    2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2    3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2    4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2    5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2    6
     8 (RCVRH,W(20)),                                                   CWW2    7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2    8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2    9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2   10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2   11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2   12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2   13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2   14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2   15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2   16
C     COMMON DECK "B1" INSERTED HERE                                    CB1     2
      INTEGER UMX,UNTBL,UITBL,UFRMTBL,IDSU(10)                          CB1     4
      COMMON/B1/UMX,UNTBL(10),UITBL(10),UFRMTBL(10),UGP(10)             CB1     5
      EQUIVALENCE (UGP,IDSU)                                            CB1     6
C     COMMON DECK "B2" INSERTED HERE                                    CB2     2
      INTEGER DUMX,DUNTBL,DUITBL,DUFRMTB,IDSDU(10)                      CB2     4
      COMMON/B2/DUMX,DUNTBL(10),DUITBL(10),DUFRMTB(10),DUGP(10)         CB2     5
      EQUIVALENCE (DUGP,IDSDU)                                          CB2     6
C     COMMON DECK "B3" INSERTED HERE                                    CB3     2
      INTEGER CMX,CNTBL,CITBL,CFRMTBL,IDSC(10)                          CB3     4
      COMMON/B3/CMX,CNTBL(10),CITBL(10),CFRMTBL(10),CGP(512)            CB3     5
      EQUIVALENCE (CGP,IDSC),(ANC,CGP(11))                              CB3     6
C     COMMON DECK "B4" INSERTED HERE                                    CB4     2
      INTEGER DCMX,DCNTBL,DCITBL,DCFRMTB,IDSDC(10)                      CB4     4
      COMMON/B4/DCMX,DCNTBL(10),DCITBL(10),DCFRMTB(10),DCGP(10)         CB4     5
      EQUIVALENCE (DCGP,IDSDC)                                          CB4     6
```

```
C     COMMON DECK "B5" INSERTED HERE                                    CB5      2
      INTEGER TMX,TNTBL,TITBL,TFRMTBL,IDST(10)                          CB5      4
      COMMON/B5/TMX,TNTBL(10),TITBL(10),TFRMTBL(10),TGP(262)            CB5      5
      EQUIVALENCE (TGP,IDST),(ANT,TGP(11))                              CB5      6
C     COMMON DECK "B6" INSERTED HERE                                    CB6      2
      INTEGER DTMX,DTNTBL,DTITBL,DTFRMTB,IDSDT(10)                      CB6      4
      COMMON/B6/DTMX,DTNTBL(10),DTITBL(10),DTFRMTB(10),DTGP(10)         CB6      5
      EQUIVALENCE (DTGP,IDSDT)                                          CB6      6
C     COMMON DECK "B7" INSERTED HERE                                    CB7      2
      INTEGER MMX,MNTBL,MITBL,MFRMTBL,IDSM(10)                          CB7      4
      REAL MGP                                                          CB7      5
      COMMON/B7/MMX,MNTBL(10),MITBL(10),MFRMTBL(10),MGP(10)             CB7      6
      EQUIVALENCE (MGP,IDSM)                                            CB7      7
C     COMMON DECK "B9" INSERTED HERE                                    CB8      2
      INTEGER GMX,GNTBL,GITBL,GFRMTBL,IDSG(10)                          CB8      4
      COMMON/B9/GMX,GNTBL(10),GITBL(10),GFRMTBL(10),GGP(113)            CB8      5
      EQUIVALENCE (GGP,IDSG),(ANG,GGP(11))                              CB8      6
C                                                                       READW   17
C     COMMON DECK "B10" INSERTED HERE                                   CB9      2
      INTEGER DGMX,DGNTBL,DGITBL,DGFRMTB,IDSDG(10)                      CB9      4
      COMMON/B10/DGMX,DGNTBL(10),DGITBL(10),DGFRMTB(10),DGGP(10)        CB9      5
      EQUIVALENCE (DGGP,IDSDG)                                          CB9      6
C                                                                       READW   19
C     COMMON DECK "B8" INSERTED HERE                                    CB10     2
      INTEGER RMX,RNTBL,RITBL,RFRMTBL,IDSR(10)                          CB10     4
      COMMON/B8/RMX,RNTBL(10),RITBL(10),RFRMTBL(10),RGP(10)             CB10     5
      EQUIVALENCE (RGP,IDSR)                                            CB10     6
C                                                                       READW   21
C     COMMON DECK "CB17" INSERTED HERE                                  CB17     2
      INTEGER VMX,VNTBL,VITBL,VFRMTBL,IDSV(10)                          CB17     4
      COMMON/B17/VMX,VNTBL(10),VITBL(10),VFRMTBL(10),VGP(53)            CB17     5
      EQUIVALENCE (VGP,IDSV),(ANV,VGP(11))                              CB17     6
C     COMMON DECK "CB18" INSERTED HERE                                  CB18     2
      INTEGER DVMX,DVNTBL,DVITBL,DVFRMTB,IDSDV(10)                      CB18     4
      COMMON/B18/DVMX,DVNTBL(10),DVITBL(10),DVFRMTB(10),DVGP(11)        CB18     5
      EQUIVALENCE (DVGP,IDSDV),(ANDV,DVGP(11))                          CB18     6
C     COMMON DECK "CB19" INSERTED HERE                                  CB19     2
      INTEGER PRMX,PRNTBL,PRITBL,PRFRMTB,IDSPR(10)                      CB19     4
      COMMON/B19/PRMX,PRNTBL(10),PRITBL(10),PRFRMTB(10),PRGP(11)        CB19     5
      EQUIVALENCE (PRGP,IDSPR),(ANP,PRGP(11))                           CB19     6
C     COMMON DECK "CB20" INSERTED HERE                                  CB20     2
      INTEGER DPMX,DPNTBL,DPITBL,DPFRMTB,IDSDP(10)                      CB20     4
      COMMON/B20/DPMX,DPNTBL(10),DPITBL(10),DPFRMTB(10),DPGP(11)        CB20     5
      EQUIVALENCE (DPGP,IDSDP),(ANDP,DPGP(11))                          CB20     6
C                                                                       READW   26
      PARAMETER (MXCMTS=83,BIGVAL=1.E30)                                READW   27
      CHARACTER*38 WFRMT,WNOTES(2)                                      READW   28
      CHARACTER*80 LINEX,NUMSTG                                         READW   29
      CHARACTER*100 STMP1,STMP2                                         READW   30
      INTEGER STRIM,WCOMTS(MXCMTS)                                      READW   31
      CHARACTER*1 DEG,KM,NM,FEET,CYCLE,PER,MSKMS                        READW   32
C     COMMON DECK "RAYDEV" INSERTED HERE                                CRAYDEV2
C     DEVICE ASSIGNED TO RAYTRC INPUT FILE                              CRAYDEV4
      COMMON/RAYDEV/NRYIND,NDEVTMP,NFRMAT,NDEVGRP,NDEVBIN               CRAYDEV5
      LOGICAL NWOK,AB,UCON                                              READW   34
```

```
C
C     COMMON DECK "FLAG" INSERTED HERE
      LOGICAL NEWWR,NEWWP,NEWTRC,PENET
      COMMON /FLG/ NTYP,NEWWR,NEWWP,NEWTRC,PENET,LINES,IHOP,HPUNCH
      COMMON/FLGP/NSET
C
      CHARACTER PC*1
C
      INTEGER XMX,XNTBL(10),XITBL(10),XFRMTBL(10)
      REAL XGP(11)
C
      DATA XMX/2/
      DATA XNTBL/1,11,12,7*0/
      DATA XITBL/1,9*0/
      DATA XFRMTBL/1,2,8*0/
C
      DATA WCOMTS,WNOTES/MXCMTS*1, ' ',' INPUT OVERRIDDEN'/
C
      AB=.FALSE.
      READW=0.0
C
      IF(NDEVTMP.GT.0) REWIND NDEVTMP
C
      READ(NRYIND,1000,END=3) ID
 1000 FORMAT (BZ,10A8)
      IF(NDEVTMP.GT.0) WRITE(NDEVTMP,1000) ID
      GO TO 4
C
3     READW=1.0
      IF(NFRMAT.LT.0) RETURN
C
      PRINT 1040
      WRITE(3,1040)
1040  FORMAT(' END OF INPUT DATA')
      CALL ENDPLT
33    IF(PLT.NE.0.0) CALL DDEND
      STOP
C
4     READ(NRYIND,'(A)',END=10001) STMP1
C
      READ(STMP1,1100) NW,WWW,LINEX
1100  FORMAT (BZ,I3,E14.7,A)
      IF(NDEVTMP.GT.0.AND.(NFRMAT.NE.-2.OR.NW.LT.0))
     1     WRITE(NDEVTMP,'(A)') STMP1
10001 IF (NW.EQ.0) GO TO 10
      IF(NW.GT.MAXW) GO TO 3400
C
      IF (NW.LE.MAXW .AND. NW.GT.0) GO TO 5
C     TABULAR INPUT DATA
      NWOK=.FALSE.
C 'OPEN' THE TEMP FILE
1150  FORMAT(I3,T18,A)
      NWP=-NW
      IF(NFRMAT.EQ.-2) THEN
C     USE DUMMY ARRAYS TO ABSORB TABULAR DATA
```

```
         CALL READW1(AB,NWOK,NW,XMX,XNTBL,XITBL,XFRMTBL,XGP)
      ELSEIF(NWP.EQ.1) THEN
         CALL READW1(AB,NWOK,NW,UMX,UNTBL,UITBL,UFRMTBL,UGP)
      ELSEIF(NWP.EQ.2) THEN
         CALL READW1(AB,NWOK,NW,DUMX,DUNTBL,DUITBL,DUFRMTB,DUGP)
      ELSEIF(NWP.EQ.3) THEN
         CALL READW1(AB,NWOK,NW,CMX,CNTBL,CITBL,CFRMTBL,CGP)
      ELSEIF(NWP.EQ.4) THEN
         CALL READW1(AB,NWOK,NW,DCMX,DCNTBL,DCITBL,DCFRMTB,DCGP)
      ELSEIF(NWP.EQ.5) THEN
            CALL READW1(AB,NWOK,NW,TMX,TNTBL,TITBL,TFRMTBL,TGP)
      ELSEIF(NWP.EQ.6) THEN
         CALL READW1(AB,NWOK,NW,DTMX,DTNTBL,DTITBL,DTFRMTB,DTGP)
      ELSEIF(NWP.EQ.7) THEN
         CALL READW1(AB,NWOK,NW,MMX,MNTBL,MITBL,MFRMTBL,MGP)
      ELSEIF(NWP.EQ.8) THEN
         CALL READW1(AB,NWOK,NW,RMX,RNTBL,RITBL,RFRMTBL,RGP)
      ELSEIF(NWP.EQ.9) THEN
         CALL READW1(AB,NWOK,NW,GMX,GNTBL,GITBL,GFRMTBL,GGP)
      ELSEIF(NWP.EQ.10) THEN
         CALL READW1(AB,NWOK,NW,DGMX,DGNTBL,DGITBL,DGFRMTB,DGGP)
      ELSEIF(NWP.EQ.17) THEN
         CALL READW1(AB,NWOK,NW,VMX,VNTBL,VITBL,VFRMTBL,VGP)
      ELSEIF(NWP.EQ.18) THEN
         CALL READW1(AB,NWOK,NW,DVMX,DVNTBL,DVITBL,DVFRMTB,DVGP)
      ELSEIF(NWP.EQ.19) THEN
         CALL READW1(AB,NWOK,NW,PRMX,PRNTBL,PRITBL,PRFRMTB,PRGP)
      ELSEIF(NWP.EQ.20) THEN
         CALL READW1(AB,NWOK,NW,DPMX,DPNTBL,DPITBL,DPFRMTB,DPGP)
      ELSE
         CALL SREAD1(AB,NWOK,NW)
      ENDIF
      IF(NWOK) GO TO 4
C
3400  WRITE(3,4000) NW,MAXW
4000  FORMAT(15H1THE SUBSCRIPT ,I3,  ' ON THE W-ARRAY INPUT IS OUT OF BO
     1UNDS.  ALLOWABLE VALUES ARE -8 THROUGH,',I4,'.' )
      CALL EXIT
C
5     READ(LINEX,70) DEG,KM,NM,FEET,CYCLE,PER,MSKMS
70    FORMAT(7A)
      W(NW)=WWW
C
C     CHECK FOR A 'TERRAIN RELATIVE' HEIGHT SPECIFICATION.
C     IF SO ADD FLAG VALUE 10**40 TO BE TESTED FOR LATER IN 'RAYTRC'
C
      IF(MSKMS.EQ.'T' .AND. WWW.EQ.0.0) W(NW)=1.E-40
      IF(WWW.EQ.0.0) GO TO 4
      IF(MSKMS.EQ.'T') WWW=WWW*1.E40
C
C     CHECK FOR KEYWORD UNITS SPECIFICATION, IF SO PERFORM CONVERSION
C
      IF(.NOT.UCON(KU,LINEX(:10),CONV)) GO TO 60
75    W(NW)=WWW*CONV
      GO TO 4
```

```
C
60    IF (DEG.EQ.'1') WWW=WWW*RAD
      IF (KM.EQ.'1') WWW=WWW/EARTHR
      IF (NM.EQ.'1') WWW=WWW*1.852
      IF (FEET.EQ.'1') WWW=WWW*3.048006096E-4
      IF(CYCLE.EQ.'1') WWW=WWW*PIT2
      IF(PER.EQ.'1') WWW=PIT2/WWW
      IF(MSKMS.EQ.'1') WWW=WWW*1.E-3
      W(NW)=WWW
      GO TO 4
C
10    IF(.NOT.AB) RETURN
      PRINT 1200
1200  FORMAT(' A DATA FORMATTING ERROR PREVENTS CONTINUED EXECUTION')
      CALL EXIT
C
      ENTRY SETW(XX)
C     THIS ENTRY IS CALLED ONCE BEFORE THE FIRST RUN SET IS READ
C
C INITIALIZE SOME MATHEMATICAL CONSTANTS
      PI=4.0*ATAN(1.0)
      PIT2=2.*PI
      PID2=PI/2.
      DEGS=180./PI
      RAD=PI/180.
      ALN10=ALOG(10.)
CC********* INITIALIZE SOME VARIABLES IN THE W ARRAY
      MAXW=NWARSZ
      CALL CLEAR(W,MAXW)
      PLON=0.
      PLAT=PID2
      EARTHR=6370.
      INTYP=3.
      MAXERR=1.E-4
      ERATIO=50.
      STEP1=1.
      STPMAX=100.
      STPMIN=1.E-8
      FACTR=0.5
      HITLET=.15
      MAXSTP=1000.0
      HOP=1.
      HMAX=500.0
      EXTINC=999.999
      PAGLN=66.0
C
      CALL SETUCON
C
      RETURN
C
      ENTRY SETOVR(XX)
C     THIS ENTRY IS CALLED AFTER EACH RUN SET IS READ
C
C     PERFORM OVERRIDE ASSIGNMENTS TO W-ELEMENTS
      CALL OVERRD(MAXSTP,0.0,1000.0,WCOMTS(23),2,1)
```

```
      CALL OVERRD(ERATIO,0.0,50.0,WCOMTS(43),2,1)
      CALL OVERRD(FACTR,0.0,0.5,WCOMTS(47),2,1)
      CALL OVERRD(SKIP,0.0,BIGVAL,WCOMTS(71),2,1)
      CALL OVERRD(HITLET,0.0,.15,WCOMTS(75),2,1)
      IF(PAGLN.LT.30.0) PAGLN=0.0
      CALL OVERRD(PAGLN,0.0,66.0,WCOMTS(77),2,1)
      CALL OVERRD(PFACTR,0.0,1.0,WCOMTS(82),2,1)
      RETURN
C
      ENTRY PRINTW(XX)
C     THIS ENTRY IS CALLED AFTER EACH RUN SET IS READ
C     TO PRINT VALUES OF THE W-ARRAY IN A FORMAT WHICH SHOWS
C     FULL ACCURACY AND THE EFFECTS OF ANY CONVERSIONS OR OVERRIDES.
C
C     GO TO NEXT PAGE, PUT OUT COPY OF INPUT DATA FOR THIS RUN SET
C     ASSUMING WE HAD A TEMPORARY FILE TO USE
C
      IF(NDEVTMP.LE.0) GO TO 1065
      CALL NEWPAG(NPAG,INT(PAGLN),PC)
      LNSXPG=LINES+INT(PAGLN)
      CALL PUTHDR(3,PC,NPAG)
      CALL PUTDVR(3)
      CALL PUTKBK(3,1)
      CALL PUTKCT(3,'INPUT DATA FILE FOR RUN SET NUMBER'
     1  //NUMSTG(NSET,1,'(I5)'))
      CALL PUTKBK(3,1)
      CALL PUTDVR(3)
      CALL PUTKBK(3,1)
      REWIND NDEVTMP
C
1060  READ(NDEVTMP,'(A)',END=1065) LINEX
      IF(LINES.GE.LNSXPG) THEN
         CALL NEWPAG(NPAG,INT(PAGLN),PC)
         LNSXPG=LINES+INT(PAGLN)
         CALL PUTHDR(3,PC,NPAG)
         CALL PUTKBK(3,1)
      ENDIF
      CALL PUTKST(3,' '//LINEX)
      GO TO 1060
1065  CONTINUE
C
C     GO TO NEXT PAGE, PUT OUT W-ELEMENTS SHOWING FULL PRECISION
C     USE A TWO COLUMN FORMAT
C
      CALL NEWPAG(NPAG,INT(PAGLN),PC)
      CALL PUTHDR(3,PC,NPAG)
      CALL PUTDVR(3)
      CALL PUTKBK(3,1)
      CALL PUTKCT(3,'INITIAL VALUES FOR THE W ARRAY')
      CALL PUTKCT(3,'ONLY NONZERO VALUES PRINTED'
     1  //' -- ALL ANGLES IN RADIANS')
      CALL PUTKBK(3,1)
      CALL PUTDVR(3)
      LINSPP=INT(PAGLN)-7
C
```

```
      WFRMT='(BZ,I3,E14.7,A)'
      IF(NFRMAT.EQ.0) WFRMT='(I4,E24.15,A)'
C
      NWE=0
C     ALLOW MAXIMUM OF 10 PAGES OF OUTPUT(1 IS ENOUGH)
      DO 18 NPGS=1,10
C     ALLOW 3 LINES FOR INSIDE HEADER(SEE BELOW)
         LINSPP=LINSPP-3
C     NEXT ELEMENT TO SCAN IS ONE MORE THAN LAST ONE
         LWE=NWE+1
C
C     INITIALIZE SECOND COLUMN START INDEX
         NXCL=0
C     INITIALIZE TOTAL ELEMENT COUNTER
         NELS=0
C     LOOP TO FIND BREAK POINTS FOR FIRST/SECOND COLUMNS
         DO 14 I=LWE,MAXW
            IF(W(I).EQ.0.0) GO TO 14
            NWE=I
            NELS=NELS+1
            IF(NELS.EQ.LINSPP) NXCL=I
            IF(NELS.EQ.2*LINSPP) GO TO 15
14       CONTINUE
15    IF(NELS.EQ.0) GO TO 22
C
         IF(NPGS.GT.1) THEN
            CALL NEWPAG(NPAG,INT(PAGLN),PC)
            CALL PUTHDR(3,PC,NPAG)
            LINSPP=INT(PAGLN)-1
         ENDIF
C
C     INSERT 'INNER' HEADER
         CALL PUTKBK(3,2)
         STMP1=' N              W(N)'
         CALL PUTKST(3,STMP1(:65)//'  '//STMP1)
C
      NX=NXCL-1
      IF(NX.LT.LWE) NX=NWE
      DO 17 NW=LWE,NX
      NCOMT=MIN0(NW,MXCMTS)
      IF(W(NW).NE.0) THEN
        WRITE(STMP1,WFRMT) NW,W(NW),WNOTES(WCOMTS(NCOMT))
        STMP2=' '
        IF(NXCL.EQ.0) GO TO 23
        DO 19 I=NXCL,MAXW
        IF(W(I).NE.0) THEN
          NCOMT=MIN0(I,MXCMTS)
          WRITE(STMP2,WFRMT) I,W(I),WNOTES(WCOMTS(NCOMT))
          NXCL=I+1
          GO TO 23
        ENDIF
19      CONTINUE
23      CALL PUTKST(3,STMP1(:65)//'  '//STMP2)
      ENDIF
17    CONTINUE
```

```
      IF(NELS.EQ.0) GO TO 22                                            READW307
18    CONTINUE                                                          READW308
C                                                                       READW309
22    CONTINUE                                                          READW310
      END                                                               READW311
```

```
      SUBROUTINE CLEAR(A,N)                                             CLEAR  2
C     SET N ELEMENTS OF ARRAY A TO ZERO                                 CLEAR  3
      REAL A(N)                                                         CLEAR  4
      IF(N.LE.0) RETURN                                                 CLEAR  5
      DO    10 I=1,N                                                    CLEAR  6
10    A(I)=0.0                                                          CLEAR  7
      RETURN                                                            CLEAR  8
      END                                                               CLEAR  9
```

```
      FUNCTION ND2B(INDEC)                                              ND2B   2
C                                                                       ND2B   3
C     CONVERT A NUMERIC DECIMAL DIGIT STRING TO A BIT STRING            ND2B   4
C     WITH EACH BIT SET BY A CORRESPONDING DECIMAL DIGIT.               ND2B   5
C                                                                       ND2B   6
      ND2B=0                                                            ND2B   7
      IF(INDEC.LE.0) RETURN                                             ND2B   8
      M=INDEC                                                           ND2B   9
      MB=1                                                              ND2B  10
C                                                                       ND2B  11
10    IF(MOD(M,10).NE.0) ND2B=ND2B+MB                                   ND2B  12
      MB=MB*2                                                           ND2B  13
      M=M/10                                                            ND2B  14
      IF(M.GT.0) GO TO 10                                               ND2B  15
      END                                                               ND2B  16
```

```
      LOGICAL FUNCTION UCON(JC1,U,CONV)                                 UCON   2
      LOGICAL SETUCON                                                   UCON   3
C     PROVIDES KEYWORD UNITS CONVERSION FOR W-ARRAY INPUT.              UCON   4
C     UNITS SPECIFICATION MUST BE IN THE FORM 'DV UN' WHERE DV IS THE   UCON   5
C     TYPE OF VARIABLE(SUCH AS AN FOR ANGLE) AND UN IS THE UNITS CHOICE UCON   6
C     SUCH AS RD FOR RADIANS). RETURN VALUE IS THE CONVERSION FACTOR.   UCON   7
C     COMMON DECK "CUCON" INSERTED HERE                                 CUCON  2
      COMMON/UCONV/CNVV(4,4)                                            CUCON  4
      CHARACTER PCV*3,CNVC*2                                            CUCON  5
      COMMON/UCONC/PCV(4),CNVC(4,4)                                     CUCON  6
C     COMMON DECK "CONST" INSERTED HERE                                 CCONST 2
      COMMON/PCONST/CREF,RGAS,GAMMA                                     CCONST 4
```

```
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10
C     COMMON DECK "WWR" INSERTED HERE
      PARAMETER (NWARSZ=1000)
      COMMON/WW/ID(10),MAXW,W(NWARSZ)
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),
     8 (RCVRH,W(20)),
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),
     6 (HMIN,W(27)),(RGMAX,W(28)),
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))
      CHARACTER U*(*),PU*3,PV*2
C
      DATA NPV,NPU/3,4/
      UCON=.FALSE.
      CONV=1.0
C
      READ(U,100) PU,PV
100   FORMAT(A,A)
C
      DO 2010 J=1,NPV
2010     IF(PU.EQ.PCV(J)) GO TO 2015
      JC1=4
      RETURN
2015  JC1=J
      DO 2020 K=1,NPU
2020     IF(PV.EQ.CNVC(J,K)) GO TO 2025
      JC1=4
      RETURN
2025  UCON=.TRUE.
      CONV=CNVV(JC1,K)
      RETURN
C
C     INITIAL CONVERSION CONSTANTS
      ENTRY SETUCON(JC1,U,CONV)
      SETUCON=.TRUE.
C
      CNVV(1,2) = RAD
      CNVV(1,3) = 1.0/EARTHR
      CNVV(2,2) = 1.0E-3
      CNVV(2,3) = 3.048006096E-4
      CNVV(2,4) = 1.852
      CNVV(3,2) = PIT2
      CNVV(3,3) = PIT2*1.0E3
      CNVV(3,4) = PIT2*RAD
      END
```

```
      SUBROUTINE TRACE                                                  TRACE  3
C                                                 VERSION 3/4/86        TRACE  4
C     CALCULATES ONE RAYPATH FOR THE REQUESTED NUMBER OF HOPS           TRACE  5
C     (CROSSINGS OR CLOSEST APPROACH TO RECEIVER SURFACE; A CLOSEST     TRACE  6
C     APPROACH COUNTS AS TWO HOPS).                                     TRACE  7
C     THIS VERSION IS LIMITED TO USE WITH ATMOSPHERIC MODELS            TRACE  8
      REAL NORYST                                                       TRACE  9
      PARAMETER (NORYST=0.0,MXPRGS=200)                                 TRACE 10
C                                                                       TRACE 11
      COMMON /RK/ N,STEP,MODE,E1MAX,E1MIN,E2MAX,E2MIN,FACT,RSTART       TRACE 12
C     COMMON DECK "FLAG" INSERTED HERE                                  TRACE 13
      LOGICAL NEWWR,NEWWP,NEWTRC,PENET                                  TRACE 14
      COMMON /FLG/ NTYP,NEWWR,NEWWP,NEWTRC,PENET,LINES,IHOP,HPUNCH      TRACE 15
C     COMMON DECK "TRAC" INSERTED HERE                                  CTRAC  2
      LOGICAL GROUND,SURF,PERIGE,THERE,MINDIS,NEWRAY                    CTRAC  4
      COMMON /TRAC/ GROUND,SURF,PERIGE,THERE,MINDIS,NEWRAY,SMT,OSMT     CTRAC  5
      COMMON/TRAC/ROLD(20),DROLD(20),TOLD,ZDOT,D2Z,RAD,RAD1             CTRAC  6
C     COMMON DECK "RR" INSERTED HERE                                    TRACE 17
      REAL MODREC                                                       TRACE 18
      COMMON/RR/ MODREC(4)                                              TRACE 19
      COMMON/RR/F,PFR,PFRR,PFRTH,PFRPH                                  TRACE 20
      COMMON/RR/PFTH,PFPH,PFTHTH,PFPHPH,PFTHPH,FSELECT,FTIME            TRACE 21
C     COMMON DECK "GG" INSERTED HERE                                    TRACE 22
      REAL MODG                                                         TRACE 23
      COMMON/GG/MODG(4)                                                 TRACE 24
      COMMON/GG/G,PGR,PGRR,PGRTH,PGRPH                                  TRACE 25
      COMMON/GG/PGTH,PGPH,PGTHTH,PGPHPH,PGTHPH,GSELECT,GTIME            TRACE 26
C     COMMON DECK "RKTIME" INSERTED HERE                                TRACE 27
      COMMON/CRKTIME/RKTIME                                             TRACE 28
C     COMMON DECK "RINPLEX" INSERTED HERE                               TRACE 29
      REAL KAY2,KAY2I                                                   TRACE 30
      COMPLEX PNP,POLAR,LPOLAR                                          TRACE 31
      LOGICAL SPACE                                                     TRACE 32
      COMMON /RIN/ MODRIN(8),RAYNAME(2,3),TYPE(3),SPACE                 TRACE 33
      COMMON/RIN/OMEGMIN,OMEGMAX,KAY2,KAY2I                             TRACE 34
      COMMON/RIN/PNP(10),POLAR,LPOLAR,SGN                               TRACE 35
      COMMON R(20),T,STP,DRDT(20)                                       TRACE 36
C     COMMON DECK "WWR" INSERTED HERE                                   TRACE 37
      PARAMETER (NWARSZ=1000)                                           TRACE 38
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   TRACE 39
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                TRACE 40
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    TRACE 41
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      TRACE 42
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          TRACE 43
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         TRACE 44
     8 (RCVRH,W(20)),                                                   TRACE 45
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))TRACE 46
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      TRACE 47
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      TRACE 48
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     TRACE 49
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       TRACE 50
```

```
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))
C     THE FOLLOWING IS A LOCAL COMMON ONLY
C      COMMON DECK "TRLOCAL" INSERTED HERE
       COMMON/TRLOCAL/RSIGN,HOME,FDOT,GDOT,GOLD,GDOLD
       LOGICAL PCROSS,HOME,WASNT
       LOGICAL LAUNCH,APOGEE,PLTENB,EXTON
       EQUIVALENCE (R(2),TH),(R(3),PHI)
C
       CHARACTER*10 XCOND,DEFCND
       PARAMETER (DEFCND=' MAX HOPS')
C
       REAL PRGHST(3,MXPRGS)
       EQUIVALENCE(RKTIME,IRKTIME), (APOG,PRIGEE,W(80))
       DATA GROUND/.FALSE./
C
C      INCREMENT EVENT COUNT
       IRKTIME=IRKTIME+1
       PLTENB=APOG.EQ.0.0
C      ENABLE EXTINCTION TEST IF LIMIT IS SET AND VARIABLE IS BEING INTE
       EXTON=(EXTINC.GT.0.0) .AND. (W(58).NE.0.0)
C      POINT INDEX TO ABSORTION VARIABLE
       NR=7
       IF(W(57).NE.0.0) NR=8
       RMAX=HMAX+EARTHR
       RMIN=HMIN+EARTHR
       NHOP=HOP
       MAX=MAXSTP
       NSKIP=SKIP
C********* INITIALIZE PARAMETERS FOR INTEGRATION SUBROUTINE RKAM
       MODE=INTYP
       STEP=STEP1
       E1MAX=MAXERR
       E1MIN=MAXERR/ERATIO
       E2MAX=STPMAX
       E2MIN=STPMIN
       FACT=FACTR
       RSTART=1.
       CALL HAMLTN
       FDOT=DOT(F)
C
C      CHECK FOR EQUALITY WITH RECEIVER HEIGHT WITHIN MACHINE PRECISION
       THERE=F.EQ.0.0
C
       HOME=FDOT*F.GE.0.
C********* IHOP=0 TELLS PRINTR TO PRINT HEADING AND PUNCH A TRANSMITTER
C********* RAYSET AND NEWTRC=TRUE TELLS RAYPLT TO START A NEW RAY
       NEWTRC=.TRUE.
       IHOP=0
C********* RESET PERIGEE PLOT COUNTER
       NPRGS=0
C      USE CURRENT RELATIVE POSITION OF RAY TO PREDICT
C      SIGN OF NEXT INTERSECTION OF RAY WITH RECEIVER HEIGHT
```

```
C     (WILL SOON BE REVERSED).
      RSIGN=SIGN(1.0,F)
C     IF RAY IS LAUNCHED FROM RECEIVER HEIGHT, USE DIRECTION INSTEAD
      IF(THERE) RSIGN=SIGN(1.0,FDOT)
      GDOT=DOT(G)
C     TEST CASE FOR RECEIVER ON THE GROUND
      IF(F.EQ.G) RSIGN=1.0
C     IF RECEIVER IS ON TERRAIN THEN DIRECTION OF FIRST
C     RECEIVER CROSSING IS DOWNWARD
      CALL PRINTR('TXMTR',RAYSET)
      IF (PLTENB) CALL RAYPLT
      NEWRAY=.TRUE.
      STH0=SIN(TH)
      CTH0=COS(TH)
      PHI0=PHI
C
C     SET DEFAULT RAY EVENTS FOR 'PRINTR'
      XCOND=DEFCND
      XSET=NORYST
C
C********* LOOP ON NUMBER OF HOPS
10    IHOP=IHOP+1
C      REVERSE SIGN AT EACH CROSSING OF RECEIVER HEIGHT
      RSIGN=-RSIGN
      IF (IHOP.GT.NHOP) GO TO 100
      PENET=.FALSE.
C********* LOOP ON MAXIMUM NUMBER OF STEPS PER HOP
      DO 79 J=1,MAX
C     LAUNCH=TRUE ONLY WHEN TRANSMITTER ON GROUND WITH DOWNWARD
C     TRANSMISSION(POSSIBLE ONLY ON 1ST STEP).
      LAUNCH=G.EQ.0.0 .AND. GDOT.LT.0.0
C
C     SAVE CURRENT STATE VALUES FOR LATER COMPARISONS
   12 DO 13 L=1,6
      ROLD(L)=R(L)
   13 DROLD(L)=DRDT(L)
      FDOLD=FDOT
      GDOLD=GDOT
      FOLD=F
      GOLD=G
      TOLD=T
C
C     PROCESS NEXT RAY POINT
      CALL RKAM
      FDOT=DOT(F)
      WASNT=.NOT.HOME
      HOME=FDOT*F.GE.0.
C
C
C     LOOK FOR DOWNGOING CROSSING OF RECEIVER SURFACE
      IF(.NOT.LAUNCH.AND.F.LT.0.0.AND.RSIGN.LT.0.0) GO TO 50
C     CHECK FOR CASE OF CLOSEST APPROACH
      IF (HOME.AND.WASNT) GO TO 30
C     DETERMINE IF A GROUND CROSSING HAS OCCURED
      IF(PCROSS(G,GDOT)) GO TO 20
```

```
C     LOOK FOR UPGOING CROSSING OF RECEIVER SURFACE
      IF(.NOT.LAUNCH.AND.F.GT.0.0.AND.RSIGN.GT.0.0) GO TO 50
      GO TO 18
C
C     CHECK FOR PERIGEE CONDITION
18    IF (DROLD(1).GE.0..OR.DRDT(1).LE.0.) GO TO 25
      CALL PRINTR(' PERIGEE ',NORYST)
      IF(PRIGEE.EQ.0.0) GO TO 25
C
      IF(NPRGS.GE.MXPRGS) CALL STOPIT('PRG LIMIT')
      NPRGS=NPRGS+1
      CALL RMOVE(PRGHST(1,NPRGS),R,3)
C
25    APOGEE=DROLD(1).GT.0..AND.DRDT(1).LT.0.
      PLTENB=APOG.EQ.0.0.OR.APOGEE
C
      IF(APOGEE) CALL PRINTR(' APOGEE  ',NORYST)
      IF (DROLD(2)*DRDT(2).LT.0.) CALL PRINTR(' MAX LAT ',NORYST)
      IF (DROLD(3)*DRDT(3).LT.0.) CALL PRINTR(' MAX LONG',NORYST)
      DO 14 I=4,6
      IF (ROLD(I)*R(I).LT.0.) CALL PRINTR(' WAVE REV',NORYST)
   14 CONTINUE
      GO TO 75
C********* RAY WENT UNDERGROUND
C     USE 'FULL' REGULAR PARABOLIC FIT(ENTRY 'FIT')
20    CALL RCROSS(-1.,G,GDOT,'GGRND REF',GROUND)
      GO TO 75
C********* RAY MAY HAVE MADE A CLOSEST APPROACH
C     USE 'FULL' REGULAR PARABOLIC FIT
30    CALL FIT(F,FOLD,FDOLD)
      IF(RAD.GE.0.0) GO TO 501
C     ESTIMATE TIME OF CLOSEST APPROACH
      TP=T-ZDOT/D2Z
      CALL GRAZE(F,RSIGN,TP)
      FDOT=ZDOT
      IF (THERE) GO TO 51
C
C     SET DRDT(1)=0 TO AVOID INCORRECT APOGEE OR PERIGEE PRINTOUT, A NEW
C     VALUE WILL BE COMPUTED BEFORE FURTHER ANALYSIS IS DONE
40    DRDT(1)=0.
      HPUNCH=R(1)-EARTHR
      CALL PRINTR('MMIN DIST',RAYSET)
      IF(PLTENB) CALL RAYPLT
      IF (IHOP.GE.NHOP) GO TO 100
      IHOP=IHOP+1
      RSIGN=-RSIGN
      CALL PRINTR ('MMIN DIST',RAYSET)
      GO TO 89
C********* RAY CROSSED RECEIVER HEIGHT
50    CALL FIT(F,FOLD,FDOLD)
C     ESTIMATE GROUP TIME WHEN RAY CROSSES RECEIVER HEIGHT
C     (IN THE CORRECT DIRECTION).
501   TC=T-2.0*F/(ZDOT+SIGN(RAD1,RSIGN))
      CALL BACKUP(F,RSIGN,TC)
      FDOT=ZDOT
```

```
      IF(.NOT.THERE) GO TO 40
51    HPUNCH=R(1)-EARTHR
      CALL PRINTR('RRCVR    ',RAYSET)
      IF (PLTENB) CALL RAYPLT
      IF (GET(F).NE.GET(G)) GO TO 89
C     RECEIVER IS ON TERRAINE
      IF (IHOP.GE.NHOP) GO TO 100
      IHOP=IHOP+1
      RSIGN=-RSIGN
C********* GROUND REFLECT
      CALL REFLECT(G)
      CALL HAMLTN
      FDOT=DOT(F)
      RSTART=1.
      HPUNCH=R(1)-EARTHR
      CALL PRINTR('GGRND REF',RAYSET)
      THERE=.TRUE.
      HPUNCH=R(1)-EARTHR
      CALL PRINTR ('RRCVR    ',RAYSET)
      GO TO 89
C*********
75    IF (PLTENB) CALL RAYPLT
      PLTENB=APOG.EQ.0.0
      IF(EXTON) THEN
         IF (R(NR).GE.EXTINC) XCOND='EEXTINC'
      ENDIF
      IF (R(1).GT.RMAX.AND.F.GT.0.0.AND.FDOT.GT.0.) XCOND='UMAX HT'
      IF (R(1).LT.RMIN.AND.F.LT.0.0.AND.FDOT.LT.0.) XCOND='DMIN HT'
      RANGE=EARTHR*ACOS(COS(TH)*CTH0+SIN(TH)*STH0*COS(PHI-PHI0))
      IF(RGMAX.GT.0.0.AND.RGMAX.LT.RANGE) XCOND='FMAX RANGE'
      IF(XCOND.NE.DEFCND) GO TO 90
C
      IF (MOD(J,NSKIP).EQ.0) CALL PRINTR('          ',NORYST)
   79 CONTINUE
C********* EXCEEDED MAXIMUM NUMBER OF STEPS
      HPUNCH=R(1)-EARTHR
      CALL PRINTR('SSTEP MAX',RAYSET)
      GO TO 100
C*********
   89 HOME=.TRUE.
      GDOT=DOT(G)
      GO TO 10
C********* RAY PENETRATED COMPUTATIONAL AREA BOUNDARY AND WAS HEADING OU
90    PENET=.TRUE.
      HPUNCH=R(1)-EARTHR
      XSET=RAYSET
C
C     NORMAL EXIT
100   CALL PRINTR(XCOND,XSET)
      IF(NPRGS.LE.0) RETURN
C
      CALL DASH
      NEWTRC=.TRUE.
      DO 110 I=1,NPRGS
      CALL RMOVE(R,PRGHST(1,I),3)
```

```
110   CALL RAYPLT                                                              TRACE271
C                                                                              TRACE272
      CALL RESET('DASH')                                                       TRACE273
      END                                                                      TRACE274
```

```
      LOGICAL FUNCTION PCROSS(Z,ZDT)                                           TRACE275
C     RETURNS 'TRUE' IF PARABOLIC 'FIT2'(SEE 'FIT' ROUTINE)                   TRACE276
C     INDICATES CROSSING OF SURFACE 'Z'.                                       TRACE277
      REAL ZDT(3)                                                              TRACE278
C     COMMON DECK "TRAC" INSERTED HERE                                         CTRAC  2
      LOGICAL GROUND,SURF,PERIGE,THERE,MINDIS,NEWRAY                           CTRAC  4
      COMMON /TRAC/ GROUND,SURF,PERIGE,THERE,MINDIS,NEWRAY,SMT,OSMT            CTRAC  5
      COMMON/TRAC/ROLD(20),DROLD(20),TOLD,ZDOT,D2Z,RAD,RAD1                    CTRAC  6
C                                                                              TRACE280
C     USE SPECIAL PARABOLIC FIT GIVING 'SMT' AND 'OSMT'                        TRACE281
      ZDT(1)=DOT(Z)                                                            TRACE282
      PCROSS=.TRUE.                                                            TRACE283
      IF(Z.LT.0.0) GO TO 20                                                    TRACE284
      IF(ZDT(1).LE.0.0.OR.ZDT(3).GE.0.0) GO TO 15                              TRACE285
      CALL FIT2(Z,ZDT(2),ZDT(3))                                               TRACE286
      IF(SMT.GT.Z.OR.OSMT.GT.ZDT(2)) GO TO 20                                  TRACE287
C                                                                              TRACE288
15    PCROSS=.FALSE.                                                           TRACE289
20    RETURN                                                                   TRACE290
      END                                                                      TRACE291
```

```
      SUBROUTINE RCROSS(S,Z,ZDT,EVENT,QMODE)                                   TRACE292
C     FIND ESTIMATED CROSSING POINT OF SURFACE 'Z' THEN USE                    TRACE293
C     ROUTINE 'BACKUP' TO GO THERE AND PERFORM A RAY REFLECTION                TRACE294
C     WITH ROUTINE 'REFLECT'.                                                  TRACE295
C     EXTENDS RAY PROPAGATION TO INTERSECTION WITH REFLECTING SURFACE          TRACE296
C     Z AND CALLS 'REFLECT' TO OBTAIN VECTOR COMPONENTS.                       TRACE297
C                                                                              TRACE298
      CHARACTER EVENT*8                                                        TRACE299
      REAL ZDT(3)                                                              TRACE300
      LOGICAL QMODE                                                            TRACE301
C     COMMON DECK "TRAC" INSERTED HERE                                         CTRAC  2
      LOGICAL GROUND,SURF,PERIGE,THERE,MINDIS,NEWRAY                           CTRAC  4
      COMMON /TRAC/ GROUND,SURF,PERIGE,THERE,MINDIS,NEWRAY,SMT,OSMT            CTRAC  5
      COMMON/TRAC/ROLD(20),DROLD(20),TOLD,ZDOT,D2Z,RAD,RAD1                    CTRAC  6
C     COMMON DECK "RR" INSERTED HERE                                           TRACE303
      REAL MODREC                                                              TRACE304
      COMMON/RR/ MODREC(4)                                                     TRACE305
      COMMON/RR/F,PFR,PFRR,PFRTH,PFRPH                                         TRACE306
      COMMON/RR/PFTH,PFPH,PFTHTH,PFPHPH,PFTHPH,FSELECT,FTIME                   TRACE307
C     COMMON DECK "WWR" INSERTED HERE                                          TRACE308
      PARAMETER (NWARSZ=1000)                                                  TRACE309
```

```
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   TRACE310
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                TRACE311
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    TRACE312
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      TRACE313
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          TRACE314
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         TRACE315
     8 (RCVRH,W(20)),                                                   TRACE316
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))TRACE317
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      TRACE318
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      TRACE319
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     TRACE320
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       TRACE321
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        TRACE322
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        TRACE323
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              TRACE324
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   TRACE325
C     COMMON DECK "FLAG" INSERTED HERE                                  TRACE326
      LOGICAL NEWWR,NEWWP,NEWTRC,PENET                                  TRACE327
      COMMON /FLG/ NTYP,NEWWR,NEWWP,NEWTRC,PENET,LINES,IHOP,HPUNCH      TRACE328
      COMMON R(20),T,STP,DRDT(20)                                       TRACE329
      COMMON /RK/ N,STEP,MODE,E1MAX,E1MIN,E2MAX,E2MIN,FACT,RSTART       TRACE330
      LOGICAL HOME                                                      TRACE331
C     COMMON DECK "TRLOCAL" INSERTED HERE                               TRACE332
      COMMON/TRLOCAL/RSIGN,HOME,FDOT,GDOT,GOLD,GDOLD                    TRACE333
C                                                                       TRACE334
      IF(QMODE) GO TO 60                                                TRACE335
      CALL FIT(Z,ZDT(2),ZDT(3))                                         TRACE336
      TC=T-2.0*Z/(ZDOT-RAD1)                                            TRACE337
      CALL BACKUP(Z,-1.,TC)                                             TRACE338
C********* GROUND REFLECT                                               TRACE339
60    CALL REFLECT(Z)                                                   TRACE340
      CALL HAMLTN                                                       TRACE341
      ZDT(1)=DOT(Z)                                                     TRACE342
      FDOT=DOT(F)                                                       TRACE343
      HOME=FDOT*F.GE.0.0                                                TRACE344
      RSTART=1.                                                         TRACE345
      HPUNCH=R(1)-EARTHR                                                TRACE346
      CALL PRINTR(EVENT,RAYSET)                                         TRACE347
      IF(F.NE.Z) RETURN                                                 TRACE348
C     AVOID RECEIVER CROSSING AT THE TRANSMITTER IF RECEIVER ON TERRAIN TRACE349
      RSIGN=S                                                           TRACE350
      HOME=.TRUE.                                                       TRACE351
      END                                                               TRACE352
```

```
      SUBROUTINE HAMLTN                                                 HAMLTN 2
C********* CALCULATES HAMILTONS EQUATIONS FOR RAY TRACING               HAMLTN 3
C     AND OTHER QUANTITIES TO BE INTEGRATED                             HAMLTN 4
C     COMMON DECK "GG" INSERTED HERE                                    CGG    2
      REAL MODG                                                         CGG    4
      COMMON/GG/MODG(4)                                                 CGG    5
      COMMON/GG/G,PGR,PGRR,PGRTH,PGRPH                                  CGG    6
```

```
      COMMON/GG/PGTH,PGPH,PGTHTH,PGPHPH,PGTHPH,GSELECT,GTIME             CGG     7
C     COMMON DECK "CONST" INSERTED HERE                                  CCONST  2
      COMMON/PCONST/CREF,RGAS,GAMMA                                      CCONST  4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                          CCONST  5
C     COMMON DECK "RINREAL" INSERTED HERE                                CRINREA2
      LOGICAL SPACE                                                      CRINREA4
      REAL LPOLAR,LPOLRI,KPHK,KPHKI,KAY2,KAY2I                           CRINREA5
      CHARACTER DISPM*6                                                  CRINREA6
      COMMON/RINPL/DISPM                                                 CRINREA7
      COMMON /RIN/ MODRIN(8),RAYNAME(2,3),TYPE(3),SPACE                  CRINREA8
      COMMON/RIN/OMEGMIN,OMEGMAX,KAY2,KAY2I,                             CRINREA9
     1              H,HI,PHT,PHTI,PHR,PHRI,PHTH,PHTHI,PHPH,PHPHI         CRINRE10
     2, PHOW,PHOWI,PHKR,PHKRI,PHKTH,PHKTI, PHKPH,PHKPI                   CRINRE11
     3 ,KPHK,KPHKI,POLAR,POLARI,LPOLAR,LPOLRI,SGN                        CRINRE12
      COMMON R(20),T,STP,DRDT(20)                                        HAMLTN 8
      PARAMETER (NWARSZ=1000)                                            CWW1    3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                    CWW1    4
      EQUIVALENCE (TH,R(2)),(PH,R(3)),(KR,R(4)),(KTH,R(5)),(KPH,R(6)),   HAMLTN10
     1 (DTHDT,DRDT(2)),(DPHDT,DRDT(3)),(DKRDT,DRDT(4)),(DKTHDT,DRDT(5)),HAMLTN11
     2 (DKPHDT,DRDT(6)),(HMAX,W(26)),(OW,W(6))                           HAMLTN12
      REAL KR,KTH,KPH                                                    HAMLTN13
      STH=SIN(TH)                                                        HAMLTN14
      CTH=SIN(PID2-TH)                                                   HAMLTN15
      RSTH=R(1)*STH                                                      HAMLTN16
      RCTH=R(1)*CTH                                                      HAMLTN17
      CALL DISPER                                                        HAMLTN18
      DENPHC=1.0/(PHOW*CREF)                                             HAMLTN19
      DRDT(1)=-PHKR*DENPHC                                               HAMLTN20
      DTHDT=-PHKTH*DENPHC/R(1)                                           HAMLTN21
      DPHDT=-PHKPH*DENPHC/RSTH                                           HAMLTN22
      DKRDT=PHR*DENPHC+KTH*DTHDT+KPH*STH*DPHDT                           HAMLTN23
      DKTHDT=(PHTH*DENPHC-KTH*DRDT(1)+KPH*RCTH*DPHDT)/R(1)               HAMLTN24
      DKPHDT=(PHPH*DENPHC-KPH*STH*DRDT(1)-KPH*RCTH*DTHDT)/RSTH           HAMLTN25
      NR=6                                                               HAMLTN26
C********* PHASE PATH                                                    HAMLTN27
      IF (W(57).EQ.0.) GO TO 10                                          HAMLTN28
      NR=NR+1                                                            HAMLTN29
      DRDT(NR)=-      KPHK/PHOW/OW                                       HAMLTN30
C********* ABSORPTION                                                    HAMLTN31
   10 IF (W(58).EQ.0.) GO TO 15                                          HAMLTN32
      NR=NR+1                                                            HAMLTN33
      DRDT(NR)= 10./ALN10*KPHK*KAY2I/(KR*KR+KTH*KTH+KPH*KPH)*DENPHC      HAMLTN34
C********* DOPPLER SHIFT                                                 HAMLTN35
   15 IF (W(59).EQ.0.) GO TO 20                                          HAMLTN36
      NR=NR+1                                                            HAMLTN37
      DRDT(NR)=-PHT*DENPHC/PIT2                                          HAMLTN38
C********* GEOMETRICAL PATH LENGTH                                       HAMLTN39
   20 IF (W(60).EQ.0.) GO TO 25                                          HAMLTN40
      NR=NR+1                                                            HAMLTN41
      DRDT(NR)=SQRT(PHKR**2+PHKTH**2+PHKPH**2)*ABS(DENPHC)               HAMLTN42
C********* TERRAIN FUNCTION AS COMPLETE INTEGRAL                         HAMLTN43
25    IF(ABS(W(61))+ABS(W(62))+ABS(W(63))+ABS(W(64)).EQ.0.0) GO TO 45    HAMLTN44
      CALL TOPOG                                                         HAMLTN45
C                                                                        HAMLTN46
      IF(W(61).EQ.0.) GO TO 30                                           HAMLTN47
```

```
      NR=NR+1
      DRDT(NR)=PGR*DRDT(1)+PGTH*DTHDT+PGPH*DPHDT
C********* TERRAIN TIME DERIVATIVES AS COMPLETE INTEGRALS
   30 IF(W(62).EQ.0.) GO TO 35
      NR=NR+1
C     DERIVATIVE WITH RESPECT TO R
      DRDT(NR)=PGRR*DRDT(1)+PGRTH*DTHDT+PGRPH*DPHDT
   35 IF(W(63).EQ.0.) GO TO 40
      NR=NR+1
C     DERIVATIVE WITH RESPECT TO THETA
      DRDT(NR)=PGRTH*DRDT(1)+PGTHTH*DTHDT+PGTHPH*DPHDT
   40 IF(W(64).EQ.0.) GO TO 45
      NR=NR+1
C     DERIVATIVE WITH RESPECT TO PHI
      DRDT(NR)=PGRPH*DRDT(1)+PGTHPH*DTHDT+PGPHPH*DPHDT
C********* OTHER CALCULATIONS
45    CONTINUE
      RETURN
         END
```

```
      SUBROUTINE RKAM
C     KEEPS TRACK OF INTERNAL INTEGRATION STEPS PERFORMED BY THE RKAM1
C     ROUTINE AND MAKES THEM AVAILABLE TO CALLING ROUTINES.
      COMMON/RKAMS/XV(5),FV(4,20),YU(5,20),EPM,ALPHA,MM
C     COMMON DECK "RK" INSERTED HERE
C     DEFINE SIZE REQUIRED FOR RAY STATE SAVE ARRAY
      PARAMETER (LRKAMS=87+2*100,NXRKMS=12+LRKAMS,MXEQPT=21)
      PARAMETER (NRKSAV=NXRKMS+MXEQPT-1)
      COMMON /RK/ NEQS,STEP,MODE,E1MAX,E1MIN,E2MAX,E2MIN,FACT,RSTART
C     COMMON DECK "RKTIME" INSERTED HERE
      COMMON/CRKTIME/RKTIME
      COMMON Y(20),T,SPACE,DYDT(20)
      REAL RV(1)
      DOUBLE PRECISION YU
      EQUIVALENCE(RKTIME,IRKTIME),(RV,R)
C
      REAL SVBUF(NRKSAV)
      LOGICAL SAVED
C
      DATA SAVED/.FALSE./
C     PERFORM CLOCK ADVANCE(SEE 'GET' ROUTINE)
      IRKTIME=IRKTIME+1
C
      IF(RSTART.NE.0.0 .OR. MODE.LE.2) GO TO 250
C
      NV=NV+1
      IF(NV.LE.4) GO TO 300
C
250   TOLD=T
C     PERFORM NUMERICAL INTEGRATION TO TIME 'T'
      CALL RKAM1
```

```
      IF(MODE.LE.2) GO TO 400
      IF(RSTART.NE.0.0) GO TO 260
C     SEARCH FOR REQUESTED VALUE IN PIPELINE
C     TAKING SIGN OF 'STEP' INTO ACCOUNT.
      TUP=TOLD+.25*STEP
      DO 150 NV=1,3
      IF(SPACE.GT.0.0.AND.TUP.LT.XV(NV)) GO TO 300
      IF(SPACE.LT.0.0.AND.TUP.GT.XV(NV)) GO TO 300
150   CONTINUE
C     GOT IT ALREADY, RETURN
      GO TO 400
C
260   NV=1
C     RETRIEVE REQUESTED VALUES FROM PIPELINE
300   T=XV(NV)
      DO 350 I=1,NEQS
      Y(I)=YU(NV,I)
350   DYDT(I)=FV(NV,I)
C
C     STANDARD EXIT SEQUENCE
400   RETURN
C
      ENTRY RKSAVE(SVBUF)
C
      SVBUF(1)=NV
      SVBUF(2)=NEQS
      SVBUF(3)=MM
      SVBUF(4)=SPACE
      SVBUF(5)=MODE
      CALL RMOVE(SVBUF(6),E1MAX,6)
      CALL RMOVE(SVBUF(12),XV,LRKAMS)
      CALL RMOVE(SVBUF(NXRKMS),Y,MXEQPT)
C
      SAVED=.TRUE.
      RETURN
      ENTRY RKRSTR(SVBUF)
C
      IF(.NOT.SAVED) RETURN
C
      NV=SVBUF(1)
      NEQS=SVBUF(2)
      MM=SVBUF(3)
      SPACE=SVBUF(4)
      MODE=SVBUF(5)
      CALL RMOVE(E1MAX,SVBUF(6),6)
      CALL RMOVE(XV,SVBUF(12),LRKAMS)
      CALL RMOVE(Y,SVBUF(NXRKMS),MXEQPT)
C
      WRITE(3,*) 'RESTORING NV,NEQS,MM,ALPHA,T='
     1    ,NV,NEQS,MM,ALPHA,T
      RETURN
      END
```

```
      SUBROUTINE RKAM1
C        NUMERICAL INTEGRATION OF DIFFERENTIAL EQUATIONS
C     THIS ROUTINE IS A MODIFICATION OF RKAMSUB, WHICH WAS WRITTEN
C     BY G.J. LASTMAN AND IS AVAILABLE THROUGH THE CDC CO-OP LIBRARY
C     AS 'D2 UTEX RKAMSUB'.
      COMMON /RK/ NN,SPACE,MODE,E1MAX,E1MIN,E2MAX,E2MIN,FACT,RSTART
      COMMON Y(20),T,STEP,DYDT(20)
      COMMON/RKAMS/XV(5),FV(4,20),YU(5,20),EPM,ALPHA,MM
      DIMENSION DELY(4,20),BET(4)
      DOUBLE PRECISION YU
C
      IF (RSTART.EQ.0.) GO TO 1000
      LL=1
      MM=1
      IF (MODE.EQ.1) MM=4
      ALPHA=T
      EPM=0.0
      BET(1)=0.5
      BET(2)=0.5
      BET(3)=1.0
      BET(4)=0.0
      STEP=SPACE
      R=19.0/270.0
      XV(MM)=T
      IF (E1MIN.LE.0.) E1MIN=E1MAX/50.
      IF (FACT.LE.0.) FACT=0.5
      CALL HAMLTN
      DO 320 I=1,NN
      FV(MM,I)=DYDT(I)
  320 YU(MM,I)=Y(I)
      RSTART=0.
      GO TO 1001
 1000 IF (MODE.NE.1) GO TO 2000
C
C          RUNGE-KUTTA
 1001 DO 1034 K=1,4
      DO 1350 I=1,NN
      DELY(K,I)=STEP*FV(MM,I)
      Z=YU(MM,I)
 1350 Y(I)=Z+BET(K)*DELY(K,I)
      T=BET(K)*STEP+XV(MM)
      CALL HAMLTN
      DO 1034 I=1,NN
 1034 FV(MM,I)=DYDT(I)
      DO 1039 I=1,NN
      DEL=(DELY(1,I)+2.0*DELY(2,I)+2.0*DELY(3,I)+DELY(4,I))/6.0
 1039 YU(MM+1,I)=YU(MM,I)+DEL
      MM=MM+1
      XV(MM)=XV(MM-1)+STEP
      DO 1400 I=1,NN
 1400 Y(I)=YU(MM,I)
      T=XV(MM)
      CALL HAMLTN
```

```
      IF (MODE.EQ.1) GO TO 42
      DO 150 I=1,NN
  150 FV(MM,I)=DYDT(I)
      IF (MM.LE.3) GO TO 1001
C
C            ADAMS-MOULTON
 2000 DO 2048 I=1,NN
      DEL=STEP*(55.*FV(4,I)-59.*FV(3,I)+37.*FV(2,I)-9.*FV(1,I))/24.
      Y(I)=YU(4,I)+DEL
 2048 DELY(1,I)=Y(I)
      T=XV(4)+STEP
      CALL HAMLTN
      XV(5)=T
      DO 2051 I=1,NN
      DEL=STEP*(9.*DYDT(I)+19.*FV(4,I)-5.*FV(3,I)+FV(2,I))/24.
      YU(5,I)=YU(4,I)+DEL
 2051 Y(I)=YU(5,I)
      CALL HAMLTN
      IF (MODE.LE.2) GO TO 42
C
C            ERROR ANALYSIS
      SSE=0.0
      DO 3033 I=1,NN
      EPSIL=R*ABS(Y(I)-DELY(1,I))
      IF (MODE.EQ.3.AND.Y(I).NE.0.) EPSIL=EPSIL/ABS(Y(I))
      IF (SSE.LT.EPSIL) SSE=EPSIL
 3033 CONTINUE
      IF (E1MAX.GT.SSE) GO TO 3035
      IF (ABS(STEP).LE.E2MIN) GO TO 42
      LL=1
      MM=1
      STEP=STEP*FACT
      GO TO 1001
 3035 IF (LL.LE.1.OR.SSE.GE.E1MIN.OR.E2MAX.LE.ABS(STEP)) GO TO 42
      LL=2
      MM=3
      XV(2)=XV(3)
      XV(3)=XV(5)
      DO 5363 I=1,NN
      FV(2,I)=FV(3,I)
      FV(3,I)=DYDT(I)
      YU(2,I)=YU(3,I)
 5363 YU(3,I)=YU(5,I)
      STEP=2.0*STEP
      GO TO 1001
C
C            EXIT ROUTINE
   42 LL=2
      MM=4
      DO 12 K=1,3
      XV(K)=XV(K+1)
      DO 12 I=1,NN
      FV(K,I)=FV(K+1,I)
   12 YU(K,I)=YU(K+1,I)
      XV(4)=XV(5)
```

```
      DO 52 I=1,NN                                                      RKAM1110
      FV(4,I)=DYDT(I)                                                   RKAM1111
   52 YU(4,I)=YU(5,I)                                                   RKAM1112
      IF (MODE.LE.2) RETURN                                             RKAM1113
      E=ABS(XV(4)-ALPHA)                                                RKAM1114
      IF (E.LE.EPM+.25*STEP) GO TO 2000                                 RKAM1115
      EPM=E                                                             RKAM1116
      RETURN                                                            RKAM1117
         END                                                            RKAM1118
```

```
      SUBROUTINE BACKUP(Z,RSIGN,TC)                                     BACKUP 2
C     MOVES THE RAY TO THE CLOSEST INTERSECTION WITH THE RECEIVER       BACKUP 3
C     OR TERRAIN SURFACE(VARIABLES 'FSELECT' OR 'GSELECT' IN LABELED    BACKUP 4
C     COMMONS /RR/ OR /GG/, RESPECTIVELY, TELL WHICH KIND OF SURFACE).  BACKUP 5
C                                                                       BACKUP 6
      CHARACTER*9 NBAK,NGRAZ,NTRY                                       BACKUP 7
C                                                                       BACKUP 8
      PARAMETER (PRNZTL=0.5E-4,PRNDZTL=1.E-6)                           BACKUP 9
C                                                                       BACKUP10
      COMMON /RK/ N,STEP,MODE,E1MAX,E1MIN,E2MAX,E2MIN,FACT,RSTART       BACKUP11
C     COMMON DECK "TRAC" INSERTED HERE                                  CTRAC  2
      LOGICAL GROUND,SURF,PERIGE,THERE,MINDIS,NEWRAY                    CTRAC  4
      COMMON /TRAC/ GROUND,SURF,PERIGE,THERE,MINDIS,NEWRAY,SMT,OSMT     CTRAC  5
      COMMON/TRAC/ROLD(20),DROLD(20),TOLD,ZDOT,D2Z,RAD,RAD1             CTRAC  6
      COMMON R(20),T,STP,DRDT(20)                                       BACKUP13
      PARAMETER (NWARSZ=1000)                                           CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      EQUIVALENCE (EARTHR,W(1)),(INTYP,W(41)),(STEP1,W(44))             BACKUP15
      REAL KR                                                           BACKUP16
      EQUIVALENCE (KR,R(4)),(DKRDT,DRDT(4))                             BACKUP17
      REAL INTYP                                                        BACKUP18
      LOGICAL PCNTRL                                                    BACKUP19
      DATA NBAK,NGRAZ/' BACK UP0', ' GRAZE 1'/                          BACKUP20
C                                                                       BACKUP21
      NTRY=NBAK                                                         BACKUP22
      GO TO 100                                                         BACKUP23
C                                                                       BACKUP24
      ENTRY GRAZE(Z,RSIGN,TC)                                           BACKUP25
      NTRY=NGRAZ                                                        BACKUP26
C                                                                       BACKUP27
C     DEFINE BASE STEP SIZE                                             BACKUP28
100   STPB=STP                                                          BACKUP29
C                                                                       BACKUP30
C     DEFINE BACKUP LOCATION TOLERANCES BASED ON INTEGRATION MODE       BACKUP31
C                                                                       BACKUP32
      IF(MODE.LT.3) THEN                                                BACKUP33
         TOL=STEP1                                                      BACKUP34
      ELSEIF(MODE.EQ.3) THEN                                            BACKUP35
         TOL=ABS(E1MAX*STPB)                                            BACKUP36
      ELSE                                                              BACKUP37
         TOL=E1MAX                                                      BACKUP38
```

```
      ENDIF
C     USE THIS OR MINIMUM STEP SIZE WHICHEVER IS LESS
      TOL=AMIN1(TOL,FACT*E2MIN)
C
      PCNTRL=W(73).NE.0.0
      THERE=.TRUE.
C      STEP TO ESTIMATED CROSSING AT TC.
C********* DIAGNOSTIC PRINTOUT
      IF(PCNTRL) CALL PRINTR (NTRY,0.)
      STEP=TC-T
      STEP=SIGN(AMIN1(ABS(STPB),ABS(STEP)),STEP)
      MODE=1
      RSTART=1
      TOLD=T
      CALL RMOVE(DROLD,DRDT,3)
      ZDOLD=DOT(Z)
      CALL RKAM
      RSTART=1.0
      ZDOT=DOT(Z)
      IF(NTRY.EQ.NGRAZ .OR. RSIGN*ZDOT.LE.0.0) GO TO 12
C
C********* FIND NEAREST INTERSECTION OF RAY WITH THE HEIGHT HS
      DO 10 I=1,10
      STEP=-Z/ZDOT
      STEP=SIGN(AMIN1(ABS(STPB),ABS(STEP)),STEP)
      IF (ABS(Z).LT.PRNZTL .AND. ABS(STEP).LT.TOL) GO TO 60
C********* DIAGNOSTIC PRINTOUT
      IF(PCNTRL) CALL PRINTR(' BACK UP1',0.)
      MODE=1
      RSTART=1.
      TOLD=T
      CALL RMOVE(DROLD,DRDT,3)
      ZDOLD=ZDOT
      CALL RKAM
      ZDOT=DOT(Z)
   10 RSTART=1.
C
C********* FIND NEAREST CLOSEST APPROACH OF RAY TO THE HEIGHT HS
12    THERE=.FALSE.
      DO 20 I=1,10
C     DO 'LOCAL' PARABOLIC FIT
      CALL FIT3(Z,ZOLD,ZDOLD)
      STEP=-ZDOT/D2Z
      STEP=SIGN(AMIN1(ABS(STPB),ABS(STEP)),STEP)
      IF (ABS(ZDOT).LE.PRNDZTL .AND. ABS(STEP).LT.TOL) GO TO 60
C********* DIAGNOSTIC PRINTOUT
      IF(PCNTRL) CALL PRINTR (' GRAZE 1 ',0.)
      MODE=1
      RSTART=1.
      TOLD=T
      CALL RMOVE(DROLD,DRDT,3)
      ZOLD=Z
      ZDOLD=ZDOT
      CALL RKAM
      RSTART=1.
```

```
      IF (D2Z*Z.LT.0.) GO TO 30                                         BACKUP94
      IF(KPARLEL(Z).EQ.0.0) GO TO 60                                    BACKUP95
   20 CONTINUE                                                          BACKUP96
      WRITE(3,350)                                                      BACKUP97
350   FORMAT(' ****** COULDN''T FIND CLOSEST APPROACH IN 10 STEPS')     BACKUP98
      GO TO 60                                                          BACKUP99
C                                                                       BACKU100
   30 CONTINUE                                                          BACKU101
C********** DIAGNOSTIC PRINTOUT                                         BACKU102
      IF(PCNTRL) CALL PRINTR (' BACK UP2',0.)                           BACKU103
      MODE=1                                                            BACKU104
C********** ESTIMATE DISTANCE TO NEAREST INTERSECTION GOING THE RIGHT   BACKU105
C********** DIRECTION OF RAY WITH HEIGHT HS                             BACKU106
      CALL FIT3(Z,ZOLD,ZDOLD)                                           BACKU107
      STEP=(-ZDOT+RSIGN*RAD1)/D2Z                                       BACKU108
      RSTART=1.                                                         BACKU109
      CALL RKAM                                                         BACKU110
      RSTART=1.                                                         BACKU111
C********** FIND NEAREST INTERSECTION OF RAY WITH HEIGHT HS             BACKU112
      DO 40 I=1,10                                                      BACKU113
      ZDOT=DOT(Z)                                                       BACKU114
      STEP=-Z/ZDOT                                                      BACKU115
      STEP=SIGN(AMIN1(ABS(STPB),ABS(STEP)),STEP)                        BACKU116
      IF (ABS(Z).LT.PRNZTL .AND. ABS(STEP).LT.TOL) GO TO 50             BACKU117
C********** DIAGNOSTIC PRINTOUT                                         BACKU118
      IF(PCNTRL) CALL PRINTR (' BACK UP3',0.)                           BACKU119
      MODE=1                                                            BACKU120
      RSTART=1.                                                         BACKU121
      CALL RKAM                                                         BACKU122
   40 RSTART=1.                                                         BACKU123
   50 THERE=.TRUE.                                                      BACKU124
C********** RESET STANDARD MODE AND INTEGRATION TYPE                    BACKU125
   60 MODE=INTYP                                                        BACKU126
      STEP=STPB                                                         BACKU127
      RETURN                                                            BACKU128
         END                                                            BACKU129
```

```
      FUNCTION REFLECT(Z)                                               REFLECT2
C     COMPUTES NORMAL AND PARALLEL COMPONENTS OF THE K-VECTOR AT        REFLECT3
C     REFLECTION POINTS TO A SURFACE. WIND EFFECTS ARE INCLUDED.        REFLECT4
C                                                                       REFLECT5
      REAL Z(12),KPARLEL,KNORM                                          REFLECT6
C     COMMON DECK "CC" INSERTED HERE                                    CCC    2
      REAL MODC                                                         CCC    4
      COMMON/CC/MODC(4),CS,PCST,PCSR,PCSTH,PCSPH                        CCC    5
C     COMMON DECK "WWR" INSERTED HERE                                   CWWR   2
      PARAMETER (NWARSZ=1000)                                           CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
```

```
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),           CWW2    5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),          CWW2    6
     8 (RCVRH,W(20)),                                                    CWW2    7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2    8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                       CWW2    9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                       CWW2   10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                      CWW2   11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),        CWW2   12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))         CWW2   13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),         CWW2   14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))               CWW2   15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                    CWW2   16
C     COMMON DECK "RKAM" INSERTED HERE                                   RKAMCOM2
      REAL KR,KTH,KPH                                                    RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)        RKAMCOM5
C     COMMON DECK "UU" INSERTED HERE                                     CUU     2
      REAL MODU                                                          CUU     4
      COMMON/UU/MODU(4)                                                  CUU     5
     1  ,V  ,PVT  ,PVR  ,PVTH  ,PVPH                                     CUU     6
     2  ,VR ,PVRT ,PVRR ,PVRTH ,PVRPH                                    CUU     7
     3  ,VTH,PVTHT,PVTHR,PVTHTH,PVTHPH                                   CUU     8
     4  ,VPH,PVPHT,PVPHR,PVPHTH,PVPHPH                                   CUU     9
C     COMMON DECK "FNDER" INSERTED HERE                                  CFNDER 2
      COMMON/FNDER/NZ,NPZR,NPZRR,NPZRTH,NPZRPH,NPZTH,NPZPH,NPZTHTH       CFNDER 4
      COMMON/FNDER/NPZPHPH,NPZTHPH,NSELECT,NTIME                         CFNDER 5
C                                                                        REFLEC12
      REAL NR,NTH,NPH,KRNR,KPARR,KPARTH,KPARPH                           REFLEC13
C                                                                        REFLEC14
      IENTRY=1                                                           REFLEC15
      REFLECT=0.0                                                        REFLEC16
      GO TO 5                                                            REFLEC17
C                                                                        REFLEC18
      ENTRY KPARLEL(Z)                                                   REFLEC19
      KPARLEL=0.0                                                        REFLEC20
      IENTRY=2                                                           REFLEC21
      GO TO 5                                                            REFLEC22
C                                                                        REFLEC23
      ENTRY KNORM(Z)                                                     REFLEC24
      KNORM=0.0                                                          REFLEC25
      IENTRY=3                                                           REFLEC26
C                                                                        REFLEC27
5     Z(1)=GET(Z)                                                        REFLEC28
      NR=Z(NPZR)                                                         REFLEC29
      NTH=Z(NPZTH)/R                                                     REFLEC30
      NPH=Z(NPZPH)/(R*SIN(TH))                                           REFLEC31
C                                                                        REFLEC32
      CALL RENORM(NR,1.0,3)                                              REFLEC33
C     COMPUTE THE NORMAL COMPONENT OF K-VECTOR TO SURFACE                REFLEC34
      KRNR=KR*NR+KTH*NTH+KPH*NPH                                         REFLEC35
      IF(IENTRY.NE.3) GO TO 8                                            REFLEC36
C     IF ENTRY 3 THEN WE ARE DONE                                        REFLEC37
      REFLECT=KRNR                                                       REFLEC38
      RETURN                                                             REFLEC39
C                                                                        REFLEC40
C     COMPUTE THE PARALLEL VECTOR COMPONENT                              REFLEC41
```

```
8       KPARR=KR-KRNR*NR                                              REFLEC42
        KPARTH=KTH-KRNR*NTH                                           REFLEC43
        KPARPH=KPH-KRNR*NPH                                           REFLEC44
C                                                                     REFLEC45
        IF(IENTRY.NE.2) GO TO 10                                      REFLEC46
        REFLECT=ABS(KPARR)+ABS(KPARTH)+ABS(KPARPH)                    REFLEC47
        RETURN                                                        REFLEC48
C                                                                     REFLEC49
10      OWIPAR=OW-KPARR*VR-KPARTH*VTH-KPARPH*VPH                      REFLEC50
        VNR=VR*NR+VTH*NTH+VPH*NPH                                     REFLEC51
C                                                                     REFLEC52
        FCTR=2.0*(KRNR+OWIPAR*VNR/(CS-VNR*VNR))                       REFLEC53
        KR=KR-FCTR*NR                                                 REFLEC54
        KTH=KTH-FCTR*NTH                                              REFLEC55
        KPH=KPH-FCTR*NPH                                              REFLEC56
        END                                                           REFLEC57
```

```
      SUBROUTINE FIT(Z,ZOLD,ZDOLD)                                    FIT     2
C     COMPUTES THREE TYPES OF PARABOLIC FITS TO RAY PATH RELATIVE     FIT     3
C     TO TERRAIN.                                                     FIT     4
C                                                                     FIT     5
      REAL Z(12)                                                      FIT     6
C     COMMON DECK "RKAM" INSERTED HERE                                RKAMCOM2
      REAL KR,KTH,KPH                                                 RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)     RKAMCOM5
C     COMMON DECK "TRAC" INSERTED HERE                                CTRAC   2
      LOGICAL GROUND,SURF,PERIGE,THERE,MINDIS,NEWRAY                  CTRAC   4
      COMMON /TRAC/ GROUND,SURF,PERIGE,THERE,MINDIS,NEWRAY,SMT,OSMT   CTRAC   5
      COMMON/TRAC/ROLD(20),DROLD(20),TOLD,ZDOT,D2Z,RAD,RAD1           CTRAC   6
C     COMMON DECK "FNDER" INSERTED HERE                               CFNDER  2
      COMMON/FNDER/NZ,NPZR,NPZRR,NPZRTH,NPZRPH,NPZTH,NPZPH,NPZTHTH    CFNDER  4
      COMMON/FNDER/NPZPHPH,NPZTHPH,NSELECT,NTIME                      CFNDER  5
C                                                                     FIT    10
      REAL D2(3)                                                      FIT    11
C                                                                     FIT    12
C     USE FIT OF APPENDIX 'J' OF REPORT 'WPL-103' WHICH USES          FIT    13
C     WEIGHTED ESTIMATE OF 1ST DERIVATIVE.                            FIT    14
C                                                                     FIT    15
      IENTRY=1                                                        FIT    16
      GO TO 5                                                         FIT    17
C                                                                     FIT    18
C     USE MODIFIED FIT REQUIRING HEIGHTS OF PARABOLA VERTICES FROM    FIT    19
C     CURRENT AND PREVIOUS RAY POINTS.                                FIT    20
      ENTRY FIT2(Z,ZOLD,ZDOLD)                                        FIT    21
      IENTRY=2                                                        FIT    22
      GO TO 5                                                         FIT    23
C                                                                     FIT    24
C     USE FIT OF APPENDIX U(LOCAL VALUE OF 1ST DERIVATIVE)            FIT    25
      ENTRY FIT3(Z,ZOLD,ZDOLD)                                        FIT    26
      IENTRY=3                                                        FIT    27
C                                                                     FIT    28
```

```
5       ZDOT=DOT(Z)
C
        DTI=1.0/(TPULSE-TOLD)
        DO 10 I=1,3
10      D2(I)=(DRDT(I)-DROLD(I))*DTI
C
        D2Z=Z(NPZR)*D2(1)+Z(NPZTH)*D2(2)+Z(NPZPH)*D2(3)
     1          +Z(NPZRR)*DRDT(1)*DRDT(1)
     1          +Z(NPZTHTH)*DRDT(2)*DRDT(2)
     1          +Z(NPZPHPH)*DRDT(3)*DRDT(3)
     1   +2.0*(Z(NPZRTH)*DRDT(1)*DRDT(2)
     1          +Z(NPZRPH)*DRDT(1)*DRDT(3)
     1          +Z(NPZTHPH)*DRDT(2)*DRDT(3)  )
C
C       THE STATEMENTS FROM HERE TO 'END FIT' IMPLEMENT THE
C       PARABOLIC FITS IN EQUATIONS J.1 AND U.3 OF THE TEXT.
C
        IF(IENTRY.NE.2) GO TO 30
        SMT=0.
        IF (D2Z.NE.0.) SMT=0.5*ZDOT*ZDOT/D2Z
C       USE FIT U.3 AT THE PREVIOUS POINT OF RAY PATH
        OSMT=0.
        IF (D2Z.NE.0.) OSMT=0.5*ZDOLD*ZDOLD/D2Z
        GO TO 2000
C
C       IMPLEMENTATION OF FIT FOR EQUATION J.1
C
30      IF(IENTRY.EQ.3) GO TO 1000
        ZDOTM=.5*(ZDOT+ZDOLD)
C       IMPLEMENT TESTS OF EQUATIONS J.2 AND J.3
        IF(ABS(ZDOTM).LE..05*ABS(ZDOT)) GO TO 1000
        FCTR=(Z(NZ)-ZOLD)*DTI/ZDOTM
        D2Z=FCTR*D2Z
        ZDOT=FCTR*ZDOT
C       END FIT
C
C       COMMON CODE FOR FIT AND FIT3
1000    RAD=ZDOT*ZDOT-2.0*Z(NZ)*D2Z
        RAD1=SQRT(AMAX1(RAD,0.0))
C
2000    CONTINUE
C
        END
```

```
        FUNCTION GET1(Z)
C       FUNCTIONALLY THE SAME AS 'GET' PROGRAM, SEE DOCUMENTATION THERE
C       NEEDED BECAUSE RECEIVER MODELS WILL CALL GET TO OBTAIN TERRAIN VA
C       BUT THEY THEMSELVES ARE CALLED VIA GET. HENCE HAVE RE-ENTRANCE
C       PROBLEM.
C
        REAL Z(*)
```

```
      REAL PF(10),PG(10)
C     COMMON DECK "RKAM" INSERTED HERE
      REAL KR,KTH,KPH
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)
C     COMMON DECK "FNDER" INSERTED HERE
      COMMON/FNDER/NZ,NPZR,NPZRR,NPZRTH,NPZRPH,NPZTH,NPZPH,NPZTHTH
      COMMON/FNDER/NPZPHPH,NPZTHPH,NSELECT,NTIME
C     COMMON DECK "RR" INSERTED HERE
      REAL MODREC
      COMMON/RR/ MODREC(4)
      COMMON/RR/F,PFR,PFRR,PFRTH,PFRPH
      COMMON/RR/PFTH,PFPH,PFTHTH,PFPHPH,PFTHPH,FSELECT,FTIME
C     COMMON DECK "GG" INSERTED HERE
      REAL MODG
      COMMON/GG/MODG(4)
      COMMON/GG/G,PGR,PGRR,PGRTH,PGRPH
      COMMON/GG/PGTH,PGPH,PGTHTH,PGPHPH,PGTHPH,GSELECT,GTIME
C     COMMON DECK "RKTIME" INSERTED HERE
      COMMON/CRKTIME/RKTIME
C
C     COMMON DECK "RMACH" INSERTED HERE
      COMMON/CRMACH/RMACH(5)
      EQUIVALENCE(RKTIME,IRKTIME)
C     COMMON DECK "WWR" INSERTED HERE
      PARAMETER (NWARSZ=1000)
      COMMON/WW/ID(10),MAXW,W(NWARSZ)
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),
     8 (RCVRH,W(20)),
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),
     6 (HMIN,W(27)),(RGMAX,W(28)),
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))
C
      COMMON/CGET/ZERO
C
      IENTRY=1
      GO TO 5
C
      ENTRY DOT1(Z)
      IENTRY=2
5     CONTINUE
      IF(ZERO.EQ.0.0) ZERO=EARTHR*RMACH(3)*2.0
C
      IF(ITEST(Z(NTIME)).EQ.IRKTIME) GO TO 10
C
      IF(Z(NSELECT).EQ.FSELECT) CALL RECEVER
```

```
      IF(Z(NSELECT).EQ.GSELECT) CALL TOPOG                               GET    45
      Z(NTIME)=RKTIME                                                    GET    46
C     REMOVE MACHINE ROUND OFF NOISE FROM EXACT RECEIVER LOCATIONS       GET    47
      IF(ABS(Z(NZ)).LE.ZERO) Z(NZ)=0.0                                   GET    48
C                                                                        GET    49
10    IF(IENTRY.NE.1) GO TO 20                                           GET    50
      GET1=Z(NZ)                                                         GET    51
      RETURN                                                             GET    52
C                                                                        GET    53
20    GET1=Z(NPZR)*DRDT(1)+Z(NPZTH)*DRDT(2)+Z(NPZPH)*DRDT(3)             GET    54
      RETURN                                                             GET    55
      ENTRY GETST1(Z)                                                    GET    56
C                                                                        GET    57
C     FORCE LOAD THE GET ROUTINE                                         GET    58
      CALL GET                                                           GET    59
      END                                                                GET    60
```

```
      FUNCTION GET(Z)                                                    GET    61
C     'GET' AND ENTRY 'DOT' PROVIDE A CONTROL METHOD FOR AVOIDING        GET    62
C     REDUNDANT CALLS TO THE TERRAIN AND RECEIVER MODELS. THE VALUES     GET    63
C     RETURNED ARE THE FUNCTION VALUES FOR 'F' OR 'G' OR THEIR DERIVATI GET    64
C     (VIA 'DOT' ENTRY). UNNECESSARY CALLS ARE ELIMINATED THROUGH USE    GET    65
C     OF 'TIME OF CALL' VARIABLES WHICH ARE COMPARED WITH THE CURRENT    GET    66
C     LAST CALL TIME MAINTAINED BY THE 'RKAM' PROGRAM. WHEN VALUES ARE   GET    67
C     NOT CURRENT THEY ARE UPDATED BY CALLS TO THE APPROPRIATE ROUTINES GET    68
C                                                                        GET    69
      REAL Z(*),PF(10),PG(10)                                            GET    70
C     COMMON DECK "RKAM" INSERTED HERE                                   RKAMCOM2
      REAL KR,KTH,KPH                                                    RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)        RKAMCOM5
C     COMMON DECK "FNDER" INSERTED HERE                                  CFNDER 2
      COMMON/FNDER/NZ,NPZR,NPZRR,NPZRTH,NPZRPH,NPZTH,NPZPH,NPZTHTH       CFNDER 4
      COMMON/FNDER/NPZPHPH,NPZTHPH,NSELECT,NTIME                         CFNDER 5
C     COMMON DECK "RR" INSERTED HERE                                     CRR     2
      REAL MODREC                                                        CRR     4
      COMMON/RR/ MODREC(4)                                               CRR     5
      COMMON/RR/F,PFR,PFRR,PFRTH,PFRPH                                   CRR     6
      COMMON/RR/PFTH,PFPH,PFTHTH,PFPHPH,PFTHPH,FSELECT,FTIME             CRR     7
C     COMMON DECK "GG" INSERTED HERE                                     CGG     2
      REAL MODG                                                          CGG     4
      COMMON/GG/MODG(4)                                                  CGG     5
      COMMON/GG/G,PGR,PGRR,PGRTH,PGRPH                                   CGG     6
      COMMON/GG/PGTH,PGPH,PGTHTH,PGPHPH,PGTHPH,GSELECT,GTIME             CGG     7
C     COMMON DECK "RKTIME" INSERTED HERE                                 CRKTIME2
      COMMON/CRKTIME/RKTIME                                              CRKTIME4
C                                                                        GET    76
C     COMMON DECK "RMACH" INSERTED HERE                                  CRMACH 2
      COMMON/CRMACH/RMACH(5)                                             CRMACH 4
      EQUIVALENCE(RKTIME,IRKTIME)                                        GET    78
C                                                                        GET    79
C     COMMON DECK "WWR" INSERTED HERE                                    CWWR    2
```

```
      PARAMETER (NWARSZ=1000)                                           CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
     8 (RCVRH,W(20)),                                                   CWW2   7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2   9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2  10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2  11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2  12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2  13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2  15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2  16
C                                                                       GET   81
      COMMON/CGET/ZERO                                                  GET   82
C                                                                       GET   83
      EQUIVALENCE (PF,PFR), (PG,PGR)                                    GBL    2
      DATA PF,PG/1.0,9*0.0,1.0,9*0.0/                                   GBL    3
      DATA NZ,NPZR,NPZRR,NPZRTH,NPZRPH,NPZTH,NPZPH,NPZTHTH              GBL    4
     1    ,NPZPHPH,NPZTHPH,NSELECT,NTIME                                GBL    5
     2  /1,2,3,4,5,6,7,8,9,10,11,12/                                    GBL    6
      DATA FTIME,GTIME,FSELECT,GSELECT/2*-1.0,8HRECEIVER ,7HTERRAIN /   GBL    7
      DATA ZERO/0.0/                                                    GBL    8
C                                                                       GET   86
      IENTRY=1                                                          GET   87
      GO TO 5                                                           GET   88
C                                                                       GET   89
      ENTRY DOT(Z)                                                      GET   90
      IENTRY=2                                                          GET   91
5     CONTINUE                                                          GET   92
      IF(ZERO.EQ.0.0) ZERO=EARTHR*RMACH(3)*2.0                          GET   93
C                                                                       GET   94
      IF(ITEST(Z(NTIME)).EQ.IRKTIME) GO TO 10                           GET   95
C                                                                       GET   96
      IF(Z(NSELECT).EQ.FSELECT) CALL RECEVER                            GET   97
      IF(Z(NSELECT).EQ.GSELECT) CALL TOPOG                              GET   98
      Z(NTIME)=RKTIME                                                   GET   99
C     REMOVE MACHINE ROUND OFF NOISE FROM EXACT RECEIVER LOCATIONS      GET  100
      IF(ABS(Z(NZ)).LE.ZERO) Z(NZ)=0.0                                  GET  101
C                                                                       GET  102
10    IF(IENTRY.NE.1) GO TO 20                                          GET  103
      GET=Z(NZ)                                                         GET  104
      RETURN                                                            GET  105
C                                                                       GET  106
20    GET=Z(NPZR)*DRDT(1)+Z(NPZTH)*DRDT(2)+Z(NPZPH)*DRDT(3)             GET  107
      RETURN                                                            GET  108
      ENTRY DFCNCL(Z)                                                   GET  109
C                                                                       GET  110
C     FORCE LOAD THE MACHINE DEPENDENT CONSTANTS(SEE MODULE 'DFCNST')   GET  111
C                                                                       GET  112
      CALL DFCNST                                                       GET  113
```

```
      END                                                                GET   114

      FUNCTION ITEST(I)                                                  ITEST   2
C     USED TO PASS INTEGER VALUES THROUGH FOR VARIABLES TYPED REAL       ITEST   3
C     (AS IN MIXED REAL/INTEGER ARRAYS)                                  ITEST   4
      ITEST=I                                                            ITEST   5
      END                                                                ITEST   6

      SUBROUTINE CONBLK                                                  CONBLK 3
C     DATA INITIALIZATION AND FILE OPENING SERVICE ROUTINE               CONBLK 4
C                                                                        CONBLK 5
C     COMMON DECK "HDR" INSERTED HERE                                    CHDR   2
      CHARACTER*10 INITID*80,DAT,TOD                                     CHDR   4
      COMMON/HDR/SEC                                                     CHDR   5
      COMMON/HDRC/INITID,DAT,TOD                                         CHDR   6
C                                                                        CONBLK 9
C     COMMON DECK "CPROCFL" INSERTED HERE                                CPROCFL2
      INTEGER PMX,PNTBL,PITBL,PFRMTBL,IDSP(10)                           CPROCFL4
C     PARAMETER DECK "PGROUPS"                                           PGROUPS2
      PARAMETER (NCHPG1=11,NWPV=250,NSPGP=NCHPG1+2*NWPV+1)               PGROUPS3
      PARAMETER (MNGRP=9,MXGRP=69,MXLIST=MXGRP-MNGRP+2)                  PGROUPS4
      COMMON/PROCFL/LIST(MXLIST)                                         CPROCFL6
      COMMON/PROCFL/PMX,PNTBL(10),PITBL(10),PFRMTBL(10),PGP(NSPGP)       CPROCFL7
      EQUIVALENCE (PGP,IDSP)                                             CPROCFL8
C     COMMON DECK "CC" INSERTED HERE                                     CCC    2
      REAL MODC                                                          CCC    4
      COMMON/CC/MODC(4),CS,PCST,PCSR,PCSTH,PCSPH                         CCC    5
C     COMMON DECK "GG" INSERTED HERE                                     CGG    2
      REAL MODG                                                          CGG    4
      COMMON/GG/MODG(4)                                                  CGG    5
      COMMON/GG/G,PGR,PGRR,PGRTH,PGRPH                                   CGG    6
      COMMON/GG/PGTH,PGPH,PGTHTH,PGPHPH,PGTHPH,GSELECT,GTIME             CGG    7
C     COMMON DECK "RR" INSERTED HERE                                     CRR    2
      REAL MODREC                                                        CRR    4
      COMMON/RR/ MODREC(4)                                               CRR    5
      COMMON/RR/F,PFR,PFRR,PFRTH,PFRPH                                   CRR    6
      COMMON/RR/PFTH,PFPH,PFTHTH,PFPHPH,PFTHPH,FSELECT,FTIME             CRR    7
C     COMMON DECK "TT" INSERTED HERE                                     CTT    2
      REAL MODT                                                          CTT    4
      COMMON/TT/MODT(4), T,PTT,PTR,PTTH,PTPH                             CTT    5
C     COMMON DECK "UU" INSERTED HERE                                     CUU    2
      REAL MODU                                                          CUU    4
      COMMON/UU/MODU(4)                                                  CUU    5
     1  ,V  ,PVT  ,PVR  ,PVTH  ,PVPH                                     CUU    6
     2  ,VR ,PVRT ,PVRR ,PVRTH ,PVRPH                                    CUU    7
     3  ,VTH,PVTHT,PVTHR,PVTHTH,PVTHPH                                   CUU    8
     4  ,VPH,PVPHT,PVPHR,PVPHTH,PVPHPH                                   CUU    9
```

```
C                                                                       CONBLK16
C      COMMON DECK "RAYDEV" INSERTED HERE                               CRAYDEV2
C      DEVICE ASSIGNED TO RAYTRC INPUT FILE                             CRAYDEV4
       COMMON/RAYDEV/NRYIND,NDEVTMP,NFRMAT,NDEVGRP,NDEVBIN              CRAYDEV5
C      COMMON DECK "CUCON" INSERTED HERE                                CUCON  2
       COMMON/UCONV/CNVV(4,4)                                           CUCON  4
       CHARACTER PCV*3,CNVC*2                                           CUCON  5
       COMMON/UCONC/PCV(4),CNVC(4,4)                                    CUCON  6
C                                                                       CONBLK19
C      COMMON DECK "SS" INSERTED HERE                                   CSS    2
       REAL MODSURF                                                     CSS    4
       COMMON/SS/ MODSURF(4)                                            CSS    5
       COMMON/SS/U,PUR,PURR,PURTH,PURPH                                 CSS    6
       COMMON/SS/PUTH,PUPH,PUTHTH,PUPHPH,PUTHPH,USELECT,UTIME           CSS    7
C      COMMON DECK "CONST" INSERTED HERE                                CCONST 2
       COMMON/PCONST/CREF,RGAS,GAMMA                                    CCONST 4
       COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                        CCONST 5
C      COMMON DECK "WWR" INSERTED HERE                                  CWWR   2
       PARAMETER (NWARSZ=1000)                                          CWW1   3
       COMMON/WW/ID(10),MAXW,W(NWARSZ)                                  CWW1   4
       REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                               CWW2   2
       EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),   CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
     8 (RCVRH,W(20)),                                                   CWW2   7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2   9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2  10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2  11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2  12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2  13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2  15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2  16
C      COMMON DECK "B9" INSERTED HERE                                   CB8    2
       INTEGER GMX,GNTBL,GITBL,GFRMTBL,IDSG(10)                         CB8    4
       COMMON/B9/GMX,GNTBL(10),GITBL(10),GFRMTBL(10),GGP(113)           CB8    5
       EQUIVALENCE (GGP,IDSG),(ANG,GGP(11))                             CB8    6
C      COMMON DECK "B2" INSERTED HERE                                   CB2    2
       INTEGER DUMX,DUNTBL,DUITBL,DUFRMTB,IDSDU(10)                     CB2    4
       COMMON/B2/DUMX,DUNTBL(10),DUITBL(10),DUFRMTB(10),DUGP(10)        CB2    5
       EQUIVALENCE (DUGP,IDSDU)                                         CB2    6
C      COMMON DECK "B4" INSERTED HERE                                   CB4    2
       INTEGER DCMX,DCNTBL,DCITBL,DCFRMTB,IDSDC(10)                     CB4    4
       COMMON/B4/DCMX,DCNTBL(10),DCITBL(10),DCFRMTB(10),DCGP(10)        CB4    5
       EQUIVALENCE (DCGP,IDSDC)                                         CB4    6
C      COMMON DECK "B6" INSERTED HERE                                   CB6    2
       INTEGER DTMX,DTNTBL,DTITBL,DTFRMTB,IDSDT(10)                     CB6    4
       COMMON/B6/DTMX,DTNTBL(10),DTITBL(10),DTFRMTB(10),DTGP(10)        CB6    5
       EQUIVALENCE (DTGP,IDSDT)                                         CB6    6
C      COMMON DECK "B1" INSERTED HERE                                   CB1    2
       INTEGER UMX,UNTBL,UITBL,UFRMTBL,IDSU(10)                         CB1    4
       COMMON/B1/UMX,UNTBL(10),UITBL(10),UFRMTBL(10),UGP(10)            CB1    5
       EQUIVALENCE (UGP,IDSU)                                           CB1    6
```

```
C     COMMON DECK "B3" INSERTED HERE                                    CB3       2
      INTEGER CMX,CNTBL,CITBL,CFRMTBL,IDSC(10)                          CB3       4
      COMMON/B3/CMX,CNTBL(10),CITBL(10),CFRMTBL(10),CGP(512)            CB3       5
      EQUIVALENCE (CGP,IDSC),(ANC,CGP(11))                              CB3       6
C     COMMON DECK "B5" INSERTED HERE                                    CB5       2
      INTEGER TMX,TNTBL,TITBL,TFRMTBL,IDST(10)                          CB5       4
      COMMON/B5/TMX,TNTBL(10),TITBL(10),TFRMTBL(10),TGP(262)            CB5       5
      EQUIVALENCE (TGP,IDST),(ANT,TGP(11))                              CB5       6
C     COMMON DECK "B7" INSERTED HERE                                    CB7       2
      INTEGER MMX,MNTBL,MITBL,MFRMTBL,IDSM(10)                          CB7       4
      REAL MGP                                                          CB7       5
      COMMON/B7/MMX,MNTBL(10),MITBL(10),MFRMTBL(10),MGP(10)             CB7       6
      EQUIVALENCE (MGP,IDSM)                                            CB7       7
C     COMMON DECK "RINPLEX" INSERTED HERE                               CRINPLE2
      REAL KAY2,KAY2I                                                   CRINPLE4
      COMPLEX PNP,POLAR,LPOLAR                                          CRINPLE5
      LOGICAL SPACE                                                     CRINPLE6
      CHARACTER DISPM*6                                                 CRINPLE7
      COMMON/RINPL/DISPM                                                CRINPLE8
      COMMON /RIN/ MODRIN(8),RAYNAME(2,3),TYPE(3),SPACE                 CRINPLE9
      COMMON/RIN/OMEGMIN,OMEGMAX,KAY2,KAY2I                             CRINPL10
      COMMON/RIN/PNP(10),POLAR,LPOLAR,SGN                               CRINPL11
C     COMMON DECK "MM" INSERTED HERE                                    CMM       2
      REAL M,MODM                                                       CMM       4
      COMMON/MM/MODM(4),M,PMT,PMR,PMTH,PMPH                             CMM       5
C     COMMON DECK "PP" INSERTED HERE                                    CPP       2
      REAL MODP                                                         CPP       4
      COMMON/PP/MODP(4),P,PPT,PPR,PPTH,PPPH                             CPP       5
C     COMMON DECK "AA" INSERTED HERE                                    CAA       2
      REAL MODA                                                         CAA       4
      REAL MU,MUPT,MUPR,MUPTH,MUPPH                                     CAA       5
      REAL KAP,KAPPT,KAPPR,KAPPTH,KAPPPH                                CAA       6
      COMMON/AA/MODA(4),MU,MUPT,MUPR,MUPTH,MUPPH                        CAA       7
      COMMON/AA/KAP,KAPPT,KAPPR,KAPPTH,KAPPPH                           CAA       8
C     COMMON DECK "FLAG" INSERTED HERE                                  CFLAG     2
      LOGICAL NEWWR,NEWWP,NEWTRC,PENET                                  CFLAG     4
      COMMON /FLG/ NTYP,NEWWR,NEWWP,NEWTRC,PENET,LINES,IHOP,HPUNCH      CFLAG     5
      COMMON/FLGP/NSET                                                  CFLAG     6
C     COMMON DECK "RKTIME" INSERTED HERE                                CRKTIME2
      COMMON/CRKTIME/RKTIME                                             CRKTIME4
C                                                                       CONBLK37
      COMMON/RAYCON/MCONP                                               CONBLK38
C                                                                       CONBLK39
      EQUIVALENCE(RKTIME,IRKTIME)                                       CONBLK40
C                                                                       CONBLK41
C     COMMON DECK "B10" INSERTED HERE                                   CB9       2
      INTEGER DGMX,DGNTBL,DGITBL,DGFRMTB,IDSDG(10)                      CB9       4
      COMMON/B10/DGMX,DGNTBL(10),DGITBL(10),DGFRMTB(10),DGGP(10)        CB9       5
      EQUIVALENCE (DGGP,IDSDG)                                          CB9       6
C     COMMON DECK "B8" INSERTED HERE                                    CB10      2
      INTEGER RMX,RNTBL,RITBL,RFRMTBL,IDSR(10)                          CB10      4
      COMMON/B8/RMX,RNTBL(10),RITBL(10),RFRMTBL(10),RGP(10)             CB10      5
      EQUIVALENCE (RGP,IDSR)                                            CB10      6
C     COMMON DECK "CB17" INSERTED HERE                                  CB17      2
      INTEGER VMX,VNTBL,VITBL,VFRMTBL,IDSV(10)                          CB17      4
```

```
      COMMON/B17/VMX,VNTBL(10),VITBL(10),VFRMTBL(10),VGP(53)           CB17    5
      EQUIVALENCE (VGP,IDSV),(ANV,VGP(11))                             CB17    6
C     COMMON DECK "CB18" INSERTED HERE                                 CB18    2
      INTEGER DVMX,DVNTBL,DVITBL,DVFRMTB,IDSDV(10)                     CB18    4
      COMMON/B18/DVMX,DVNTBL(10),DVITBL(10),DVFRMTB(10),DVGP(11)       CB18    5
      EQUIVALENCE (DVGP,IDSDV),(ANDV,DVGP(11))                         CB18    6
C     COMMON DECK "CB19" INSERTED HERE                                 CB19    2
      INTEGER PRMX,PRNTBL,PRITBL,PRFRMTB,IDSPR(10)                     CB19    4
      COMMON/B19/PRMX,PRNTBL(10),PRITBL(10),PRFRMTB(10),PRGP(11)       CB19    5
      EQUIVALENCE (PRGP,IDSPR),(ANP,PRGP(11))                          CB19    6
C     COMMON DECK "CB20" INSERTED HERE                                 CB20    2
      INTEGER DPMX,DPNTBL,DPITBL,DPFRMTB,IDSDP(10)                     CB20    4
      COMMON/B20/DPMX,DPNTBL(10),DPITBL(10),DPFRMTB(10),DPGP(11)       CB20    5
      EQUIVALENCE (DPGP,IDSDP),(ANDP,DPGP(11))                         CB20    6
      REAL KVECT(22)                                                   CONBLK48
      REAL VSET(20)                                                    CONBLK49
      EQUIVALENCE(KVECT,KAY2),(VSET,V)                                 CONBLK50
C                                                                      CONBLK51
      DATA KVECT/22*0.0/                                               CONBLK52
      DATA RAYNAME/6*1H /                                              CONBLK53
      DATA VSET/20*0.0/                                                CONBLK54
      DATA OMEGMIN,OMEGMAX/0.0,0.0/                                    CONBLK55
      DATA POLAR,LPOLAR/(0.0,0.0),(1.0,0.0)/                           CONBLK56
C                                                                      CONBLK57
      DATA IRKTIME/0/                                                  CONBLK58
      DATA MCONP/0/                                                    CONBLK59
      DATA CREF,GAMMA,RGAS/1.0,1.4,8.31436E-3/                         CONBLK60
      DATA NRYIND,NDEVTMP,NFRMAT,NDEVGRP,NDEVBIN/8,4,0,10,11/          CONBLK61
      DATA PCV/'AN','LN','FQ',' '/                                     CONBLK62
      DATA CNVC/'RD' ,'KM' ,'RD' ,' '                                  CONBLK63
     1          ,'DG' ,' M' ,'HZ' ,' '                                 CONBLK64
     2          ,'KM' ,'FT' ,'KH' ,' '                                 CONBLK65
     3          ,'  ' ,'NM' ,'DG' ,' '/                                CONBLK66
      DATA CNVV /16*1.0 /                                              CONBLK67
C                                                                      CONBLK68
      DATA IDSDC,IDSDT,IDSDU,IDSC,IDSM,IDST,IDSU,IDSG/80*1H /          CONBLK69
      DATA IDSDG,IDSV,IDSDV,IDSPR,IDSDP,IDSR/60*1H /                   CONBLK70
      DATA MODG/7HNO MODL,0.0,7HNO MODL,0.0/                           CONBLK71
      DATA MODREC/7HNO MODL,0.0,7HNO MODL,0.0/                         CONBLK72
      DATA MODC/7HNO MODL,0.0,7HNO MODL,0.0/                           CONBLK73
      DATA MODM/7HNO MODL,0.0,7HNO MODL,0.0/                           CONBLK74
      DATA MODT/7HNO MODL,0.0,7HNO MODL,0.0/                           CONBLK75
      DATA MODU/7HNO MODL,0.0,7HNO MODL,0.0/                           CONBLK76
      DATA MODC/7HNO MODL,0.0,7HNO MODL,0.0/                           CONBLK77
      DATA MODT/7HNO MODL,0.0,7HNO MODL,0.0/                           CONBLK78
      DATA MODM/7HNO MODL,0.0,7HNO MODL,0.0/                           CONBLK79
      DATA MODP/7HNO MODL,0.0,7HNO MODL,0.0/                           CONBLK80
      DATA MODA/7HNO MODL,0.0,7HNO MODL,0.0/                           CONBLK81
C                                                                      CONBLK82
      RETURN                                                           CONBLK91
C                                                                      CONBLK92
      ENTRY STDINI                                                     CONBLK93
      CALL OPNREP(NDEVTMP,'TAPE4')                                     CONBLK94
      CALL OPNURP(NDEVBIN,'TAPE6')                                     CONBLK95
      CALL OPNREP(9,'PUNCH')                                           CONBLK96
```

```
C
      ENTRY STDINT
      OPEN(UNIT=NRYIND,FILE='DINP',STATUS='OLD',ERR=1000)
      REWIND NRYIND
      CALL OPNREP(2,'OUTPUT')
      CALL OPNREP(3,'DOUTP')
      CALL OPNURP(NDEVGRP,'TAPE5')
C
C     TEMPORARY WE HOPE, TO MAKE CRAY VERSION WORK
      FSELECT=8HRECEIVER
      GSELECT=7HTERRAIN
C  INITIALIZE RAYSET FILE
      READ(NRYIND,'(A)',END=1000) INITID
      CALL SETW
      CALL SYSDAT(DAT)
      CALL SYSTIM(TOD)
      CALL SYSSEC(SEC)
C     INITIALIZE POINT MODEL LIST
      LIST(1)=1
      LIST(2)=0
C     FILL FORMAT CONTROL ARRAYS
      PNTBL(1)=1
C     ALLOW FOR AN 80 CHARACTER IDENT STRING(A8)
      PNTBL(2)=NCHPG1
C     ALLOW MAXIMUM 250 WORD PER VARIABLE
      PITBL(2)=NWPV
C     FOR BOTH X AND Y PLUS 1 FOR NUMBER OF VARIABLES
      PNTBL(3)=NSPGP
      RETURN
1000  STOP 'DINP FORMAT ERROR'
      END
```

```
      LOGICAL FUNCTION WCHANGE(W1,W2)
      REAL W1(400),W2(400)
C
C WCHANGE PROVIDES A TEST AGAINST TWO W-ARRAYS
C     IF BOTH WOULD PRODUCE THE SAME SET OF RAYS BY RAYTRC THEN RESULT
C     IS <FALSE>.
C
      INTEGER NDX(8)
C
      DATA NGRPS,NDX/8, 1,17, 21,26, 41,47, 100,399/
C
      WCHANGE=.FALSE.
      DO 20 N=1,NGRPS,2
         N1=NDX(N)
         N2=NDX(N+1)
      DO 20 I=N1,N2
         WCHANGE=W1(I).NE.W2(I)
         IF(WCHANGE) RETURN
20    CONTINUE
```

```
      END                                                               WCHANG21


      FUNCTION RENORM(VECTOR,NNORM,NCOMPS)                              RENORM 2
C     NORMALIZES 'NCOMPS' COMPONENT VECTOR 'VECTOR' TO MAGNITUDE        RENORM 3
C     'NNORM' AND RETURNS SQUARE ROOT OF FACTOR NEEDED.                 RENORM 4
      REAL NNORM,VECTOR(*)                                              RENORM 5
C                                                                       RENORM 6
      RENORM=0.0                                                        RENORM 7
      IF(NNORM.LE.0.0) RETURN                                           RENORM 8
C                                                                       RENORM 9
      RENORM=0.0                                                        RENORM10
      DO 10 I=1,NCOMPS                                                  RENORM11
10    RENORM=RENORM+VECTOR(I)*VECTOR(I)                                 RENORM12
      IF(RENORM.EQ.0.0) RETURN                                          RENORM13
C                                                                       RENORM14
      RENORM=SQRT(NNORM/RENORM)                                         RENORM15
      DO 20 I=1,NCOMPS                                                  RENORM16
20    VECTOR(I)=VECTOR(I)*RENORM                                        RENORM17
C                                                                       RENORM18
      RETURN                                                            RENORM19
      END                                                               RENORM20


      SUBROUTINE SET2(A,V,N)                                            SET2    2
C     SETS N COMPONENTS OF VECTER TO SINGLE VALUE V                     SET2    3
C                                                                       SET2    4
      REAL A(N)                                                         SET2    5
C                                                                       SET2    6
      DO 100 I=1,N                                                      SET2    7
100   A(I)=V                                                            SET2    8
      END                                                               SET2    9


      SUBROUTINE PRINTR(EVENT,CARD)                                     PRINTR 2
C                                                                       PRINTR 3
      CHARACTER EVENT*9,NWHY*8,CC*1,PC*1,TMP*9                          PRINTR 4
C                                                                       PRINTR 5
C     PRINTS OUTPUT AND OUTPUTS RAYSETS(MACHINE READABLE OUTPUT)        PRINTR 6
C     WHEN 'CARD' ARGUMENT NONZERO.                                     PRINTR 7
      REAL KNORM                                                        PRINTR 8
      DIMENSION G0(3,3),G1(3,3)                                         PRINTR 9
      CHARACTER*12 HEADRS(20),HEAD(20),UNITS(20),UNIT(20)               PRINTR10
      DIMENSION RPRINT(20),NPR(20)                                      PRINTR11
C     COMMON DECK "RK" INSERTED HERE                                    CRK     2
C     DEFINE SIZE REQUIRED FOR RAY STATE SAVE ARRAY                     CRK     4
```

```
      PARAMETER (LRKAMS=87+2*100,NXRKMS=12+LRKAMS,MXEQPT=21)            CRK      5
      PARAMETER (NRKSAV=NXRKMS+MXEQPT-1)                                CRK      6
      COMMON /RK/ NEQS,STEP,MODE,E1MAX,E1MIN,E2MAX,E2MIN,FACT,RSTART    CRK      7
C     COMMON DECK "CERR" INSERTED HERE                                  CERR     2
      COMMON/ERR/NERG,NERR,NERT,NERP                                    CERR     3
C     COMMON DECK "GG" INSERTED HERE                                    CGG      2
      REAL MODG                                                         CGG      4
      COMMON/GG/MODG(4)                                                 CGG      5
      COMMON/GG/G,PGR,PGRR,PGRTH,PGRPH                                  CGG      6
      COMMON/GG/PGTH,PGPH,PGTHTH,PGPHPH,PGTHPH,GSELECT,GTIME            CGG      7
C     COMMON DECK "CONST" INSERTED HERE                                 CCONST 2
      COMMON/PCONST/CREF,RGAS,GAMMA                                     CCONST 4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                         CCONST 5
C     COMMON DECK "FLAG" INSERTED HERE                                  CFLAG  2
      LOGICAL NEWWR,NEWWP,NEWTRC,PENET                                  CFLAG  4
      COMMON /FLG/ NTYP,NEWWR,NEWWP,NEWTRC,PENET,LINES,IHOP,HPUNCH      CFLAG  5
      COMMON/FLGP/NSET                                                  CFLAG  6
C     COMMON DECK "RINPLEX" INSERTED HERE                               CRINPLE2
      REAL KAY2,KAY2I                                                   CRINPLE4
      COMPLEX PNP,POLAR,LPOLAR                                          CRINPLE5
      LOGICAL SPACE                                                     CRINPLE6
      CHARACTER DISPM*6                                                 CRINPLE7
      COMMON/RINPL/DISPM                                                CRINPLE8
      COMMON /RIN/ MODRIN(8),RAYNAME(2,3),TYPE(3),SPACE                 CRINPLE9
      COMMON/RIN/OMEGMIN,OMEGMAX,KAY2,KAY2I                             CRINPL10
      COMMON/RIN/PNP(10),POLAR,LPOLAR,SGN                               CRINPL11
C     COMMON DECK "RKAM" INSERTED HERE                                  RKAMCOM2
      REAL KR,KTH,KPH                                                   RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)       RKAMCOM5
C     COMMON DECK "WW" INSERTED HERE                                    CWW      2
      PARAMETER (NWARSZ=1000)                                           CWW1     3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1     4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2     2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2     3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2     4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2     5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2     6
     8 (RCVRH,W(20)),                                                   CWW2     7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2     8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2     9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2    10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2    11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2    12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2    13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2    14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2    15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2    16
      REAL MMODEL,MFORM,MID                                             CWW3     2
C                                                                       CWW3     3
C     WIND        100-124                                               CWW3     4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)           CWW3     5
C                                                                       CWW3     6
C     DELTA WIND        125-149                                         CWW3     7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)        CWW3     8
C                                                                       CWW3     9
```

```
C      SOUND SPEED 150-174
       EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)
       EQUIVALENCE (W(153),REFC)
C
C      DELTA SOUND SPEED 175-199
       EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)
C
C      TEMPERATURE    200-224
       EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)
C
C      DELTA TEMPERATURE    225-249
       EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)
C
C      MOLECULAR   250-274
       EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)
C
C      RECEIVER HEIGHT   275-299
       EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)
C
C      TOPOGRAPHY   300-324
       EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)
C
C      DELTA TOPOGRAPHY   325-349
       EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)
C
C      UPPER SURFACE TOPOGRAPHY   350-374
       EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)
C      PLOT ENHANCEMENTS CONTROL PARAMETERS
C
       EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)
C      ABSORPTION           500-524
       EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)
C
C      DELTA ABSORPTION           525-549
       EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)
C
C      PRESSURE          550-574
       EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)
C
C      DELTA PRESSURE          575-599
       EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)
C
       EQUIVALENCE(T,TPULSE),(PHI,PH),(TH,THETA)
       DATA
     2  HEADRS(7)/'  PHASE TIME'/,UNITS(7)/'    SEC'/
     3 ,HEADRS(8)/'  ABSORPTION'/,UNITS(8)/'    DB '/
     4 ,HEADRS(9)/'   DOPPLER  '/,UNITS(9)/'  C/,S '/
     5 ,HEADRS(10)/' PATH LENGTH'/,UNITS(10)/'    KM '/
     6 ,HEADRS(11)/' TERRAIN'/,UNITS(11)/' '/
     7 ,HEADRS(12)/' TERRAIN PGR'/,UNITS(12)/' '/
     8 ,HEADRS(13)/' TERRAIN PGT'/,UNITS(13)/' '/
     9 ,HEADRS(14)/' TERRAIN PGP'/,UNITS(14)/' '/
C
C       ROUND-OFF FUNCTION
       ROUND(X)=SIGN(ABS(X)+0.5,X)
```

```
C
      NWHY=EVENT(2:)
      GO TO 10
C
C     INITIALIZATION ENTRY POINT FOR PRINTR. CALLED FOR EACH NEW
C     W-ARRAY.
      ENTRY IPRINTR
C
C********* NEW W ARRAY -- REINITIALIZE
      NEWWP=.FALSE.
      SPL=SIN (PLON-TLON)
      CPL=SIN (PID2-(PLON-TLON))
      SP=SIN (PLAT)
      CP=SIN (PID2-PLAT)
      SL=SIN (TLAT)
      CL=SIN (PID2-TLAT)
C********* MATRIX TO ROTATE COORDINATES
      G0(1,1)=CPL*SP*CL-CP*SL
      G0(1,2)=SPL*SP
      G0(1,3)=-SL*SP*CPL-CL*CP
      G0(2,1)=-SPL*CL
      G0(2,2)=CPL
      G0(2,3)=SL*SPL
      G0(3,1)=CL*CP*CPL+SP*SL
      G0(3,2)=CP*SPL
      G0(3,3)=-SL*CP*CPL+SP*CL
      DENM=G0(1,1)*G0(2,2)*G0(3,3)+G0(1,2)*G0(3,1)*G0(2,3)
     1   +G0(2,1)*G0(3,2)*G0(1,3)-G0(2,2)*G0(3,1)*G0(1,3)
     2   -G0(1,2)*G0(2,1)*G0(3,3)-G0(1,1)*G0(3,2)*G0(2,3)
C********* THE MATRIX G1 IS THE INVERSE OF THE MATRIX G
      G1(1,1)=(G0(2,2)*G0(3,3)-G0(3,2)*G0(2,3))/DENM
      G1(1,2)=(G0(3,2)*G0(1,3)-G0(1,2)*G0(3,3))/DENM
      G1(1,3)=(G0(1,2)*G0(2,3)-G0(2,2)*G0(1,3))/DENM
      G1(2,1)=(G0(3,1)*G0(2,3)-G0(2,1)*G0(3,3))/DENM
      G1(2,2)=(G0(1,1)*G0(3,3)-G0(3,1)*G0(1,3))/DENM
      G1(2,3)=(G0(2,1)*G0(1,3)-G0(1,1)*G0(2,3))/DENM
      G1(3,1)=(G0(2,1)*G0(3,2)-G0(3,1)*G0(2,2))/DENM
      G1(3,2)=(G0(3,1)*G0(1,2)-G0(1,1)*G0(3,2))/DENM
      G1(3,3)=(G0(1,1)*G0(2,2)-G0(2,1)*G0(1,2))/DENM
      R0=EARTHR+XMTRH
C********* CARTESIAN COORDINATES OF TRANSMITTER
      XR=R0*G0(1,1)
      YR=R0*G0(2,1)
      ZR=R0*G0(3,1)
      CTHR=G0(3,1)
      STHR=SIN (ACOS (CTHR))
      PHIR=ATAN2(YR,XR)
      ALPH=ATAN2(G0(3,2),G0(3,3))
C*********
      NR=0
      NP=0
      NERG=0
      NERR=0
      NERT=0
      NERP=0
```

```
C     INSURE NO GARBLE IN HEADERS
      HEAD(1)=' '
      UNIT(1)=' '
C
      DO 7 NN=7,20
      IF (W(NN+50).EQ.0.) GO TO 7
C********* DEPENDENT VARIABLE NUMBER NN IS BEING INTEGRATED
C********* NR IS THE NUMBER OF DEPENDENT VARIABLES BEING INTEGRATED
      NR=NR+1
C     ENABLE SELECTED RELATIVE ERROR PRINTOUTS FOR TERRAIN(G) OR
C     ITS DERIVATIVES WITH RESPECT TO (R)ANGE, (T)HETA OR (P)HI.
      IF(NN.EQ.11) NERG=NR
      IF(NN.EQ.12) NERR=NR
      IF(NN.EQ.13) NERT=NR
      IF(NN.EQ.14) NERP=NR
      IF (W(NN+50).NE.2.) GO TO 7
C********* DEPENDENT VARIABLE NUMBER NN IS BEING INTEGRATED AND PRINTED.
C********* NP IS THE NUMBER OF DEPENDENT VARIABLES BEING INTEGRATED AND
C********* PRINTED
      NP=NP+1
C********* SAVE THE INDEX OF THE DEPENDENT VARIABLE TO PRINT
      NPR(NP)=NR
      HEAD(NP)=HEADRS(NN)
      HEAD(NP)=HEADRS(NN)
      UNIT(NP)=UNITS(NN)
    7 CONTINUE
      NPM=MIN0(NP,3)
      NP1=NPM+1
      P=0.0
      ABSORB=0.0
      DOPP=0.0
      NEQS=NR+6
      RETURN
C
      ENTRY PRNHD1(CC)
C     PRINT PRINTR HEADER 1
C   IF NO OUTPUT NEEDED EXIT NOW
      IF(PRTSRP.NE.0.0) RETURN
C
C     ADD NUMBER OF LINES NEEDED FOR THIS HEADER
      PC=CC
      IF(CC.EQ.'1') CALL NEWPAG(NPAG,INT(PAGLN),PC)
      LINES=LINES+3
C********* PRINT COLUMN HEADINGS
C
      WRITE(3,1100) PC,(HEAD(NN),NN=1,NPM)
C
1100  FORMAT (   A,T25,'ELEVATION',T54,'AZIMUTH',T71,'ELEVATION'/
     3 ,T20,2('ABOVE',6X),T53,'DEVIATION',T72,'ANGLE'/'  ERROR   EVENT'
     4  ,T20,'SEA LEVEL  TERRAIN     RANGE',4X,2('XMTR   LOCAL',5X),
     5  'PULSE TIME',3A12)
C
      WRITE(3,1150) (UNIT(NN),NN=1,NPM)
 1150 FORMAT(13X,3(8X,'KM'),2X,2(6X,'DEG',5X,'DEG'),T88
     1 ,'SEC',3X,3(3X,A7,2X))
```

```
C
      RETURN
C
      ENTRY PRNHD2(CC)
C     PRINT PRINTR HEADER 1
C   IF NO OUTPUT NEEDED EXIT NOW
      IF(PRTSRP.NE.0.0) RETURN
C     PAGE BY HALF LENGTH
      LINSPP=PAGLN/2
      IF(LINSPP.LT.40) LINSPP=PAGLN
      PC=CC
      IF(CC.EQ.'1') CALL NEWPAG(NPAG,LINSPP,PC)
C     ADD NUMBER OF LINES NEEDED FOR THIS SUBHEADER
      LINES=LINES+1
C
      IF(ELEND.GE.ELBEG+ELSTEP) THEN
         WRITE(3,'(A,''ELEVATION ANGLE OF TRANSMISSION =''
     1    ,F10.4,'' DEG'')') PC,BETA*DEGS
      ELSEIF(AZEND.GE.AZBEG+AZSTEP) THEN
         WRITE(3,'(A,''AZIMUTH ANGLE OF TRANSMISSION =''
     1    ,F10.4,'' DEG'')') PC,AZ1*DEGS
      ENDIF
      RETURN
C
C     IF PRINTING SUPPRESSED AND RAYSETS OFF NOTHING TO DO
10    IF(PRTSRP.NE.0.0 .AND. CARD.EQ.0.0) RETURN
      CALL DISPER
      IF (CARD.EQ.0.0 .OR. IHOP.NE.0) GO TO 12
C********* OUTPUT A TRANSMITTER RAYSET
C        NOTE: THIS IS A SPECIAL CASE, ALL OTHER RAY EVENTS ARE
C           OUTPUT AT CODE 'PUNCH A RAYSET' BELOW.
C
      TLOND=TLON*DEGS
      IF (TLOND.LT.0.) TLOND=TLOND+360.
      TLATD=TLAT*DEGS
      IF (TLATD.LT.0.) TLATD=TLATD+360.
      AZ=AZ1*DEGS
      EL=BETA*DEGS
      NHOP=HOP
      NXMTRH=ROUND(XMTRH*1.E4)
      NTLATD=ROUND(TLATD*1.E3)
       NTLOND=ROUND(TLOND*1.E3)
      NRCVRH=ROUND(RCVRH*1.E4)
       NF=ROUND(OW*1.E4)
      NAZ=ROUND(AZ*1.E5)
       NEL=ROUND(EL*1.E5)
      NPOLAR1=ROUND(REAL(POLAR)*1.E2)
       NPOLAR2=ROUND(AIMAG(POLAR)*1.E2)
      WRITE(9,1201) ID(1),TYPE(NTYP),NXMTRH,NTLATD,NTLOND,NRCVRH,NF,NAZ
     1 ,NEL,NPOLAR1,NPOLAR2,NHOP,'T'
 1201 FORMAT(A3,A1,I9,2I6,2I9,2I10,5X,I5,I4,I2,A1)
C*********
12    V=1.E10
C     OBTAIN THE WORST ERROR OF THOSE ENABLED.
      IF (KAY2.NE.0.) V=(KR**2+KTH**2+KPH**2)/KAY2-1.
```

```
      ERT=1HK
      V=RERR(V,ERT,'G',NERG,G)
      V=RERR(V,ERT,'R',NERR,PGR)
      V=RERR(V,ERT,'T',NERT,PGTH)
      V=RERR(V,ERT,'P',NERP,PGPH)
C
      H=R-EARTHR
      STH=SIN (THETA)
      CTH=SIN (PID2-THETA)
C********* CARTESIAN COORDINATES OF RAY POINT, ORIGIN AT TRANSMITTER
      XP=R*STH*SIN (PID2-PHI)-XR
      YP=R*STH*SIN (PHI)-YR
      ZP=R*CTH-ZR
C********* CARTESIAN COORDINATES OF RAY POINT, ORIGIN AT TRANSMITTER AND
C********* ROTATED
      EPS=XP*G1(1,1)+YP*G1(1,2)+ZP*G1(1,3)
      ETA=XP*G1(2,1)+YP*G1(2,2)+ZP*G1(2,3)
      ZETA=XP*G1(3,1)+YP*G1(3,2)+ZP*G1(3,3)
      RCE2=ETA**2+ZETA**2
      RCE=SQRT (RCE2)
C********* GROUND RANGE
      RANGE=EARTHR*ATAN2(RCE,EARTHR+EPS+XMTRH)
C********* ANGLE OF WAVE NORMAL WITH LOCAL HORIZONTAL
      ELL=ATAN2(KR,SQRT (KTH**2+KPH**2))*DEGS
C********* STRAIGHT LINE DISTANCE FROM TRANSMITTER TO RAY POINT
      SR=SQRT (RCE2+EPS**2)
C********* TERRAIN RELATIVE HEIGHT
      GRH=GET(G)/PGR
C     REPORT GROUP TIME AS FIRST 'OPTION'
      RPRINT(1)=T
      IF (NP1.LT.2) GO TO 16
C     ADD MORE OPTIONS IF REQUESTED
      DO 15 I=2,NP1
      NN=NPR(I-1)
   15 RPRINT(I)=RKVARS(NN)
C
16    IF(V.GE.0.0) THEN
         WRITE(TMP,'(A1,E7.2)') ERT,V
      ELSE
         WRITE(TMP,'(A1,E7.1)') ERT,V
      ENDIF
C     DETERMINE WHERE TO PUT A SPACE
C     AT BEGINNING OR END OF 1ST 2 VALUES
      KT=1
      IF(TMP(1:1).EQ.'K') KT=2
C
      IF (SR.GE.1.E-6) GO TO 20
C********* TOO CLOSE TO TRANSMITTER TO CALCULATE DIRECTION FROM
C********* TRANSMITTER
      IF(PRTSRP.EQ.0.0)
     1 WRITE(3,1500)TMP(KT:9),NWHY,H,GRH,RANGE,ELL,(RPRINT(NN),NN=1,NP1)
 1500 FORMAT (1X,A8,A8,2F10.4,F11.4,26X,F8.3,4F12.4)
C     SET RAYSET VARIABLES TO UNDEFINDED VALUES FOR FLAGS
      AZDEV=999.0
      AZA=999.0
```

```
      GO TO 40
C********* ELEVATION ANGLE OF RAY POINT FROM TRANSMITTER
   20 EL=ATAN2(EPS,RCE)*DEGS
      IF (RCE.GE.1.E-6) GO TO 30
C********* NEARLY DIRECTLY ABOVE OR BELOW TRANSMITTER. CAN NOT CALCULATE
C********* AZIMUTH DIRECTION FROM TRANSMITTER ACCURATELY
      IF(PRTSRP.EQ.0.0)
     1   WRITE(3,2500)TMP(KT:9),NWHY,H,GRH,RANGE,EL,ELL
     2   ,(RPRINT(NN),NN=1,NP1)
 2500 FORMAT (1X,A8,A8,2F10.4,F11.4,17X,F9.3,F8.3,
     1 4F12.4)
      GO TO 40
C********* AZIMUTH ANGLE OF RAY POINT FROM TRANSMITTER
   30 ANGA=ATAN2(ETA,ZETA)
      AZDEV=180.-AMOD(540.-(AZ1-ANGA)*DEGS,360.)
      IF (KTH.NE.0..OR.KPH.NE.0.) GO TO 34
C********* WAVE NORMAL IS VERTICAL, SO AZIMUTH DIRECTION CANNOT BE
C********* CALCULATED
      IF(PRTSRP.EQ.0.0)
     1 WRITE(3,3000)TMP(KT:9),NWHY,H,GRH,RANGE,AZDEV,EL,ELL,(RPRINT(NN),
     1 NN=1,NP1)
 3000 FORMAT (1X,A8,A8,2F10.4,F11.4,F9.3,8X,F9.3,F8.3,
     1 4F12.4)
      GO TO 40
   34 ANA=ANGA-ALPH
      SANA=SIN (ANA)
      SPHI=SANA*STHR/STH
      CPHI=-SIN (PID2-ANA)*SIN (PID2-(PHI-PHIR))+SANA*SIN (PHI-PHIR)
     1 *CTHR
      AZA=180.-AMOD (540.-(ATAN2(SPHI,CPHI)-ATAN2(KPH,KTH))*DEGS,360.)
      IF(PRTSRP.EQ.0.0)
     1  WRITE(3,3500)TMP(KT:9),NWHY,H,GRH,RANGE,AZDEV,AZA,EL,ELL
     2   ,(RPRINT(NN),NN=1,NP1)
 3500 FORMAT (1X,A8,A8,2F10.4,F11.4,2(F9.3,F8.3),
     1 4F12.4)
C*********
   40 LINES=LINES+1
      IF (NP.LE.3) GO TO 45
C********* ADDITIONAL LINE TO PRINT REMAINING DEPENDENT INTEGRATION
C********* VARIABLES
      IF(PRTSRP.EQ.0.0)
     1   WRITE(3,4000) (RPRINT(NN),NN=4,NP)
 4000 FORMAT (99X,3F12.4)
      LINES=LINES+1
C
C     IF NO 'CARDS' WANTED OR AT TRANSMITTER, NO RAYSET OUTPUT
C
45    IF (CARD.EQ.0.0 .OR. IHOP.LT.1) RETURN
C
C********* PUNCH A RAYSET
      IF (AZDEV.LT.-90.) AZDEV=AZDEV+360.
      IF (AZA.LT.-90.) AZA=AZA+360.
      NR=0
      IF (W(57).EQ.0.) GO TO 47
C********* PHASE PATH
```

```
      NR=NR+1                                                           PRINT308
      P=RKVARS(NR)                                                      PRINT309
   47 IF (W(58).EQ.0.) GO TO 48                                         PRINT310
C********* ABSORPTION                                                   PRINT311
      NR=NR+1                                                           PRINT312
      ABSORB=RKVARS(NR)                                                 PRINT313
C********* DOPPLER SHIFT                                                PRINT314
   48 IF (W(59).NE.0.) DOPP=RKVARS(NR+1)                                PRINT315
      NHPUNCH=ROUND(HPUNCH*1.E4)                                        PRINT316
       NRANGE=ROUND(RANGE*1.E4)                                         PRINT317
      NAZDEV=ROUND(AZDEV*1.E3)                                          PRINT318
       NAZA=ROUND(AZA*1.E3)                                             PRINT319
       NELL=ROUND(ELL*1.E3)                                             PRINT320
      IF(NWHY.EQ.'GRND REF') NELL =                                     PRINT321
     1 ROUND((PID2 - ACOS(KNORM(G)/SQRT(KR*KR+KTH*KTH+KPH*KPH)))        PRINT322
     2   *DEGS*1.E3)                                                    PRINT323
      NABSORB=AMIN1(999999.0,ROUND(ABSORB*1.E3))                        PRINT324
       NDOPP=ROUND(DOPP*1.E3)                                           PRINT325
      NPOLAR1=ROUND(REAL(POLAR)*1.E2)                                   PRINT326
       NPOLAR2=ROUND(AIMAG(POLAR)*1.E2)                                 PRINT327
      JP=ROUND(P*1.E5)                                                  PRINT328
      JT=ROUND(T*1.E5)                                                  PRINT329
      WRITE(9,4501) NHPUNCH,NRANGE,NAZDEV,NAZA,NELL,JT,JP,NABSORB       PRINT330
     1 ,NDOPP,NPOLAR1,NPOLAR2,IHOP,EVENT(1:1)                           PRINT331
 4501 FORMAT(2I9,3I6,2I10,2I6,I5,I4,I2,A1)                              PRINT332
      RETURN                                                            PRINT333
      END                                                               PRINT334
```

```
      SUBROUTINE ATMOSHD                                                ATMOSHD2
C     PRINTS PAGE HEADINGS                                              ATMOSHD3
C     IF W(72) IS NEGATIVE, ONLY ONE HEADER OUTPUT IS INCLUDED IN RAYSE ATMOSHD4
      CHARACTER PCC*1,PC*1,BLANKS*100,DIVIDR*132,BANNER(8)*80           ATMOSHD5
      CHARACTER NUMSTG*80,STMP*80                                       ATMOSHD6
      LOGICAL NOPUNCH                                                   ATMOSHD7
      INTEGER STRIM                                                     ATMOSHD8
C                                                                       ATMOSHD9
C     TWO ENTRY POINTS ARE PROVIDED. ONE FOR THE FIRST PAGE HEADER      ATMOSH10
C     OF THE COMPUTATIONAL PRINTOUT AND FOR THE RAYSET FILE. THE        ATMOSH11
C     SECOND ENTRY IS FOR ALL SUBSEQUENT PAGES OF THE COMPUTATIONAL     ATMOSH12
C     PRINTOUT.                                                         ATMOSH13
C                                                                       ATMOSH14
C     COMMON DECK "AA" INSERTED HERE                                    CAA     2
      REAL MODA                                                         CAA     4
      REAL MU,MUPT,MUPR,MUPTH,MUPPH                                     CAA     5
      REAL KAP,KAPPT,KAPPR,KAPPTH,KAPPPH                                CAA     6
      COMMON/AA/MODA(4),MU,MUPT,MUPR,MUPTH,MUPPH                        CAA     7
      COMMON/AA/KAP,KAPPT,KAPPR,KAPPTH,KAPPPH                           CAA     8
C     COMMON DECK "PP" INSERTED HERE                                    CPP     2
      REAL MODP                                                         CPP     4
      COMMON/PP/MODP(4),P,PPT,PPR,PPTH,PPPH                             CPP     5
C                                                                       ATMOSH17
```

```
C     COMMON DECK "SS" INSERTED HERE                                    CSS      2
      REAL MODSURF                                                      CSS      4
      COMMON/SS/ MODSURF(4)                                             CSS      5
      COMMON/SS/U,PUR,PURR,PURTH,PURPH                                  CSS      6
      COMMON/SS/PUTH,PUPH,PUTHTH,PUPHPH,PUTHPH,USELECT,UTIME            CSS      7
C     COMMON DECK "CONST" INSERTED HERE                                 CCONST   2
      COMMON/PCONST/CREF,RGAS,GAMMA                                     CCONST   4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                         CCONST   5
C     COMMON DECK "WWR" INSERTED HERE                                   CWWR     2
      PARAMETER (NWARSZ=1000)                                           CWW1     3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1     4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2     2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2     3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2     4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2     5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2     6
     8 (RCVRH,W(20)),                                                   CWW2     7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2     8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2     9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2    10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2    11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2    12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2    13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2    14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2    15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2    16
C     COMMON DECK "GG" INSERTED HERE                                    CGG      2
      REAL MODG                                                         CGG      4
      COMMON/GG/MODG(4)                                                 CGG      5
      COMMON/GG/G,PGR,PGRR,PGRTH,PGRPH                                  CGG      6
      COMMON/GG/PGTH,PGPH,PGTHTH,PGPHPH,PGTHPH,GSELECT,GTIME            CGG      7
C     COMMON DECK "RR" INSERTED HERE                                    CRR      2
      REAL MODREC                                                       CRR      4
      COMMON/RR/ MODREC(4)                                              CRR      5
      COMMON/RR/F,PFR,PFRR,PFRTH,PFRPH                                  CRR      6
      COMMON/RR/PFTH,PFPH,PFTHTH,PFPHPH,PFTHPH,FSELECT,FTIME            CRR      7
C     COMMON DECK "B9" INSERTED HERE                                    CB8      2
      INTEGER GMX,GNTBL,GITBL,GFRMTBL,IDSG(10)                          CB8      4
      COMMON/B9/GMX,GNTBL(10),GITBL(10),GFRMTBL(10),GGP(113)            CB8      5
      EQUIVALENCE (GGP,IDSG),(ANG,GGP(11))                              CB8      6
C     COMMON DECK "B2" INSERTED HERE                                    CB2      2
      INTEGER DUMX,DUNTBL,DUITBL,DUFRMTB,IDSDU(10)                      CB2      4
      COMMON/B2/DUMX,DUNTBL(10),DUITBL(10),DUFRMTB(10),DUGP(10)         CB2      5
      EQUIVALENCE (DUGP,IDSDU)                                          CB2      6
C     COMMON DECK "B4" INSERTED HERE                                    CB4      2
      INTEGER DCMX,DCNTBL,DCITBL,DCFRMTB,IDSDC(10)                      CB4      4
      COMMON/B4/DCMX,DCNTBL(10),DCITBL(10),DCFRMTB(10),DCGP(10)         CB4      5
      EQUIVALENCE (DCGP,IDSDC)                                          CB4      6
C     COMMON DECK "B6" INSERTED HERE                                    CB6      2
      INTEGER DTMX,DTNTBL,DTITBL,DTFRMTB,IDSDT(10)                      CB6      4
      COMMON/B6/DTMX,DTNTBL(10),DTITBL(10),DTFRMTB(10),DTGP(10)         CB6      5
      EQUIVALENCE (DTGP,IDSDT)                                          CB6      6
C     COMMON DECK "B1" INSERTED HERE                                    CB1      2
      INTEGER UMX,UNTBL,UITBL,UFRMTBL,IDSU(10)                          CB1      4
      COMMON/B1/UMX,UNTBL(10),UITBL(10),UFRMTBL(10),UGP(10)             CB1      5
```

```
      EQUIVALENCE (UGP,IDSU)                                            CB1      6
C     COMMON DECK "B3" INSERTED HERE                                    CB3      2
      INTEGER CMX,CNTBL,CITBL,CFRMTBL,IDSC(10)                          CB3      4
      COMMON/B3/CMX,CNTBL(10),CITBL(10),CFRMTBL(10),CGP(512)            CB3      5
      EQUIVALENCE (CGP,IDSC),(ANC,CGP(11))                              CB3      6
C     COMMON DECK "B5" INSERTED HERE                                    CB5      2
      INTEGER TMX,TNTBL,TITBL,TFRMTBL,IDST(10)                          CB5      4
      COMMON/B5/TMX,TNTBL(10),TITBL(10),TFRMTBL(10),TGP(262)            CB5      5
      EQUIVALENCE (TGP,IDST),(ANT,TGP(11))                              CB5      6
C     COMMON DECK "B7" INSERTED HERE                                    CB7      2
      INTEGER MMX,MNTBL,MITBL,MFRMTBL,IDSM(10)                          CB7      4
      REAL MGP                                                          CB7      5
      COMMON/B7/MMX,MNTBL(10),MITBL(10),MFRMTBL(10),MGP(10)             CB7      6
      EQUIVALENCE (MGP,IDSM)                                            CB7      7
C     COMMON DECK "HDR" INSERTED HERE                                   CHDR     2
      CHARACTER*10 INITID*80,DAT,TOD                                    CHDR     4
      COMMON/HDR/SEC                                                    CHDR     5
      COMMON/HDRC/INITID,DAT,TOD                                        CHDR     6
C     COMMON DECK "RINPLEX" INSERTED HERE                               CRINPLE2
      REAL KAY2,KAY2I                                                   CRINPLE4
      COMPLEX PNP,POLAR,LPOLAR                                          CRINPLE5
      LOGICAL SPACE                                                     CRINPLE6
      CHARACTER DISPM*6                                                 CRINPLE7
      COMMON/RINPL/DISPM                                                CRINPLE8
      COMMON /RIN/ MODRIN(8),RAYNAME(2,3),TYPE(3),SPACE                 CRINPLE9
      COMMON/RIN/OMEGMIN,OMEGMAX,KAY2,KAY2I                             CRINPL10
      COMMON/RIN/PNP(10),POLAR,LPOLAR,SGN                               CRINPL11
C     COMMON DECK "CC" INSERTED HERE                                    CCC      2
      REAL MODC                                                         CCC      4
      COMMON/CC/MODC(4),CS,PCST,PCSR,PCSTH,PCSPH                        CCC      5
C     COMMON DECK "MM" INSERTED HERE                                    CMM      2
      REAL M,MODM                                                       CMM      4
      COMMON/MM/MODM(4),M,PMT,PMR,PMTH,PMPH                             CMM      5
C     COMMON DECK "FLAG" INSERTED HERE                                  CFLAG    2
      LOGICAL NEWWR,NEWWP,NEWTRC,PENET                                  CFLAG    4
      COMMON /FLG/ NTYP,NEWWR,NEWWP,NEWTRC,PENET,LINES,IHOP,HPUNCH      CFLAG    5
      COMMON/FLGP/NSET                                                  CFLAG    6
C     COMMON DECK "TT" INSERTED HERE                                    CTT      2
      REAL MODT                                                         CTT      4
      COMMON/TT/MODT(4), T,PTT,PTR,PTTH,PTPH                            CTT      5
C     COMMON DECK "UU" INSERTED HERE                                    CUU      2
      REAL MODU                                                         CUU      4
      COMMON/UU/MODU(4)                                                 CUU      5
     1  ,V   ,PVT   ,PVR   ,PVTH   ,PVPH                                CUU      6
     2  ,VR  ,PVRT  ,PVRR  ,PVRTH  ,PVRPH                               CUU      7
     3  ,VTH,PVTHT,PVTHR,PVTHTH,PVTHPH                                  CUU      8
     4  ,VPH,PVPHT,PVPHR,PVPHTH,PVPHPH                                  CUU      9
C                                                                       ATMOSH38
C     COMMON DECK "B10" INSERTED HERE                                   CB9      2
      INTEGER DGMX,DGNTBL,DGITBL,DGFRMTB,IDSDG(10)                      CB9      4
      COMMON/B10/DGMX,DGNTBL(10),DGITBL(10),DGFRMTB(10),DGGP(10)        CB9      5
      EQUIVALENCE (DGGP,IDSDG)                                          CB9      6
C     COMMON DECK "B8" INSERTED HERE                                    CB10     2
      INTEGER RMX,RNTBL,RITBL,RFRMTBL,IDSR(10)                          CB10     4
      COMMON/B8/RMX,RNTBL(10),RITBL(10),RFRMTBL(10),RGP(10)             CB10     5
```

```
      EQUIVALENCE (RGP,IDSR)                                            CB10    6
C     COMMON DECK "CB17" INSERTED HERE                                  CB17    2
      INTEGER VMX,VNTBL,VITBL,VFRMTBL,IDSV(10)                          CB17    4
      COMMON/B17/VMX,VNTBL(10),VITBL(10),VFRMTBL(10),VGP(53)            CB17    5
      EQUIVALENCE (VGP,IDSV),(ANV,VGP(11))                              CB17    6
C     COMMON DECK "CB18" INSERTED HERE                                  CB18    2
      INTEGER DVMX,DVNTBL,DVITBL,DVFRMTB,IDSDV(10)                      CB18    4
      COMMON/B18/DVMX,DVNTBL(10),DVITBL(10),DVFRMTB(10),DVGP(11)        CB18    5
      EQUIVALENCE (DVGP,IDSDV),(ANDV,DVGP(11))                          CB18    6
C     COMMON DECK "CB19" INSERTED HERE                                  CB19    2
      INTEGER PRMX,PRNTBL,PRITBL,PRFRMTB,IDSPR(10)                      CB19    4
      COMMON/B19/PRMX,PRNTBL(10),PRITBL(10),PRFRMTB(10),PRGP(11)        CB19    5
      EQUIVALENCE (PRGP,IDSPR),(ANP,PRGP(11))                           CB19    6
C     COMMON DECK "CB20" INSERTED HERE                                  CB20    2
      INTEGER DPMX,DPNTBL,DPITBL,DPFRMTB,IDSDP(10)                      CB20    4
      COMMON/B20/DPMX,DPNTBL(10),DPITBL(10),DPFRMTB(10),DPGP(11)        CB20    5
      EQUIVALENCE (DPGP,IDSDP),(ANDP,DPGP(11))                          CB20    6
C                                                                       ATMOSH45
      PARAMETER (NBNRLS=8,NBLNS=8)                                      ATMOSH46
C                                                                       ATMOSH47
      DATA BLANKS/' '/                                                  ATMOSH48
      DATA BANNER/                                                      ATMOSH49
     1  '***** H A R P A *****'                                         ATMOSH50
     2 ,'HAMILTONIAN ACOUSTIC RAY-TRACING PROGRAM FOR THE ATMOSPHERE'   ATMOSH51
     3 ,' '                                                             ATMOSH52
     4 ,'BY'                                                            ATMOSH53
     5 ,'R. M. JONES,  J. P. RILEY  AND  T. M. GEORGES'                 ATMOSH54
     6 ,'WAVE PROPAGATION LABORATORY'                                   ATMOSH55
     7 ,'NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION'               ATMOSH56
     8 ,'BOULDER, COLORADO 80303'/                                      ATMOSH57
      DATA NOPUNCH/.FALSE./                                             ATMOSH58
      DATA IBLK/1H /                                                    ATMOSH59
C                                                                       ATMOSH60
      ENTRY HEADER1                                                     ATMOSH61
C                                                                       ATMOSH62
C     COMPUTE EFFECTIVE LINES COUNT BASED ON FIXED PAGE SIZE            ATMOSH63
      CALL NEWPAG(NPAG,INT(PAGLN),PC)                                   ATMOSH64
      CALL SFILL(DIVIDR,LEN(DIVIDR),'-')                                ATMOSH65
      DIVIDR(1:1)=' '                                                   ATMOSH66
      NTYP=2                                                            ATMOSH67
      IF(RAY.NE.0.0) NTYP=2.0+SIGN(1.0,RAY)                             ATMOSH68
C                                                                       ATMOSH69
1600  FORMAT(2(2(A8,F7.1,1X),2X))                                       ATMOSH70
      CALL PUTKST(3,'1'                                                 ATMOSH71
     1  //DAT//TOD//BLANKS(:100)//'PAGE'//NUMSTG(NPAG,1,'(I5)'))        ATMOSH72
      CALL PUTKST(3,DIVIDR)                                             ATMOSH73
      CALL PUTKBK(3,1)                                                  ATMOSH74
      DO 15 I=1,NBNRLS                                                  ATMOSH75
15    CALL PUTKCT(3,BANNER(I))                                          ATMOSH76
      CALL PUTKBK(3,1)                                                  ATMOSH77
      CALL PUTKST(3,DIVIDR)                                             ATMOSH78
      CALL PUTKBK(3,NBLNS)                                              ATMOSH79
C                                                                       ATMOSH80
      CALL PUTKST(3,BLANKS(:57)//'RUN SET NUMBER'                       ATMOSH81
     1  //NUMSTG(NSET,1,'(I5)'))                                        ATMOSH82
```

```
      CALL PUTKBK(3,1)                                                  ATMOSH83
      WRITE(STMP,'(10A8)') ID                                           ATMOSH84
      CALL PUTKST(3,BLANKS(:52)//'ATMOSPHERIC MODEL ID -- '//STMP(:3))  ATMOSH85
      CALL PUTKBK(3,1)                                                  ATMOSH86
      CALL PUTKST(3,                                                    ATMOSH87
     1 BLANKS(:25)//'ATMOSPHERIC MODEL DESCRIPTION -- '//STMP(7:))      ATMOSH88
      CALL PUTKBK(3,1)                                                  ATMOSH89
C                                                                       ATMOSH90
      WRITE(STMP,'(2A8)') (RAYNAME(I,NTYP),I=1,2)                       ATMOSH91
      CALL PUTKST(3,STMP)                                               ATMOSH92
      CALL PUTKBK(3,1)                                                  ATMOSH93
      CALL PUTKST(3,DIVIDR)                                             ATMOSH94
      CALL PUTKBK(3,1)                                                  ATMOSH95
      CALL PUTKST(3,                                                    ATMOSH96
     1  BLANKS(:8)//'MODEL                SUBROUTINE    DATA SET'       ATMOSH97
     1  //'         DESCRIPTION')                                       ATMOSH98
      CALL PUTKST(3,                                                    ATMOSH99
     1  BLANKS(:8)//'TYPE                    NAME          ID')         ATMOS100
      CALL PUTKBK(3,1)                                                  ATMOS101
      CALL PUTKST(3,DIVIDR)                                             ATMOS102
      CALL PUTKBK(3,1)                                                  ATMOS103
      CALL PUTKST(3,' DISPERSION RELATION             '//DISPM          ATMOS104
     1  //BLANKS(:16)//NUMSTG(MODRIN,8,'(8A8)'))                        ATMOS105
      CALL PUTDES(3,'BACKGROUND WIND VELOCITY',MODU,IDSU)               ATMOS106
      CALL PUTDES(3,'WIND VELOCITY PERTURBATION',MODU(3),IDSDU)         ATMOS107
      CALL PUTDES(3,'BACKGROUND SOUND SPEED',MODC,IDSC)                 ATMOS108
      CALL PUTDES(3,'SOUND SPEED PERTURBATION',MODC(3),IDSDC)           ATMOS109
      CALL PUTDES(3,'BACKGROUND TEMPERATURE',MODT,IDST)                 ATMOS110
      CALL PUTDES(3,'TEMPERATURE PERTURBATION',MODT(3),IDSDT)           ATMOS111
      CALL PUTDES(3,'MOLECULAR WEIGHT',MODM,IDSM)                       ATMOS112
      CALL PUTDES(3,'BACKGROUND TERRAIN',MODG,IDSG)                     ATMOS113
      CALL PUTDES(3,'TERRAIN PERTURBATION',MODG(3),IDSDG)               ATMOS114
      CALL PUTKST(3,' BACKGROUND VISCOSITY/')                           ATMOS115
      CALL PUTDES(3,'  THERMAL CONDUCTIVITY',MODA,IDSV)                 ATMOS116
      CALL PUTKST(3,' VISCOSITY/THERMAL')                               ATMOS117
      CALL PUTDES(3,'  CONDUCTIVITY PERTURBATION',MODA(3),IDSDV)        ATMOS118
      CALL PUTDES(3,'BACKGROUND PRESSURE',MODP,IDSPR)                   ATMOS119
      CALL PUTDES(3,'PRESSURE PERTURBATION',MODP(3),IDSDP)              ATMOS120
      CALL PUTDES(3,'RECEIVER SURFACE',MODREC,IDSR)                     ATMOS121
      CALL PUTKST(3,DIVIDR)                                             ATMOS122
C                                                                       ATMOS123
      NOPUNCH=NOPUNCH.AND.RAYSET.LT.0.0                                 ATMOS124
      IF(NOPUNCH) RETURN                                                ATMOS125
      NOPUNCH=RAYSET.LT.0.0                                             ATMOS126
C                                                                       ATMOS127
      WRITE(9,1200) ID,DAT,TOD                                          ATMOS128
      WRITE(9,1600) MODU,MODC,MODT,MODM,MODP,MODA                       ATMOS129
     1    ,MODG,MODREC                                                  ATMOS130
      IF(IDSU(1) .NE. IBLK) WRITE(9,1200) IDSU                          ATMOS131
      IF(IDSDU(1) .NE. IBLK) WRITE(9,1200) IDSDU                        ATMOS132
      IF(IDSC(1) .NE. IBLK) WRITE(9,1200) IDSC                          ATMOS133
      IF(IDSDC(1) .NE. IBLK) WRITE(9,1200) IDSDC                        ATMOS134
      IF(IDST(1) .NE. IBLK) WRITE(9,1200) IDST                          ATMOS135
      IF(IDSDT(1) .NE. IBLK) WRITE(9,1200) IDSDT                        ATMOS136
      IF(IDSM(1) .NE. IBLK) WRITE(9,1200) IDSM                          ATMOS137
```

```
      IF(IDSG(1) .NE. IBLK) WRITE(9,1200) IDSG
C
 1000 FORMAT (A1,10A8,24X,2A,' PAGE',I4)
1200  FORMAT(10A8,2(A8,2X))
C
      RETURN
C
      ENTRY HEADER2
C
C     COMPUTE EFFECTIVE LINES COUNT BASED ON FIXED PAGE SIZE
      CALL NEWPAG(NPAG,INT(PAGLN),PC)
      LINES=LINES+5
C
      WRITE(3,1000) PC,ID,DAT,TOD,NPAG
      WRITE(3,2400) AZ1*DEGS,TLAT*DEGS,OW/PIT2,BETA*DEGS
     1         ,TLON*DEGS,MAXERR
 2400 FORMAT (
     1  /' AZIMUTH ANGLE OF TRANSMISSION   =',F12.6,' DEG'
     2  ,'      TRANSMITTER  LATITUDE  =',F12.6,' DEG'
     3  ,'    FREQUENCY =',F12.6,' HZ'
     4  /' ELEVATION ANGLE OF TRANSMISSION =',F12.6,' DEG'
     5  ,'      TRANSMITTER  LONGITUDE =',F12.6,' DEG'
     6  ,'    SINGLE STEP ERROR =',1PG13.6/)
C
      RETURN
C
      ENTRY PUTDVR(NUNIT)
      CALL PUTKST(NUNIT,DIVIDR)
      RETURN
C
      ENTRY PUTHDR(NUNIT,PCC,NP)
      CALL PUTKST(NUNIT,
     1  PCC//DAT//TOD//BLANKS(:100)//'PAGE'//NUMSTG(NP,1,'(I5)'))
      RETURN
      END
```

```
      SUBROUTINE PUTDES(NUNIT,DES,MOD,ID)
      CHARACTER DES*(*),TITLE*30,SMODL*20,NUMSTG*80
      REAL MOD(2),ID(10)
      TITLE=DES
      WRITE(SMODL,'(A8,F10.2)') MOD
      CALL PUTKST(NUNIT, ' '//TITLE//SMODL//NUMSTG(ID,10,'(10A8)'))
      END
```

```
      CHARACTER*80 FUNCTION NUMSTG(V,N,FRM)
      INTEGER V(N)
      CHARACTER FRM*(*)
```

```
      NUMSTG=' '
      WRITE(NUMSTG,FRM) V
      RETURN
      END
```

```
      SUBROUTINE SFILL(STG,LN,C)
      CHARACTER STG*(*),C*1
C
      DO 10 I=1,LN
10    STG(I:I)=C
      RETURN
      END
```

```
      INTEGER FUNCTION STRIM(C)
      CHARACTER*(*) C
C
      L=LEN(C)
      DO 100 I=L,1,-1
100   IF(C(I:I) .NE. ' ') GO TO 200
      I=0
200   STRIM=I
      END
```

```
      FUNCTION RERR(V,ERT,ELAB,NKV,PREF)
C     RETURNS RELATIVE ERROR OF VARIABLE RKVAR(NKV) IF PREF<>0 AND
C     THE ERROR IS GREATER THAN PREVIOUS ERROR 'V'.
C     COMMON DECK "RKAM" INSERTED HERE
      REAL KR,KTH,KPH
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)
C
      RERR=V
      IF(NKV.LE.0) RETURN
      IF(PREF.EQ.0.) RETURN
      V1=RKVARS(NKV)/PREF-1.
      IF(ABS(V1).LT.ABS(V)) RETURN
C
      ERT=ELAB
      RERR=V1
      END
```

```
      SUBROUTINE RERROR(ROUTIN,STR,VAL)                                  RERROR 2
C     REPORTS ERROR CONDITIONS AND STOPS PROGRAM.                        RERROR 3
C                                                                        RERROR 4
      PRINT 10,ROUTIN,STR,VAL                                            RERROR 5
10    FORMAT(39H ERROR CONDITION IN RAYTRACE ROUTINE <                   RERROR 6
     1  ,A8,10H> DUE TO " ,A10,9H", VALUE=  ,F8.2)                       RERROR 7
      STOP                                                               RERROR 8
      END                                                                RERROR 9
```

```
      SUBROUTINE STOPIT(A)                                               STOPIT 2
C     PRINTS CONDITION AND STOPS PROGRAM                                 STOPIT 3
C     AFTER CALLING THE SYSTEM POST MORTEM DUMP.                         STOPIT 4
C                                                                        STOPIT 5
      CHARACTER A*(*)                                                    STOPIT 6
C                                                                        STOPIT 7
      PRINT 100, A                                                       STOPIT 8
      CALL MORTEM                                                        STOPIT 9
100   FORMAT('*** STOPIT WITH CONDITION <',A,'>')                        STOPIT10
      STOP                                                               STOPIT11
      END                                                                STOPIT12
```

```
      SUBROUTINE PUTKST(NUNIT,STRG)                                      PUTKST 2
C     WRITE LINE OF OUTPUT TO PRINTER UNIT ADDING TO LINE COUNT          PUTKST 3
C                                                                        PUTKST 4
      CHARACTER STRG*(*)                                                 PUTKST 5
      CHARACTER BLANKS*100,STMP*80                                       PUTKST 6
      INTEGER STRIM                                                      PUTKST 7
      CHARACTER PC*1                                                     PUTKST 8
C                                                                        PUTKST 9
C     COMMON DECK "FLAG" INSERTED HERE                                   CFLAG  2
      LOGICAL NEWWR,NEWWP,NEWTRC,PENET                                   CFLAG  4
      COMMON /FLG/ NTYP,NEWWR,NEWWP,NEWTRC,PENET,LINES,IHOP,HPUNCH       CFLAG  5
      COMMON/FLGP/NSET                                                   CFLAG  6
C                                                                        PUTKST11
      DATA BLANKS/' '/                                                   PUTKST12
C                                                                        PUTKST13
C     PUT OUT A STRING WITH LINE COUNT INCREMENT                         PUTKST14
      LINES=LINES+1                                                      PUTKST15
      LN=MIN0(LEN(STRG),132)                                             PUTKST16
      WRITE(NUNIT,'(A)') STRG(:LN)                                       PUTKST17
      RETURN                                                             PUTKST18
C                                                                        PUTKST19
      ENTRY PUTKCT(NUNIT,STRG)                                           PUTKST20
C     PUT OUT A CENTERED LINE AND COUNT                                  PUTKST21
      NTRM=MAX0(1,STRIM(STRG))                                           PUTKST22
      NBLKS=66-(NTRM+1)/2                                                PUTKST23
```

```
      LINES=LINES+1
      LN=MIN0(NTRM,132-NBLKS)
      STMP=STRG
      WRITE(NUNIT,'(A)') BLANKS(:NBLKS)//STMP(:LN)
      RETURN
      ENTRY PUTKBK(NUNIT,NBKS)
C     PUT A BLANK LINE TO PRINTER
C
      LINES=LINES+1
      DO 100 I=1,NBKS
100      WRITE(NUNIT,'(1X)')
      RETURN
C
      ENTRY NEWPAG(NP,LINSPP,PC)
C     THE LOGIC HERE ASSUMES THAT A FORM FEED IS ALWAYS NEEDED
      PC='1'
C     COMPUTE EFFECTIVE LINES COUNT AFTER NEXT FORM FEED
      NPAG=(LINES+LINSPP-1)/LINSPP
      LINES=NPAG*LINSPP
      NP=NPAG+1
      RETURN
      END
```

```
      SUBROUTINE OPNREP(IUN,FNAME)
C     CYBER VERSION OF OPEN FOR REPLACE
C     SINCE THIS OPERATION SEEMS TO BE COMPILER DEPENDENT
C     WE MAKE A SUBROUTINE OUT OF IT.
C     THIS OPEN OPERATION ALLOWS FOR AN EXISTING VERSION OF
C     A FILE. IF THE FILE EXISTS IT IS OVERWRITTEN.
      CHARACTER*(*) FNAME
      OPEN(IUN,FILE=FNAME)
      REWIND IUN
      RETURN
      ENTRY OPNURP(IUN,FNAME)
      OPEN(IUN,FILE=FNAME,FORM='UNFORMATTED')
      REWIND IUN
      END
```

```
      SUBROUTINE OVERRD(VAR,TST,DEFT,NFLG,NFVLEQ,NFVLNE)
C
C     OVERRIDE SUPPORT ROUTINE
C     TEST 'VAR' AGAINST 'TST', IF EQUAL SET TO DEFAULT 'DEFT'
C     AND ALSO SET INTEGER FLAG 'NFLG' TO VALUE 'NFVLEQ' ELSE 'NFVLNE'
C
      IF(VAR.EQ.TST) THEN
         VAR=DEFT
         NFLG=NFVLEQ
```

```
      ELSE                                                              OVERRD11
         NFLG=NFVLNE                                                    OVERRD12
      ENDIF                                                             OVERRD13
      RETURN                                                            OVERRD14
      END                                                               OVERRD15
```

```
      SUBROUTINE SFILTR(C,S,KSET)                                       SFILTR 2
      CHARACTER*(*) C,S,KSET                                            SFILTR 3
C                                                                       SFILTR 4
      LN=LEN(C)                                                         SFILTR 5
      J=0                                                               SFILTR 6
      DO 10 I=1,LN                                                      SFILTR 7
        IF(INDEX(KSET,C(I:I)).EQ.0) THEN                                SFILTR 8
           J=J+1                                                        SFILTR 9
           S(J:J)=C(I:I)                                                SFILTR10
        ENDIF                                                           SFILTR11
10    CONTINUE                                                          SFILTR12
      RETURN                                                            SFILTR13
      END                                                               SFILTR14
```

```
      FUNCTION ALCOSH(X)                                                ALCOSH 2
C     COMPUTE LOG(COSH(X)) AND USE LARGE ARGUMENT APPROXIMATION         ALCOSH 3
C     WHEN POSSIBLE.                                                    ALCOSH 4
      DATA ALOG2/.6931471806/                                           ALCOSH 5
C                                                                       ALCOSH 6
      IF(ABS(X).GT.50.0) GO TO 10                                       ALCOSH 7
      EX=EXP(X)                                                         ALCOSH 8
      ALCOSH=ALOG((EX+1.0/EX)*.5)                                       ALCOSH 9
      RETURN                                                            ALCOSH10
10    ALCOSH=ABS(X)-ALOG2                                               ALCOSH11
      END                                                               ALCOSH12
```

```
      SUBROUTINE GAUSEL (C,NRD,NRR,NCC,NSF)                             GAUSEL 2
C     MATRIX INVERSION BY METHOD OF GAUSSIAN PIVOT                      GAUSEL 3
C     SOLVES NCC-NRR SETS OF NRR EQUATIONS                              GAUSEL 4
C********* SAME AS SUBROUTINE GAUSSEL WRITTEN BY L. DAVID LEWIS ********GAUSEL 5
      DIMENSION C(NRD,NCC),L(128,2)                                     GAUSEL 6
C          BITS = 2.**-18                                               GAUSEL 7
      DATA BITS/3.8146972656E-6/                                        GAUSEL 8
      NR=NRR                                                            GAUSEL 9
      NC=NCC                                                            GAUSEL10
      IF(NC.LT.NR.OR.NR.GT.128.OR.NR.LE.0) CALL EXIT                    GAUSEL11
C                                                                       GAUSEL12
```

```
C           INITIALIZE.
      NSF=0
      NRM=NR-1
      NRP=NR+1
      D=1.
      LSD=1
      DO 1 KR=1,NR
      L(KR,1)=KR
1     L(KR,2)=0
      IF(NR.EQ.1) GO TO 42
C
C           ELIMINATION PHASE.
      DO 41 KP=1,NRM
      KPP=KP+1
      PM=0.
      MPN=0
C
C           SEARCH COLUMN KP FROM DIAGONAL DOWN FOR MAX PIVOT.
      DO 2 KR=KP,NR
      LKR=L(KR,1)
      PT=ABS(C(LKR,KP))
      IF(PT.LE.PM) GO TO 2
      PM=PT
      MPN=KR
      LMP=LKR
2     CONTINUE
C
C           IF MAX PIVOT IS ZERO, MATRIX IS SINGULAR.
      IF(MPN.EQ.0) GO TO 9
      NSF=NSF+1
      IF(MPN.EQ.KP) GO TO 3
C
C           NEW ROW NUMBER KP HAS MAX PIVOT.
      LSD=-LSD
      L(KP,2)=L(KP,1)
      L(MPN,1)=L(KP,1)
      L(KP,1)=LMP
C
C           ROW OPERATIONS TO ZERO COLUMN KP BELOW DIAGONAL.
3     MKP=L(KP,1)
      P=C(MKP,KP)
      D=D*P
      DO 41 KR=KPP,NR
      MKR=L(KR,1)
      Q=C(MKR,KP)/P
      IF(Q.EQ.0.) GO TO 41
C
C           SUBTRACT Q * PIVOT ROW FROM ROW KR.
      DO 4 LC=KPP,NC
      R=Q*C(MKP,LC)
      C(MKR,LC)=C(MKR,LC)-R
4     IF(ABS(C(MKR,LC)).LT.ABS(R)*BITS) C(MKR,LC)=0.
41    CONTINUE
C
C           LOWER RIGHT HAND CORNER.
```

```
42     LNR=L(NR,1)                                                     GAUSEL68
       P=C(LNR,NR)                                                     GAUSEL69
       IF(P.EQ.0.) GO TO 9                                             GAUSEL70
       NSF=NSF+1                                                       GAUSEL71
       D=D*P*LSD                                                       GAUSEL72
       IF(NR.EQ.NC) GO TO 8                                            GAUSEL73
C                                                                      GAUSEL74
C          BACK SOLUTION PHASE.                                        GAUSEL75
       DO 61 MC=NRP,NC                                                 GAUSEL76
       C(LNR,MC)=C(LNR,MC)/P                                           GAUSEL77
       IF(NR.EQ.1) GO TO 61                                            GAUSEL78
       DO 6 LL=1,NRM                                                   GAUSEL79
       KR=NR-LL                                                        GAUSEL80
       MR=L(KR,1)                                                      GAUSEL81
       KRP=KR+1                                                        GAUSEL82
       DO 5 MS=KRP,NR                                                  GAUSEL83
       LMS=L(MS,1)                                                     GAUSEL84
       R=C(MR,MS)*C(LMS,MC)                                            GAUSEL85
       C(MR,MC)=C(MR,MC)-R                                             GAUSEL86
5      IF(ABS(C(MR,MC)).LT.ABS(R)*BITS) C(MR,MC)=0.                    GAUSEL87
6      C(MR,MC)=C(MR,MC)/C(MR,KR)                                      GAUSEL88
61     CONTINUE                                                        GAUSEL89
C                                                                      GAUSEL90
C          SHUFFLE SOLUTION ROWS BACK TO NATURAL ORDER.                GAUSEL91
       DO 71 LL=1,NRM                                                  GAUSEL92
       KR=NR-LL                                                        GAUSEL93
       MKR=L(KR,2)                                                     GAUSEL94
       IF(MKR.EQ.0) GO TO 71                                           GAUSEL95
       MKP=L(KR,1)                                                     GAUSEL96
       DO 7 LC=NRP,NC                                                  GAUSEL97
       Q=C(MKR,LC)                                                     GAUSEL98
       C(MKR,LC)=C(MKP,LC)                                             GAUSEL99
7      C(MKP,LC)=Q                                                     GAUSE100
71     CONTINUE                                                        GAUSE101
C                                                                      GAUSE102
C          NORMAL AND SINGULAR RETURNS.  GOOD SOLUTION COULD HAVE D=0. GAUSE103
8      C(1,1)=D                                                        GAUSE104
       GO TO 91                                                        GAUSE105
9      C(1,1)=0.                                                       GAUSE106
   91 RETURN                                                           GAUSE107
          END                                                          GAUSE108
```

```
       SUBROUTINE RAYPLT                                               RAYPLT 2
C      MAIN PLOTTING PROGRAM; INITIALIZES, READS INPUT, PLOTS          RAYPLT 3
C      PROJECTIONS OF RAYS ON A VERTICAL OR HORIZONTAL PLANE.          RAYPLT 4
C         ABS(PLT)=1. PLOTS PROJECTION OF RAYPATH ON VERTICAL PLANE    RAYPLT 5
C                     RECTANGULAR EXPANSION BY FACTOR 'PFACTR'         RAYPLT 6
C                 =2. PLOTS PROJECTION OF RAYPATH ON GROUND            RAYPLT 7
C                 =3. VERICAL PROJECTION USING RADIAL EXPANSION BY FACTO RAYPLT 8
C                     'PFACTR'                                         RAYPLT 9
C      COMMON DECK "FILEC" INSERTED HERE                               CFILEC 2
```

```
      COMMON /FILEC/NPLTDP                                              CFILEC 4
      COMMON /PLT/ XL,XR,YB,YT,PRESET                                   RAYPLT11
      COMMON/PLT/RMIN,RMAX,ALPHA,APLT                                   RAYPLT12
C                                                                       RAYPLT13
C     COMMON DECK "CONST" INSERTED HERE                                 CCONST 2
      COMMON/PCONST/CREF,RGAS,GAMMA                                     CCONST 4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                         CCONST 5
C     COMMON DECK "FLAG" INSERTED HERE                                  CFLAG  2
      LOGICAL NEWWR,NEWWP,NEWTRC,PENET                                  CFLAG  4
      COMMON /FLG/ NTYP,NEWWR,NEWWP,NEWTRC,PENET,LINES,IHOP,HPUNCH      CFLAG  5
      COMMON/FLGP/NSET                                                  CFLAG  6
C     COMMON DECK "RKAM" INSERTED HERE                                  RKAMCOM2
      REAL KR,KTH,KPH                                                   RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)       RKAMCOM5
C     COMMON DECK "WW" INSERTED HERE                                    CWW    2
      PARAMETER (NWARSZ=1000)                                           CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
     8 (RCVRH,W(20)),                                                   CWW2   7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2   9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2  10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2  11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2  12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2  13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2  15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2  16
      REAL MMODEL,MFORM,MID                                             CWW3   2
C                                                                       CWW3   3
C     WIND        100-124                                               CWW3   4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)           CWW3   5
C                                                                       CWW3   6
C     DELTA WIND        125-149                                         CWW3   7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)        CWW3   8
C                                                                       CWW3   9
C     SOUND SPEED 150-174                                               CWW3  10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)           CWW3  11
      EQUIVALENCE (W(153),REFC)                                         CWW3  12
C                                                                       CWW3  13
C     DELTA SOUND SPEED 175-199                                         CWW3  14
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)        CWW3  15
C                                                                       CWW3  16
C     TEMPERATURE    200-224                                            CWW3  17
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)           CWW3  18
C                                                                       CWW3  19
C     DELTA TEMPERATURE    225-249                                      CWW3  20
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)        CWW3  21
C                                                                       CWW3  22
C     MOLECULAR   250-274                                               CWW3  23
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)           CWW3  24
```

```
C                                                                       CWW3   25
C     RECEIVER HEIGHT   275-299                                         CWW3   26
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)           CWW3   27
C                                                                       CWW3   28
C     TOPOGRAPHY   300-324                                              CWW3   29
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)           CWW3   30
C                                                                       CWW3   31
C     DELTA TOPOGRAPHY   325-349                                        CWW3   32
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)        CWW3   33
C                                                                       CWW3   34
C     UPPER SURFACE TOPOGRAPHY   350-374                                CWW3   35
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)           CWW3   36
C     PLOT ENHANCEMENTS CONTROL PARAMETERS                              CWW3   37
C                                                                       CWW3   38
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)                       CWW3   39
C     ABSORPTION          500-524                                       CWW3   40
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)           CWW3   41
C                                                                       CWW3   42
C     DELTA ABSORPTION          525-549                                 CWW3   43
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)        CWW3   44
C                                                                       CWW3   45
C     PRESSURE          550-574                                         CWW3   46
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)           CWW3   47
C                                                                       CWW3   48
C     DELTA PRESSURE          575-599                                   CWW3   49
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)        CWW3   50
C                                                                       CWW3   51
      REAL LTIC                                                         RAYPLT18
      INTEGER IWDMP(411)                                                RAYPLT19
      EQUIVALENCE(IWDMP,ID)                                             RAYPLT20
C                                                                       RAYPLT21
      EXTERNAL PLOT,PLTLB,PLTLBN,PLTHLB                                 RAYPLT22
C                                                                       RAYPLT23
      DATA NWDSRK/4/                                                    RAYPLT24
C                                                                       RAYPLT25
      IF (.NOT.NEWWR) GO TO 8                                           RAYPLT26
      APLT=ABS(PLT)                                                     RAYPLT27
C                                                                       RAYPLT28
C          NEW W ARRAY -- REINITIALIZE                                  RAYPLT29
      NEWWR=.FALSE.                                                     RAYPLT30
      PRESET=1.                                                         RAYPLT31
C     IF NO ACTIVE PLOTTING,WE ARE STILL DUMPING DATA                   RAYPLT32
      IF(PLT.EQ.0) GO TO 5                                              RAYPLT33
C     INITIALIZE ANNOTATION MODEL                                       RAYPLT34
C  CONVERT COORDINATES OF VERTICAL PLANE FROM GEOGRAPHIC TO GEOMAGNETIC RAYPLT35
      SW=SIN (PLAT)                                                     RAYPLT36
      CW=SIN (PID2-PLAT)                                                RAYPLT37
      SLM=SIN (LLAT)                                                    RAYPLT38
      CLM=SIN (PID2-LLAT)                                               RAYPLT39
      SRM=SIN (RLAT)                                                    RAYPLT40
      CRM=SIN (PID2-RLAT)                                               RAYPLT41
      CDPHI=SIN (PID2-(LLON-PLON))                                      RAYPLT42
      PHL=ATAN2(SIN (LLON-PLON)*CLM,CDPHI*SW*CLM-CW*SLM)                RAYPLT43
      CTHL=CDPHI*CW*CLM+SW*SLM                                          RAYPLT44
      STHL=SIN (ACOS (CTHL))                                            RAYPLT45
```

```
      CDPHI=SIN (PID2-(RLON-PLON))                                      RAYPLT46
      PHR=ATAN2(SIN (RLON-PLON)*CRM,CDPHI*SW*CRM-CW*SRM)                RAYPLT47
      CTHR=CDPHI*CW*CRM+SW*SRM                                          RAYPLT48
      STHR=SIN (ACOS (CTHR))                                            RAYPLT49
      CLR=CTHL*CTHR+STHL*STHR*SIN (PID2-(PHL-PHR))                      RAYPLT50
      ALPHA=.5*ACOS (CLR)                                               RAYPLT51
      SLR=SQRT (1.-CLR**2)                                              RAYPLT52
C                                                                       RAYPLT53
      IF (APLT.EQ.2.) GO TO 3                                           RAYPLT54
C                                                                       RAYPLT55
C     VERTICAL PROJECTIONS ONLY                                         RAYPLT56
      T=HB                                                              RAYPLT57
      HB=AMIN1(T,HT)                                                    RAYPLT58
      HT=AMAX1(T,HT)                                                    RAYPLT59
      RMIN=EARTHR+HB                                                    RAYPLT60
      YT1=YT                                                            RAYPLT61
      RMAX=EARTHR+HT                                                    RAYPLT62
C                                                                       RAYPLT63
      IF(APLT.NE.4.) GO TO 100                                          RAYPLT64
C                                                                       RAYPLT65
C     SCALE FOR CARTESIAN PROJECTION(=4)                                RAYPLT66
      XR=ALPHA                                                          RAYPLT67
      XR1=XR                                                            RAYPLT68
      XL=-XR                                                            RAYPLT69
      XL1=XL                                                            RAYPLT70
      YB=RMIN                                                           RAYPLT71
      YT=RMAX                                                           RAYPLT72
      GO TO 5                                                           RAYPLT73
C                                                                       RAYPLT74
C     PROJECTIONS 1 AND 3                                               RAYPLT75
100   XR1=RMIN*SIN (ALPHA)                                              RAYPLT76
      XL1=-XR1                                                          RAYPLT77
      YB=RMIN*SIN (PID2-ALPHA)                                          RAYPLT78
      IF(APLT.EQ.3.) RMAX=RMIN+(RMAX-RMIN)*PFACTR                       RAYPLT79
      XR=AMAX1(RMAX*SIN(ALPHA),(RMAX-YB)/2.)                            RAYPLT80
      XL=-XR                                                            RAYPLT81
      YT1=RMAX*(YB/RMIN)                                                RAYPLT82
      YT=2.0*XR                                                         RAYPLT83
      IF(APLT.EQ.1) YT=YT/PFACTR                                        RAYPLT84
      YT=YT+YB                                                          RAYPLT85
      GO TO 5                                                           RAYPLT86
C                                                                       RAYPLT87
C     HORIZONTAL PROJECTION(=2)                                         RAYPLT88
3     ALPH1=ATAN2(STHR*SIN (PHR-PHL),(CTHR-CTHL*CLR)/STHL)              RAYPLT89
      XL=0.0                                                            RAYPLT90
      XL1=0.0                                                           RAYPLT91
      XR=EARTHR*2.0*ALPHA                                               RAYPLT92
      XR1=XR                                                            RAYPLT93
C USE 90% OF X-RANGE FOR Y-RANGE                                        RAYPLT94
      RMAX=0.5*(0.90*XR)/PFACTR                                         RAYPLT95
      YT=RMAX                                                           RAYPLT96
      RMIN=-RMAX                                                        RAYPLT97
      YB=RMIN                                                           RAYPLT98
C                                                                       RAYPLT99
5     IF(NPLTDP.LE.0) GO TO 8                                           RAYPL100
```

```
      WRITE(NPLTDP) 1,NWDSRK,1,411
      WRITE(NPLTDP) (IWDMP(I),I=1,411)
C
8     NEW=0
      IF (NEWTRC) NEW=1
      NEWTRC=.FALSE.
C
      IF(NPLTDP.LE.0) GO TO 88
      IF(NEW.NE.1) GO TO 84
      WRITE(NPLTDP) 1,NWDSRK,17,25
      WRITE(NPLTDP) (IWDMP(I),I=17,25)
C
84    WRITE(NPLTDP) 3-NEW,R,TH,PH
C
88    IF(PLT.EQ.0) RETURN
C
      STH=SIN (TH)
      CTH=SIN (PID2-TH)
      CR=CTHR*CTH+STHR*STH*SIN (PID2-(PHR-PH))
      CL=CTHL*CTH+STHL*STH*SIN (PID2-(PHL-PH))
      CEA=ATAN2(CR-CL*CLR,CL*SLR)
C
      IF(APLT.NE.4.) GO TO 150
      CALL PLOT(CEA-ALPHA,R,NEW)
      RETURN
C
150   IF(APLT.EQ.2.) GO TO 10
      RX=R
      IF(APLT.EQ.3.) RX=RMIN+(R-RMIN)*PFACTR
      CALL PLOT(CEA-ALPHA,RX,NEW)
      RETURN
C
10    SL=SQRT(AMAX1(0.,1.-CL**2))
      TMP1=STH*SIN (PH-PHL)
      TMP2=(CTH-CTHL*CL)/STHL
      ALPH2=0.
      IF (TMP1.NE.0..OR.TMP2.NE.0.) ALPH2=ATAN2(TMP1,TMP2)
      CALL PLOT (EARTHR*CEA,EARTHR*ASIN(SL*SIN (ALPH1-ALPH2)),NEW)
      RETURN
C
C          DRAW AXES AND CALL FOR LABELING AND TERMINATION OF THIS PLOT
      ENTRY ENDPLT
C
C     IF NEWWR IS STILL TRUE, NO PLOTS WHERE PRODUCED
      IF(NEWWR) RETURN
      NEWWR=.TRUE.
C
C     SIGNAL END OF PLOT
      IF(NPLTDP.GT.0) WRITE(NPLTDP) 4,(NWDSRK,I=2,NWDSRK)
C
      IF(PLT.EQ.0) RETURN
C
      TICKX=0.01*(YT-YB)
      DTIC=TIC*EARTHR
      CALL SETXY(APLT,-ALPHA,RMIN,ALPHA,RMAX)
```

```
C
      IF(APLT.EQ.4.) GO TO 200
      IF(APLT.EQ.2.) GO TO 25
C
C     CURVLINEAR PROJECTIONS(=1,3)
      CALL ARCTIC(-ALPHA,ALPHA,RMIN,TICKX,PLTHLB)
      CALL ARCTIC(-ALPHA,ALPHA,RMAX,-TICKX,PLTHLB)
      GO TO 300
C
C     CARTESIAN PROJECTION PUT IN BOTTOM BOUNDARY
200   CALL TIKLINE(-ALPHA,RMIN,ALPHA,RMIN,TIC,-TICKX,PLTHLB)
C     PUT IN TOP BOUNDARY
      CALL TIKLINE(-ALPHA,RMAX,ALPHA,RMAX,TIC,TICKX,PLTHLB)
C
300   TIKL=.02*ALPHA
      LTIC=TICV
      IF(APLT.EQ.3.) LTIC=TICV*PFACTR
C     PUT IN LEFT BOUNDARY
      CALL TIKLINE(-ALPHA,RMIN,-ALPHA,RMAX,LTIC,TIKL,PLTLB)
C     PUT IN RIGHT BOUNDARY
      CALL TIKLINE(ALPHA,RMIN,ALPHA,RMAX,LTIC,-TIKL,PLTLBN)
C
      GO TO 50
C   DRAW TICKS, BOX FOR HORIZONTAL PLOT
25    CALL DRAWTKS(DTIC,TICV,XL,XR,YB,YT,PLOT)
C
50    IF(APLT.EQ.2.0) THEN
        CALL PLTANOT(ID,OW/PIT2,XFQMDL,YFQMDL,XL,YB,XR,YT,DEGS,
     2      LLAT,LLON,RLAT,RLON,HB,HT,APLT,DTIC,DTIC/PFACTR,PLOT)
      ELSE
        CALL PLTANOT(ID,OW/PIT2,XFQMDL,YFQMDL,-ALPHA,RMIN,ALPHA,RMAX,
     2      DEGS,LLAT,LLON,RLAT,RLON,HB,HT,APLT,DTIC,TICV,PLOT)
      ENDIF
C
      CALL LABPLT
      CALL PLTEND
      RETURN
      END
```

```
      SUBROUTINE PLOT (X,Y,NEW)
C     PLOTS ONE VECTOR FROM CURRENT PLOT POSITION TO POINT(X,Y)
C     TAKING BORDER CROSSINGS INTO ACCOUNT.
      COMMON /PLT/ XMIN0,XMAX0,YMIN0,YMAX0,RESET
      COMMON/PLT/RMIN,RMAX,ALPHA,APLT
      COMMON /DD/ INT,IOR,IT,IS,IC,ICC,IX,IY
C DEFINE NOMINAL PLOTTING AREA(ZERO SUFFIXES) AND AN
C OUTER CLIPPING BOUNDARY BEYOND WHICH NO VECTORS EXTEND.
      DATA XOLD,YOLD/0.0,0.0/
C 90% FOR Y RANGE
C
C           COMPUTE SCALE FACTORS
```

```
    1 IF (RESET.EQ.0.) GO TO 5
      RESET=0.
      IF(APLT.EQ.2.) THEN
         MRNGE=723
         MINX0=165
         MINY0=140
      ELSE
         MRNGE=813
         MINX0=165
         MINY0=140
      ENDIF
      IF(APLT.EQ.4.) MINY0=0
C
      MAXX0=MINX0+MRNGE
      MAXY0=MINY0+MRNGE
C
      XSCALE=(MAXX0-MINX0)/(XMAX0-XMIN0)
      YSCALE=(MAXY0-MINY0)/(YMAX0-YMIN0)
      XMIN=XMIN0
      YMIN=YMIN0
      XMAX=XMAX0
      YMAX=YMAX0
      IF(APLT.EQ.2.) GO TO 5
C
      XMIN=-ALPHA
      XMAX=ALPHA
      YMIN=RMIN
      YMAX=RMAX
      IF(APLT.NE.4) GO TO 5
      YSCALE=.85*YSCALE
      MINY0=MINY0+60
C
C          START A NEW LINE
C          HORIZONTAL DISPLACEMENT
5     XS=X-XOLD
      YS=Y-YOLD
      S=1.0
      IF(NEW.EQ.0) GO TO 10
      IF(X.GE.XMIN.AND.X.LE.XMAX.AND.Y.GE.YMIN.AND.Y.LE.YMAX) GO TO 48
      GO TO 50
C
10    IF (XS) 11,12,16
C          NEGATIVE
   11 X1=XMAX
      X2=XMIN
      GO TO 20
C          ZERO
   12 IF (YS) 13,50,14
   13 S1=(YMAX-YOLD)/YS
      S2=(YMIN-YOLD)/YS
      GO TO 40
   14 S1=(YMIN-YOLD)/YS
      S2=(YMAX-YOLD)/YS
      GO TO 40
C          POSITIVE
```

```
   16 X1=XMIN
      X2=XMAX
C
C           VERTICAL DISPLACEMENT
   20 IF (YS) 21,22,26
C           NEGATIVE
   21 Y1=YMAX
      Y2=YMIN
      GO TO 30
C           ZERO
   22 S1=(X1-XOLD)/XS
      S2=(X2-XOLD)/XS
      GO TO 40
C           POSITIVE
   26 Y1=YMIN
      Y2=YMAX
C
   30 S1=AMAX1((X1-XOLD)/XS,(Y1-YOLD)/YS)
      S2=AMIN1((X2-XOLD)/XS,(Y2-YOLD)/YS)
C
C           PLOT LINE -- CHECKING FOR BORDER CROSSINGS
40    IF (S2.LT.0..OR.S1.GT.1.) GO TO 50
      IF (S1.LT.0.) GO TO 42
C           PREVIOUS POINT OFF GRAPH
      XP=XOLD+XS*S1
      YP=YOLD+YS*S1
      IF(APLT.EQ.2.0.OR.APLT.EQ.4.0) GO TO 41
      T=XP
      XP=YP*SIN(T)
      YP=YP*COS(T)
C
41    IX=MINX0+(XP-XMIN0)*XSCALE+0.5
      IY=MINY0+(YP-YMIN0)*YSCALE+0.5
      CALL DDBP
      GO TO 48
C
42    IF (S2.GT.1.) GO TO 48
C           CURRENT POINT OFF GRAPH
      S=S2
C           CURRENT POINT ON GRAPH
48    XP=XOLD+XS*S
      YP=YOLD+YS*S
      IF(APLT.EQ.2.0.OR.APLT.EQ.4.0) GO TO 49
      T=XP
      XP=YP*SIN(T)
      YP=YP*COS(T)
49    IX=MINX0+(XP-XMIN0)*XSCALE+0.5
      IY=MINY0+(YP-YMIN0)*YSCALE+0.5
      IF(NEW.EQ.0) CALL DDVC
      IF(NEW.NE.0) CALL DDBP
C
C           EXIT ROUTINE
   50 XOLD=X
      YOLD=Y
      RETURN
```

```
C                                                                 PLOT 124
C          TERMINATE THE CURRENT PLOT                             PLOT 125
      ENTRY PLTEND(X,Y,NEW)                                       PLOT 126
      CALL DDFR                                                   PLOT 127
C                                                                 PLOT 128
      RETURN                                                      PLOT 129
         END                                                      PLOT 130
```

```
      SUBROUTINE LABPLT                                           LABPLT 2
C          LABEL THE CURRENT PLOT                                 LABPLT 3
      CHARACTER*80 LABEL,CHID                                     LABPLT 4
      CHARACTER*4 ANGRANG,ANOTE                                   LABPLT 5
C     COMMON DECK "SS" INSERTED HERE                              CSS    2
      REAL MODSURF                                                CSS    4
      COMMON/SS/ MODSURF(4)                                       CSS    5
      COMMON/SS/U,PUR,PURR,PURTH,PURPH                            CSS    6
      COMMON/SS/PUTH,PUPH,PUTHTH,PUPHPH,PUTHPH,USELECT,UTIME      CSS    7
C     COMMON DECK "GG" INSERTED HERE                              CGG    2
      REAL MODG                                                   CGG    4
      COMMON/GG/MODG(4)                                           CGG    5
      COMMON/GG/G,PGR,PGRR,PGRTH,PGRPH                            CGG    6
      COMMON/GG/PGTH,PGPH,PGTHTH,PGPHPH,PGTHPH,GSELECT,GTIME      CGG    7
C     COMMON DECK "RR" INSERTED HERE                              CRR    2
      REAL MODREC                                                 CRR    4
      COMMON/RR/ MODREC(4)                                        CRR    5
      COMMON/RR/F,PFR,PFRR,PFRTH,PFRPH                            CRR    6
      COMMON/RR/PFTH,PFPH,PFTHTH,PFPHPH,PFTHPH,FSELECT,FTIME      CRR    7
C     COMMON DECK "UU" INSERTED HERE                              CUU    2
      REAL MODU                                                   CUU    4
      COMMON/UU/MODU(4)                                           CUU    5
     1  ,V  ,PVT  ,PVR  ,PVTH  ,PVPH                              CUU    6
     2  ,VR ,PVRT ,PVRR ,PVRTH ,PVRPH                             CUU    7
     3  ,VTH,PVTHT,PVTHR,PVTHTH,PVTHPH                            CUU    8
     4  ,VPH,PVPHT,PVPHR,PVPHTH,PVPHPH                            CUU    9
C     COMMON DECK "CC" INSERTED HERE                              CCC    2
      REAL MODC                                                   CCC    4
      COMMON/CC/MODC(4),CS,PCST,PCSR,PCSTH,PCSPH                  CCC    5
C     COMMON DECK "TT" INSERTED HERE                              CTT    2
      REAL MODT                                                   CTT    4
      COMMON/TT/MODT(4), T,PTT,PTR,PTTH,PTPH                      CTT    5
C     COMMON DECK "MM" INSERTED HERE                              CMM    2
      REAL M,MODM                                                 CMM    4
      COMMON/MM/MODM(4),M,PMT,PMR,PMTH,PMPH                       CMM    5
C     COMMON DECK "HDR" INSERTED HERE                             CHDR   2
      CHARACTER*10 INITID*80,DAT,TOD                              CHDR   4
      COMMON/HDR/SEC                                              CHDR   5
      COMMON/HDRC/INITID,DAT,TOD                                  CHDR   6
C     COMMON DECK "RINPLEX" INSERTED HERE                         CRINPLE2
      REAL KAY2,KAY2I                                             CRINPLE4
      COMPLEX PNP,POLAR,LPOLAR                                    CRINPLE5
      LOGICAL SPACE                                               CRINPLE6
```

```
      CHARACTER DISPM*6                                                 CRINPLE7
      COMMON/RINPL/DISPM                                                CRINPLE8
      COMMON /RIN/ MODRIN(8),RAYNAME(2,3),TYPE(3),SPACE                 CRINPLE9
      COMMON/RIN/OMEGMIN,OMEGMAX,KAY2,KAY2I                             CRINPL10
      COMMON/RIN/PNP(10),POLAR,LPOLAR,SGN                               CRINPL11
      COMMON /DD/ INT,IOR,IT,IS,IC,ICC,IX,IY                            LABPLT15
C     COMMON DECK "CONST" INSERTED HERE                                 CCONST 2
      COMMON/PCONST/CREF,RGAS,GAMMA                                     CCONST 4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                         CCONST 5
C     COMMON DECK "FLAG" INSERTED HERE                                  CFLAG  2
      LOGICAL NEWWR,NEWWP,NEWTRC,PENET                                  CFLAG  4
      COMMON /FLG/ NTYP,NEWWR,NEWWP,NEWTRC,PENET,LINES,IHOP,HPUNCH      CFLAG  5
      COMMON/FLGP/NSET                                                  CFLAG  6
C     COMMON DECK "WW" INSERTED HERE                                    CWW    2
      PARAMETER (NWARSZ=1000)                                           CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
     8 (RCVRH,W(20)),                                                   CWW2   7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2   9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2  10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2  11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2  12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2  13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2  15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2  16
      REAL MMODEL,MFORM,MID                                             CWW3   2
C                                                                       CWW3   3
C     WIND         100-124                                              CWW3   4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)           CWW3   5
C                                                                       CWW3   6
C     DELTA WIND         125-149                                        CWW3   7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)        CWW3   8
C                                                                       CWW3   9
C     SOUND SPEED 150-174                                               CWW3  10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)           CWW3  11
      EQUIVALENCE (W(153),REFC)                                         CWW3  12
C                                                                       CWW3  13
C     DELTA SOUND SPEED 175-199                                         CWW3  14
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)        CWW3  15
C                                                                       CWW3  16
C     TEMPERATURE    200-224                                            CWW3  17
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)           CWW3  18
C                                                                       CWW3  19
C     DELTA TEMPERATURE    225-249                                      CWW3  20
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)        CWW3  21
C                                                                       CWW3  22
C     MOLECULAR   250-274                                               CWW3  23
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)           CWW3  24
C                                                                       CWW3  25
```

```
C     RECEIVER HEIGHT   275-299                                         CWW3  26
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)           CWW3  27
C                                                                       CWW3  28
C     TOPOGRAPHY   300-324                                              CWW3  29
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)           CWW3  30
C                                                                       CWW3  31
C     DELTA TOPOGRAPHY   325-349                                        CWW3  32
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)        CWW3  33
C                                                                       CWW3  34
C     UPPER SURFACE TOPOGRAPHY   350-374                                CWW3  35
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)           CWW3  36
C     PLOT ENHANCEMENTS CONTROL PARAMETERS                              CWW3  37
C                                                                       CWW3  38
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)                       CWW3  39
C     ABSORPTION          500-524                                       CWW3  40
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)           CWW3  41
C                                                                       CWW3  42
C     DELTA ABSORPTION          525-549                                 CWW3  43
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)        CWW3  44
C                                                                       CWW3  45
C     PRESSURE          550-574                                         CWW3  46
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)           CWW3  47
C                                                                       CWW3  48
C     DELTA PRESSURE          575-599                                   CWW3  49
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)        CWW3  50
C                                                                       CWW3  51
C                                                                       LABPLT19
C     COMMON DECK "AA" INSERTED HERE                                    CAA    2
      REAL MODA                                                         CAA    4
      REAL MU,MUPT,MUPR,MUPTH,MUPPH                                     CAA    5
      REAL KAP,KAPPT,KAPPR,KAPPTH,KAPPPH                                CAA    6
      COMMON/AA/MODA(4),MU,MUPT,MUPR,MUPTH,MUPPH                        CAA    7
      COMMON/AA/KAP,KAPPT,KAPPR,KAPPTH,KAPPPH                           CAA    8
C     COMMON DECK "PP" INSERTED HERE                                    CPP    2
      REAL MODP                                                         CPP    4
      COMMON/PP/MODP(4),P,PPT,PPR,PPTH,PPPH                             CPP    5
C                                                                       LABPLT22
      WRITE(LABEL,900) ID                                               LABPLT23
900   FORMAT(10A8)                                                      LABPLT24
      CHID=LABEL                                                        LABPLT25
      LABEL=CHID(5:)                                                    LABPLT26
      IOR=0                                                             LABPLT27
      IT=0                                                              LABPLT28
      IS=2                                                              LABPLT29
      IX=0                                                              LABPLT30
      IY=1023                                                           LABPLT31
      CALL DDTEXT (8,LABEL)                                             LABPLT32
      IX=1090                                                           LABPLT33
      IY=0                                                              LABPLT34
      CALL DDTEXT (2,DAT)                                               LABPLT35
      IY=1023                                                           LABPLT36
      F=OW/PIT2                                                         LABPLT37
      NANGLE=10                                                         LABPLT38
      ANGRANG='EL ='                                                    LABPLT39
      ANOTE='AZ ='                                                      LABPLT40
```

```
      ANEL=0.
      IF(ELSTEP.NE.0.0) ANEL=(ELEND-ELBEG)/ELSTEP+1.5
      IF(ANEL.GT.1.0.OR.PLT.LT.0.0) GO TO 100
         ANGRANG='AZ ='
         ANOTE='EL ='
         NANGLE=14
100   WRITE(LABEL,1300)  CHID(1:3),F,ANOTE,W(NANGLE)*DEGS
1300  FORMAT ('MODEL = ',A,'  ,FREQ =',F9.3,' HZ, ',A,F7.3,' DEG')
      IX=0
      IY=IY-32
      CALL DDTEXT (7,LABEL)
C
C INDEX OF OPPOSITE ANGLE
      NANGLE=(10+14-NANGLE)+1
      NANGLE2=NANGLE+2
      WRITE(LABEL,1400) ANGRANG,(W(I)*DEGS,I=NANGLE,NANGLE2)
1400  FORMAT(A,F7.2,' DEG TO',F7.2,' DEG, STEP =',F7.2,' DEG')
      IY=IY-32
      CALL DDTEXT (7,LABEL)
C
      WRITE(LABEL,1500) XMTRH,TLAT*DEGS,TLON*DEGS
1500  FORMAT('XMTR HT =',F6.2,' KM ,LAT =',F6.2,' DEG, LONG ='
     1   ,F6.2,' DEG')
      IY=IY-32
      CALL DDTEXT (7,LABEL)
C
      IY=IY-32
      WRITE(LABEL,'(10A8)') MODRIN
      CALL DDTEXT(8,LABEL)
C
      IX=1050
      CALL DDTEXT(1,'MODELS')
      IY=IY-15
C     LOOP FOR 8 MODELS AND PERTURBATIONS
      DO 1700 K=1,16
      I=(K-1)/2+1
C     GENERATE ALTERNATING 1,2;3,4 SERIES FOR MODEL AND PERTURBATION
      J1=2*(K+1-I*2)+1
      J2=J1+1
      IF(I.EQ.1) WRITE(LABEL,1600)  (MODU(J),J=J1,J2)
      IF(I.EQ.2) WRITE(LABEL,1600)  (MODC(J),J=J1,J2)
      IF(I.EQ.3) WRITE(LABEL,1600)  (MODT(J),J=J1,J2)
      IF(I.EQ.4 .AND. J1.EQ.1)
     1 WRITE(LABEL,1600)  (MODM(J),J=J1,J2)
C
      IF(I.EQ.5) WRITE(LABEL,1600)  (MODG(J),J=J1,J2)
      IF(I.EQ.6) WRITE(LABEL,1600)  (MODA(J),J=J1,J2)
      IF(I.EQ.7) WRITE(LABEL,1600)  (MODP(J),J=J1,J2)
C NO FURTHER OUTPUT FOR MODELS WITHOUT PERTURBATIONS
      IF(J1.GT.1) GO TO 1610
      IF(I.EQ.8) WRITE(LABEL,1600)  (MODREC(J),J=J1,J2)
1600  FORMAT(2(2(A8,2X,F5.1,1X),2X))
C
1610  IF(LABEL(1:1).EQ.' ') GO TO 1700
      IY=IY-32
```

```
      CALL DDTEXT(2,LABEL)                                              LABPLT96
1700  LABEL(1:1)=' '                                                    LABPLT97
C                                                                       LABPLT98
         END                                                            LABPLT99
```

```
      SUBROUTINE PLTHLB(X,Y,NC)                                         PLTHLB 2
C     HORIZONTAL TICK ANNOTATION ROUTINE FOR RAYPLOT.                   PLTHLB 3
C                                                                       PLTHLB 4
C                                                                       PLTHLB 5
      EXTERNAL PLOT                                                     PLTHLB 6
C                                                                       PLTHLB 7
      CALL PLTANH(X,Y,NC,PLOT)                                          PLTHLB 8
      END                                                               PLTHLB 9
```

```
      SUBROUTINE PLTANH(X,Y,NC,PLOT)                                    PLTANH 2
C                                                                       PLTANH 3
C     TIC LABELING ROUTINE FOR HORIZONTAL PLOT PROJECTIONS              PLTANH 4
C                                                                       PLTANH 5
      CHARACTER ANNOT*10                                                PLTANH 6
      COMMON/DD/IN,IOR,IT,IS,IC,ICC,IX,IY                               PLTANH 7
C     COMMON DECK "WWR" INSERTED HERE                                   CWWR    2
      PARAMETER (NWARSZ=1000)                                           CWW1    3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1    4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2    2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2    3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2    4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2    5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2    6
     8 (RCVRH,W(20)),                                                   CWW2    7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2    8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2    9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2   10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2   11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2   12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2   13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2   14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2   15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2   16
C     COMMON DECK "CONST" INSERTED HERE                                 CCONST 2
      COMMON/PCONST/CREF,RGAS,GAMMA                                     CCONST 4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                         CCONST 5
      COMMON/LABCLT/PROJCT,THMIN,THMAX,RMIN,RMAX                        PLTANH10
C                                                                       PLTANH11
      COMMON/RAYCON/MCONP                                               PLTANH12
C                                                                       PLTANH13
      DATA LNC/-100/                                                    PLTANH14
C                                                                       PLTANH15
```

```
      IF(NC.LE.0 .OR. NC.GT.2) GO TO 100
C
C     NORMALIZE LETTER SIZE FACTOR TO .15 INCHES
      HLETF=HITLET/.15
C
      IF(LNC.NE.NC) LICM=-100
      LNC=NC
C
      CALL PLOT(X,Y,1)
      IX=IX-80*HLETF
      ICM=IX
      IF(NC.GT.1) THEN
         ICM=IY
         IX=IX-40
      ENDIF
C
C     INSURE THAT OVERLAPS OF ANNOTATIONS DO NOT OCCUR
      IF(IABS(ICM-LICM).LT.INT(80*HLETF)) GO TO 100
C
      IF(PROJCT.EQ.2.0) GO TO 25
      IF(HB.GE.0.0.AND.Y.GT.(RMIN+RMAX)/2) GO TO 100
      IF(HB.LT.0.0.AND.Y.LT.(RMIN+RMAX)/2) GO TO 100
C
25    F=DEGS
      TMP=(THMAX-THMIN)*DEGS
      IF(TMP.LT.10.) F =EARTHR
      FT=F
      IF(PROJCT.EQ.2.0 .AND. MCONP.EQ.0) F=F/EARTHR
      V=X-THMIN
      IF(NC.GT.1) V=Y-(RMIN+RMAX)/2.
      IF(ABS(AMOD(TIC*FT+.0001,1.)).GT..001) GO TO 60
C
      WRITE(ANNOT,50) INT(V*F+SIGN(.5,V))
50    FORMAT(I5)
      GO TO 90
C
60    WRITE(ANNOT,80) V*F
80    FORMAT(F8.2)
C
90    LICM=ICM
C     ALLOW 8 RASTERS FOR THE LETTER HEIGHT
      H=HB
      IF(PROJCT.EQ.2.0) H=0.
      IF(NC.EQ.1) IY=IY-8*HLETF-SIGN(52.,H)
      IOR=0
95    FORMAT(3G13.6,2I5,1X,A10)
99    FORMAT(2G13.6,5I5,1X,A10)
      CALL DDTEXT(1,ANNOT)
C
100   CALL PLOT(X,Y,MIN0(1,NC))
      END
```

```
      SUBROUTINE SETXY(PROJ,XMIN,YMIN,XMAX,YMAX)
C     PLOT INITIALIZATION; SETS PROJECTION PARAMETERS
C
      COMMON/LABCLT/PROJCT,THMIN,THMAX,RMIN,RMAX
C
C
C   INITIAL ANNOTATION MODEL
      CALL SETANN
C
      PROJCT=PROJ
      THMIN=XMIN
      RMIN=YMIN
      THMAX=XMAX
      RMAX=YMAX
      END
```

```
      SUBROUTINE TIKLINE(XL1,YB,XL,YT1,TICV,TIKSZ,PLOT)
C     DRAWS STRAIGHT LINE WITH TICKS AT INTERVALS
C
C
      XDF=XL-XL1
      YDF=YT1-YB
      DST=SQRT(XDF*XDF+YDF*YDF)
      TICE=DST
      T=0.0
      IF(TICV.EQ.0.0) GO TO 50
      TICE=TICV
      T=TIKSZ/TICV
C
50    TICVX=TICE*XDF/DST
      TICVY=TICE*YDF/DST
      TX=TICVY*T
      TY=-TICVX*T
      NTIC=1+DST/TICE
      CALL PLOT(XL1,YB,-1)
      DO 100 I=0,NTIC-1
        X=XL1+I*TICVX
        Y=YB+I*TICVY
        CALL PLOT(X,Y,0)
        CALL PLOT(X+TX,Y+TY,0)
100   CALL PLOT(X,Y,1)
      CALL PLOT(XL,YT1,0)
      CALL PLOT(XL,YT1,1)
      END
```

```
      SUBROUTINE PLTANOT(ID,FREQ,XMF,YMF,XL,YB,XR,YT
```

```
     1     ,DEGS,LLAT,LLON,RLAT,RLON,ALTLOW,ALTHI,PLT,DTICH,DTICV,PLOT)  PLTANOT3
C      PUTS STANDARD ANNOTATIONS ON PLOTS                               PLTANOT4
C                                                                       PLTANOT5
       COMMON/LABCLT/PROJCT,THMIN,THMAX,RMIN,RMAX                       PLTANOT6
       REAL LLAT,LLON                                                   PLTANOT7
       CHARACTER LABEL*80                                               PLTANOT8
       COMMON /DD/ INT,IOR,IT,IS,IC,ICC,IX,IY                           PLTANOT9
C      COMMON DECK "ANNOT" INSERTED HERE                                ANNOT  2
       CHARACTER*10 ANOTES,HNOTES                                       ANNOT  4
       COMMON/ANNCTL/LENA(4),LENHA(3)                                   ANNOT  5
       COMMON/ANNCTC/ANOTES(2,4),HNOTES(4,3)                            ANNOT  6
C                                                                       PLTANO11
       IF(PLT.EQ.2. .OR. XMF.LE.0.0 .OR. YMF.LE.0.0) GO TO 45           PLTANO12
       CALL PLOT(XL+XMF*(XR-XL),YB+YMF*(YT-YB),1)                       PLTANO13
       IF(FREQ.LE.0.0) GO TO 30                                         PLTANO14
C                                                                       PLTANO15
       WRITE(LABEL,25) FREQ                                             PLTANO16
25     FORMAT('FREQ =',F9.3)                                            PLTANO17
       CALL DDTEXT(2,LABEL)                                             PLTANO18
       IY=IY+40                                                         PLTANO19
C                                                                       PLTANO20
30     WRITE(LABEL,35) ID                                               PLTANO21
35     FORMAT('MODEL = ',A3)                                            PLTANO22
       CALL DDTEXT(2,LABEL)                                             PLTANO23
C                                                                       PLTANO24
45     NSPY=97                                                          PLTANO25
       IOR=0                                                            PLTANO26
C                                                                       PLTANO27
C FIRST THE LOW ALTITUDE-LATITUDE ANNOTATION                            PLTANO28
50     CALL PLOT(XL,YB,1)                                               PLTANO29
       IX=IX-155                                                        PLTANO30
       IY=IY-NSPY                                                       PLTANO31
       WRITE(LABEL,1840) DEGS*LLON                                      PLTANO32
       CALL DDTEXT(2,LABEL)                                             PLTANO33
       WRITE(LABEL,1850) DEGS*LLAT                                      PLTANO34
       IY=IY-32                                                         PLTANO35
       CALL DDTEXT(2,LABEL)                                             PLTANO36
C                                                                       PLTANO37
C NEXT THE RIGHT LATITUDE ANNOTATION                                    PLTANO38
       CALL PLOT(XR,YB,1)                                               PLTANO39
       IX=IX-130                                                        PLTANO40
       IY=IY-NSPY                                                       PLTANO41
       WRITE(LABEL,1840) DEGS*RLON                                      PLTANO42
       CALL DDTEXT(2,LABEL)                                             PLTANO43
       WRITE(LABEL,1850) DEGS*RLAT                                      PLTANO44
       IY=IY-32                                                         PLTANO45
       CALL DDTEXT(2,LABEL)                                             PLTANO46
C                                                                       PLTANO47
       XMID=(XL+XR)/2.0                                                 PLTANO48
       IF(PLT.EQ.2.0) GO TO 55                                          PLTANO49
C                                                                       PLTANO50
C PUT THE HORIZONTAL TIC LABEL                                          PLTANO51
       Y=YT                                                             PLTANO52
       IF(ALTLOW.GE.0.0) Y=YB                                           PLTANO53
       CALL PLOT(XMID,Y,1)                                              PLTANO54
```

```
C                                                                       PLTANO55
      IF(ALTLOW.LT.0.0) IY=IY+80                                        PLTANO56
      IF(ALTLOW.GE.0.0) IY=IY-95                                        PLTANO57
      GO TO 60                                                          PLTANO58
C                                                                       PLTANO59
55    CALL PLOT(XMID,YB,1)                                              PLTANO60
      IY=IY-95                                                          PLTANO61
C                                                                       PLTANO62
60    IX=IX-235                                                         PLTANO63
      NOTEA=1                                                           PLTANO64
      TMP=(THMAX-THMIN)*DEGS                                            PLTANO65
      IF(TMP.GT.10.) NOTEA=2                                            PLTANO66
C                                                                       PLTANO67
      CALL DDTEXT (LENHA(NOTEA),HNOTES(1,NOTEA))                        PLTANO68
C                                                                       PLTANO69
C PUT THE VERTICAL TIC LABEL                                            PLTANO70
      CALL PLOT(XL,(YB+YT)/2.0,1)                                       PLTANO71
      IOR=1                                                             PLTANO72
      IF(PLT.NE.2.) GO TO 100                                           PLTANO73
C                                                                       PLTANO74
C   HORIZONTAL PLOT  PUT Y-AXIS ANNOTATION                              PLTANO75
C                                                                       PLTANO76
      IX=IX-125                                                         PLTANO77
      IY=200                                                            PLTANO78
      CALL DDTEXT (LENHA(3),HNOTES(1,3))                                PLTANO79
      GO TO 200                                                         PLTANO80
C                                                                       PLTANO81
100   IX=IX-125                                                         PLTANO82
      IY=IY-75                                                          PLTANO83
      NOTEA=1                                                           PLTANO84
      IF(ALTLOW.GE.0.0) NOTEA=3                                         PLTANO85
      IF(ABS(ALTHI-ALTLOW).GE.1.) NOTEA=NOTEA+1                         PLTANO86
      CALL DDTEXT (LENA(NOTEA),ANOTES(1,NOTEA))                         PLTANO87
C                                                                       PLTANO88
200   IOR=0                                                             PLTANO89
1840  FORMAT(F7.2,' DEG E.')                                            PLTANO90
1850  FORMAT(F7.2,' DEG N.')                                            PLTANO91
C                                                                       PLTANO92
      END                                                               PLTANO93
```

```
      SUBROUTINE DRAWTKS(DTICH,DTICV,XLPR,XR,YBPR,YT,PLOT)              DRAWTKS2
C                                                                       DRAWTKS3
C     HORIZONTAL PLOT PROJECTION SUPPORT ROUTINE.                       DRAWTKS4
C     DRAW BOUNDARY TO PLOT AREA, TICS AND TIC LABELS.                  DRAWTKS5
C                                                                       DRAWTKS6
      EXTERNAL PLOT                                                     DRAWTKS7
C                                                                       DRAWTKS8
      XL=XLPR                                                           DRAWTKS9
      YB=YBPR                                                           DRAWTK10
      YMID=.5*(YB+YT)                                                   DRAWTK11
      TICX=.01*(YT-YB)                                                  DRAWTK12
```

```
      TICY=.01*(XR-XL)                                                  DRAWTK13
      YBP=YB                                                            DRAWTK14
      IF(DTICV.GT.0.) YBP=YMID-AINT((YMID-YB)/DTICV)*DTICV              DRAWTK15
C                                                                       DRAWTK16
      NTICX=(XR-XL)/DTICH                                               DRAWTK17
      NTICY=1                                                           DRAWTK18
      IF(DTICV.GT.0.) NTICY=(YT-YBP+DTICV)/DTICV                        DRAWTK19
C&     PRINT *,"DRAWTKS ",XLPR,XR,YBPR,YBP,YT,DTICH,DTICV,NTICX,NTICY   DRAWTK20
      IOF=0                                                             DRAWTK21
C                                                                       DRAWTK22
C     TWO PASSES BOTTOM AND LEFT THEN TOP AND RIGHT                     DRAWTK23
      DO 100 J=1,2                                                      DRAWTK24
      CALL PLOT(XL,YBPR,1)                                              DRAWTK25
      DO 30 I=1,NTICY                                                   DRAWTK26
      Y=YBP+(I-1)*DTICV                                                 DRAWTK27
      CALL PLOT(XL,Y,0)                                                 DRAWTK28
      CALL PLOT(XL+TICY,Y,0)                                            DRAWTK29
30    CALL PLTANH(XL,Y,IOF+2,PLOT)                                      DRAWTK30
      CALL PLOT(XL,YT,0)                                                DRAWTK31
C                                                                       DRAWTK32
      CALL PLOT(XLPR,YB,1)                                              DRAWTK33
C                                                                       DRAWTK34
      DO 40 I=1,NTICX                                                   DRAWTK35
      X=XLPR+I*DTICH                                                    DRAWTK36
      CALL PLOT(X,YB,0)                                                 DRAWTK37
      CALL PLOT(X,YB+TICX,0)                                            DRAWTK38
      CALL PLTANH(X,YB,IOF+1,PLOT)                                      DRAWTK39
40    CALL PLOT(X,YB,1)                                                 DRAWTK40
      CALL PLOT(XR,YB,0)                                                DRAWTK41
      YB=YT                                                             DRAWTK42
      XL=XR                                                             DRAWTK43
      TICX=-TICX                                                        DRAWTK44
      TICY=-TICY                                                        DRAWTK45
100   IOF=10                                                            DRAWTK46
C                                                                       DRAWTK47
      CALL PLOT(XLPR,YMID,1)                                            DRAWTK48
      CALL PLOT(XR,YMID,0)                                              DRAWTK49
      END                                                               DRAWTK50
```

```
      SUBROUTINE PLTLB(X,PY,NCP)                                        PLTLB   2
C     PUT VERTICAL TIC ANNOTATIONS ON RAY PLOT                          PLTLB   3
C     THIS IS A SPECIAL PURPOSE SUBSTITUTE FOR THE 'PLOT' ROUTINE       PLTLB   4
C     IT COMPUTES THE NEAREST ROUNDED TIC POSITIONS FOR VERTICAL LINES. PLTLB   5
C     WHEN THE NC PARAMETER IS > 0 AN ANNOTATION IS GENERATED           PLTLB   6
C     AT THE TIC POSITION. THE ADDITIONAL NC VALUE < 0 ALLOWS FOR       PLTLB   7
C     SIMPLE PEN UP MOTION TO THE X,Y POSITION.                         PLTLB   8
C                                                                       PLTLB   9
      CHARACTER ANNOT*10                                                PLTLB  10
C                                                                       PLTLB  11
C     COMMON DECK "WWR" INSERTED HERE                                   CWWR    2
      PARAMETER (NWARSZ=1000)                                           CWW1    3
```

```
      COMMON/WW/ID(10),MAXW,W(NWARSZ)
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),
     8 (RCVRH,W(20)),
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),
     6 (HMIN,W(27)),(RGMAX,W(28)),
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))
      COMMON /DD/ IINT,IOR,IT,IS,IC,ICC,IX,IY
      COMMON/LABCLT/PROJCT,THMIN,THMAX,RMIN,RMAX
C
      DATA LIY/-100/
C
      NC=NCP
      GO TO 10
C
      ENTRY PLTLBN(X,PY,NCP)
      NC=0
C
10    Y=PY
C
      IF(NCP.LT.0) GO TO 200
      F=1.
      IF(AMAX1(ABS(RMAX-EARTHR),ABS(RMIN-EARTHR)).LT.1.) F=1000.
      Y=INT((Y-EARTHR)*F)/F+EARTHR
      IF(NC.EQ.0) GO TO 100
C
      V=Y
      IF(PROJCT.EQ.3.) V=RMIN+(Y-RMIN)/PFACTR
      IF(ABS(AINT(TICV*F)-TICV*F).GT..001) GO TO 60
C
      WRITE(ANNOT,50) INT(ABS(V-EARTHR)*F)
50    FORMAT(I3)
      GO TO 90
C
60    WRITE(ANNOT,80) ABS(V-EARTHR)*F
80    FORMAT(F6.2)
C
90    CALL PLOT(X,PY,1)
      CALL PLOT(X,Y,1)
C     INSURE THAT OVERLAPS OF ANNOTATIONS DO NOT OCCUR
      IF(IABS(IY-LIY).LT.80) GO TO 100
      LIY=IY
      IX=IX-100
      IOR=0
      CALL DDTEXT(1,ANNOT)
C
```

```
100    CALL PLOT(X,Y,NC)                                                 PLTLB 52
       RETURN                                                            PLTLB 53
C                                                                        PLTLB 54
200    CALL PLOT(X,Y,1)                                                  PLTLB 55
       RETURN                                                            PLTLB 56
       END                                                               PLTLB 57


       SUBROUTINE ARCTIC(THMIN,THMAX,HEIGHT,TICY,PLOT)                   ARCTIC 2
C      DRAW RANGE AXIS IN RAY TRACE PLOT. INCLUDES ANY CURVILINEAR       ARCTIC 3
C      PROJECTIONS PROVIDED IN DDGRAPH.                                  ARCTIC 4
C      COMMON DECK "WWR" INSERTED HERE                                   CWWR   2
       PARAMETER (NWARSZ=1000)                                           CWW1   3
       COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
       REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
       EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
      1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
      2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
      3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
      8 (RCVRH,W(20)),                                                   CWW2   7
      4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
      5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2   9
      6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2  10
      8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2  11
      6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2  12
      7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2  13
      9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2  14
      1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2  15
      2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2  16
       COMMON /DDSCALE/ XMIN,XMAX,YMIN,YMAX,MINX,MAXX,MINY,MAXY,SCX,SCY, ARCTIC 6
      1 NSCX,NXCY,MSCX,MSCY,ISCX,ISCY                                    ARCTIC 7
C                                                                        ARCTIC 8
       NTIC=2                                                            ARCTIC 9
       IF(TIC.NE.0.) NTIC=1+(THMAX-THMIN)/TIC                            ARCTIC10
       NLINE=MAX0(2,100/NTIC)                                            ARCTIC11
C                                                                        ARCTIC12
       TICN=TIC/(NLINE-2)                                                ARCTIC13
       DO 10 I=1,NTIC                                                    ARCTIC14
       X=THMIN+(I-1)*TIC                                                 ARCTIC15
       CALL PLOT(X,HEIGHT+TICY,1)                                        ARCTIC16
       DO 10 J=2,NLINE                                                   ARCTIC17
       XJ=X+(J-2)*TICN                                                   ARCTIC18
       IF(XJ.GT.THMAX) GO TO 15                                          ARCTIC19
10     CALL PLOT(XJ,HEIGHT,0)                                            ARCTIC20
C                                                                        ARCTIC21
15     CALL PLOT(THMAX,HEIGHT,0)                                         ARCTIC22
       CALL PLOT(THMAX,HEIGHT+TICY,1)                                    ARCTIC23
       RETURN                                                            ARCTIC24
       END                                                               ARCTIC25
```

```
      BLOCK DATA PLOTBL                                                 PLOTBL 2
C     BLOCK DATA INITIALIZING PLOT ROUTINE COUNTERS AND                 PLOTBL 3
C     EXTREME VALUES VARIABLES.                                         PLOTBL 4
      COMMON/KNKN/KNBP,KNVC,KNDT                                        PLOTBL 5
      COMMON/DDLIM/MXIX,MXIY,MNIX,MNIY                                  PLOTBL 6
      DATA MXIX,MXIY,MNIX,MNIY/2*-1000,2*1000/                          PLOTBL 7
      DATA KNBP,KNVC,KNDT/3*0/                                          PLOTBL 8
      END                                                               PLOTBL 9
```

```
      SUBROUTINE DDINIT(N,TEXT)                                         DDINIT 2
C     INITIALIZES PLOTTING PROCESS(DDPLOT)                              DDINIT 3
C     WRITE PARAMETERS TO GRAPHICS CALLS FILE                           DDINIT 4
C     COMMON DECK "RAYDEV" INSERTED HERE                                CRAYDEV2
C     DEVICE ASSIGNED TO RAYTRC INPUT FILE                              CRAYDEV4
      COMMON/RAYDEV/NRYIND,NDEVTMP,NFRMAT,NDEVGRP,NDEVBIN               CRAYDEV5
      CHARACTER TEXT*(*)                                                DDINIT 6
C                                                                       DDINIT 7
      DATA IDD/0/                                                       DDINIT 8
10    FORMAT(5A10)                                                      DDINIT 9
      IF(IDD.EQ.0) REWIND NDEVGRP                                       DDINIT10
      IDD=1                                                             DDINIT11
      WRITE(NDEVGRP) 0,0,0                                              DDINIT12
      WRITE(NDEVGRP) M,LEN(TEXT),TEXT                                   DDINIT13
      END                                                               DDINIT14
```

```
      SUBROUTINE DDBP                                                   DDBP    2
C     SETS A VECTOR ORIGIN(DDPLOT)                                      DDBP    3
C     WRITE PARAMETERS TO GRAPHICS CALLS FILE                           DDBP    4
C     COMMON DECK "RAYDEV" INSERTED HERE                                CRAYDEV2
C     DEVICE ASSIGNED TO RAYTRC INPUT FILE                              CRAYDEV4
      COMMON/RAYDEV/NRYIND,NDEVTMP,NFRMAT,NDEVGRP,NDEVBIN               CRAYDEV5
      COMMON/DD/IN,IOR,IT,IS,IC,ICC,IX,IY                               DDBP    6
      COMMON/KNKN/KNBP,KNVC,KNDT                                        DDBP    7
      COMMON/DDLIM/MXIX,MXIY,MNIX,MNIY                                  DDBP    8
      MNIX=MIN0(MNIX,IX)                                                DDBP    9
      MXIX=MAX0(MXIX,IX)                                                DDBP   10
      MNIY=MIN0(MNIY,IY)                                                DDBP   11
      MXIY=MAX0(MXIY,IY)                                                DDBP   12
      IF(IX.LT.0.OR.IX.GT.1023.OR.IY.LT.0.OR.IY.GT.1023                 DDBP   13
     1  ) PRINT 10,'DDBP ',KNBP,IX,IY                                   DDBP   14
10    FORMAT(A10,3I5)                                                   DDBP   15
      KNBP=KNBP+1                                                       DDBP   16
      WRITE(NDEVGRP) 1,IX,IY                                            DDBP   17
      END                                                               DDBP   18
```

```
      SUBROUTINE DDVC                                                   DDVC     2
C     PLOTS A VECTOR(DDPLOT)                                            DDVC     3
C     WRITE PARAMETERS TO GRAPHICS CALLS FILE                           DDVC     4
C     COMMON DECK "RAYDEV" INSERTED HERE                                CRAYDEV2
C     DEVICE ASSIGNED TO RAYTRC INPUT FILE                              CRAYDEV4
      COMMON/RAYDEV/NRYIND,NDEVTMP,NFRMAT,NDEVGRP,NDEVBIN               CRAYDEV5
      COMMON/DD/IN,IOR,IT,IS,IC,ICC,IX,IY                               DDVC     6
      COMMON/KNKN/KNBP,KNVC,KNDT                                        DDVC     7
      COMMON/DDLIM/MXIX,MXIY,MNIX,MNIY                                  DDVC     8
      MNIX=MIN0(MNIX,IX)                                                DDVC     9
      MXIX=MAX0(MXIX,IX)                                                DDVC    10
      MNIY=MIN0(MNIY,IY)                                                DDVC    11
      MXIY=MAX0(MXIY,IY)                                                DDVC    12
      IF(IX.LT.0.OR.IX.GT.1023.OR.IY.LT.0.OR.IY.GT.1023                 DDVC    13
     1  ) PRINT 10,'DDVC ',KNVC,IX,IY                                   DDVC    14
10    FORMAT(A10,3I5)                                                   DDVC    15
      KNVC=KNVC+1                                                       DDVC    16
      WRITE(NDEVGRP) 2,IX,IY                                            DDVC    17
      END                                                               DDVC    18
```

```
      SUBROUTINE DDTEXT(N,TEXT)                                         DDTEXT 2
C     WRITES A CHARACTER ARRAY PACKED A10(DDPLOT)                       DDTEXT 3
C     WRITE PARAMETERS TO GRAPHICS CALLS FILE                           DDTEXT 4
C     COMMON DECK "RAYDEV" INSERTED HERE                                CRAYDEV2
C     DEVICE ASSIGNED TO RAYTRC INPUT FILE                              CRAYDEV4
      COMMON/RAYDEV/NRYIND,NDEVTMP,NFRMAT,NDEVGRP,NDEVBIN               CRAYDEV5
      CHARACTER TEXT*(*)                                                DDTEXT 6
      COMMON/DD/IN,IOR,IT,IS,IC,ICC,IX,IY                               DDTEXT 7
      COMMON/KNKN/KNBP,KNVC,KNDT                                        DDTEXT 8
100   FORMAT(A10,2I5,I2,1X,8A10)                                        DDTEXT 9
      KNDT=KNDT+1                                                       DDTEXT10
      WRITE(NDEVGRP) 3,IX,IY                                            DDTEXT11
      WRITE(NDEVGRP) IOR,N,N*10,(TEXT(I:I),I=1,N*10)                    DDTEXT12
      END                                                               DDTEXT13
```

```
      SUBROUTINE DDTAB                                                  DDTAB  2
C     INITIALIZES TABULAR PLOTTING(DDPLOT)                              DDTAB  3
C     WRITE PARAMETERS TO GRAPHICS CALLS FILE                           DDTAB  4
C     COMMON DECK "RAYDEV" INSERTED HERE                                CRAYDEV2
C     DEVICE ASSIGNED TO RAYTRC INPUT FILE                              CRAYDEV4
      COMMON/RAYDEV/NRYIND,NDEVTMP,NFRMAT,NDEVGRP,NDEVBIN               CRAYDEV5
      END                                                               DDTAB  6
```

```
      SUBROUTINE DDFR                                                   DDFR    2
C     ADVANCE ONE PLOTTING FRAME(DDPLOT)                                DDFR    3
C     WRITE PARAMETERS TO GRAPHICS CALLS FILE                           DDFR    4
C     COMMON DECK "RAYDEV" INSERTED HERE                                CRAYDEV2
C     DEVICE ASSIGNED TO RAYTRC INPUT FILE                              CRAYDEV4
      COMMON/RAYDEV/NRYIND,NDEVTMP,NFRMAT,NDEVGRP,NDEVBIN                CRAYDEV5
10    FORMAT(A10)                                                       DDFR    6
      WRITE(NDEVGRP) -2,0,0                                             DDFR    7
      END                                                               DDFR    8
```

```
      SUBROUTINE DDEND                                                  DDEND   2
C     EMPTIES PLOT BUFFER AND RELEASES PLOTTING COMMAND FILE(DDPLOT)    DDEND   3
C     WRITE PARAMETERS TO GRAPHICS CALLS FILE                           DDEND   4
C     COMMON DECK "RAYDEV" INSERTED HERE                                CRAYDEV2
C     DEVICE ASSIGNED TO RAYTRC INPUT FILE                              CRAYDEV4
      COMMON/RAYDEV/NRYIND,NDEVTMP,NFRMAT,NDEVGRP,NDEVBIN                CRAYDEV5
      COMMON/KNKN/KNBP,KNVC,KNDT                                        DDEND   6
      COMMON/DDLIM/MXIX,MXIY,MNIX,MNIY                                  DDEND   7
10    FORMAT(3A10,5I5)                                                  DDEND   8
      WRITE(NDEVGRP) -1,0,0                                             DDEND   9
      END                                                               DDEND  10
```

```
      SUBROUTINE DASH                                                   DASH    2
C     ACTIVATE DASHED LINE CONNECTIONS(DISSPLA)                         DASH    3
C     WRITE PARAMETERS TO GRAPHICS CALLS FILE                           DASH    4
C     COMMON DECK "RAYDEV" INSERTED HERE                                CRAYDEV2
C     DEVICE ASSIGNED TO RAYTRC INPUT FILE                              CRAYDEV4
      COMMON/RAYDEV/NRYIND,NDEVTMP,NFRMAT,NDEVGRP,NDEVBIN                CRAYDEV5
      WRITE(NDEVGRP) 37,0,0                                             DASH    6
      END                                                               DASH    7
```

```
      SUBROUTINE RESET(S)                                               RESET   2
C     RESETS AN OPTION TO ITS DEFAULT VALUE(DISSPLA)                    RESET   3
C     WRITE PARAMETERS TO GRAPHICS CALLS FILE                           RESET   4
C     COMMON DECK "RAYDEV" INSERTED HERE                                CRAYDEV2
C     DEVICE ASSIGNED TO RAYTRC INPUT FILE                              CRAYDEV4
      COMMON/RAYDEV/NRYIND,NDEVTMP,NFRMAT,NDEVGRP,NDEVBIN                CRAYDEV5
      CHARACTER S*10                                                    RESET   6
      WRITE(NDEVGRP) 38,0,0                                             RESET   7
```

```
      WRITE(NDEVGRP) S                                               RESET  8
      END                                                            RESET  9


      SUBROUTINE HEIGHT(H)                                           HEIGHT 2
C     SETS REFERENCE CHARACTER HEIGHT IN INCHES(DISSPLA)             HEIGHT 3
C     WRITE PARAMETERS TO GRAPHICS CALLS FILE                        HEIGHT 4
C     COMMON DECK "RAYDEV" INSERTED HERE                             CRAYDEV2
C     DEVICE ASSIGNED TO RAYTRC INPUT FILE                           CRAYDEV4
      COMMON/RAYDEV/NRYIND,NDEVTMP,NFRMAT,NDEVGRP,NDEVBIN             CRAYDEV5
      WRITE(NDEVGRP) 13,H,0                                          HEIGHT 6
      END                                                            HEIGHT 7


      SUBROUTINE MX1ALF(T1,T2)                                       MX1ALF 2
C     SPECIFY USE OF ALTERNATE CHARACTER SET NUMBER 1(DISSPLA)       MX1ALF 3
C     WRITE PARAMETERS TO GRAPHICS CALLS FILE                        MX1ALF 4
C     COMMON DECK "RAYDEV" INSERTED HERE                             CRAYDEV2
C     DEVICE ASSIGNED TO RAYTRC INPUT FILE                           CRAYDEV4
      COMMON/RAYDEV/NRYIND,NDEVTMP,NFRMAT,NDEVGRP,NDEVBIN             CRAYDEV5
      WRITE(NDEVGRP) 11,T1,T2                                        MX1ALF 6
      END                                                            MX1ALF 7


      SUBROUTINE MX2ALF(T1,T2)                                       MX2ALF 2
C     SPECIFY USE OF ALTERNATE CHARACTER SET NUMBER 2(DISSPLA)       MX2ALF 3
C     WRITE PARAMETERS TO GRAPHICS CALLS FILE                        MX2ALF 4
C     COMMON DECK "RAYDEV" INSERTED HERE                             CRAYDEV2
C     DEVICE ASSIGNED TO RAYTRC INPUT FILE                           CRAYDEV4
      COMMON/RAYDEV/NRYIND,NDEVTMP,NFRMAT,NDEVGRP,NDEVBIN             CRAYDEV5
      WRITE(NDEVGRP) 12,T1,T2                                        MX2ALF 6
      END                                                            MX2ALF 7


      SUBROUTINE SCMPLX                                              SCMPLX 2
C     SPECIFY USE OF SIMPLEX CHARACTER SET(DISSPLA)                  SCMPLX 3
C     WRITE PARAMETERS TO GRAPHICS CALLS FILE                        SCMPLX 4
C     COMMON DECK "RAYDEV" INSERTED HERE                             CRAYDEV2
C     DEVICE ASSIGNED TO RAYTRC INPUT FILE                           CRAYDEV4
      COMMON/RAYDEV/NRYIND,NDEVTMP,NFRMAT,NDEVGRP,NDEVBIN             CRAYDEV5
      WRITE(NDEVGRP) 10,0,0                                          SCMPLX 6
      END                                                            SCMPLX 7
```

DISPERSION-RELATION ROUTINES (Tape File 4)

```
      SUBROUTINE ANWNL                                                  ANWNL  8
C     DISPERSION RELATION FOR ACOUSTIC WAVES NO WIND, NO LOSSES         ANWNL  9
C     COMMON DECK "RKAM" INSERTED HERE                                  RKAMCOM2
      REAL KR,KTH,KPH                                                   RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)       RKAMCOM5
C     COMMON DECK "CC" INSERTED HERE                                    CCC    2
      REAL MODC                                                         CCC    4
      COMMON/CC/MODC(4),CS,PCST,PCSR,PCSTH,PCSPH                        CCC    5
C     COMMON DECK "CONST" INSERTED HERE                                 CCONST 2
      COMMON/PCONST/CREF,RGAS,GAMMA                                     CCONST 4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                         CCONST 5
C     COMMON DECK "RK" INSERTED HERE                                    CRK    2
C     DEFINE SIZE REQUIRED FOR RAY STATE SAVE ARRAY                     CRK    4
      PARAMETER (LRKAMS=87+2*100,NXRKMS=12+LRKAMS,MXEQPT=21)            CRK    5
      PARAMETER (NRKSAV=NXRKMS+MXEQPT-1)                                CRK    6
      COMMON /RK/ NEQS,STEP,MODE,E1MAX,E1MIN,E2MAX,E2MIN,FACT,RSTART    CRK    7
C     COMMON DECK "RINREAL" INSERTED HERE                               CRINREA2
      LOGICAL SPACE                                                     CRINREA4
      REAL LPOLAR,LPOLRI,KPHK,KPHKI,KAY2,KAY2I                          CRINREA5
      CHARACTER DISPM*6                                                 CRINREA6
      COMMON/RINPL/DISPM                                                CRINREA7
      COMMON /RIN/ MODRIN(8),RAYNAME(2,3),TYPE(3),SPACE                 CRINREA8
      COMMON/RIN/OMEGMIN,OMEGMAX,KAY2,KAY2I,                            CRINREA9
     1                 H,HI,PHT,PHTI,PHR,PHRI,PHTH,PHTHI,PHPH,PHPHI     CRINRE10
     2, PHOW,PHOWI,PHKR,PHKRI,PHKTH,PHKTI, PHKPH,PHKPI                  CRINRE11
     3 ,KPHK,KPHKI,POLAR,POLARI,LPOLAR,LPOLRI,SGN                       CRINRE12
C     COMMON DECK "UU" INSERTED HERE                                    CUU    2
      REAL MODU                                                         CUU    4
      COMMON/UU/MODU(4)                                                 CUU    5
     1  ,V  ,PVT  ,PVR  ,PVTH  ,PVPH                                    CUU    6
     2  ,VR ,PVRT ,PVRR ,PVRTH ,PVRPH                                   CUU    7
     3  ,VTH,PVTHT,PVTHR,PVTHTH,PVTHPH                                  CUU    8
     4  ,VPH,PVPHT,PVPHR,PVPHTH,PVPHPH                                  CUU    9
C     COMMON DECK "WW" INSERTED HERE                                    CWW    2
      PARAMETER (NWARSZ=1000)                                           CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
     8 (RCVRH,W(20)),                                                   CWW2   7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2   9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2  10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2  11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2  12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2  13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2  15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2  16
      REAL MMODEL,MFORM,MID                                             CWW3   2
C                                                                       CWW3   3
C     WIND        100-124                                               CWW3   4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)           CWW3   5
```

```
C
C     DELTA WIND          125-149
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)
C
C     SOUND SPEED 150-174
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)
      EQUIVALENCE (W(153),REFC)
C
C     DELTA SOUND SPEED 175-199
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)
C
C     TEMPERATURE     200-224
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)
C
C     DELTA TEMPERATURE     225-249
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)
C
C     MOLECULAR   250-274
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)
C
C     RECEIVER HEIGHT   275-299
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)
C
C     TOPOGRAPHY   300-324
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)
C
C     DELTA TOPOGRAPHY   325-349
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)
C
C     UPPER SURFACE TOPOGRAPHY   350-374
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)
C     PLOT ENHANCEMENTS CONTROL PARAMETERS
C
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)
C     ABSORPTION          500-524
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)
C
C     DELTA ABSORPTION          525-549
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)
C
C     PRESSURE          550-574
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)
C
C     DELTA PRESSURE          575-599
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)
C
C
      REAL KS,MKS
      REAL KVECT(24)
      REAL VSET(20)
      EQUIVALENCE(KVECT,KAY2),(VSET,V)
C
      DATA MODRIN/8HACOUSTIC ,8H WAVE ** ,8H** NO WI ,8HND *** N
     1  ,8HO LOSSES ,3*1H /
      DATA DISPM/'ANWNL'/
```

```
      DATA TYPE/3*1H0 /                                                 ANNBL   5
C                                                                       ANWNL  25
      ENTRY IDISPER                                                     ANWNL  26
      CALL ISPEED                                                       ANWNL  27
      CALL IRECVR                                                       ANWNL  28
      CALL ITOPOG                                                       ANWNL  29
      CALL ISURFAC                                                      ANWNL  30
      RETURN                                                            ANWNL  31
C                                                                       ANWNL  32
      ENTRY DISPER                                                      ANWNL  33
      ENTRY RINDEX                                                      ANWNL  34
C                                                                       ANWNL  35
      SPACE=.FALSE.                                                     ANWNL  36
C                                                                       ANWNL  37
      KS=KR*KR+KTH*KTH+KPH*KPH                                          ANWNL  38
C     SOUND SPEED                                                       ANWNL  39
      CALL SPEED                                                        ANWNL  40
      OWS=OW*OW                                                         ANWNL  41
      KAY2=OWS/CS                                                       ANWNL  42
C                                                                       ANWNL  43
      H=OW*OW - CS*KS                                                   ANWNL  44
      MKS=-KS                                                           ANWNL  45
      PHT=MKS*PCST                                                      ANWNL  46
      PHR=MKS*PCSR                                                      ANWNL  47
      PHTH=MKS*PCSTH                                                    ANWNL  48
      PHPH=MKS*PCSPH                                                    ANWNL  49
C                                                                       ANWNL  50
      PHOW=2.0*OW                                                       ANWNL  51
      CS2=-2.0*CS                                                       ANWNL  52
      PHKR=CS2*KR                                                       ANWNL  53
      PHKTH=CS2*KTH                                                     ANWNL  54
      PHKPH=CS2*KPH                                                     ANWNL  55
      KPHK=CS2*KS                                                       ANWNL  56
      RETURN                                                            ANWNL  57
      END                                                               ANWNL  58
```

```
      SUBROUTINE AWWNL                                                  AWWNL   8
C     DISPERSION RELATION FOR ACOUSTIC WAVES WITH WIND, NO LOSSES       AWWNL   9
C     COMMON DECK "RKAM" INSERTED HERE                                  RKAMCOM2
      REAL KR,KTH,KPH                                                   RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)       RKAMCOM5
C     COMMON DECK "CC" INSERTED HERE                                    CCC     2
      REAL MODC                                                         CCC     4
      COMMON/CC/MODC(4),CS,PCST,PCSR,PCSTH,PCSPH                        CCC     5
C     COMMON DECK "CONST" INSERTED HERE                                 CCONST  2
      COMMON/PCONST/CREF,RGAS,GAMMA                                     CCONST  4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                         CCONST  5
C     COMMON DECK "RK" INSERTED HERE                                    CRK     2
C     DEFINE SIZE REQUIRED FOR RAY STATE SAVE ARRAY                     CRK     4
      PARAMETER (LRKAMS=87+2*100,NXRKMS=12+LRKAMS,MXEQPT=21)            CRK     5
      PARAMETER (NRKSAV=NXRKMS+MXEQPT-1)                                CRK     6
```

```
      COMMON /RK/ NEQS,STEP,MODE,E1MAX,E1MIN,E2MAX,E2MIN,FACT,RSTART    CRK      7
C     COMMON DECK "RINREAL" INSERTED HERE                               CRINREA2
      LOGICAL SPACE                                                     CRINREA4
      REAL LPOLAR,LPOLRI,KPHK,KPHKI,KAY2,KAY2I                          CRINREA5
      CHARACTER DISPM*6                                                 CRINREA6
      COMMON/RINPL/DISPM                                                CRINREA7
      COMMON /RIN/ MODRIN(8),RAYNAME(2,3),TYPE(3),SPACE                 CRINREA8
      COMMON/RIN/OMEGMIN,OMEGMAX,KAY2,KAY2I,                            CRINREA9
     1              H,HI,PHT,PHTI,PHR,PHRI,PHTH,PHTHI,PHPH,PHPHI        CRINRE10
     2, PHOW,PHOWI,PHKR,PHKRI,PHKTH,PHKTI, PHKPH,PHKPI                  CRINRE11
     3 ,KPHK,KPHKI,POLAR,POLARI,LPOLAR,LPOLRI,SGN                       CRINRE12
C     COMMON DECK "UU" INSERTED HERE                                    CUU      2
      REAL MODU                                                         CUU      4
      COMMON/UU/MODU(4)                                                 CUU      5
     1  ,V  ,PVT  ,PVR  ,PVTH  ,PVPH                                    CUU      6
     2  ,VR ,PVRT ,PVRR ,PVRTH ,PVRPH                                   CUU      7
     3  ,VTH,PVTHT,PVTHR,PVTHTH,PVTHPH                                  CUU      8
     4  ,VPH,PVPHT,PVPHR,PVPHTH,PVPHPH                                  CUU      9
C     COMMON DECK "WW" INSERTED HERE                                    CWW      2
      PARAMETER (NWARSZ=1000)                                           CWW1     3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1     4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2     2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2     3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2     4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2     5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2     6
     8 (RCVRH,W(20)),                                                   CWW2     7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2     8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2     9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2    10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2    11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2    12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2    13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2    14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2    15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2    16
      REAL MMODEL,MFORM,MID                                             CWW3     2
C                                                                       CWW3     3
C     WIND         100-124                                              CWW3     4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)           CWW3     5
C                                                                       CWW3     6
C     DELTA WIND         125-149                                        CWW3     7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)        CWW3     8
C                                                                       CWW3     9
C     SOUND SPEED 150-174                                               CWW3    10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)           CWW3    11
      EQUIVALENCE (W(153),REFC)                                         CWW3    12
C                                                                       CWW3    13
C     DELTA SOUND SPEED 175-199                                         CWW3    14
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)        CWW3    15
C                                                                       CWW3    16
C     TEMPERATURE     200-224                                           CWW3    17
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)           CWW3    18
C                                                                       CWW3    19
C     DELTA TEMPERATURE     225-249                                     CWW3    20
```

```
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)        CWW3   21
C                                                                       CWW3   22
C     MOLECULAR   250-274                                               CWW3   23
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)           CWW3   24
C                                                                       CWW3   25
C     RECEIVER HEIGHT   275-299                                         CWW3   26
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)           CWW3   27
C                                                                       CWW3   28
C     TOPOGRAPHY   300-324                                              CWW3   29
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)           CWW3   30
C                                                                       CWW3   31
C     DELTA TOPOGRAPHY   325-349                                        CWW3   32
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)        CWW3   33
C                                                                       CWW3   34
C     UPPER SURFACE TOPOGRAPHY   350-374                                CWW3   35
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)           CWW3   36
C     PLOT ENHANCEMENTS CONTROL PARAMETERS                              CWW3   37
C                                                                       CWW3   38
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)                       CWW3   39
C     ABSORPTION          500-524                                       CWW3   40
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)           CWW3   41
C                                                                       CWW3   42
C     DELTA ABSORPTION          525-549                                 CWW3   43
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)        CWW3   44
C                                                                       CWW3   45
C     PRESSURE          550-574                                         CWW3   46
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)           CWW3   47
C                                                                       CWW3   48
C     DELTA PRESSURE          575-599                                   CWW3   49
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)        CWW3   50
C                                                                       CWW3   51
C                                                                       AWWNL 17
      REAL KS,KV                                                        AWWNL 18
      REAL KVECT(24)                                                    AWWNL 19
      EQUIVALENCE(KVECT,KAY2)                                           AWWNL 20
C                                                                       AWWNL 21
      DATA MODRIN/8HACOUSTIC ,8H WAVE ** ,8H* WITH W ,8HIND ****        AWNBL  2
     1  ,8H** NO LO ,4HSSES ,2*1H /                                     AWNBL  3
      DATA DISPM/'AWWNL'/                                               AWNBL  4
      DATA TYPE/3*1H1 /                                                 AWNBL  5
C                                                                       AWWNL 24
      ENTRY IDISPER                                                     AWWNL 25
      CALL ISPEED                                                       AWWNL 26
      CALL IWINDR                                                       AWWNL 27
      CALL IRECVR                                                       AWWNL 28
      CALL ITOPOG                                                       AWWNL 29
      CALL ISURFAC                                                      AWWNL 30
      RETURN                                                            AWWNL 31
C                                                                       AWWNL 32
      ENTRY DISPER                                                      AWWNL 33
C                                                                       AWWNL 34
      SPACE=.FALSE.                                                     AWWNL 35
C                                                                       AWWNL 36
      KS=KR*KR+KTH*KTH+KPH*KPH                                          AWWNL 37
C     WIND VELOCITY                                                     AWWNL 38
```

```
      CALL WINDR                                                        AWWNL 39
      KV=KR*VR+KTH*VTH+KPH*VPH                                          AWWNL 40
      VLS=KV*KV/KS                                                      AWWNL 41
C     SOUND SPEED                                                       AWWNL 42
      CALL SPEED                                                        AWWNL 43
      OWS=OW*OW                                                         AWWNL 44
      KAY2=OWS/(SQRT(CS)+KV/SQRT(KS))**2                                AWWNL 45
C                                                                       AWWNL 46
      OWI=OW-KV                                                         AWWNL 47
      H=OWI*OWI - CS*KS                                                 AWWNL 48
      POWIT=-KR*PVRT - KTH*PVTHT - KPH*PVPHT                            AWWNL 49
      PHT=2.0*OWI*POWIT - KS*PCST                                       AWWNL 50
      POWIR=-KR*PVRR - KTH*PVTHR - KPH*PVPHR                            AWWNL 51
      PHR=2.0*OWI*POWIR - KS*PCSR                                       AWWNL 52
      POWITH=-KR*PVRTH - KTH*PVTHTH - KPH*PVPHTH                        AWWNL 53
      PHTH=2.0*OWI*POWITH - KS*PCSTH                                    AWWNL 54
      POWIPH=-KR*PVRPH - KTH*PVTHPH - KPH*PVPHPH                        AWWNL 55
      PHPH=2.0*OWI*POWIPH - KS*PCSPH                                    AWWNL 56
C                                                                       AWWNL 57
      PHOW=2.0*OWI                                                      AWWNL 58
      PHKR=-2.0*(OWI*VR + CS*KR)                                        AWWNL 59
      PHKTH=-2.0*(OWI*VTH + CS*KTH)                                     AWWNL 60
      PHKPH=-2.0*(OWI*VPH + CS*KPH)                                     AWWNL 61
      KPHK=-2.0*(OWI*KV + CS*KS)                                        AWWNL 62
      RETURN                                                            AWWNL 63
      END                                                               AWWNL 64
```

```
      SUBROUTINE ANWWL                                                  ANWWL  8
C     DISPERSION RELATION FOR ACOUSTIC WAVES NO WIND, WITH LOSSES       ANWWL  9
C     COMMON DECK "RKAM" INSERTED HERE                                  RKAMCOM2
      REAL KR,KTH,KPH                                                   RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)       RKAMCOM5
C     COMMON DECK "CC" INSERTED HERE                                    CCC    2
      REAL MODC                                                         CCC    4
      COMMON/CC/MODC(4),CS,PCST,PCSR,PCSTH,PCSPH                        CCC    5
C     COMMON DECK "CONST" INSERTED HERE                                 CCONST 2
      COMMON/PCONST/CREF,RGAS,GAMMA                                     CCONST 4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                         CCONST 5
C     COMMON DECK "RK" INSERTED HERE                                    CRK    2
C     DEFINE SIZE REQUIRED FOR RAY STATE SAVE ARRAY                     CRK    4
      PARAMETER (LRKAMS=87+2*100,NXRKMS=12+LRKAMS,MXEQPT=21)            CRK    5
      PARAMETER (NRKSAV=NXRKMS+MXEQPT-1)                                CRK    6
      COMMON /RK/ NEQS,STEP,MODE,E1MAX,E1MIN,E2MAX,E2MIN,FACT,RSTART    CRK    7
C     COMMON DECK "RINREAL" INSERTED HERE                               CRINREA2
      LOGICAL SPACE                                                     CRINREA4
      REAL LPOLAR,LPOLRI,KPHK,KPHKI,KAY2,KAY2I                          CRINREA5
      CHARACTER DISPM*6                                                 CRINREA6
      COMMON/RINPL/DISPM                                                CRINREA7
      COMMON /RIN/ MODRIN(8),RAYNAME(2,3),TYPE(3),SPACE                 CRINREA8
      COMMON/RIN/OMEGMIN,OMEGMAX,KAY2,KAY2I,                            CRINREA9
     1           H,HI,PHT,PHTI,PHR,PHRI,PHTH,PHTHI,PHPH,PHPHI           CRINRE10
```

```
     2, PHOW,PHOWI,PHKR,PHKRI,PHKTH,PHKTI, PHKPH,PHKPI                   CRINRE11
     3 ,KPHK,KPHKI,POLAR,POLARI,LPOLAR,LPOLRI,SGN                          CRINRE12
C     COMMON DECK "UU" INSERTED HERE                                       CUU     2
      REAL MODU                                                            CUU     4
      COMMON/UU/MODU(4)                                                    CUU     5
     1  ,V  ,PVT  ,PVR  ,PVTH  ,PVPH                                       CUU     6
     2  ,VR ,PVRT ,PVRR ,PVRTH ,PVRPH                                      CUU     7
     3  ,VTH,PVTHT,PVTHR,PVTHTH,PVTHPH                                     CUU     8
     4  ,VPH,PVPHT,PVPHR,PVPHTH,PVPHPH                                     CUU     9
C     COMMON DECK "WW" INSERTED HERE                                       CWW     2
      PARAMETER (NWARSZ=1000)                                              CWW1    3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                      CWW1    4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                   CWW2    2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),      CWW2    3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),        CWW2    4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),            CWW2    5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),           CWW2    6
     8 (RCVRH,W(20)),                                                      CWW2    7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2    8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                         CWW2    9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                         CWW2   10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                        CWW2   11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),          CWW2   12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))           CWW2   13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),           CWW2   14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))                 CWW2   15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                      CWW2   16
      REAL MMODEL,MFORM,MID                                                CWW3    2
C                                                                          CWW3    3
C     WIND         100-124                                                 CWW3    4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)              CWW3    5
C                                                                          CWW3    6
C     DELTA WIND         125-149                                           CWW3    7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)           CWW3    8
C                                                                          CWW3    9
C     SOUND SPEED 150-174                                                  CWW3   10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)              CWW3   11
      EQUIVALENCE (W(153),REFC)                                            CWW3   12
C                                                                          CWW3   13
C     DELTA SOUND SPEED 175-199                                            CWW3   14
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)           CWW3   15
C                                                                          CWW3   16
C     TEMPERATURE    200-224                                               CWW3   17
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)              CWW3   18
C                                                                          CWW3   19
C     DELTA TEMPERATURE    225-249                                         CWW3   20
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)           CWW3   21
C                                                                          CWW3   22
C     MOLECULAR   250-274                                                  CWW3   23
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)              CWW3   24
C                                                                          CWW3   25
C     RECEIVER HEIGHT   275-299                                            CWW3   26
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)              CWW3   27
C                                                                          CWW3   28
C     TOPOGRAPHY   300-324                                                 CWW3   29
```

```
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)             CWW3   30
C                                                                         CWW3   31
C     DELTA TOPOGRAPHY   325-349                                          CWW3   32
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)          CWW3   33
C                                                                         CWW3   34
C     UPPER SURFACE TOPOGRAPHY   350-374                                  CWW3   35
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)             CWW3   36
C     PLOT ENHANCEMENTS CONTROL PARAMETERS                                CWW3   37
C                                                                         CWW3   38
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)                         CWW3   39
C     ABSORPTION          500-524                                         CWW3   40
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)             CWW3   41
C                                                                         CWW3   42
C     DELTA ABSORPTION          525-549                                   CWW3   43
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)          CWW3   44
C                                                                         CWW3   45
C     PRESSURE          550-574                                           CWW3   46
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)             CWW3   47
C                                                                         CWW3   48
C     DELTA PRESSURE          575-599                                     CWW3   49
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)          CWW3   50
C                                                                         CWW3   51
C                                                                         ANWWL  17
C     COMMON DECK "AA" INSERTED HERE                                      CAA     2
      REAL MODA                                                           CAA     4
      REAL MU,MUPT,MUPR,MUPTH,MUPPH                                       CAA     5
      REAL KAP,KAPPT,KAPPR,KAPPTH,KAPPPH                                  CAA     6
      COMMON/AA/MODA(4),MU,MUPT,MUPR,MUPTH,MUPPH                          CAA     7
      COMMON/AA/KAP,KAPPT,KAPPR,KAPPTH,KAPPPH                             CAA     8
C                                                                         ANWWL  19
C     COMMON DECK "PP" INSERTED HERE                                      CPP     2
      REAL MODP                                                           CPP     4
      COMMON/PP/MODP(4),P,PPT,PPR,PPTH,PPPH                               CPP     5
C                                                                         ANWWL  21
C     COMMON DECK "MM" INSERTED HERE                                      CMM     2
      REAL M,MODM                                                         CMM     4
      COMMON/MM/MODM(4),M,PMT,PMR,PMTH,PMPH                               CMM     5
C                                                                         ANWWL  23
      REAL KS,MKS                                                         ANWWL  24
      REAL KVECT(24)                                                      ANWWL  25
      REAL VSET(20)                                                       ANWWL  26
      EQUIVALENCE(KVECT,KAY2),(VSET,V)                                    ANWWL  27
C                                                                         ANWWL  28
      DATA MODRIN/8HACOUSTIC ,8H WAVE ** ,8H** NO WI ,8HND *** W          ANWBL   2
     1 ,8HWITH LOS ,3HSES ,2*1H /                                         ANWBL   3
      DATA DISPM/'ANWWL'/                                                 ANWBL   4
      DATA TYPE/3*1H2 /                                                   ANWBL   5
C                                                                         ANWWL  31
      ENTRY IDISPER                                                       ANWWL  32
      CALL ISPEED                                                         ANWWL  33
      CALL IRECVR                                                         ANWWL  34
      CALL ITOPOG                                                         ANWWL  35
      CALL ISURFAC                                                        ANWWL  36
C                                                                         ANWWL  37
      CALL IPRES                                                          ANWWL  38
```

```
      CALL IABSRP
      RETURN
C
      ENTRY DISPER
      ENTRY RINDEX
C
      SPACE=.FALSE.
C
      KS=KR*KR+KTH*KTH+KPH*KPH
C     SOUND SPEED
      CALL SPEED
      OWS=OW*OW
      KAY2=OWS/CS
C
      H=OW*OW - CS*KS
      MKS=-KS
      PHT=MKS*PCST
      PHR=MKS*PCSR
      PHTH=MKS*PCSTH
      PHPH=MKS*PCSPH
C
      PHOW=2.0*OW
      CS2=-2.0*CS
      PHKR=CS2*KR
      PHKTH=CS2*KTH
      PHKPH=CS2*KPH
      KPHK=CS2*KS
C
      CALL PRES
      CALL ABSRP
      GMS=GAMMA-1.0
      KAY2I=-(OWI/(GAMMA*P))*(4./3.*MU+GMS*GMS*M/(GAMMA*RGAS)*KAP)*KAY2
      RETURN
      END
```

```
      SUBROUTINE AWWWL
C     DISPERSION RELATION FOR ACOUSTIC WAVES WITH WIND, WITH LOSSES
C     COMMON DECK "RKAM" INSERTED HERE
      REAL KR,KTH,KPH
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)
C     COMMON DECK "CC" INSERTED HERE
      REAL MODC
      COMMON/CC/MODC(4),CS,PCST,PCSR,PCSTH,PCSPH
C     COMMON DECK "CONST" INSERTED HERE
      COMMON/PCONST/CREF,RGAS,GAMMA
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10
C     COMMON DECK "RK" INSERTED HERE
C     DEFINE SIZE REQUIRED FOR RAY STATE SAVE ARRAY
      PARAMETER (LRKAMS=87+2*100,NXRKMS=12+LRKAMS,MXEQPT=21)
      PARAMETER (NRKSAV=NXRKMS+MXEQPT-1)
      COMMON /RK/ NEQS,STEP,MODE,E1MAX,E1MIN,E2MAX,E2MIN,FACT,RSTART
```

```
C     COMMON DECK "RINREAL" INSERTED HERE
      LOGICAL SPACE                                                        CRINREA2
      REAL LPOLAR,LPOLRI,KPHK,KPHKI,KAY2,KAY2I                             CRINREA4
      CHARACTER DISPM*6                                                    CRINREA5
      COMMON/RINPL/DISPM                                                   CRINREA6
      COMMON /RIN/ MODRIN(8),RAYNAME(2,3),TYPE(3),SPACE                    CRINREA7
      COMMON/RIN/OMEGMIN,OMEGMAX,KAY2,KAY2I,                               CRINREA8
     1             H,HI,PHT,PHTI,PHR,PHRI,PHTH,PHTHI,PHPH,PHPHI            CRINREA9
     2, PHOW,PHOWI,PHKR,PHKRI,PHKTH,PHKTI, PHKPH,PHKPI                     CRINRE10
     3 ,KPHK,KPHKI,POLAR,POLARI,LPOLAR,LPOLRI,SGN                          CRINRE11
C     COMMON DECK "UU" INSERTED HERE                                       CRINRE12
      REAL MODU                                                            CUU     2
      COMMON/UU/MODU(4)                                                    CUU     4
     1  ,V  ,PVT  ,PVR  ,PVTH  ,PVPH                                       CUU     5
     2  ,VR ,PVRT ,PVRR ,PVRTH ,PVRPH                                      CUU     6
     3  ,VTH,PVTHT,PVTHR,PVTHTH,PVTHPH                                     CUU     7
     4  ,VPH,PVPHT,PVPHR,PVPHTH,PVPHPH                                     CUU     8
C     COMMON DECK "WW" INSERTED HERE                                       CUU     9
      PARAMETER (NWARSZ=1000)                                              CWW     2
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                      CWW1    3
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                   CWW1    4
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),       CWW2    2
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),         CWW2    3
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),             CWW2    4
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),            CWW2    5
     8 (RCVRH,W(20)),                                                      CWW2    6
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))   CWW2    7
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                         CWW2    8
     6 (HMIN,W(27)),(RGMAX,W(28)),                                         CWW2    9
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                        CWW2   10
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),          CWW2   11
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))           CWW2   12
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),           CWW2   13
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))                 CWW2   14
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                      CWW2   15
      REAL MMODEL,MFORM,MID                                                CWW2   16
C                                                                          CWW3    2
C     WIND         100-124                                                 CWW3    3
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)              CWW3    4
C                                                                          CWW3    5
C     DELTA WIND        125-149                                            CWW3    6
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)           CWW3    7
C                                                                          CWW3    8
C     SOUND SPEED 150-174                                                  CWW3    9
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)              CWW3   10
      EQUIVALENCE (W(153),REFC)                                            CWW3   11
C                                                                          CWW3   12
C     DELTA SOUND SPEED 175-199                                            CWW3   13
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)           CWW3   14
C                                                                          CWW3   15
C     TEMPERATURE     200-224                                              CWW3   16
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)              CWW3   17
C                                                                          CWW3   18
C     DELTA TEMPERATURE     225-249                                        CWW3   19
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)           CWW3   20
                                                                           CWW3   21
```

```
C                                                                       CWW3   22
C     MOLECULAR   250-274                                               CWW3   23
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)           CWW3   24
C                                                                       CWW3   25
C     RECEIVER HEIGHT   275-299                                         CWW3   26
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)           CWW3   27
C                                                                       CWW3   28
C     TOPOGRAPHY   300-324                                              CWW3   29
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)           CWW3   30
C                                                                       CWW3   31
C     DELTA TOPOGRAPHY   325-349                                        CWW3   32
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)        CWW3   33
C                                                                       CWW3   34
C     UPPER SURFACE TOPOGRAPHY   350-374                                CWW3   35
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)           CWW3   36
C     PLOT ENHANCEMENTS CONTROL PARAMETERS                              CWW3   37
C                                                                       CWW3   38
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)                       CWW3   39
C     ABSORPTION        500-524                                         CWW3   40
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)           CWW3   41
C                                                                       CWW3   42
C     DELTA ABSORPTION         525-549                                  CWW3   43
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)        CWW3   44
C                                                                       CWW3   45
C     PRESSURE         550-574                                          CWW3   46
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)           CWW3   47
C                                                                       CWW3   48
C     DELTA PRESSURE         575-599                                    CWW3   49
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)        CWW3   50
C                                                                       CWW3   51
C                                                                       AWWWL  17
C     COMMON DECK "AA" INSERTED HERE                                    CAA     2
      REAL MODA                                                         CAA     4
      REAL MU,MUPT,MUPR,MUPTH,MUPPH                                     CAA     5
      REAL KAP,KAPPT,KAPPR,KAPPTH,KAPPPH                                CAA     6
      COMMON/AA/MODA(4),MU,MUPT,MUPR,MUPTH,MUPPH                        CAA     7
      COMMON/AA/KAP,KAPPT,KAPPR,KAPPTH,KAPPPH                           CAA     8
C                                                                       AWWWL  19
C     COMMON DECK "PP" INSERTED HERE                                    CPP     2
      REAL MODP                                                         CPP     4
      COMMON/PP/MODP(4),P,PPT,PPR,PPTH,PPPH                             CPP     5
C                                                                       AWWWL  21
C     COMMON DECK "MM" INSERTED HERE                                    CMM     2
      REAL M,MODM                                                       CMM     4
      COMMON/MM/MODM(4),M,PMT,PMR,PMTH,PMPH                             CMM     5
C                                                                       AWWWL  23
      REAL KS,KV                                                        AWWWL  24
      REAL KVECT(24)                                                    AWWWL  25
      EQUIVALENCE(KVECT,KAY2)                                           AWWWL  26
C                                                                       AWWWL  27
      DATA MODRIN/8HACOUSTIC ,8H WAVE ** ,8H* WITH W ,8HIND ****        AWWBL   2
     1 ,8H** WITH  ,6HLOSSES ,2*1H /                                    AWWBL   3
      DATA DISPM/'AWWWL'/                                               AWWBL   4
      DATA TYPE/3*1H3 /                                                 AWWBL   5
C                                                                       AWWWL  30
```

```
      ENTRY IDISPER
      CALL ISPEED
      CALL IWINDR
      CALL IRECVR
      CALL ITOPOG
      CALL ISURFAC
C
      CALL IPRES
      CALL IABSRP
      RETURN
C
      ENTRY DISPER
C
      SPACE=.FALSE.
C
      KS=KR*KR+KTH*KTH+KPH*KPH
C     WIND VELOCITY
      CALL WINDR
      KV=KR*VR+KTH*VTH+KPH*VPH
      VLS=KV*KV/KS
C     SOUND SPEED
      CALL SPEED
      OWS=OW*OW
      KAY2=OWS/(SQRT(CS)+KV/SQRT(KS))**2
C
      OWI=OW-KV
      H=OWI*OWI - CS*KS
      POWIT=-KR*PVRT - KTH*PVTHT - KPH*PVPHT
      PHT=2.0*OWI*POWIT - KS*PCST
      POWIR=-KR*PVRR - KTH*PVTHR - KPH*PVPHR
      PHR=2.0*OWI*POWIR - KS*PCSR
      POWITH=-KR*PVRTH - KTH*PVTHTH - KPH*PVPHTH
      PHTH=2.0*OWI*POWITH - KS*PCSTH
      POWIPH=-KR*PVRPH - KTH*PVTHPH - KPH*PVPHPH
      PHPH=2.0*OWI*POWIPH - KS*PCSPH
C
      PHOW=2.0*OWI
      PHKR=-2.0*(OWI*VR + CS*KR)
      PHKTH=-2.0*(OWI*VTH + CS*KTH)
      PHKPH=-2.0*(OWI*VPH + CS*KPH)
      KPHK=-2.0*(OWI*KV + CS*KS)
C
      CALL PRES
      CALL ABSRP
      GMS=GAMMA-1.0
      KAY2I=-(OW/(GAMMA*P))*(4./3.*MU+GMS*GMS*M/(GAMMA*RGAS)*KAP)*KAY2
      RETURN
      END
```

```
      SUBROUTINE WLINEAR                                                WLINEAR8
C     LINEAR WIND VELOCITY PROFILE                                      WLINEAR9
C     PROVIDES CONSTANT RADIAL, ZONAL AND MERIDONAL WINDS               WLINEA10
C     EXCEPT THAT A POSSIBLE LINEAR HEIGHT GRADIENT OF THE ZONAL        WLINEA11
C     COMPONENT IS ALLOWED.                                             WLINEA12
C     COMMON DECK "RKAM" INSERTED HERE                                  RKAMCOM2
      REAL KR,KTH,KPH                                                   RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)       RKAMCOM5
C     COMMON DECK "UU" INSERTED HERE                                    CUU    2
      REAL MODU                                                         CUU    4
      COMMON/UU/MODU(4)                                                 CUU    5
     1   ,V   ,PVT   ,PVR   ,PVTH   ,PVPH                               CUU    6
     2   ,VR  ,PVRT  ,PVRR  ,PVRTH  ,PVRPH                              CUU    7
     3   ,VTH,PVTHT,PVTHR,PVTHTH,PVTHPH                                 CUU    8
     4   ,VPH,PVPHT,PVPHR,PVPHTH,PVPHPH                                 CUU    9
C     COMMON DECK "WW" INSERTED HERE                                    CWW    2
      PARAMETER (NWARSZ=1000)                                           CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
     8 (RCVRH,W(20)),                                                   CWW2   7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2   9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2  10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2  11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2  12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2  13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2  15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2  16
      REAL MMODEL,MFORM,MID                                             CWW3   2
C                                                                       CWW3   3
C     WIND          100-124                                             CWW3   4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)           CWW3   5
C                                                                       CWW3   6
C     DELTA WIND          125-149                                       CWW3   7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)        CWW3   8
C                                                                       CWW3   9
C     SOUND SPEED 150-174                                               CWW3  10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)           CWW3  11
      EQUIVALENCE (W(153),REFC)                                         CWW3  12
C                                                                       CWW3  13
C     DELTA SOUND SPEED 175-199                                         CWW3  14
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)        CWW3  15
C                                                                       CWW3  16
C     TEMPERATURE    200-224                                            CWW3  17
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)           CWW3  18
C                                                                       CWW3  19
C     DELTA TEMPERATURE    225-249                                      CWW3  20
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)        CWW3  21
C                                                                       CWW3  22
C     MOLECULAR   250-274                                               CWW3  23
```

```
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)
C
C     RECEIVER HEIGHT   275-299
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)
C
C     TOPOGRAPHY   300-324
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)
C
C     DELTA TOPOGRAPHY   325-349
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)
C
C     UPPER SURFACE TOPOGRAPHY   350-374
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)
C     PLOT ENHANCEMENTS CONTROL PARAMETERS
C
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)
C     ABSORPTION          500-524
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)
C
C     DELTA ABSORPTION          525-549
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)
C
C     PRESSURE          550-574
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)
C
C     DELTA PRESSURE          575-599
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)
C
C     COMMON DECK "B1" INSERTED HERE
      INTEGER UMX,UNTBL,UITBL,UFRMTBL,IDSU(10)
      COMMON/B1/UMX,UNTBL(10),UITBL(10),UFRMTBL(10),UGP(10)
      EQUIVALENCE (UGP,IDSU)
      EQUIVALENCE (W(103),UR0)
      EQUIVALENCE (W(104),UTH0),(W(105),UPH0),(W(106),PUPHZ0)
C
      DATA RECOGU/1.0/
      DATA UMX/1/
      DATA UNTBL/1,11,8*0/
      DATA UITBL/1,9*0/
      DATA UFRMTBL/1,9*0/
C
      ENTRY IWINDR
C
      IF(RECOGU .NE. UMODEL)
     1    CALL RERROR('SPEED   ','WRNG MODEL',RECOGU)
C
      MODU(1)=7HWLINEAR
      MODU(2)=UID
      CALL IPWINDR
C
      RETURN
C
      ENTRY WINDR
      H = R - EARTHR
C
```

```
      CALL CLEAR(V,20)                                                  WLINEA38
C                                                                       WLINEA39
       VR = UR0                                                         WLINEA40
       VTH = UTH0                                                       WLINEA41
       VPH = (UPH0 + PUPHZ0 * H)                                        WLINEA42
      V=SQRT(VR*VR+VTH*VTH+VPH*VPH)                                     WLINEA43
      PVR=PUPHZ0                                                        WLINEA44
      IF(V.NE.0.0) PVR=VPH/V*PUPHZ0                                     WLINEA45
      PVPHR = PUPHZ0                                                    WLINEA46
      CALL PWINDR                                                       WLINEA47
       RETURN                                                           WLINEA48
       END                                                              WLINEA49




      SUBROUTINE WTIDE                                                  WTIDE  8
C     WIND VELOCITY MODEL                                               WTIDE  9
C     PROVIDES WIND FIELD OF THE ATMOSPHERIC TIDES BY ZONAL AND         WTIDE 10
C     MERIDONAL HEIGHT PROFILES THAT ARE SINUSOIDAL AND IN PHASE        WTIDE 11
C     QUADRATURE.                                                       WTIDE 12
C                                                                       WTIDE 13
      REAL LAMZ                                                         WTIDE 14
C     COMMON DECK "RKAM" INSERTED HERE                                  RKAMCOM2
      REAL KR,KTH,KPH                                                   RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)       RKAMCOM5
C     COMMON DECK "UU" INSERTED HERE                                    CUU    2
      REAL MODU                                                         CUU    4
      COMMON/UU/MODU(4)                                                 CUU    5
     1  ,V   ,PVT  ,PVR  ,PVTH  ,PVPH                                   CUU    6
     2  ,VR  ,PVRT ,PVRR ,PVRTH ,PVRPH                                  CUU    7
     3  ,VTH,PVTHT,PVTHR,PVTHTH,PVTHPH                                  CUU    8
     4  ,VPH,PVPHT,PVPHR,PVPHTH,PVPHPH                                  CUU    9
C     COMMON DECK "WW" INSERTED HERE                                    CWW    2
      PARAMETER (NWARSZ=1000)                                           CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
     8 (RCVRH,W(20)),                                                   CWW2   7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2   9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2  10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2  11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2  12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2  13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2  15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2  16
      REAL MMODEL,MFORM,MID                                             CWW3   2
C                                                                       CWW3   3
C     WIND          100-124                                             CWW3   4
```

```
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)           CWW3   5
C                                                                       CWW3   6
C     DELTA WIND          125-149                                       CWW3   7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)        CWW3   8
C                                                                       CWW3   9
C     SOUND SPEED 150-174                                               CWW3  10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)           CWW3  11
      EQUIVALENCE (W(153),REFC)                                         CWW3  12
C                                                                       CWW3  13
C     DELTA SOUND SPEED 175-199                                         CWW3  14
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)        CWW3  15
C                                                                       CWW3  16
C     TEMPERATURE    200-224                                            CWW3  17
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)           CWW3  18
C                                                                       CWW3  19
C     DELTA TEMPERATURE    225-249                                      CWW3  20
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)        CWW3  21
C                                                                       CWW3  22
C     MOLECULAR   250-274                                               CWW3  23
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)           CWW3  24
C                                                                       CWW3  25
C     RECEIVER HEIGHT   275-299                                         CWW3  26
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)           CWW3  27
C                                                                       CWW3  28
C     TOPOGRAPHY   300-324                                              CWW3  29
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)           CWW3  30
C                                                                       CWW3  31
C     DELTA TOPOGRAPHY   325-349                                        CWW3  32
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)        CWW3  33
C                                                                       CWW3  34
C     UPPER SURFACE TOPOGRAPHY   350-374                                CWW3  35
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)           CWW3  36
C     PLOT ENHANCEMENTS CONTROL PARAMETERS                              CWW3  37
C                                                                       CWW3  38
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)                       CWW3  39
C     ABSORPTION          500-524                                       CWW3  40
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)           CWW3  41
C                                                                       CWW3  42
C     DELTA ABSORPTION          525-549                                 CWW3  43
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)        CWW3  44
C                                                                       CWW3  45
C     PRESSURE          550-574                                         CWW3  46
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)           CWW3  47
C                                                                       CWW3  48
C     DELTA PRESSURE          575-599                                   CWW3  49
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)        CWW3  50
C                                                                       CWW3  51
C     COMMON DECK "B1" INSERTED HERE                                    CB1    2
      INTEGER UMX,UNTBL,UITBL,UFRMTBL,IDSU(10)                          CB1    4
      COMMON/B1/UMX,UNTBL(10),UITBL(10),UFRMTBL(10),UGP(10)             CB1    5
      EQUIVALENCE (UGP,IDSU)                                            CB1    6
C     COMMON DECK "CONST" INSERTED HERE                                 CCONST 2
      COMMON/PCONST/CREF,RGAS,GAMMA                                     CCONST 4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                         CCONST 5
      EQUIVALENCE (W(103),UPH0)                                         WTIDE 20
```

```
      EQUIVALENCE (W(104),UTH0),(W(105),LAMZ),(W(106),TTAU),(W(107),TAU)WTIDE 21
C                                                                       WTIDE 22
      DATA RECOGU/5.0/                                                  WTIDE 23
      DATA UMX/1/                                                       WDEBL  2
      DATA UNTBL/1,11,8*0/                                              WDEBL  3
      DATA UITBL/1,9*0/                                                 WDEBL  4
      DATA UFRMTBL/1,9*0/                                               WDEBL  5
C                                                                       WTIDE 26
C                                                                       WTIDE 27
      ENTRY IWINDR                                                      WTIDE 28
C                                                                       WTIDE 29
      IF(RECOGU .NE. UMODEL)                                            WTIDE 30
     1    CALL RERROR('SPEED  ','WRNG MODEL',RECOGU)                    WTIDE 31
C                                                                       WTIDE 32
      MODU(1)=5HWTIDE                                                   WTIDE 33
      MODU(2)=UID                                                       WTIDE 34
      CALL IPWINDR                                                      WTIDE 35
C                                                                       WTIDE 36
      Q=PID2/LAMZ                                                       WTIDE 37
      S=PID2/TAU                                                        WTIDE 38
      RETURN                                                            WTIDE 39
C                                                                       WTIDE 40
      ENTRY WINDR                                                       WTIDE 41
C                                                                       WTIDE 42
      CALL CLEAR(V,20)                                                  WTIDE 43
C                                                                       WTIDE 44
      ARG=PID2*((R-EARTHR)/LAMZ+TTAU)                                   WTIDE 45
C                                                                       WTIDE 46
      CSA=COS(ARG)                                                      WTIDE 47
      SSA=SIN(ARG)                                                      WTIDE 48
      VTH=UTH0*SSA                                                      WTIDE 49
      VPH=UPH0*CSA                                                      WTIDE 50
C                                                                       WTIDE 51
      PVTHR=CSA*Q*UTH0                                                  WTIDE 52
      PVPHR=SSA*Q*UPH0                                                  WTIDE 53
C      V=SQRT(VTH*VTH+VPH*VPH)                                          WTIDE 54
      PVTHT=CSA*S*UTH0                                                  WTIDE 55
      PVPHT=SSA*S*UPH0                                                  WTIDE 56
      CALL PWINDR                                                       WTIDE 57
       RETURN                                                           WTIDE 58
       END                                                              WTIDE 59
```

```
      SUBROUTINE ULOGZ2                                                 ULOGZ2 8
C     LOGARITHMIC WIND VELOCITY PROFILE                                 ULOGZ2 9
C     SINGULARITY IS AVOIDED BY A LINEAR TANGENT UNIQUELY               ULOGZ210
C     DETERMINED BY CONTINUITY OF VALUE AND SLOPE GOING THROUGH         ULOGZ211
C     THE ORIGIN.                                                       ULOGZ212
      REAL K                                                            ULOGZ213
      EQUIVALENCE (USTAR,W(103)),(K,W(104)),(Z0,W(105))                 ULOGZ214
C                                                                       ULOGZ215
C     COMMON DECK "RKAM" INSERTED HERE                                  RKAMCOM2
```

```
      REAL KR,KTH,KPH                                                   RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)       RKAMCOM5
C     COMMON DECK "UU" INSERTED HERE                                    CUU    2
      REAL MODU                                                         CUU    4
      COMMON/UU/MODU(4)                                                 CUU    5
     1  ,V  ,PVT  ,PVR  ,PVTH  ,PVPH                                    CUU    6
     2  ,VR ,PVRT ,PVRR ,PVRTH ,PVRPH                                   CUU    7
     3  ,VTH,PVTHT,PVTHR,PVTHTH,PVTHPH                                  CUU    8
     4  ,VPH,PVPHT,PVPHR,PVPHTH,PVPHPH                                  CUU    9
C     COMMON DECK "WW" INSERTED HERE                                    CWW    2
      PARAMETER (NWARSZ=1000)                                           CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
     8 (RCVRH,W(20)),                                                   CWW2   7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2   9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2  10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2  11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2  12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2  13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2  15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2  16
      REAL MMODEL,MFORM,MID                                             CWW3   2
C                                                                       CWW3   3
C     WIND          100-124                                             CWW3   4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)           CWW3   5
C                                                                       CWW3   6
C     DELTA WIND          125-149                                       CWW3   7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)        CWW3   8
C                                                                       CWW3   9
C     SOUND SPEED 150-174                                               CWW3  10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)           CWW3  11
      EQUIVALENCE (W(153),REFC)                                         CWW3  12
C                                                                       CWW3  13
C     DELTA SOUND SPEED 175-199                                         CWW3  14
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)        CWW3  15
C                                                                       CWW3  16
C     TEMPERATURE    200-224                                            CWW3  17
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)           CWW3  18
C                                                                       CWW3  19
C     DELTA TEMPERATURE    225-249                                      CWW3  20
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)        CWW3  21
C                                                                       CWW3  22
C     MOLECULAR   250-274                                               CWW3  23
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)           CWW3  24
C                                                                       CWW3  25
C     RECEIVER HEIGHT   275-299                                         CWW3  26
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)           CWW3  27
C                                                                       CWW3  28
C     TOPOGRAPHY   300-324                                              CWW3  29
```

```
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)          CWW3   30
C                                                                      CWW3   31
C     DELTA TOPOGRAPHY   325-349                                       CWW3   32
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)       CWW3   33
C                                                                      CWW3   34
C     UPPER SURFACE TOPOGRAPHY   350-374                               CWW3   35
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)          CWW3   36
C     PLOT ENHANCEMENTS CONTROL PARAMETERS                             CWW3   37
C                                                                      CWW3   38
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)                      CWW3   39
C     ABSORPTION        500-524                                        CWW3   40
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)          CWW3   41
C                                                                      CWW3   42
C     DELTA ABSORPTION        525-549                                  CWW3   43
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)       CWW3   44
C                                                                      CWW3   45
C     PRESSURE         550-574                                         CWW3   46
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)          CWW3   47
C                                                                      CWW3   48
C     DELTA PRESSURE        575-599                                    CWW3   49
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)       CWW3   50
C                                                                      CWW3   51
C     COMMON DECK "B1" INSERTED HERE                                   CB1     2
      INTEGER UMX,UNTBL,UITBL,UFRMTBL,IDSU(10)                         CB1     4
      COMMON/B1/UMX,UNTBL(10),UITBL(10),UFRMTBL(10),UGP(10)            CB1     5
      EQUIVALENCE (UGP,IDSU)                                           CB1     6
C     COMMON DECK "GG" INSERTED HERE                                   CGG     2
      REAL MODG                                                        CGG     4
      COMMON/GG/MODG(4)                                                CGG     5
      COMMON/GG/G,PGR,PGRR,PGRTH,PGRPH                                 CGG     6
      COMMON/GG/PGTH,PGPH,PGTHTH,PGPHPH,PGTHPH,GSELECT,GTIME           CGG     7
      DATA RECOGU/6.0/                                                 ULOGZ221
      DATA UMX/1/                                                      UGZ2BL 2
      DATA UNTBL/1,11,8*0/                                             UGZ2BL 3
      DATA UITBL/1,9*0/                                                UGZ2BL 4
      DATA UFRMTBL/1,9*0/                                              UGZ2BL 5
C                                                                      ULOGZ224
      ENTRY IWINDR                                                     ULOGZ225
C                                                                      ULOGZ226
      IF(RECOGU .NE. UMODEL)                                           ULOGZ227
     1    CALL RERROR('ULOGZ2 ','WRNG MODEL',RECOGU)                   ULOGZ228
C                                                                      ULOGZ229
      MODU(1)=6HULOGZ2                                                 ULOGZ230
      MODU(2)=UID                                                      ULOGZ231
C                                                                      ULOGZ232
      Z0E=Z0*EXP(1.0)                                                  ULOGZ233
      C=USTAR/K                                                        ULOGZ234
      CZ0E=C/Z0E                                                       ULOGZ235
      CALL IPWINDR                                                     ULOGZ236
C                                                                      ULOGZ237
      RETURN                                                           ULOGZ238
C                                                                      ULOGZ239
       ENTRY WINDR                                                     ULOGZ240
      CALL CLEAR(V,20)                                                 ULOGZ241
      CALL TOPOG                                                       ULOGZ242
```

```
      Z = G/PGR                                                         ULOGZ243
      IF(Z.GT.Z0E) GO TO 100                                            ULOGZ244
C                                                                       ULOGZ245
      VPH=CZ0E*Z                                                        ULOGZ246
      PVPHR=CZ0E*PGR                                                    ULOGZ247
      PVPHTH=CZ0E*PGTH                                                  ULOGZ248
      PVPHPH=CZ0E*PGPH                                                  ULOGZ249
      GO TO 120                                                         ULOGZ250
C                                                                       ULOGZ251
100   VPH=C*ALOG(Z/Z0)                                                  ULOGZ252
      CZ=C/Z                                                            ULOGZ253
      PVPHR=CZ*PGR                                                      ULOGZ254
      PVPHTH=CZ*PGTH                                                    ULOGZ255
      PVPHPH=CZ*PGPH                                                    ULOGZ256
C                                                                       ULOGZ257
120   V=ABS(VPH)                                                        ULOGZ258
      PVR=SIGN(PVPHR,VPH)                                               ULOGZ259
C                                                                       ULOGZ260
      CALL PWINDR                                                       ULOGZ261
       RETURN                                                           ULOGZ262
       END                                                              ULOGZ263
```

```
       SUBROUTINE VVORTX3                                               VVORTX39
C     WIND VELOCITY MODEL                                               VVORTX10
C     VERTICAL VORTEX WIND PERTURBATION WITH VISCOUS CORE AND           VVORTX11
C     MULTIPLIES VELOCITY FIELD BY A GUASSIAN HEIGHT PROFILE.           VVORTX12
C                                                                       VVORTX13
C     COMMON DECK "CONST" INSERTED HERE                                 CCONST 2
      COMMON/PCONST/CREF,RGAS,GAMMA                                     CCONST 4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                         CCONST 5
C     COMMON DECK "RKAM" INSERTED HERE                                  RKAMCOM2
      REAL KR,KTH,KPH                                                   RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)       RKAMCOM5
C     COMMON DECK "UU" INSERTED HERE                                    CUU     2
      REAL MODU                                                         CUU     4
      COMMON/UU/MODU(4)                                                 CUU     5
     1  ,V  ,PVT  ,PVR  ,PVTH  ,PVPH                                    CUU     6
     2  ,VR ,PVRT ,PVRR ,PVRTH ,PVRPH                                   CUU     7
     3  ,VTH,PVTHT,PVTHR,PVTHTH,PVTHPH                                  CUU     8
     4  ,VPH,PVPHT,PVPHR,PVPHTH,PVPHPH                                  CUU     9
C     COMMON DECK "WW" INSERTED HERE                                    CWW     2
      PARAMETER (NWARSZ=1000)                                           CWW1    3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1    4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2    2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2    3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2    4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2    5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2    6
     8 (RCVRH,W(20)),                                                   CWW2    7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2    8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2    9
```

```
     6 (HMIN,W(27)),(RGMAX,W(28)),
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))
      REAL MMODEL,MFORM,MID
C
C     WIND          100-124
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)
C
C     DELTA WIND          125-149
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)
C
C     SOUND SPEED 150-174
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)
      EQUIVALENCE (W(153),REFC)
C
C     DELTA SOUND SPEED 175-199
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)
C
C     TEMPERATURE    200-224
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)
C
C     DELTA TEMPERATURE    225-249
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)
C
C     MOLECULAR    250-274
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)
C
C     RECEIVER HEIGHT   275-299
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)
C
C     TOPOGRAPHY   300-324
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)
C
C     DELTA TOPOGRAPHY   325-349
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)
C
C     UPPER SURFACE TOPOGRAPHY   350-374
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)
C     PLOT ENHANCEMENTS CONTROL PARAMETERS
C
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)
C     ABSORPTION          500-524
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)
C
C     DELTA ABSORPTION          525-549
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)
C
C     PRESSURE          550-574
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)
C
C     DELTA PRESSURE          575-599
```

```
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)        CWW3   50
C                                                                       CWW3   51
       EQUIVALENCE (U0,W(103)),(R0,W(104)),(TH0,W(105)),(PH0,W(106))    VVORTX18
       EQUIVALENCE (HWIDTH,W(107)),(HVMAX,W(108))                       VVORTX19
C     COMMON DECK "B1" INSERTED HERE                                    CB1     2
      INTEGER UMX,UNTBL,UITBL,UFRMTBL,IDSU(10)                          CB1     4
      COMMON/B1/UMX,UNTBL(10),UITBL(10),UFRMTBL(10),UGP(10)             CB1     5
      EQUIVALENCE (UGP,IDSU)                                            CB1     6
      DATA RECOGU/9.0/                                                  VVORTX21
      DATA UMX/1/                                                       VRTX3BL2
      DATA UNTBL/1,11,8*0/                                              VRTX3BL3
      DATA UITBL/1,9*0/                                                 VRTX3BL4
      DATA UFRMTBL/1,9*0/                                               VRTX3BL5
C                                                                       VVORTX24
      ENTRY IWINDR                                                      VVORTX25
C                                                                       VVORTX26
      IF(RECOGU .NE. UMODEL)                                            VVORTX27
     1    CALL RERROR('SPEED   ','WRNG MODEL',RECOGU)                   VVORTX28
C                                                                       VVORTX29
      MODU(1)=7HVVORTX3                                                 VVORTX30
      MODU(2)=UID                                                       VVORTX31
       DENOM=0.0                                                        VVORTX32
       IF(HWIDTH.NE.0.0) DENOM=1.0/HWIDTH**2                            VVORTX33
      CALL IPWINDR                                                      VVORTX34
C                                                                       VVORTX35
      RETURN                                                            VVORTX36
C                                                                       VVORTX37
       ENTRY WINDR                                                      VVORTX38
      CALL CLEAR(V,20)                                                  VVORTX39
      DR=R-EARTHR-HVMAX                                                 VVORTX40
       DTH = TH - (PID2-TH0)                                            VVORTX41
       DPH = PH - PH0                                                   VVORTX42
       RAD2 = EARTHR * SQRT(DTH * DTH + DPH * DPH)                      VVORTX43
       A = 1.397                                                        VVORTX44
       B = - 1.26                                                       VVORTX45
      EXPO=RAD2/R0                                                      VVORTX46
      EXPO=B*EXPO*EXPO                                                  VVORTX47
      EXB=0.0                                                           VVORTX48
       IF(EXPO .GT. -675.0) EXB = EXP(EXPO)                             VVORTX49
      FX=1.-EXB                                                         VVORTX50
       DUM = A * EARTHR * U0 * R0 / RAD2 * * 2                          VVORTX51
      FZ=EXP(-DR*DR*DENOM)                                              VVORTX52
      DFDZ=-2.*DR*DENOM                                                 VVORTX53
      DUM=FZ*DUM                                                        VVORTX54
C                                                                       VVORTX55
       DUX = FX / RAD2 + RAD2 * B * EXB / R0 * * 2                      VVORTX56
       VTH = - DUM * FX * DPH                                           VVORTX57
       VPH = DUM * FX * DTH                                             VVORTX58
      V=SQRT(VTH*VTH + VPH*VPH)                                         VVORTX59
       DUM2=2.*DUM*EARTHR*EARTHR                                        VVORTX60
       PVTHTH = DUM2 * DTH * DPH / RAD2 * DUX                           VVORTX61
       PVPHPH = - PVTHTH                                                VVORTX62
       PVTHPH = DPH**2 * DUM2 / RAD2 * DUX -DUM*FX                      VVORTX63
       PVPHTH = - DTH**2* DUM2 / RAD2* DUX + DUM*FX                     VVORTX64
C                                                                       VVORTX65
```

```
      PVTH=(VTH*PVTHTH + VPH*PVPHTH)/V                                  VVORTX66
      PVPH=(VTH*PVTHPH + VPH*PVPHPH)/V                                  VVORTX67
C                                                                       VVORTX68
      PVTHR=VTH*DFDZ                                                    VVORTX69
      PVPHR=VPH*DFDZ                                                    VVORTX70
      PVR=(VTH*PVTHR+VPH*PVPHR)/V                                       VVORTX71
C                                                                       VVORTX72
      CALL PWINDR                                                       VVORTX73
       RETURN                                                           VVORTX74
       END                                                              VVORTX75


      SUBROUTINE WGAUSS2                                                WGAUSS28
C     WIND VELOCITY MODEL                                               WGAUSS29
C     EXPONENTIALLY DECAYING EFFECT IN ALL THREE DIRECTIONS.            WGAUSS10
C     COMMON DECK "CONST" INSERTED HERE                                 CCONST 2
      COMMON/PCONST/CREF,RGAS,GAMMA                                     CCONST 4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                         CCONST 5
C     COMMON DECK "RKAM" INSERTED HERE                                  RKAMCOM2
      REAL KR,KTH,KPH                                                   RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)       RKAMCOM5
C     COMMON DECK "UU" INSERTED HERE                                    CUU    2
      REAL MODU                                                         CUU    4
      COMMON/UU/MODU(4)                                                 CUU    5
     1  ,V   ,PVT   ,PVR   ,PVTH   ,PVPH                                CUU    6
     2  ,VR  ,PVRT  ,PVRR  ,PVRTH  ,PVRPH                               CUU    7
     3  ,VTH,PVTHT,PVTHR,PVTHTH,PVTHPH                                  CUU    8
     4  ,VPH,PVPHT,PVPHR,PVPHTH,PVPHPH                                  CUU    9
C     COMMON DECK "WW" INSERTED HERE                                    CWW    2
      PARAMETER (NWARSZ=1000)                                           CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
     8 (RCVRH,W(20)),                                                   CWW2   7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2   9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2  10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2  11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2  12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2  13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2  15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2  16
      REAL MMODEL,MFORM,MID                                             CWW3   2
C                                                                       CWW3   3
C     WIND          100-124                                             CWW3   4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)           CWW3   5
C                                                                       CWW3   6
C     DELTA WIND          125-149                                       CWW3   7
```

```
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)        CWW3    8
C                                                                       CWW3    9
C     SOUND SPEED 150-174                                               CWW3   10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)           CWW3   11
      EQUIVALENCE (W(153),REFC)                                         CWW3   12
C                                                                       CWW3   13
C     DELTA SOUND SPEED 175-199                                         CWW3   14
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)        CWW3   15
C                                                                       CWW3   16
C     TEMPERATURE    200-224                                            CWW3   17
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)           CWW3   18
C                                                                       CWW3   19
C     DELTA TEMPERATURE    225-249                                      CWW3   20
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)        CWW3   21
C                                                                       CWW3   22
C     MOLECULAR   250-274                                               CWW3   23
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)           CWW3   24
C                                                                       CWW3   25
C     RECEIVER HEIGHT   275-299                                         CWW3   26
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)           CWW3   27
C                                                                       CWW3   28
C     TOPOGRAPHY   300-324                                              CWW3   29
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)           CWW3   30
C                                                                       CWW3   31
C     DELTA TOPOGRAPHY   325-349                                        CWW3   32
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)        CWW3   33
C                                                                       CWW3   34
C     UPPER SURFACE TOPOGRAPHY   350-374                                CWW3   35
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)           CWW3   36
C     PLOT ENHANCEMENTS CONTROL PARAMETERS                              CWW3   37
C                                                                       CWW3   38
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)                       CWW3   39
C     ABSORPTION           500-524                                      CWW3   40
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)           CWW3   41
C                                                                       CWW3   42
C     DELTA ABSORPTION           525-549                                CWW3   43
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)        CWW3   44
C                                                                       CWW3   45
C     PRESSURE           550-574                                        CWW3   46
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)           CWW3   47
C                                                                       CWW3   48
C     DELTA PRESSURE           575-599                                  CWW3   49
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)        CWW3   50
C                                                                       CWW3   51
C     COMMON DECK "B1" INSERTED HERE                                    CB1     2
      INTEGER UMX,UNTBL,UITBL,UFRMTBL,IDSU(10)                          CB1     4
      COMMON/B1/UMX,UNTBL(10),UITBL(10),UFRMTBL(10),UGP(10)             CB1     5
      EQUIVALENCE (UGP,IDSU)                                            CB1     6
      EQUIVALENCE (UPH0,W(103)),(WH,W(104)),(WTH,W(105))                WGAUSS16
      EQUIVALENCE  (WPH,W(106)),(H0,W(107)),(WGTH0,W(108)),(PH0,W(109)) WGAUSS17
      DATA RECOGU/7.0/                                                  WGAUSS18
      DATA UMX/1/                                                       WUSS2L 2
      DATA UNTBL/1,11,8*0/                                              WUSS2L 3
      DATA UITBL/1,9*0/                                                 WUSS2L 4
      DATA UFRMTBL/1,9*0/                                               WUSS2L 5
```

```
C                                                                       WGAUSS21
      ENTRY IWINDR                                                      WGAUSS22
C                                                                       WGAUSS23
      IF(RECOGU .NE. UMODEL)                                            WGAUSS24
     1    CALL RERROR('SPEED   ','WRNG MODEL',RECOGU)                   WGAUSS25
C                                                                       WGAUSS26
      MODU(1)=7HWGAUSS2                                                 WGAUSS27
      MODU(2)=UID                                                       WGAUSS28
      CALL IPWINDR                                                      WGAUSS29
C                                                                       WGAUSS30
      WIDH=0.0                                                          WGAUSS31
      WIDTH=0.0                                                         WGAUSS32
      WIDPH=0.0                                                         WGAUSS33
      THO= PID2-WGTHO                                                   WGAUSS34
      IF(WH.NE.0.0) WIDH=1.0/WH                                         WGAUSS35
      IF(WTH.NE.0.0) WIDTH=1.0/WTH                                      WGAUSS36
      IF(WPH.NE.0.0) WIDPH=1.0/WPH                                      WGAUSS37
      RETURN                                                            WGAUSS38
C                                                                       WGAUSS39
       ENTRY WINDR                                                      WGAUSS40
      CALL CLEAR(V,20)                                                  WGAUSS41
       H = R - EARTHR                                                   WGAUSS42
      DFH=(H-HO)*WIDH                                                   WGAUSS43
      DFTH=(TH-THO)*WIDTH                                               WGAUSS44
      DFPH=(PH-PHO)*WIDPH                                               WGAUSS45
      EXPO=-(DFH*DFH+DFTH*DFTH+DFPH*DFPH)                               WGAUSS46
      EXPN=0.0                                                          WGAUSS47
      IF(EXPO.GT.-200.0) EXPN=EXP(EXPO)                                 WGAUSS48
       VPH = UPHO*EXPN                                                  WGAUSS49
       PVPHR = - 2. * VPH * DFH*WIDH                                    WGAUSS50
       PVPHTH = - 2. * VPH * DFTH*WIDTH                                 WGAUSS51
       PVPHPH = - 2. * VPH * DFPH*WIDPH                                 WGAUSS52
       END                                                              WGAUSS53
```

```
      SUBROUTINE NPWIND                                                 NPWIND 8
C     DO-NOTHING WIND-VELOCITY PERTURBATION MODEL                       NPWIND 9
C     COMMON DECK "UU" INSERTED HERE                                    CUU    2
      REAL MODU                                                         CUU    4
      COMMON/UU/MODU(4)                                                 CUU    5
     1  ,V  ,PVT  ,PVR  ,PVTH  ,PVPH                                    CUU    6
     2  ,VR ,PVRT ,PVRR ,PVRTH ,PVRPH                                   CUU    7
     3  ,VTH,PVTHT,PVTHR,PVTHTH,PVTHPH                                  CUU    8
     4  ,VPH,PVPHT,PVPHR,PVPHTH,PVPHPH                                  CUU    9
C     COMMON DECK "WW" INSERTED HERE                                    CWW    2
      PARAMETER (NWARSZ=1000)                                           CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
```

```
     8 (RCVRH,W(20)),
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),
     6 (HMIN,W(27)),(RGMAX,W(28)),
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))
      REAL MMODEL,MFORM,MID
C
C     WIND         100-124
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)
C
C     DELTA WIND          125-149
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)
C
C     SOUND SPEED 150-174
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)
      EQUIVALENCE (W(153),REFC)
C
C     DELTA SOUND SPEED 175-199
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)
C
C     TEMPERATURE    200-224
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)
C
C     DELTA TEMPERATURE    225-249
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)
C
C     MOLECULAR   250-274
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)
C
C     RECEIVER HEIGHT   275-299
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)
C
C     TOPOGRAPHY   300-324
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)
C
C     DELTA TOPOGRAPHY   325-349
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)
C
C     UPPER SURFACE TOPOGRAPHY   350-374
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)
C     PLOT ENHANCEMENTS CONTROL PARAMETERS
C
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)
C     ABSORPTION          500-524
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)
C
C     DELTA ABSORPTION          525-549
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)
C
C     PRESSURE         550-574
```

```
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)           CWW3   47
C                                                                       CWW3   48
C     DELTA PRESSURE        575-599                                     CWW3   49
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)        CWW3   50
C                                                                       CWW3   51
C     COMMON DECK "B2" INSERTED HERE                                    CB2     2
      INTEGER DUMX,DUNTBL,DUITBL,DUFRMTB,IDSDU(10)                      CB2     4
      COMMON/B2/DUMX,DUNTBL(10),DUITBL(10),DUFRMTB(10),DUGP(10)         CB2     5
      EQUIVALENCE (DUGP,IDSDU)                                          CB2     6
      DATA RECOGDU/0.0/                                                 NPWIND13
      DATA DUMX/1/                                                      NINDBL 2
      DATA DUNTBL/1,11,8*0/                                             NINDBL 3
      DATA DUITBL/1,9*0/                                                NINDBL 4
      DATA DUFRMTB/1,9*0/                                               NINDBL 5
C                                                                       NPWIND16
      ENTRY IPWINDR                                                     NPWIND17
      IF(RECOGDU .NE. DUMODEL)                                          NPWIND18
     1    CALL RERROR('DWINDR ','WRNG MODEL',RECOGDU)                   NPWIND19
      MODU(3)=6HNPWIND                                                  NPWIND20
      MODU(4)=DUID                                                      NPWIND21
      RETURN                                                            NPWIND22
C                                                                       NPWIND23
      ENTRY PWINDR                                                      NPWIND24
      RETURN                                                            NPWIND25
      END                                                               NPWIND26
```

```
      SUBROUTINE GAMRTDM                                                GAMRTDM8
C     SOUND SPEED MODEL, C**2=GAMMA*R*T/M                               GAMRTDM9
C     COMMON DECK "CC" INSERTED HERE                                    CCC     2
      REAL MODC                                                         CCC     4
      COMMON/CC/MODC(4),CS,PCST,PCSR,PCSTH,PCSPH                        CCC     5
C     COMMON DECK "TT" INSERTED HERE                                    CTT     2
      REAL MODT                                                         CTT     4
      COMMON/TT/MODT(4), T,PTT,PTR,PTTH,PTPH                            CTT     5
C     COMMON DECK "MM" INSERTED HERE                                    CMM     2
      REAL M,MODM                                                       CMM     4
      COMMON/MM/MODM(4),M,PMT,PMR,PMTH,PMPH                             CMM     5
C     COMMON DECK "CONST" INSERTED HERE                                 CCONST 2
      COMMON/PCONST/CREF,RGAS,GAMMA                                     CCONST 4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                         CCONST 5
C     COMMON DECK "WW" INSERTED HERE                                    CWW     2
      PARAMETER (NWARSZ=1000)                                           CWW1    3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1    4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2    2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2    3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2    4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2    5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2    6
     8 (RCVRH,W(20)),                                                   CWW2    7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2    8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2    9
```

```
     6 (HMIN,W(27)),(RGMAX,W(28)),
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))
      REAL MMODEL,MFORM,MID
C
C     WIND         100-124
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)
C
C     DELTA WIND         125-149
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)
C
C     SOUND SPEED 150-174
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)
      EQUIVALENCE (W(153),REFC)
C
C     DELTA SOUND SPEED 175-199
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)
C
C     TEMPERATURE    200-224
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)
C
C     DELTA TEMPERATURE     225-249
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)
C
C     MOLECULAR    250-274
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)
C
C     RECEIVER HEIGHT   275-299
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)
C
C     TOPOGRAPHY   300-324
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)
C
C     DELTA TOPOGRAPHY   325-349
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)
C
C     UPPER SURFACE TOPOGRAPHY   350-374
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)
C     PLOT ENHANCEMENTS CONTROL PARAMETERS
C
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)
C     ABSORPTION          500-524
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)
C
C     DELTA ABSORPTION          525-549
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)
C
C     PRESSURE         550-574
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)
C
C     DELTA PRESSURE          575-599
```

```
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)         CWW3  50
C                                                                        CWW3  51
C                                                                        GAMRTD15
C     COMMON DECK "B3" INSERTED HERE                                     CB3      2
      INTEGER CMX,CNTBL,CITBL,CFRMTBL,IDSC(10)                           CB3      4
      COMMON/B3/CMX,CNTBL(10),CITBL(10),CFRMTBL(10),CGP(512)             CB3      5
      EQUIVALENCE (CGP,IDSC),(ANC,CGP(11))                               CB3      6
      DATA CMX/1/                                                        GRTDMBL2
      DATA CNTBL/1,11,8*0/                                               GRTDMBL3
      DATA CITBL/1,9*0/                                                  GRTDMBL4
      DATA CFRMTBL/1,9*0/                                                GRTDMBL5
      DATA RECOGC/1.0/                                                   GAMRTD19
C                                                                        GAMRTD20
      ENTRY ISPEED                                                       GAMRTD21
C                                                                        GAMRTD22
      IF(REFC.GT.0.0) CREF=REFC                                          GAMRTD23
      IF(RECOGC .NE. CMODEL)                                             GAMRTD24
     1     CALL RERROR('SPEED   ','WRNG MODEL',RECOGC)                   GAMRTD25
      MODC(1)=7HGAMRTDM                                                  GAMRTD26
      MODC(2)=CID                                                        GAMRTD27
      CALL ITEMP                                                         GAMRTD28
      CALL IMOLWT                                                        GAMRTD29
      CALL IPSPEED                                                       GAMRTD30
C                                                                        GAMRTD31
      RETURN                                                             GAMRTD32
C                                                                        GAMRTD33
      ENTRY SPEED                                                        GAMRTD34
      CALL TEMP                                                          GAMRTD35
      CALL MOLWT                                                         GAMRTD36
      CS=GAMMA*RGAS*T/M                                                  GAMRTD37
      PCST=CS*(PTT/T - PMT/M)                                            GAMRTD38
      PCSR=CS*(PTR/T - PMR/M)                                            GAMRTD39
      PCSTH=CS*(PTTH/T - PMTH/M)                                         GAMRTD40
      PCSPH=CS*(PTPH/T - PMPH/M)                                         GAMRTD41
C                                                                        GAMRTD42
C                                                                        GAMRTD43
      CALL PSPEED                                                        GAMRTD44
      RETURN                                                             GAMRTD45
      END                                                                GAMRTD46
```

```
      SUBROUTINE CSTANH                                                  CSTANH10
C     SPEED PROFILE REPRESENTED BY A SEQUENCE OF LINEAR SEGMENTS         CSTANH11
C     SMOOTHLY JOINED BY HYPERBOLIC FUNCTIONS. PARAMETERS ARE INPUT      CSTANH12
C     AS TABULAR DATA WITH SLOPES COMPUTED FROM SPEED DATA.              CSTANH13
C                                                                        CSTANH14
      REAL ALC(20),Z(19),B(19),DL(19)                                    CSTANH15
C     COMMON DECK "RKAM" INSERTED HERE                                   RKAMCOM2
      REAL KR,KTH,KPH                                                    RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)        RKAMCOM5
C     COMMON DECK "B3" INSERTED HERE                                     CB3      2
      INTEGER CMX,CNTBL,CITBL,CFRMTBL,IDSC(10)                           CB3      4
```

```
      COMMON/B3/CMX,CNTBL(10),CITBL(10),CFRMTBL(10),CGP(512)             CB3     5
      EQUIVALENCE (CGP,IDSC),(ANC,CGP(11))                               CB3     6
C     COMMON DECK "CC" INSERTED HERE                                     CCC     2
      REAL MODC                                                          CCC     4
      COMMON/CC/MODC(4),CS,PCST,PCSR,PCSTH,PCSPH                         CCC     5
C     COMMON DECK "CONST" INSERTED HERE                                  CCONST  2
      COMMON/PCONST/CREF,RGAS,GAMMA                                      CCONST  4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                          CCONST  5
C     COMMON DECK "WW" INSERTED HERE                                     CWW     2
      PARAMETER (NWARSZ=1000)                                            CWW1    3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                    CWW1    4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                 CWW2    2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),     CWW2    3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),       CWW2    4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),           CWW2    5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),          CWW2    6
     8 (RCVRH,W(20)),                                                    CWW2    7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2    8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                       CWW2    9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                       CWW2   10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                      CWW2   11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),        CWW2   12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))         CWW2   13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),         CWW2   14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))               CWW2   15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                    CWW2   16
      REAL MMODEL,MFORM,MID                                              CWW3    2
C                                                                        CWW3    3
C     WIND          100-124                                              CWW3    4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)            CWW3    5
C                                                                        CWW3    6
C     DELTA WIND          125-149                                        CWW3    7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)         CWW3    8
C                                                                        CWW3    9
C     SOUND SPEED 150-174                                                CWW3   10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)            CWW3   11
      EQUIVALENCE (W(153),REFC)                                          CWW3   12
C                                                                        CWW3   13
C     DELTA SOUND SPEED 175-199                                          CWW3   14
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)         CWW3   15
C                                                                        CWW3   16
C     TEMPERATURE    200-224                                             CWW3   17
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)            CWW3   18
C                                                                        CWW3   19
C     DELTA TEMPERATURE    225-249                                       CWW3   20
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)         CWW3   21
C                                                                        CWW3   22
C     MOLECULAR   250-274                                                CWW3   23
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)            CWW3   24
C                                                                        CWW3   25
C     RECEIVER HEIGHT   275-299                                          CWW3   26
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)            CWW3   27
C                                                                        CWW3   28
C     TOPOGRAPHY   300-324                                               CWW3   29
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)            CWW3   30
```

```
C
C     DELTA TOPOGRAPHY   325-349
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)
C
C     UPPER SURFACE TOPOGRAPHY   350-374
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)
C     PLOT ENHANCEMENTS CONTROL PARAMETERS
C
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)
C     ABSORPTION        500-524
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)
C
C     DELTA ABSORPTION         525-549
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)
C
C     PRESSURE          550-574
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)
C
C     DELTA PRESSURE          575-599
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)
C
C
      EQUIVALENCE (Z0,CGP(12)),(CS0,CGP(32)),(DL0,CGP(52))
      EQUIVALENCE (Z,CGP(13)),(B,CGP(33)),(DL,CGP(53))
C
      DATA RECOGC,N/2.0,0/
      DATA PCST,PCSTH,PCSPH/3*0.0/
      DATA ANC/0.0/
      DATA CMX/2/
      DATA CNTBL/1,11,72,7*0/
      DATA CITBL/1,20,8*0/
      DATA CFRMTBL/1,2,8*0/
C
      ENTRY ISPEED
C
      IF(REFC.GT.0.0) CREF=REFC
C
      CALL IPSPEED
C
C     IF HAD PREVIOUS CALL BUT NOTHING THIS TIME, EXIT NOW
C     RETAINING PREVIOUS TABULAR DATA COUNT
      IF(N.GT.0 .AND. ANC.EQ.0.0) RETURN
C
      IF(RECOGC .NE. CMODEL)
     1    CALL RERROR('SPEED   ','WRNG MODEL',RECOGC)
      MODC(1)=6HCSTANH
      MODC(2)=CID
      N=ANC/3
      IF(ANC.NE.3*N.OR.N.LE.0)
     1    CALL RERROR('CSTANH','BAD NUMBER',ANC+2.0)
      N=N-2
      ANC=0.0
C
C
C     CONVERT 'C' ARRAY INPUT(OVERLAYS 'B' ARRAY) TO 'B' ARRAY
```

```
C                                                                       CSTANH51
      ZM1=Z0                                                            CSTANH52
      CS0=CS0*CS0                                                       CSTANH53
      CSM1=CS0                                                          CSTANH54
      NP1=N+1                                                           CSTANH55
      DO 10 I=1,NP1                                                     CSTANH56
         ZR=Z(I)                                                        CSTANH57
         ALC(I)=ALCOSH((ZR-Z0) / DL(I))                                 CSTANH58
         CS=B(I)**2                                                     CSTANH59
         B(I)=(CS-CSM1)/(ZR-ZM1)                                        CSTANH60
         ZM1=ZR                                                         CSTANH61
10       CSM1=CS                                                        CSTANH62
C                                                                       CSTANH63
      RETURN                                                            CSTANH64
C                                                                       CSTANH65
      ENTRY SPEED                                                       CSTANH66
C                                                                       CSTANH67
      IF(N.LE.0)                                                        CSTANH68
     1    CALL RERROR('CSTANH','BAD N VALUE',FLOAT(N))                  CSTANH69
C                                                                       CSTANH70
      ZR=R-EARTHR                                                       CSTANH71
      SUM = 0.                                                          CSTANH72
      PCSR=B(1)                                                         CSTANH73
      DO 1 I = 1, N                                                     CSTANH74
      SAV=0.5*(B(I+1)-B(I))                                             CSTANH75
      PCSR= PCSR+ SAV * (1.+TANH ((ZR-Z(I)) /DL(I)))                    CSTANH76
    1 SUM = SUM+DL(I) * SAV *(ALCOSH((ZR-Z(I))/DL(I))-ALC(I))           CSTANH77
      CS = CS0+SUM + 0.5*(B(1) + B(N + 1)) * (ZR-Z0)                    CSTANH78
C                                                                       CSTANH79
      CALL PSPEED                                                       CSTANH80
      RETURN                                                            CSTANH81
      END                                                               CSTANH82
```

```
      SUBROUTINE NPSPEED                                                NPSPEED8
C     DO-NOTHING SOUND SPEED PERTURBATION MODEL                         NPSPEED9
C     COMMON DECK "CC" INSERTED HERE                                    CCC      2
      REAL MODC                                                         CCC      4
      COMMON/CC/MODC(4),CS,PCST,PCSR,PCSTH,PCSPH                        CCC      5
C     COMMON DECK "WW" INSERTED HERE                                    CWW      2
      PARAMETER (NWARSZ=1000)                                           CWW1     3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1     4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2     2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2     3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2     4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2     5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2     6
     8 (RCVRH,W(20)),                                                   CWW2     7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2     8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2     9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2    10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2    11
```

```
      6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),
      7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))
      9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),
      1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))
      2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))
       REAL MMODEL,MFORM,MID
C
C      WIND         100-124
       EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)
C
C      DELTA WIND          125-149
       EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)
C
C      SOUND SPEED 150-174
       EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)
       EQUIVALENCE (W(153),REFC)
C
C      DELTA SOUND SPEED 175-199
       EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)
C
C      TEMPERATURE    200-224
       EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)
C
C      DELTA TEMPERATURE    225-249
       EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)
C
C      MOLECULAR   250-274
       EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)
C
C      RECEIVER HEIGHT   275-299
       EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)
C
C      TOPOGRAPHY   300-324
       EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)
C
C      DELTA TOPOGRAPHY   325-349
       EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)
C
C      UPPER SURFACE TOPOGRAPHY   350-374
       EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)
C      PLOT ENHANCEMENTS CONTROL PARAMETERS
C
       EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)
C      ABSORPTION          500-524
       EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)
C
C      DELTA ABSORPTION           525-549
       EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)
C
C      PRESSURE          550-574
       EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)
C
C      DELTA PRESSURE          575-599
       EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)
C
```

```
C                                                                      NPSPEE12
C     COMMON DECK "B2" INSERTED HERE                                   CB2     2
      INTEGER DUMX,DUNTBL,DUITBL,DUFRMTB,IDSDU(10)                     CB2     4
      COMMON/B2/DUMX,DUNTBL(10),DUITBL(10),DUFRMTB(10),DUGP(10)        CB2     5
      EQUIVALENCE (DUGP,IDSDU)                                         CB2     6
C                                                                      NPSPEE14
      DATA DUMX/1/                                                     NPEEDBL2
      DATA DUNTBL/1,11,8*0/                                            NPEEDBL3
      DATA DUITBL/1,9*0/                                               NPEEDBL4
      DATA DUFRMTB/1,9*0/                                              NPEEDBL5
C                                                                      NPSPEE17
      DATA RECOGDC/0.0/                                                NPSPEE18
C                                                                      NPSPEE19
      ENTRY IPSPEED                                                    NPSPEE20
      IF(RECOGDC .NE. DCMODEL)                                         NPSPEE21
     1    CALL RERROR('DSPEED ','WRNG MODEL',RECOGDC)                  NPSPEE22
      MODC(3)=7HNPSPEED                                                NPSPEE23
      MODC(4)=DCID                                                     NPSPEE24
      RETURN                                                           NPSPEE25
C                                                                      NPSPEE26
      ENTRY PSPEED                                                     NPSPEE27
      RETURN                                                           NPSPEE28
      END                                                              NPSPEE29
```

```
      SUBROUTINE CBLOB2                                                CBLOB2 8
C     SOUND SPEED PERTURBATION MODEL                                   CBLOB2 9
C     MULTIPLICATIVE PERTURBATION WITH EXPONENTIALLY DECAYING          CBLOB210
C     EFFECT IN ALL THREE DIRECTIONS. GIVE LATITUDE                    CBLOB211
C     INSTEAD OF CO-LATITUDE.                                          CBLOB212
C     COMMON DECK "CONST" INSERTED HERE                                CCONST 2
      COMMON/PCONST/CREF,RGAS,GAMMA                                    CCONST 4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                        CCONST 5
C     COMMON DECK "RKAM" INSERTED HERE                                 RKAMCOM2
      REAL KR,KTH,KPH                                                  RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)      RKAMCOM5
C     COMMON DECK "B4" INSERTED HERE                                   CB4     2
      INTEGER DCMX,DCNTBL,DCITBL,DCFRMTB,IDSDC(10)                     CB4     4
      COMMON/B4/DCMX,DCNTBL(10),DCITBL(10),DCFRMTB(10),DCGP(10)        CB4     5
      EQUIVALENCE (DCGP,IDSDC)                                         CB4     6
C                                                                      CBLOB216
C     COMMON DECK "CC" INSERTED HERE                                   CCC     2
      REAL MODC                                                        CCC     4
      COMMON/CC/MODC(4),CS,PCST,PCSR,PCSTH,PCSPH                       CCC     5
C     COMMON DECK "WW" INSERTED HERE                                   CWW     2
      PARAMETER (NWARSZ=1000)                                          CWW1    3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                  CWW1    4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                               CWW2    2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),   CWW2    3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),     CWW2    4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),         CWW2    5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),        CWW2    6
```

```
     8 (RCVRH,W(20)),                                                   CWW2   7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2   9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2  10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2  11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2  12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2  13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2  15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2  16
      REAL MMODEL,MFORM,MID                                             CWW3   2
C                                                                       CWW3   3
C     WIND          100-124                                             CWW3   4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)           CWW3   5
C                                                                       CWW3   6
C     DELTA WIND          125-149                                       CWW3   7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)        CWW3   8
C                                                                       CWW3   9
C     SOUND SPEED 150-174                                               CWW3  10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)           CWW3  11
      EQUIVALENCE (W(153),REFC)                                         CWW3  12
C                                                                       CWW3  13
C     DELTA SOUND SPEED 175-199                                         CWW3  14
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)        CWW3  15
C                                                                       CWW3  16
C     TEMPERATURE    200-224                                            CWW3  17
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)           CWW3  18
C                                                                       CWW3  19
C     DELTA TEMPERATURE    225-249                                      CWW3  20
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)        CWW3  21
C                                                                       CWW3  22
C     MOLECULAR   250-274                                               CWW3  23
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)           CWW3  24
C                                                                       CWW3  25
C     RECEIVER HEIGHT   275-299                                         CWW3  26
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)           CWW3  27
C                                                                       CWW3  28
C     TOPOGRAPHY   300-324                                              CWW3  29
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)           CWW3  30
C                                                                       CWW3  31
C     DELTA TOPOGRAPHY   325-349                                        CWW3  32
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)        CWW3  33
C                                                                       CWW3  34
C     UPPER SURFACE TOPOGRAPHY   350-374                                CWW3  35
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)           CWW3  36
C     PLOT ENHANCEMENTS CONTROL PARAMETERS                              CWW3  37
C                                                                       CWW3  38
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)                       CWW3  39
C     ABSORPTION           500-524                                      CWW3  40
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)           CWW3  41
C                                                                       CWW3  42
C     DELTA ABSORPTION           525-549                                CWW3  43
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)        CWW3  44
C                                                                       CWW3  45
C     PRESSURE           550-574                                        CWW3  46
```

```
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)           CWW3   47
C                                                                       CWW3   48
C     DELTA PRESSURE        575-599                                     CWW3   49
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)        CWW3   50
C                                                                       CWW3   51
C                                                                       CBLOB219
      EQUIVALENCE (C0,W(178)), (Z0,W(179)),(CBTH0,W(180))               CBLOB220
      EQUIVALENCE (PH0,W(181)),(WZ,W(182)),(WTH,W(183)),(WPH,W(184))    CBLOB221
      DATA RECOGDC/2.0/                                                 CBLOB222
      DATA DCMX/1/                                                      COB2BL 2
      DATA DCNTBL/1,11,8*0/                                             COB2BL 3
      DATA DCITBL/1,9*0/                                                COB2BL 4
      DATA DCFRMTB/1,9*0/                                               COB2BL 5
C                                                                       CBLOB225
      ENTRY IPSPEED                                                     CBLOB226
      IF(RECOGDC .NE. DCMODEL)                                          CBLOB227
     1    CALL RERROR('DSPEED ','WRNG MODEL',RECOGDC)                   CBLOB228
C                                                                       CBLOB229
      MODC(3)=6HCBLOB2                                                  CBLOB230
      MODC(4)=DCID                                                      CBLOB231
C                                                                       CBLOB232
      FWZ=0.0                                                           CBLOB233
      FWTH=0.0                                                          CBLOB234
      FWPH=0.0                                                          CBLOB235
      TH0= PID2-CBTH0                                                   CBLOB236
      IF(WZ.NE.0.0) FWZ=2.0/WZ/WZ                                       CBLOB237
      IF(WTH.NE.0.0) FWTH=2.0/WTH/WTH                                   CBLOB238
      IF(WPH.NE.0.0) FWPH=2.0/WPH/WPH                                   CBLOB239
      RETURN                                                            CBLOB240
C                                                                       CBLOB241
      ENTRY PSPEED                                                      CBLOB242
C                                                                       CBLOB243
      IF(C0.EQ.0.0) RETURN                                              CBLOB244
C                                                                       CBLOB245
      DZ=R-EARTHR-Z0                                                    CBLOB246
      DTH=TH-TH0                                                        CBLOB247
      DPH=PH-PH0                                                        CBLOB248
      DEXPO=0.0                                                         CBLOB249
      EXPO=-0.5*(DZ*DZ*FWZ+DTH*DTH*FWTH+DPH*DPH*FWPH)                   CBLOB250
      IF(EXPO .GT. -200.0) DEXPO=C0*EXP(EXPO)                           CBLOB251
      DEL=1.0+DEXPO                                                     CBLOB252
C                                                                       CBLOB253
      PCSR=PCSR*DEL-CS*DEXPO*FWZ*DZ                                     CBLOB254
      PCSTH=PCSTH*DEL-CS*DEXPO*FWTH*DTH                                 CBLOB255
      PCSPH=PCSPH*DEL-CS*DEXPO*FWPH*DPH                                 CBLOB256
      CS=CS*DEL                                                         CBLOB257
      RETURN                                                            CBLOB258
      END                                                               CBLOB259


      SUBROUTINE TLINEAR                                                TLINEAR8
C     LINEAR TEMPERATURE PROFILE                                        TLINEAR9
```

```
C     COMMON DECK "RKAM" INSERTED HERE                                  RKAMCOM2
      REAL KR,KTH,KPH                                                   RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)       RKAMCOM5
C     COMMON DECK "MM" INSERTED HERE                                    CMM    2
      REAL M,MODM                                                       CMM    4
      COMMON/MM/MODM(4),M,PMT,PMR,PMTH,PMPH                             CMM    5
C     COMMON DECK "TT" INSERTED HERE                                    CTT    2
      REAL MODT                                                         CTT    4
      COMMON/TT/MODT(4), T,PTT,PTR,PTTH,PTPH                            CTT    5
C     COMMON DECK "WW" INSERTED HERE                                    CWW    2
      PARAMETER (NWARSZ=1000)                                           CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
     8 (RCVRH,W(20)),                                                   CWW2   7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2   9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2  10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2  11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2  12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2  13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2  15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2  16
      REAL MMODEL,MFORM,MID                                             CWW3   2
C                                                                       CWW3   3
C     WIND         100-124                                              CWW3   4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)           CWW3   5
C                                                                       CWW3   6
C     DELTA WIND          125-149                                       CWW3   7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)        CWW3   8
C                                                                       CWW3   9
C     SOUND SPEED 150-174                                               CWW3  10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)           CWW3  11
      EQUIVALENCE (W(153),REFC)                                         CWW3  12
C                                                                       CWW3  13
C     DELTA SOUND SPEED 175-199                                         CWW3  14
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)        CWW3  15
C                                                                       CWW3  16
C     TEMPERATURE    200-224                                            CWW3  17
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)           CWW3  18
C                                                                       CWW3  19
C     DELTA TEMPERATURE    225-249                                      CWW3  20
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)        CWW3  21
C                                                                       CWW3  22
C     MOLECULAR   250-274                                               CWW3  23
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)           CWW3  24
C                                                                       CWW3  25
C     RECEIVER HEIGHT   275-299                                         CWW3  26
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)           CWW3  27
C                                                                       CWW3  28
C     TOPOGRAPHY   300-324                                              CWW3  29
```

```
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)
C
C     DELTA TOPOGRAPHY   325-349
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)
C
C     UPPER SURFACE TOPOGRAPHY   350-374
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)
C     PLOT ENHANCEMENTS CONTROL PARAMETERS
C
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)
C     ABSORPTION          500-524
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)
C
C     DELTA ABSORPTION          525-549
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)
C
C     PRESSURE          550-574
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)
C
C     DELTA PRESSURE          575-599
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)
C
       EQUIVALENCE (TGND, W(203)), (A, W(204))
C     COMMON DECK "B5" INSERTED HERE
      INTEGER TMX,TNTBL,TITBL,TFRMTBL,IDST(10)
      COMMON/B5/TMX,TNTBL(10),TITBL(10),TFRMTBL(10),TGP(262)
      EQUIVALENCE (TGP,IDST),(ANT,TGP(11))
      DATA RECOGT/1.0/
      DATA ANT/0.0/
      DATA TMX/1/
      DATA TNTBL/1,11,8*0/
      DATA TITBL/1,9*0/
      DATA TFRMTBL/1,9*0/
      ENTRY ITEMP
C
      IF(RECOGT .NE. TMODEL)
     1    CALL RERROR('TEMP    ','WRNG MODEL',RECOGT)
C
      MODT(1)=7HTLINEAR
      MODT(2)=TID
      CALL IPTEMP
C
      RETURN
C
       ENTRY TEMP
       H = R - EARTHR
       T = TGND + A * H
      CALL CLEAR(PTT,4)
      PTR=A
C
      CALL PTEMP
       RETURN
       END
```

```
      SUBROUTINE TTANH5                                                 TTANH5 9
C     TEMPERATURE PROFILE REPRESENTED BY A SEQUENCE OF LINEAR SEGMENTS  TTANH510
C     SMOOTHLY JOINED BY HYPERBOLIC FUNCTIONS. PARAMETERS ARE INPUT     TTANH511
C     AS TABULAR DATA WITH SLOPES COMPUTED FROM TEMPERATURE DATA.       TTANH512
C     REFERENCE TEMPERATURE T0 IS READ FROM TABULAR DATA.               TTANH513
      DIMENSION C(20), TM(19), Z(19), DL(19)                            TTANH514
C     COMMON DECK "RKAM" INSERTED HERE                                  RKAMCOM2
      REAL KR,KTH,KPH                                                   RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)       RKAMCOM5
C     COMMON DECK "TT" INSERTED HERE                                    CTT    2
      REAL MODT                                                         CTT    4
      COMMON/TT/MODT(4), T,PTT,PTR,PTTH,PTPH                            CTT    5
C     COMMON DECK "WW" INSERTED HERE                                    CWW    2
      PARAMETER (NWARSZ=1000)                                           CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
     8 (RCVRH,W(20)),                                                   CWW2   7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2   9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2  10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2  11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2  12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2  13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2  15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2  16
      REAL MMODEL,MFORM,MID                                             CWW3   2
C                                                                       CWW3   3
C     WIND          100-124                                             CWW3   4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)           CWW3   5
C                                                                       CWW3   6
C     DELTA WIND          125-149                                       CWW3   7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)        CWW3   8
C                                                                       CWW3   9
C     SOUND SPEED 150-174                                               CWW3  10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)           CWW3  11
      EQUIVALENCE (W(153),REFC)                                         CWW3  12
C                                                                       CWW3  13
C     DELTA SOUND SPEED 175-199                                         CWW3  14
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)        CWW3  15
C                                                                       CWW3  16
C     TEMPERATURE     200-224                                           CWW3  17
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)           CWW3  18
C                                                                       CWW3  19
C     DELTA TEMPERATURE     225-249                                     CWW3  20
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)        CWW3  21
C                                                                       CWW3  22
C     MOLECULAR   250-274                                               CWW3  23
```

```
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)           CWW3   24
C                                                                       CWW3   25
C     RECEIVER HEIGHT   275-299                                         CWW3   26
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)           CWW3   27
C                                                                       CWW3   28
C     TOPOGRAPHY   300-324                                              CWW3   29
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)           CWW3   30
C                                                                       CWW3   31
C     DELTA TOPOGRAPHY   325-349                                        CWW3   32
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)        CWW3   33
C                                                                       CWW3   34
C     UPPER SURFACE TOPOGRAPHY   350-374                                CWW3   35
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)           CWW3   36
C     PLOT ENHANCEMENTS CONTROL PARAMETERS                              CWW3   37
C                                                                       CWW3   38
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)                       CWW3   39
C     ABSORPTION          500-524                                       CWW3   40
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)           CWW3   41
C                                                                       CWW3   42
C     DELTA ABSORPTION           525-549                                CWW3   43
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)        CWW3   44
C                                                                       CWW3   45
C     PRESSURE           550-574                                        CWW3   46
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)           CWW3   47
C                                                                       CWW3   48
C     DELTA PRESSURE           575-599                                  CWW3   49
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)        CWW3   50
C                                                                       CWW3   51
C     COMMON DECK "B5" INSERTED HERE                                    CB5     2
      INTEGER TMX,TNTBL,TITBL,TFRMTBL,IDST(10)                          CB5     4
      COMMON/B5/TMX,TNTBL(10),TITBL(10),TFRMTBL(10),TGP(262)            CB5     5
      EQUIVALENCE (TGP,IDST),(ANT,TGP(11))                              CB5     6
      EQUIVALENCE                (Z0,TGP(12)),(TM,TGP(33))              TTANH519
      EQUIVALENCE (Z,TGP(13)),(C,TGP(32)),(DL,TGP(53))                  TTANH520
C                                                                       TTANH521
      DATA RECOGT,N/7.0,0/                                              TTANH522
      DATA ANT/0.0/                                                     TNH5BL 2
      DATA TMX/2/                                                       TNH5BL 3
      DATA TNTBL/1,11,72,7*0/                                           TNH5BL 4
      DATA TITBL/1,20,8*0/                                              TNH5BL 5
      DATA TFRMTBL/1,2,8*0/                                             TNH5BL 6
C                                                                       TTANH525
      COSH (X) = (EXP (X) + 1. / (EXP (X))) / 2.                        TTANH526
C                                                                       TTANH527
      ENTRY ITEMP                                                       TTANH528
C                                                                       TTANH529
      CALL IPTEMP                                                       TTANH530
C                                                                       TTANH531
C     IF HAD PREVIOUS CALL BUT NOTHING THIS TIME, EXIT NOW              TTANH532
C     RETAINING PREVIOUS TABULAR DATA COUNT                             TTANH533
      IF(N.GT.0 .AND. ANT.EQ.0.0) RETURN                                TTANH534
C                                                                       TTANH535
      IF(RECOGT .NE. TMODEL)                                            TTANH536
     1    CALL RERROR('TEMP    ','WRNG MODEL',RECOGT)                   TTANH537
C                                                                       TTANH538
```

```
      MODT(1)=7HTTANH5
      MODT(2)=TID
C
      N=(ANT+1)/3 - 2
C
      IF(N.LE.0)
     1    CALL RERROR('TTANH5','BAD N VALUE',FLOAT(N))
C
      ANT=0.0
C
C
C     CONVERT 'T' ARRAY INPUT(OVERLAYS 'C' ARRAY) TO 'C' ARRAY
C
      T0=C(1)
      TIM1=T0
      ZIM1=0.0
      NP1=N+1
      DO 10 I=1,NP1
         TI=TM(I)
         ZI=Z(I)
         C(I)=(TI-TIM1)/(ZI-ZIM1)
         TIM1=TI
10       ZIM1=ZI
C
      RETURN
C
      ENTRY TEMP
      H = R - EARTHR
      SUM = 0.
C
C     LOOP TO SUM OVER ALL COEFFICIENTS
C     USE SPECIAL FUNCTION 'ALCOSH' WHICH ALLOWS FOR LARGE ARGUMENTS.
      DO 1 I = 1, N
    1 SUM = SUM + DL(I) * (C(I + 1) - C(I)) / 2. *(ALCOSH((H - Z
     1(I)) / DL(I)) - ALCOSH((Z(I)-Z0) / DL(I)))
      T = T0 + SUM + (C(1) + C(N + 1)) * (H - Z0) * 0.5
      SUM = 0.
      DO 2 I = 1, N
    2 SUM = SUM + (C(I + 1) - C(I)) / 2. * (1. + TANH ((H - Z(I)) /DL
     1 (I)))
      PTT=0.0
      PTR = C(1) + SUM
      PTTH=0.0
      PTPH=0.0
C
      CALL PTEMP
      RETURN
      END


      SUBROUTINE TTABLE
C     TABULAR TEMPERATURE PROFILE THAT MAKES A CUBIC INTERPOLATION
```

```
C     BETWEEN POINTS TO INSURE A CONTINUOUS TERMPERATURE GRADIENT        TTABLE10
      DIMENSION HPC(250),FN2C(250),ALPHA(250),TTBETA(250),GAMM(250),     TTABLE11
     1 DELTA(250),SLOPE(250),MAT(4,5)                                    TTABLE12
      REAL MAT                                                           TTABLE13
C     COMMON DECK "CONST" INSERTED HERE                                  CCONST 2
      COMMON/PCONST/CREF,RGAS,GAMMA                                      CCONST 4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                          CCONST 5
C     COMMON DECK "RKAM" INSERTED HERE                                   RKAMCOM2
      REAL KR,KTH,KPH                                                    RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)        RKAMCOM5
C     COMMON DECK "TT" INSERTED HERE                                     CTT     2
      REAL MODT                                                          CTT     4
      COMMON/TT/MODT(4), T,PTT,PTR,PTTH,PTPH                             CTT     5
C     COMMON DECK "WW" INSERTED HERE                                     CWW     2
      PARAMETER (NWARSZ=1000)                                            CWW1    3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                    CWW1    4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                 CWW2    2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),     CWW2    3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),       CWW2    4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),           CWW2    5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),          CWW2    6
     8 (RCVRH,W(20)),                                                    CWW2    7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25)) CWW2    8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                       CWW2    9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                       CWW2   10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                      CWW2   11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),        CWW2   12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))         CWW2   13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),         CWW2   14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))               CWW2   15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                    CWW2   16
      REAL MMODEL,MFORM,MID                                              CWW3    2
C                                                                        CWW3    3
C     WIND          100-124                                              CWW3    4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)            CWW3    5
C                                                                        CWW3    6
C     DELTA WIND          125-149                                        CWW3    7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)         CWW3    8
C                                                                        CWW3    9
C     SOUND SPEED 150-174                                                CWW3   10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)            CWW3   11
      EQUIVALENCE (W(153),REFC)                                          CWW3   12
C                                                                        CWW3   13
C     DELTA SOUND SPEED 175-199                                          CWW3   14
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)         CWW3   15
C                                                                        CWW3   16
C     TEMPERATURE    200-224                                             CWW3   17
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)            CWW3   18
C                                                                        CWW3   19
C     DELTA TEMPERATURE    225-249                                       CWW3   20
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)         CWW3   21
C                                                                        CWW3   22
C     MOLECULAR   250-274                                                CWW3   23
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)            CWW3   24
C                                                                        CWW3   25
```

```
C     RECEIVER HEIGHT   275-299                                          CWW3  26
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)            CWW3  27
C                                                                        CWW3  28
C     TOPOGRAPHY   300-324                                               CWW3  29
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)            CWW3  30
C                                                                        CWW3  31
C     DELTA TOPOGRAPHY   325-349                                         CWW3  32
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)         CWW3  33
C                                                                        CWW3  34
C     UPPER SURFACE TOPOGRAPHY   350-374                                 CWW3  35
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)            CWW3  36
C     PLOT ENHANCEMENTS CONTROL PARAMETERS                               CWW3  37
C                                                                        CWW3  38
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)                        CWW3  39
C     ABSORPTION         500-524                                         CWW3  40
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)            CWW3  41
C                                                                        CWW3  42
C     DELTA ABSORPTION         525-549                                   CWW3  43
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)         CWW3  44
C                                                                        CWW3  45
C     PRESSURE         550-574                                           CWW3  46
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)            CWW3  47
C                                                                        CWW3  48
C     DELTA PRESSURE         575-599                                     CWW3  49
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)         CWW3  50
C                                                                        CWW3  51
C     COMMON DECK "B5" INSERTED HERE                                     CB5    2
      INTEGER TMX,TNTBL,TITBL,TFRMTBL,IDST(10)                           CB5    4
      COMMON/B5/TMX,TNTBL(10),TITBL(10),TFRMTBL(10),TGP(262)             CB5    5
      EQUIVALENCE (TGP,IDST),(ANT,TGP(11))                               CB5    6
      EQUIVALENCE (AN,TGP(11)),(HPC,TGP(12)),(FN2C,TGP(262))             TTABLE19
C                                                                        TTABLE20
      DATA RECOGT,NOC/6.0,0/                                             TTABLE21
      DATA ANT/0.0/                                                      TTLEBL 2
      DATA TMX/2/                                                        TTLEBL 3
      DATA TNTBL/1,11,512,7*0/                                           TTLEBL 4
      DATA TITBL/1,250,8*0/                                              TTLEBL 5
      DATA TFRMTBL/1,2,8*0/                                              TTLEBL 6
C                                                                        TTABLE24
      ENTRY ITEMP                                                        TTABLE25
C                                                                        TTABLE26
C                                                                        TTABLE27
      CALL IPTEMP                                                        TTABLE28
C                                                                        TTABLE29
C     IF HAD PREVIOUS CALL BUT NOTHING THIS TIME, EXIT NOW               TTABLE30
C     RETAINING PREVIOUS TABULAR DATA COUNT                              TTABLE31
      IF(NOC.GT.0 .AND. ANT.LE.0.0) RETURN                               TTABLE32
C                                                                        TTABLE33
      IF(RECOGT.NE.TMODEL)                                               TTABLE34
     1    CALL RERROR('TEMP   ','WRNG MODEL',RECOGT)                     TTABLE35
C                                                                        TTABLE36
      NOC=ANT/2                                                          TTABLE37
      IF(ANT.NE.2*NOC .OR. NOC.LE.1)                                     TTABLE38
     1     CALL RERROR('TTABLE','BAD NUMBER',ANT)                        TTABLE39
      ANT=0.0                                                            TTABLE40
```

```
C
      MODT(1)=6HTTABLE
      MODT(2)=TID
C
C
      SLOPE(1)=(FN2C(2)-FN2C(1))/(HPC(2)-HPC(1))
      SLOPE(NOC)=0.
      NMAX=1
      DO 6 I=2,NOC
      IF (FN2C(I).GT.FN2C(NMAX)) NMAX=I
      IF (I.EQ.NOC) GO TO 4
      DO 3 J=1,3
      M=I+J-2
      MAT(J,1)=1.
      MAT(J,2)=HPC(M)
      MAT(J,3)=HPC(M)**2
    3 MAT(J,4)=FN2C(M)
      CALL GAUSEL (MAT,4,3,4,NRANK)
      IF (NRANK.LT.3) GO TO 60
      SLOPE(I)=MAT(2,4)+2.*MAT(3,4)*HPC(I)
    4 DO 5 J=1,2
      M=I+J-2
      MAT(J,1)=1.
      MAT(J,2)=HPC(M)
      MAT(J,3)=HPC(M)**2
      MAT(J,4)=HPC(M)**3
      MAT(J,5)=FN2C(M)
      L=J+2
      MAT(L,1)=0.
      MAT(L,2)=1.
      MAT(L,3)=2.*HPC(M)
      MAT(L,4)=3.*HPC(M)**2
    5 MAT(L,5)=SLOPE(M)
      CALL GAUSEL (MAT,4,4,5,NRANK)
      IF (NRANK.LT.4) GO TO 60
      ALPHA(I)=MAT(1,5)
      TTBETA(I)=MAT(2,5)
      GAMM(I)=MAT(3,5)
    6 DELTA(I)=MAT(4,5)
      HMAX=HPC(NMAX)
      NH=2
C
      RETURN
C
   60 PRINT 6000, I,HPC(I)
 6000 FORMAT(' THE',I4,'TH POINT IN THE TEMPERATURE PROFILE HAS'
     1,' THE HEIGHT',F8.2,' KM, WHICH IS THE SAME AS ANOTHER POINT.')
      CALL EXIT
C
      ENTRY TEMP
C
C
      IF(NOC.LE.0)
     1    CALL RERROR('TTABLE','BAD N VALUE',FLOAT(NOC))
C
```

```
      H=R-EARTHR                                                        TTABLE96
      PTT=0.0                                                           TTABLE97
      PTR=0.0                                                           TTABLE98
      PTTH=0.0                                                          TTABLE99
      PTPH=0.0                                                          TTABL100
      IF (H.GE.HPC(1)) GO TO 12                                         TTABL101
   11 NH=2                                                              TTABL102
      T=FN2C(1)+SLOPE(1)*(H-HPC(1))                                     TTABL103
      PTR=SLOPE(1)                                                      TTABL104
      RETURN                                                            TTABL105
   12 IF (H.GE.HPC(NOC)) GO TO 18                                       TTABL106
      NSTEP=1                                                           TTABL107
      IF (H.LT.HPC(NH-1)) NSTEP=-1                                      TTABL108
   15 IF (HPC(NH-1).LE.H.AND.H.LT.HPC(NH)) GO TO 16                     TTABL109
      NH=NH+NSTEP                                                       TTABL110
      GO TO 15                                                          TTABL111
   16 T=(ALPHA(NH)+H*(TTBETA(NH)+H*(GAMM(NH)+H*DELTA(NH))))             TTABL112
      PTR=(TTBETA(NH)+H*(2.*GAMM(NH)+H*3.*DELTA(NH)))                   TTABL113
      RETURN                                                            TTABL114
   18 T=FN2C(NOC)                                                       TTABL115
C                                                                       TTABL116
      CALL PTEMP                                                        TTABL117
      RETURN                                                            TTABL118
         END                                                            TTABL119
```

```
      SUBROUTINE NTEMP                                                  NTEMP  9
C     DO-NOTHING TEMPERATURE MODEL                                      NTEMP 10
C     COMMON DECK "TT" INSERTED HERE                                    CTT    2
      REAL MODT                                                         CTT    4
      COMMON/TT/MODT(4), T,PTT,PTR,PTTH,PTPH                            CTT    5
C     COMMON DECK "WW" INSERTED HERE                                    CWW    2
      PARAMETER (NWARSZ=1000)                                           CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
     8 (RCVRH,W(20)),                                                   CWW2   7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2   9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2  10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2  11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2  12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2  13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2  15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2  16
      REAL MMODEL,MFORM,MID                                             CWW3   2
C                                                                       CWW3   3
C     WIND          100-124                                             CWW3   4
```

```
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)
C
C     DELTA WIND        125-149
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)
C
C     SOUND SPEED 150-174
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)
      EQUIVALENCE (W(153),REFC)
C
C     DELTA SOUND SPEED 175-199
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)
C
C     TEMPERATURE    200-224
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)
C
C     DELTA TEMPERATURE    225-249
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)
C
C     MOLECULAR   250-274
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)
C
C     RECEIVER HEIGHT   275-299
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)
C
C     TOPOGRAPHY   300-324
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)
C
C     DELTA TOPOGRAPHY   325-349
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)
C
C     UPPER SURFACE TOPOGRAPHY   350-374
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)
C     PLOT ENHANCEMENTS CONTROL PARAMETERS
C
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)
C     ABSORPTION          500-524
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)
C
C     DELTA ABSORPTION          525-549
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)
C
C     PRESSURE          550-574
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)
C
C     DELTA PRESSURE          575-599
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)
C
C     COMMON DECK "B5" INSERTED HERE
      INTEGER TMX,TNTBL,TITBL,TFRMTBL,IDST(10)
      COMMON/B5/TMX,TNTBL(10),TITBL(10),TFRMTBL(10),TGP(262)
      EQUIVALENCE (TGP,IDST),(ANT,TGP(11))
C
      DATA RECOGT/0.0/
      DATA TMX/1/
      DATA TNTBL/1,11,8*0/
```

```
      DATA TITBL/1,9*0/                                                 NMPBL  4
      DATA TFRMTBL/1,9*0/                                               NMPBL  5
C                                                                       NTEMP 18
      ENTRY ITEMP                                                       NTEMP 19
      IF(RECOGT .NE. TMODEL)                                            NTEMP 20
     1    CALL RERROR('TEMP   ','WRNG MODEL',RECOGT)                    NTEMP 21
C                                                                       NTEMP 22
      MODT(1)=5HNTEMP                                                   NTEMP 23
      MODT(2)=DTID                                                      NTEMP 24
      RETURN                                                            NTEMP 25
C                                                                       NTEMP 26
      ENTRY TEMP                                                        NTEMP 27
      RETURN                                                            NTEMP 28
      END                                                               NTEMP 29



      SUBROUTINE TBLOB2                                                 TBLOB2 8
C     TEMPERATURE PERTURBATION MODEL                                    TBLOB2 9
C     MULTIPLICATIVE PERTURBATION WITH EXPONENTIALLY DECAYING           TBLOB210
C     EFFECT IN ALL THREE DIRECTIONS. GIVE LATITUDE                     TBLOB211
C     INSTEAD OF CO-LATITUDE.                                           TBLOB212
C     COMMON DECK "CONST" INSERTED HERE                                 CCONST 2
      COMMON/PCONST/CREF,RGAS,GAMMA                                     CCONST 4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                         CCONST 5
C     COMMON DECK "RKAM" INSERTED HERE                                  RKAMCOM2
      REAL KR,KTH,KPH                                                   RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)       RKAMCOM5
C     COMMON DECK "B6" INSERTED HERE                                    CB6    2
      INTEGER DTMX,DTNTBL,DTITBL,DTFRMTB,IDSDT(10)                      CB6    4
      COMMON/B6/DTMX,DTNTBL(10),DTITBL(10),DTFRMTB(10),DTGP(10)         CB6    5
      EQUIVALENCE (DTGP,IDSDT)                                          CB6    6
C                                                                       TBLOB216
C     COMMON DECK "TT" INSERTED HERE                                    CTT    2
      REAL MODT                                                         CTT    4
      COMMON/TT/MODT(4), T,PTT,PTR,PTTH,PTPH                            CTT    5
C     COMMON DECK "WW" INSERTED HERE                                    CWW    2
      PARAMETER (NWARSZ=1000)                                           CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
     8 (RCVRH,W(20)),                                                   CWW2   7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2   9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2  10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2  11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2  12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2  13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2  15
```

```
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))              CWW2  16
      REAL MMODEL,MFORM,MID                                         CWW3   2
C                                                                   CWW3   3
C     WIND          100-124                                         CWW3   4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)       CWW3   5
C                                                                   CWW3   6
C     DELTA WIND           125-149                                  CWW3   7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)    CWW3   8
C                                                                   CWW3   9
C     SOUND SPEED 150-174                                           CWW3  10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)       CWW3  11
      EQUIVALENCE (W(153),REFC)                                     CWW3  12
C                                                                   CWW3  13
C     DELTA SOUND SPEED 175-199                                     CWW3  14
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)    CWW3  15
C                                                                   CWW3  16
C     TEMPERATURE     200-224                                       CWW3  17
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)       CWW3  18
C                                                                   CWW3  19
C     DELTA TEMPERATURE     225-249                                 CWW3  20
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)    CWW3  21
C                                                                   CWW3  22
C     MOLECULAR    250-274                                          CWW3  23
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)       CWW3  24
C                                                                   CWW3  25
C     RECEIVER HEIGHT    275-299                                    CWW3  26
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)       CWW3  27
C                                                                   CWW3  28
C     TOPOGRAPHY    300-324                                         CWW3  29
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)       CWW3  30
C                                                                   CWW3  31
C     DELTA TOPOGRAPHY    325-349                                   CWW3  32
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)    CWW3  33
C                                                                   CWW3  34
C     UPPER SURFACE TOPOGRAPHY    350-374                           CWW3  35
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)       CWW3  36
C     PLOT ENHANCEMENTS CONTROL PARAMETERS                          CWW3  37
C                                                                   CWW3  38
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)                   CWW3  39
C     ABSORPTION          500-524                                   CWW3  40
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)       CWW3  41
C                                                                   CWW3  42
C     DELTA ABSORPTION          525-549                             CWW3  43
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)    CWW3  44
C                                                                   CWW3  45
C     PRESSURE          550-574                                     CWW3  46
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)       CWW3  47
C                                                                   CWW3  48
C     DELTA PRESSURE          575-599                               CWW3  49
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)    CWW3  50
C                                                                   CWW3  51
C                                                                   TBLOB219
      EQUIVALENCE (C0,W(228)), (Z0,W(229)),(TBTH0,W(230))           TBLOB220
      EQUIVALENCE (PH0,W(231)),(WZ,W(232)),(WTH,W(233)),(WPH,W(234))TBLOB221
C                                                                   TBLOB222
```

```
      DATA RECOGDT/2.0/                                         TBLOB223
      DATA DTMX/1/                                              TOB2BL 2
      DATA DTNTBL/1,11,8*0/                                     TOB2BL 3
      DATA DTITBL/1,9*0/                                        TOB2BL 4
      DATA DTFRMTB/1,9*0/                                       TOB2BL 5
C                                                               TBLOB226
      ENTRY IPTEMP                                              TBLOB227
      IF(RECOGDT .NE. DTMODEL)                                  TBLOB228
     1    CALL RERROR('DTEMP  ','WRNG MODEL',RECOGDT)           TBLOB229
C                                                               TBLOB230
C                                                               TBLOB231
      MODT(3)=6HTBLOB2                                          TBLOB232
      MODT(4)=DTID                                              TBLOB233
C                                                               TBLOB234
      FWZ=0.0                                                   TBLOB235
      FWTH=0.0                                                  TBLOB236
      FWPH=0.0                                                  TBLOB237
      TH0= PID2-TBTH0                                           TBLOB238
      IF(WZ.NE.0.0) FWZ=2.0/WZ/WZ                               TBLOB239
      IF(WTH.NE.0.0) FWTH=2.0/WTH/WTH                           TBLOB240
      IF(WPH.NE.0.0) FWPH=2.0/WPH/WPH                           TBLOB241
      RETURN                                                    TBLOB242
C                                                               TBLOB243
      ENTRY PTEMP                                               TBLOB244
C                                                               TBLOB245
      IF(C0.EQ.0.0) RETURN                                      TBLOB246
C                                                               TBLOB247
      DZ=R-EARTHR-Z0                                            TBLOB248
      DTH=TH-TH0                                                TBLOB249
      DPH=PH-PH0                                                TBLOB250
      DEXPO=0.0                                                 TBLOB251
      EXPO=-0.5*(DZ*DZ*FWZ+DTH*DTH*FWTH+DPH*DPH*FWPH)           TBLOB252
      IF(EXPO .GT. -200.0) DEXPO=C0*EXP(EXPO)                   TBLOB253
      DEL=1.0+DEXPO                                             TBLOB254
C                                                               TBLOB255
      PTR=PTR*DEL-T*DEXPO*FWZ*DZ                                TBLOB256
      PTTH=PTTH*DEL-T*DEXPO*FWTH*DTH                            TBLOB257
      PTPH=PTPH*DEL-T*DEXPO*FWPH*DPH                            TBLOB258
      T=T*DEL                                                   TBLOB259
      RETURN                                                    TBLOB260
      END                                                       TBLOB261
```

```
      SUBROUTINE NPTEMP                                         NPTEMP 8
C     DO-NOTHING TEMPERATURE PERTURBATION MODEL                 NPTEMP 9
C     COMMON DECK "TT" INSERTED HERE                            CTT    2
      REAL MODT                                                 CTT    4
      COMMON/TT/MODT(4), T,PTT,PTR,PTTH,PTPH                    CTT    5
C     COMMON DECK "WW" INSERTED HERE                            CWW    2
      PARAMETER (NWARSZ=1000)                                   CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                           CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                        CWW2   2
```

```
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),
     8 (RCVRH,W(20)),
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),
     6 (HMIN,W(27)),(RGMAX,W(28)),
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))
      REAL MMODEL,MFORM,MID
C
C     WIND         100-124
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)
C
C     DELTA WIND         125-149
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)
C
C     SOUND SPEED 150-174
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)
      EQUIVALENCE (W(153),REFC)
C
C     DELTA SOUND SPEED 175-199
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)
C
C     TEMPERATURE    200-224
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)
C
C     DELTA TEMPERATURE    225-249
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)
C
C     MOLECULAR   250-274
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)
C
C     RECEIVER HEIGHT   275-299
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)
C
C     TOPOGRAPHY   300-324
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)
C
C     DELTA TOPOGRAPHY   325-349
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)
C
C     UPPER SURFACE TOPOGRAPHY   350-374
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)
C     PLOT ENHANCEMENTS CONTROL PARAMETERS
C
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)
C     ABSORPTION          500-524
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)
C
```

```
C     DELTA ABSORPTION          525-549                                 CWW3  43
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)        CWW3  44
C                                                                       CWW3  45
C     PRESSURE          550-574                                         CWW3  46
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)           CWW3  47
C                                                                       CWW3  48
C     DELTA PRESSURE          575-599                                   CWW3  49
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)        CWW3  50
C                                                                       CWW3  51
C                                                                       NPTEMP12
C     COMMON DECK "B6" INSERTED HERE                                    CB6    2
      INTEGER DTMX,DTNTBL,DTITBL,DTFRMTB,IDSDT(10)                      CB6    4
      COMMON/B6/DTMX,DTNTBL(10),DTITBL(10),DTFRMTB(10),DTGP(10)         CB6    5
      EQUIVALENCE (DTGP,IDSDT)                                          CB6    6
C                                                                       NPTEMP14
      DATA DTMX/1/                                                      NEMPBL 2
      DATA DTNTBL/1,11,8*0/                                             NEMPBL 3
      DATA DTITBL/1,9*0/                                                NEMPBL 4
      DATA DTFRMTB/1,9*0/                                               NEMPBL 5
C                                                                       NPTEMP17
      DATA RECOGDT/0.0/                                                 NPTEMP18
C                                                                       NPTEMP19
      ENTRY IPTEMP                                                      NPTEMP20
      IF(RECOGDT .NE. DTMODEL)                                          NPTEMP21
     1    CALL RERROR('DTEMP   ','WRNG MODEL',RECOGDT)                  NPTEMP22
C                                                                       NPTEMP23
      MODT(3)=6HNPTEMP                                                  NPTEMP24
      MODT(4)=DTID                                                      NPTEMP25
      RETURN                                                            NPTEMP26
C                                                                       NPTEMP27
      ENTRY PTEMP                                                       NPTEMP28
      RETURN                                                            NPTEMP29
      END                                                               NPTEMP30
```

```
      SUBROUTINE MCONST                                                 MCONST 8
C     CONSTANT MOLECULAR WEIGHT MODEL                                   MCONST 9
C     COMMON DECK "WW" INSERTED HERE                                    CWW    2
      PARAMETER (NWARSZ=1000)                                           CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
     8 (RCVRH,W(20)),                                                   CWW2   7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2   9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2  10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2  11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2  12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2  13
```

```
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))               CWW2  15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                    CWW2  16
      REAL MMODEL,MFORM,MID                                              CWW3   2
C                                                                        CWW3   3
C     WIND          100-124                                              CWW3   4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)            CWW3   5
C                                                                        CWW3   6
C     DELTA WIND         125-149                                         CWW3   7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)         CWW3   8
C                                                                        CWW3   9
C     SOUND SPEED 150-174                                                CWW3  10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)            CWW3  11
      EQUIVALENCE (W(153),REFC)                                          CWW3  12
C                                                                        CWW3  13
C     DELTA SOUND SPEED 175-199                                          CWW3  14
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)         CWW3  15
C                                                                        CWW3  16
C     TEMPERATURE    200-224                                             CWW3  17
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)            CWW3  18
C                                                                        CWW3  19
C     DELTA TEMPERATURE    225-249                                       CWW3  20
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)         CWW3  21
C                                                                        CWW3  22
C     MOLECULAR   250-274                                                CWW3  23
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)            CWW3  24
C                                                                        CWW3  25
C     RECEIVER HEIGHT   275-299                                          CWW3  26
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)            CWW3  27
C                                                                        CWW3  28
C     TOPOGRAPHY   300-324                                               CWW3  29
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)            CWW3  30
C                                                                        CWW3  31
C     DELTA TOPOGRAPHY   325-349                                         CWW3  32
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)         CWW3  33
C                                                                        CWW3  34
C     UPPER SURFACE TOPOGRAPHY   350-374                                 CWW3  35
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)            CWW3  36
C     PLOT ENHANCEMENTS CONTROL PARAMETERS                               CWW3  37
C                                                                        CWW3  38
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)                        CWW3  39
C     ABSORPTION          500-524                                        CWW3  40
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)            CWW3  41
C                                                                        CWW3  42
C     DELTA ABSORPTION          525-549                                  CWW3  43
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)         CWW3  44
C                                                                        CWW3  45
C     PRESSURE         550-574                                           CWW3  46
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)            CWW3  47
C                                                                        CWW3  48
C     DELTA PRESSURE          575-599                                    CWW3  49
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)         CWW3  50
C                                                                        CWW3  51
C     COMMON DECK "MM" INSERTED HERE                                     CMM    2
      REAL M,MODM                                                        CMM    4
```

```
      COMMON/MM/MODM(4),M,PMT,PMR,PMTH,PMPH                              CMM     5
      REAL  MLCNST                                                       MCONST12
      EQUIVALENCE (W(253),MLCNST)                                        MCONST13
C     COMMON DECK "B7" INSERTED HERE                                     CB7     2
      INTEGER MMX,MNTBL,MITBL,MFRMTBL,IDSM(10)                           CB7     4
      REAL MGP                                                           CB7     5
      COMMON/B7/MMX,MNTBL(10),MITBL(10),MFRMTBL(10),MGP(10)              CB7     6
      EQUIVALENCE (MGP,IDSM)                                             CB7     7
      DATA RECOGM/1.0/                                                   MCONST15
      DATA MMX/1/                                                        MNSTBL 2
      DATA MNTBL/1,11,8*0/                                               MNSTBL 3
      DATA MITBL/1,9*0/                                                  MNSTBL 4
      DATA MFRMTBL/1,9*0/                                                MNSTBL 5
C                                                                        MCONST18
      ENTRY IMOLWT                                                       MCONST19
C                                                                        MCONST20
      IF(RECOGM .NE. MMODEL)                                             MCONST21
     1    CALL RERROR('IMOLWT ','WRNG MODEL',RECOGM)                     MCONST22
C                                                                        MCONST23
      MODM(1)=6HMCONST                                                   MCONST24
      MODM(2)=MID                                                        MCONST25
C                                                                        MCONST26
      RETURN                                                             MCONST27
C                                                                        MCONST28
      ENTRY MOLWT                                                        MCONST29
      M=MLCNST                                                           MCONST30
      CALL CLEAR(PMT,4)                                                  MCONST31
      END                                                                MCONST32
```

```
      SUBROUTINE GHORIZ                                                  GHORIZ 8
C     TERRAIN MODEL USING FIXED OFFSET TO EARTHR'S SURFACE               GHORIZ 9
C     COMMON DECK "GG" INSERTED HERE                                     CGG     2
      REAL MODG                                                          CGG     4
      COMMON/GG/MODG(4)                                                  CGG     5
      COMMON/GG/G,PGR,PGRR,PGRTH,PGRPH                                   CGG     6
      COMMON/GG/PGTH,PGPH,PGTHTH,PGPHPH,PGTHPH,GSELECT,GTIME             CGG     7
C     COMMON DECK "WW" INSERTED HERE                                     CWW     2
      PARAMETER (NWARSZ=1000)                                            CWW1    3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                    CWW1    4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                 CWW2    2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),     CWW2    3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),       CWW2    4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),           CWW2    5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),          CWW2    6
     8 (RCVRH,W(20)),                                                    CWW2    7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2    8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                       CWW2    9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                       CWW2   10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                      CWW2   11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),        CWW2   12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))         CWW2   13
```

```
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),      CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))            CWW2  15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                 CWW2  16
      REAL MMODEL,MFORM,MID                                           CWW3   2
C                                                                     CWW3   3
C     WIND          100-124                                           CWW3   4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)         CWW3   5
C                                                                     CWW3   6
C     DELTA WIND          125-149                                     CWW3   7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)      CWW3   8
C                                                                     CWW3   9
C     SOUND SPEED 150-174                                             CWW3  10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)         CWW3  11
      EQUIVALENCE (W(153),REFC)                                       CWW3  12
C                                                                     CWW3  13
C     DELTA SOUND SPEED 175-199                                       CWW3  14
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)      CWW3  15
C                                                                     CWW3  16
C     TEMPERATURE    200-224                                          CWW3  17
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)         CWW3  18
C                                                                     CWW3  19
C     DELTA TEMPERATURE    225-249                                    CWW3  20
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)      CWW3  21
C                                                                     CWW3  22
C     MOLECULAR   250-274                                             CWW3  23
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)         CWW3  24
C                                                                     CWW3  25
C     RECEIVER HEIGHT   275-299                                       CWW3  26
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)         CWW3  27
C                                                                     CWW3  28
C     TOPOGRAPHY   300-324                                            CWW3  29
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)         CWW3  30
C                                                                     CWW3  31
C     DELTA TOPOGRAPHY   325-349                                      CWW3  32
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)      CWW3  33
C                                                                     CWW3  34
C     UPPER SURFACE TOPOGRAPHY   350-374                              CWW3  35
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)         CWW3  36
C     PLOT ENHANCEMENTS CONTROL PARAMETERS                            CWW3  37
C                                                                     CWW3  38
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)                     CWW3  39
C     ABSORPTION          500-524                                     CWW3  40
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)         CWW3  41
C                                                                     CWW3  42
C     DELTA ABSORPTION          525-549                               CWW3  43
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)      CWW3  44
C                                                                     CWW3  45
C     PRESSURE          550-574                                       CWW3  46
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)         CWW3  47
C                                                                     CWW3  48
C     DELTA PRESSURE          575-599                                 CWW3  49
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)      CWW3  50
C                                                                     CWW3  51
C     COMMON DECK "RKAM" INSERTED HERE                                RKAMCOM2
      REAL KR,KTH,KPH                                                 RKAMCOM4
```

```
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)        RKAMCOM5
C                                                                          GHORIZ13
      EQUIVALENCE (W(303), Z0)                                             GHORIZ14
C                                                                          GHORIZ15
C     COMMON DECK "B9" INSERTED HERE                                       CB8      2
      INTEGER GMX,GNTBL,GITBL,GFRMTBL,IDSG(10)                             CB8      4
      COMMON/B9/GMX,GNTBL(10),GITBL(10),GFRMTBL(10),GGP(113)               CB8      5
      EQUIVALENCE (GGP,IDSG),(ANG,GGP(11))                                 CB8      6
C                                                                          GHORIZ17
      DATA RECOGG/1.0/                                                     GHORIZ18
C                                                                          GHORIZ19
      DATA GMX/1/                                                          GRIZBL 2
      DATA GNTBL/1,11,8*0/                                                 GRIZBL 3
      DATA GITBL/1,9*0/                                                    GRIZBL 4
      DATA GFRMTBL/1,9*0/                                                  GRIZBL 5
C                                                                          GHORIZ22
      ENTRY ITOPOG                                                         GHORIZ23
      IF(RECOGG .NE. GMODEL)                                               GHORIZ24
     1    CALL RERROR('GHORIZ ','WRNG MODEL',RECOGG)                       GHORIZ25
      MODG(1)=6HGHORIZ                                                     GHORIZ26
      MODG(2)=GID                                                          GHORIZ27
      RETURN                                                               GHORIZ28
      ENTRY TOPOG                                                          GHORIZ29
      G=R-W(1)-Z0                                                          GHORIZ30
      PGR=1.0                                                              GHORIZ31
      CALL CLEAR(PGRR,8)                                                   GHORIZ32
      END                                                                  GHORIZ33
```

```
      SUBROUTINE GLORENZ                                                   GLORENZ8
C     TERRAIN MODEL USING LORENZIAN SHAPED HORIZONTAL SURFACE LOCATED A    GLORENZ9
C     ARBITRARY GEOGRAPHICAL LOCATION.                                     GLOREN10
C                                                                          GLOREN11
C     COMMON DECK "CONST" INSERTED HERE                                    CCONST 2
      COMMON/PCONST/CREF,RGAS,GAMMA                                        CCONST 4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                            CCONST 5
C     COMMON DECK "B9" INSERTED HERE                                       CB8      2
      INTEGER GMX,GNTBL,GITBL,GFRMTBL,IDSG(10)                             CB8      4
      COMMON/B9/GMX,GNTBL(10),GITBL(10),GFRMTBL(10),GGP(113)               CB8      5
      EQUIVALENCE (GGP,IDSG),(ANG,GGP(11))                                 CB8      6
C     COMMON DECK "RKAM" INSERTED HERE                                     RKAMCOM2
      REAL KR,KTH,KPH                                                      RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)          RKAMCOM5
C     COMMON DECK "GG" INSERTED HERE                                       CGG      2
      REAL MODG                                                            CGG      4
      COMMON/GG/MODG(4)                                                    CGG      5
      COMMON/GG/G,PGR,PGRR,PGRTH,PGRPH                                     CGG      6
      COMMON/GG/PGTH,PGPH,PGTHTH,PGPHPH,PGTHPH,GSELECT,GTIME               CGG      7
C     COMMON DECK "B10" INSERTED HERE                                      CB9      2
      INTEGER DGMX,DGNTBL,DGITBL,DGFRMTB,IDSDG(10)                         CB9      4
      COMMON/B10/DGMX,DGNTBL(10),DGITBL(10),DGFRMTB(10),DGGP(10)           CB9      5
      EQUIVALENCE (DGGP,IDSDG)                                             CB9      6
```

```
C     COMMON DECK "WW" INSERTED HERE                                    CWW      2
      PARAMETER (NWARSZ=1000)                                           CWW1     3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1     4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2     2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2     3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2     4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2     5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2     6
     8 (RCVRH,W(20)),                                                   CWW2     7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2     8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2     9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2    10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2    11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2    12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2    13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2    14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2    15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2    16
      REAL MMODEL,MFORM,MID                                             CWW3     2
C                                                                       CWW3     3
C     WIND          100-124                                             CWW3     4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)           CWW3     5
C                                                                       CWW3     6
C     DELTA WIND          125-149                                       CWW3     7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)        CWW3     8
C                                                                       CWW3     9
C     SOUND SPEED 150-174                                               CWW3    10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)           CWW3    11
      EQUIVALENCE (W(153),REFC)                                         CWW3    12
C                                                                       CWW3    13
C     DELTA SOUND SPEED 175-199                                         CWW3    14
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)        CWW3    15
C                                                                       CWW3    16
C     TEMPERATURE    200-224                                            CWW3    17
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)           CWW3    18
C                                                                       CWW3    19
C     DELTA TEMPERATURE    225-249                                      CWW3    20
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)        CWW3    21
C                                                                       CWW3    22
C     MOLECULAR   250-274                                               CWW3    23
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)           CWW3    24
C                                                                       CWW3    25
C     RECEIVER HEIGHT   275-299                                         CWW3    26
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)           CWW3    27
C                                                                       CWW3    28
C     TOPOGRAPHY   300-324                                              CWW3    29
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)           CWW3    30
C                                                                       CWW3    31
C     DELTA TOPOGRAPHY   325-349                                        CWW3    32
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)        CWW3    33
C                                                                       CWW3    34
C     UPPER SURFACE TOPOGRAPHY   350-374                                CWW3    35
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)           CWW3    36
C     PLOT ENHANCEMENTS CONTROL PARAMETERS                              CWW3    37
C                                                                       CWW3    38
```

```
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)                       CWW3   39
C     ABSORPTION          500-524                                       CWW3   40
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)           CWW3   41
C                                                                       CWW3   42
C     DELTA ABSORPTION          525-549                                 CWW3   43
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)        CWW3   44
C                                                                       CWW3   45
C     PRESSURE          550-574                                         CWW3   46
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)           CWW3   47
C                                                                       CWW3   48
C     DELTA PRESSURE          575-599                                   CWW3   49
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)        CWW3   50
C                                                                       CWW3   51
      EQUIVALENCE (GCZAMP,W(303))                                       GLOREN18
      EQUIVALENCE (GCLAMZ,W(304)) ,(GCTHDL,W(305))                      GLOREN19
      EQUIVALENCE (GCBASE,W(306))                                       GLOREN20
C                                                                       GLOREN21
      DATA RECOGG/4.0/                                                  GLOREN22
      DATA GMX/1/                                                       GRENZBL2
      DATA GNTBL/1,11,8*0/                                              GRENZBL3
      DATA GITBL/1,9*0/                                                 GRENZBL4
      DATA GFRMTBL/1,9*0/                                               GRENZBL5
C                                                                       GLOREN25
C                                                                       GLOREN26
      ENTRY ITOPOG                                                      GLOREN27
C                                                                       GLOREN28
      IF(RECOGG .NE. GMODEL)                                            GLOREN29
     1    CALL RERROR('GROUND ','WRNG MODEL',RECOGG)                    GLOREN30
      MODG(1)=7HGLORENZ                                                 GLOREN31
      MODG(2)=GID                                                       GLOREN32
C                                                                       GLOREN33
      GCTH0=PID2-GCLAMZ                                                 GLOREN34
      GCINV=1.0/GCTHDL                                                  GLOREN35
      CALL IPTOPOG                                                      GLOREN36
      RETURN                                                            GLOREN37
C                                                                       GLOREN38
      ENTRY TOPOG                                                       GLOREN39
C                                                                       GLOREN40
      CALL CLEAR(PGRTH,7)                                               GLOREN41
C                                                                       GLOREN42
      ETA=(TH-GCTH0)*GCINV                                              GLOREN43
      ETA2=ETA*ETA                                                      GLOREN44
      GBINOM=1.0/(1.0 + ETA2)                                           GLOREN45
      Z=GCZAMP*GBINOM                                                   GLOREN46
      G=R-EARTHR-Z-GCBASE                                               GLOREN47
C                                                                       GLOREN48
      PGR=1.0                                                           GLOREN49
      GBINOMB=GBINOM*GCINV                                              GLOREN50
      PGTH=2.0*Z*ETA*GBINOMB                                            GLOREN51
      PGTHTH=2.0*Z*GBINOMB*GBINOMB*(1.0-3.0*ETA2)                       GLOREN52
C                                                                       GLOREN53
      RETURN                                                            GLOREN54
      END                                                               GLOREN55
```

```
      SUBROUTINE GTANH                                                  GTANH  8
C     TERRAIN PROFILE REPRESENTED BY A SEQUENCE OF LINEAR SEGMENTS      GTANH  9
C     SMOOTHLY JOINED BY HYPERBOLIC FUNCTIONS. PARAMETERS ARE INPUT     GTANH 10
C     AS TABULAR DATA WITH SLOPES COMPUTED FROM TERRAIN DATA.           GTANH 11
C     COMMON DECK "CONST" INSERTED HERE                                 CCONST 2
      COMMON/PCONST/CREF,RGAS,GAMMA                                     CCONST 4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                         CCONST 5
C     TERRAIN MODEL                                                     GTANH 13
      REAL C(49), LAM0,LMI,LMIM1,LM(49), DL(49),ALC(50)                 GTANH 14
C     COMMON DECK "RKAM" INSERTED HERE                                  RKAMCOM2
      REAL KR,KTH,KPH                                                   RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)       RKAMCOM5
C     COMMON DECK "GG" INSERTED HERE                                    CGG    2
      REAL MODG                                                         CGG    4
      COMMON/GG/MODG(4)                                                 CGG    5
      COMMON/GG/G,PGR,PGRR,PGRTH,PGRPH                                  CGG    6
      COMMON/GG/PGTH,PGPH,PGTHTH,PGPHPH,PGTHPH,GSELECT,GTIME            CGG    7
C     COMMON DECK "B9" INSERTED HERE                                    CB8    2
      INTEGER GMX,GNTBL,GITBL,GFRMTBL,IDSG(10)                          CB8    4
      COMMON/B9/GMX,GNTBL(10),GITBL(10),GFRMTBL(10),GGP(113)            CB8    5
      EQUIVALENCE (GGP,IDSG),(ANG,GGP(11))                              CB8    6
C     COMMON DECK "WW" INSERTED HERE                                    CWW    2
      PARAMETER (NWARSZ=1000)                                           CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
     8 (RCVRH,W(20)),                                                   CWW2   7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2   9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2  10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2  11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2  12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2  13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2  15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2  16
      REAL MMODEL,MFORM,MID                                             CWW3   2
C                                                                       CWW3   3
C     WIND        100-124                                               CWW3   4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)           CWW3   5
C                                                                       CWW3   6
C     DELTA WIND        125-149                                         CWW3   7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)        CWW3   8
C                                                                       CWW3   9
C     SOUND SPEED 150-174                                               CWW3  10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)           CWW3  11
      EQUIVALENCE (W(153),REFC)                                         CWW3  12
C                                                                       CWW3  13
C     DELTA SOUND SPEED 175-199                                         CWW3  14
```

```
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)
C
C     TEMPERATURE    200-224
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)
C
C     DELTA TEMPERATURE    225-249
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)
C
C     MOLECULAR   250-274
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)
C
C     RECEIVER HEIGHT   275-299
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)
C
C     TOPOGRAPHY   300-324
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)
C
C     DELTA TOPOGRAPHY   325-349
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)
C
C     UPPER SURFACE TOPOGRAPHY   350-374
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)
C     PLOT ENHANCEMENTS CONTROL PARAMETERS
C
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)
C     ABSORPTION         500-524
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)
C
C     DELTA ABSORPTION          525-549
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)
C
C     PRESSURE          550-574
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)
C
C     DELTA PRESSURE          575-599
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)
C
      EQUIVALENCE (LAM0,GGP(12)),(Z0,GGP(62)),(DL0,GGP(112))
      EQUIVALENCE (LM,GGP(13)),(C,GGP(63)),(DL,GGP(113))
C
      DATA RECOGG/3.0/
      DATA ANG/0.0/
      DATA GMX/2/
      DATA GNTBL/1,11,162,7*0/
      DATA GITBL/1,50,8*0/
      DATA GFRMTBL/1,2,8*0/
C
      ENTRY ITOPOG
C
      CALL IPTOPOG
C
C     IF HAD PREVIOUS CALL BUT NOTHING THIS TIME, EXIT NOW
C     RETAINING PREVIOUS TABULAR DATA COUNT
      IF(N.GT.0 .AND. ANG.EQ.0.0) RETURN
C
```

```
      IF(RECOGG .NE. GMODEL)
     1    CALL RERROR('TOPO    ','WRNG MODEL',RECOGG)
      MODG(1)=5HGTANH
      MODG(2)=GID
      N=ANG/3
      IF(ANG.NE.3*N.OR.N.LE.0)
     1      CALL RERROR('GTANH','BAD NUMBER',ANG+2.0)
      N=N-2
      ANG=0.0
C
C
C     CONVERT 'Z' ARRAY INPUT(OVERLAYS 'C' ARRAY) TO 'C' ARRAY
C
      ZM1=Z0
      LAM0=PID2-LAM0
      LMIM1=LAM0
      NP1=N+1
      DO 10 I=1,NP1
         Z=C(I)
         LMI=PID2-LM(I)
         LM(I)=LMI
         ALC(I)=ALCOSH((LMI-LAM0) / DL(I))
         C(I)=(Z-ZM1)/(LMI-LMIM1)
         ZM1=Z
10       LMIM1=LMI
C
      RETURN
C
      ENTRY TOPOG
C
      IF(N.LE.0)
     1    CALL RERROR('GTANH','BAD N VALUE',FLOAT(N))
C
      SUM = 0.
      DO 1 I = 1, N
    1 SUM = SUM + DL(I) * (C(I + 1) - C(I)) / 2. *(ALCOSH(((TH-LM
     1(I)) / DL(I))) - ALC(I))
      Z = Z0 - SUM + (C(1) + C(N + 1)) * (TH-LAM0) / 2.
      G=R-EARTHR-Z
      PGR=1.0
C
      PGTH = C(1)
      DO 2 I = 1, N
    2 PGTH= PGTH+ (C(I + 1) - C(I)) / 2. * (1. + TANH ((LM(I)- TH) /DL
     1 (I)))
      PGTH=-PGTH
      PGTHTH=0.0
      DO 3 I=1,N
3     PGTHTH=PGTHTH+
     1    (C(I+1)-C(I))/2.*(1.0-TANH((LM(I)-TH)/DL(I))**2)/DL(I)
C
      CALL PTOPOG
      RETURN
      END
```

```
      SUBROUTINE NPTERR                                                 NPTERR 8
C     DO-NOTHING TERRAIN PERTURBATION MODEL                             NPTERR 9
C     COMMON DECK "GG" INSERTED HERE                                    CGG     2
      REAL MODG                                                         CGG     4
      COMMON/GG/MODG(4)                                                 CGG     5
      COMMON/GG/G,PGR,PGRR,PGRTH,PGRPH                                  CGG     6
      COMMON/GG/PGTH,PGPH,PGTHTH,PGPHPH,PGTHPH,GSELECT,GTIME            CGG     7
C                                                                       NPTERR11
C     COMMON DECK "B10" INSERTED HERE                                   CB9     2
      INTEGER DGMX,DGNTBL,DGITBL,DGFRMTB,IDSDG(10)                      CB9     4
      COMMON/B10/DGMX,DGNTBL(10),DGITBL(10),DGFRMTB(10),DGGP(10)        CB9     5
      EQUIVALENCE (DGGP,IDSDG)                                          CB9     6
C                                                                       NPTERR13
      DATA DGMX/1/                                                      NERRBL 2
      DATA DGNTBL/1,11,8*0/                                             NERRBL 3
      DATA DGITBL/1,9*0/                                                NERRBL 4
      DATA DGFRMTB/1,9*0/                                               NERRBL 5
C                                                                       NPTERR16
      ENTRY IPTOPOG                                                     NPTERR17
C                                                                       NPTERR18
      MODG(3)=6HNPTERR                                                  NPTERR19
      ENTRY PTOPOG                                                      NPTERR20
      END                                                               NPTERR21
```

```
      SUBROUTINE MUARDC                                                 MUARDC 8
C     ARDC BACKGROUND ABSORPTION FORMULA                                MUARDC 9
C                                                                       MUARDC10
C     COMMON DECK "WW" INSERTED HERE                                    CWW     2
      PARAMETER (NWARSZ=1000)                                           CWW1    3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1    4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2    2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2    3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2    4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2    5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2    6
     8 (RCVRH,W(20)),                                                   CWW2    7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2    8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2    9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2   10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2   11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2   12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2   13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2   14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2   15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2   16
      REAL MMODEL,MFORM,MID                                             CWW3    2
C                                                                       CWW3    3
```

```
C       WIND          100-124                                                   CWW3     4
        EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)                 CWW3     5
C                                                                               CWW3     6
C       DELTA WIND          125-149                                             CWW3     7
        EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)              CWW3     8
C                                                                               CWW3     9
C       SOUND SPEED 150-174                                                     CWW3    10
        EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)                 CWW3    11
        EQUIVALENCE (W(153),REFC)                                               CWW3    12
C                                                                               CWW3    13
C       DELTA SOUND SPEED 175-199                                               CWW3    14
        EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)              CWW3    15
C                                                                               CWW3    16
C       TEMPERATURE    200-224                                                  CWW3    17
        EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)                 CWW3    18
C                                                                               CWW3    19
C       DELTA TEMPERATURE    225-249                                            CWW3    20
        EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)              CWW3    21
C                                                                               CWW3    22
C       MOLECULAR   250-274                                                     CWW3    23
        EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)                 CWW3    24
C                                                                               CWW3    25
C       RECEIVER HEIGHT   275-299                                               CWW3    26
        EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)                 CWW3    27
C                                                                               CWW3    28
C       TOPOGRAPHY   300-324                                                    CWW3    29
        EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)                 CWW3    30
C                                                                               CWW3    31
C       DELTA TOPOGRAPHY   325-349                                              CWW3    32
        EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)              CWW3    33
C                                                                               CWW3    34
C       UPPER SURFACE TOPOGRAPHY   350-374                                      CWW3    35
        EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)                 CWW3    36
C       PLOT ENHANCEMENTS CONTROL PARAMETERS                                    CWW3    37
C                                                                               CWW3    38
        EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)                             CWW3    39
C       ABSORPTION          500-524                                             CWW3    40
        EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)                 CWW3    41
C                                                                               CWW3    42
C       DELTA ABSORPTION          525-549                                       CWW3    43
        EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)              CWW3    44
C                                                                               CWW3    45
C       PRESSURE          550-574                                               CWW3    46
        EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)                 CWW3    47
C                                                                               CWW3    48
C       DELTA PRESSURE          575-599                                         CWW3    49
        EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)              CWW3    50
C                                                                               CWW3    51
C                                                                               MUARDC12
        EQUIVALENCE (W(503),BETAV),(W(504),SUTH),(W(505),PRNDTL)                MUARDC13
C                                                                               MUARDC14
C       COMMON DECK "RINREAL" INSERTED HERE                                     CRINREA2
        LOGICAL SPACE                                                           CRINREA4
        REAL LPOLAR,LPOLRI,KPHK,KPHKI,KAY2,KAY2I                                CRINREA5
        CHARACTER DISPM*6                                                       CRINREA6
```

```
      COMMON/RINPL/DISPM                                                CRINREA7
      COMMON /RIN/ MODRIN(8),RAYNAME(2,3),TYPE(3),SPACE                 CRINREA8
      COMMON/RIN/OMEGMIN,OMEGMAX,KAY2,KAY2I,                            CRINREA9
     1               H,HI,PHT,PHTI,PHR,PHRI,PHTH,PHTHI,PHPH,PHPHI       CRINRE10
     2, PHOW,PHOWI,PHKR,PHKRI,PHKTH,PHKTI, PHKPH,PHKPI                  CRINRE11
     3 ,KPHK,KPHKI,POLAR,POLARI,LPOLAR,LPOLRI,SGN                       CRINRE12
C     COMMON DECK "CONST" INSERTED HERE                                 CCONST 2
      COMMON/PCONST/CREF,RGAS,GAMMA                                     CCONST 4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                         CCONST 5
C     COMMON DECK "TT" INSERTED HERE                                    CTT    2
      REAL MODT                                                         CTT    4
      COMMON/TT/MODT(4), T,PTT,PTR,PTTH,PTPH                            CTT    5
C     COMMON DECK "AA" INSERTED HERE                                    CAA    2
      REAL MODA                                                         CAA    4
      REAL MU,MUPT,MUPR,MUPTH,MUPPH                                     CAA    5
      REAL KAP,KAPPT,KAPPR,KAPPTH,KAPPPH                                CAA    6
      COMMON/AA/MODA(4),MU,MUPT,MUPR,MUPTH,MUPPH                        CAA    7
      COMMON/AA/KAP,KAPPT,KAPPR,KAPPTH,KAPPPH                           CAA    8
C                                                                       MUARDC19
C     COMMON DECK "MM" INSERTED HERE                                    CMM    2
      REAL M,MODM                                                       CMM    4
      COMMON/MM/MODM(4),M,PMT,PMR,PMTH,PMPH                             CMM    5
C                                                                       MUARDC21
C     COMMON DECK "CB17" INSERTED HERE                                  CB17   2
      INTEGER VMX,VNTBL,VITBL,VFRMTBL,IDSV(10)                          CB17   4
      COMMON/B17/VMX,VNTBL(10),VITBL(10),VFRMTBL(10),VGP(53)            CB17   5
      EQUIVALENCE (VGP,IDSV),(ANV,VGP(11))                              CB17   6
C                                                                       MUARDC23
      DATA VMX/1/                                                       MRDCBL 2
      DATA VNTBL/1,11,8*0/                                              MRDCBL 3
      DATA VITBL/1,9*0/                                                 MRDCBL 4
      DATA VFRMTBL/1,9*0/                                               MRDCBL 5
      DATA RECOGA/1.0/                                                  MUARDC26
C                                                                       MUARDC27
      ENTRY IABSRP                                                      MUARDC28
      IF(RECOGA .NE. AMODEL)                                            MUARDC29
     1    CALL RERROR('ABSRP ','WRNG MODEL',RECOGA)                     MUARDC30
      MODA(1)=6HMUARDC                                                  MUARDC31
      MODA(2)=AID                                                       MUARDC32
C     SET ALL VICOSITY/CONDUCTIVITY VALUES TO ZERO                      MUARDC33
C     INITIALLY                                                         MUARDC34
      CALL CLEAR(MU,10)                                                 MUARDC35
      CALL IPABSRP                                                      MUARDC36
      RETURN                                                            MUARDC37
C                                                                       MUARDC38
      ENTRY ABSRP                                                       MUARDC39
      MU=BETAV*T**1.5/(SUTH+T)                                          MUARDC40
      KAP=GAMMA*RGAS*MU/((GAMMA-1.0)*M*PRNDTL)                          MUARDC41
      CALL PABSRP                                                       MUARDC42
      END                                                               MUARDC43
```

```
      SUBROUTINE NPABSR
C     DO-NOTHING ABSORPTION PERTURBATION MODEL
C     COMMON DECK "WW" INSERTED HERE
      PARAMETER (NWARSZ=1000)
      COMMON/WW/ID(10),MAXW,W(NWARSZ)
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),
     8 (RCVRH,W(20)),
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),
     6 (HMIN,W(27)),(RGMAX,W(28)),
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))
      REAL MMODEL,MFORM,MID
C
C     WIND         100-124
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)
C
C     DELTA WIND         125-149
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)
C
C     SOUND SPEED 150-174
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)
      EQUIVALENCE (W(153),REFC)
C
C     DELTA SOUND SPEED 175-199
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)
C
C     TEMPERATURE    200-224
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)
C
C     DELTA TEMPERATURE    225-249
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)
C
C     MOLECULAR   250-274
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)
C
C     RECEIVER HEIGHT   275-299
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)
C
C     TOPOGRAPHY   300-324
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)
C
C     DELTA TOPOGRAPHY   325-349
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)
C
C     UPPER SURFACE TOPOGRAPHY   350-374
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)
```

```
C     PLOT ENHANCEMENTS CONTROL PARAMETERS                                CWW3   37
C                                                                         CWW3   38
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)                         CWW3   39
C     ABSORPTION          500-524                                         CWW3   40
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)             CWW3   41
C                                                                         CWW3   42
C     DELTA ABSORPTION          525-549                                   CWW3   43
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)          CWW3   44
C                                                                         CWW3   45
C     PRESSURE          550-574                                           CWW3   46
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)             CWW3   47
C                                                                         CWW3   48
C     DELTA PRESSURE          575-599                                     CWW3   49
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)          CWW3   50
C                                                                         CWW3   51
C     COMMON DECK "AA" INSERTED HERE                                      CAA     2
      REAL MODA                                                           CAA     4
      REAL MU,MUPT,MUPR,MUPTH,MUPPH                                       CAA     5
      REAL KAP,KAPPT,KAPPR,KAPPTH,KAPPPH                                  CAA     6
      COMMON/AA/MODA(4),MU,MUPT,MUPR,MUPTH,MUPPH                          CAA     7
      COMMON/AA/KAP,KAPPT,KAPPR,KAPPTH,KAPPPH                             CAA     8
C                                                                         NPABSR12
C     COMMON DECK "CB18" INSERTED HERE                                    CB18    2
      INTEGER DVMX,DVNTBL,DVITBL,DVFRMTB,IDSDV(10)                        CB18    4
      COMMON/B18/DVMX,DVNTBL(10),DVITBL(10),DVFRMTB(10),DVGP(11)          CB18    5
      EQUIVALENCE (DVGP,IDSDV),(ANDV,DVGP(11))                            CB18    6
C                                                                         NPABSR14
      DATA DVMX/1/                                                        NBSRBL 2
      DATA DVNTBL/1,11,8*0/                                               NBSRBL 3
      DATA DVITBL/1,9*0/                                                  NBSRBL 4
      DATA DVFRMTB/1,9*0/                                                 NBSRBL 5
C                                                                         NPABSR17
      DATA RECOGDA/0.0/                                                   NPABSR18
C                                                                         NPABSR19
      ENTRY IPABSRP                                                       NPABSR20
      IF(RECOGDA .NE. DAMODEL)                                            NPABSR21
     1    CALL RERROR('DABSRP ','WRNG MODEL',RECOGDA)                     NPABSR22
      MODA(3)=6HNPABSR                                                    NPABSR23
      MODA(4)=DAID                                                        NPABSR24
      RETURN                                                              NPABSR25
C                                                                         NPABSR26
      ENTRY PABSRP                                                        NPABSR27
      RETURN                                                              NPABSR28
      END                                                                 NPABSR29


      SUBROUTINE PEXP                                                     PEXP    8
C                                                                         PEXP    9
C     SCALE HEIGHT PRESSURE MODEL                                         PEXP   10
C                                                                         PEXP   11
C     COMMON DECK "WW" INSERTED HERE                                      CWW     2
      PARAMETER (NWARSZ=1000)                                             CWW1    3
```

```
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
     8 (RCVRH,W(20)),                                                   CWW2   7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2   9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2  10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2  11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2  12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2  13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2  15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2  16
      REAL MMODEL,MFORM,MID                                             CWW3   2
C                                                                       CWW3   3
C     WIND        100-124                                               CWW3   4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)           CWW3   5
C                                                                       CWW3   6
C     DELTA WIND          125-149                                       CWW3   7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)        CWW3   8
C                                                                       CWW3   9
C     SOUND SPEED 150-174                                               CWW3  10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)           CWW3  11
      EQUIVALENCE (W(153),REFC)                                         CWW3  12
C                                                                       CWW3  13
C     DELTA SOUND SPEED 175-199                                         CWW3  14
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)        CWW3  15
C                                                                       CWW3  16
C     TEMPERATURE     200-224                                           CWW3  17
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)           CWW3  18
C                                                                       CWW3  19
C     DELTA TEMPERATURE     225-249                                     CWW3  20
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)        CWW3  21
C                                                                       CWW3  22
C     MOLECULAR   250-274                                               CWW3  23
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)           CWW3  24
C                                                                       CWW3  25
C     RECEIVER HEIGHT   275-299                                         CWW3  26
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)           CWW3  27
C                                                                       CWW3  28
C     TOPOGRAPHY   300-324                                              CWW3  29
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)           CWW3  30
C                                                                       CWW3  31
C     DELTA TOPOGRAPHY   325-349                                        CWW3  32
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)        CWW3  33
C                                                                       CWW3  34
C     UPPER SURFACE TOPOGRAPHY   350-374                                CWW3  35
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)           CWW3  36
C     PLOT ENHANCEMENTS CONTROL PARAMETERS                              CWW3  37
C                                                                       CWW3  38

      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)                       CWW3  39
C     ABSORPTION          500-524                                       CWW3  40
```

```
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)           CWW3   41
C                                                                       CWW3   42
C     DELTA ABSORPTION         525-549                                  CWW3   43
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)        CWW3   44
C                                                                       CWW3   45
C     PRESSURE        550-574                                           CWW3   46
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)           CWW3   47
C                                                                       CWW3   48
C     DELTA PRESSURE        575-599                                     CWW3   49
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)        CWW3   50
C                                                                       CWW3   51
C     COMMON DECK "RKAM" INSERTED HERE                                  RKAMCOM2
      REAL KR,KTH,KPH                                                   RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)       RKAMCOM5
C     COMMON DECK "CONST" INSERTED HERE                                 CCONST 2
      COMMON/PCONST/CREF,RGAS,GAMMA                                     CCONST 4
      COMMON/MCONST/PI,PIT2,PID2,DEGS,RAD,ALN10                         CCONST 5
C     COMMON DECK "PP" INSERTED HERE                                    CPP    2
      REAL MODP                                                         CPP    4
      COMMON/PP/MODP(4),P,PPT,PPR,PPTH,PPPH                             CPP    5
C     COMMON DECK "TT" INSERTED HERE                                    CTT    2
      REAL MODT                                                         CTT    4
      COMMON/TT/MODT(4), T,PTT,PTR,PTTH,PTPH                            CTT    5
C     COMMON DECK "MM" INSERTED HERE                                    CMM    2
      REAL M,MODM                                                       CMM    4
      COMMON/MM/MODM(4),M,PMT,PMR,PMTH,PMPH                             CMM    5
C                                                                       PEXP   18
      EQUIVALENCE (W(553),P0),(W(554),HSCALE)                           PEXP   19
C                                                                       PEXP   20
C     COMMON DECK "CB19" INSERTED HERE                                  CB19   2
      INTEGER PRMX,PRNTBL,PRITBL,PRFRMTB,IDSPR(10)                      CB19   4
      COMMON/B19/PRMX,PRNTBL(10),PRITBL(10),PRFRMTB(10),PRGP(11)        CB19   5
      EQUIVALENCE (PRGP,IDSPR),(ANP,PRGP(11))                           CB19   6
C                                                                       PEXP   22
      DATA PRMX/1/                                                      PPBL   2
      DATA PRNTBL/1,11,8*0/                                             PPBL   3
      DATA PRITBL/1,9*0/                                                PPBL   4
      DATA PRFRMTB/1,9*0/                                               PPBL   5
      DATA RECOGP/1.0/                                                  PEXP   25
C                                                                       PEXP   26
      ENTRY IPRES                                                       PEXP   27
      IF(RECOGP .NE. PMODEL)                                            PEXP   28
     1    CALL RERROR('PRESUR ','WRNG MODEL',RECOGP)                    PEXP   29
      MODP(1)=4HPEXP                                                    PEXP   30
      MODP(2)=PID                                                       PEXP   31
      CALL IPPRES                                                       PEXP   32
      RETURN                                                            PEXP   33
C                                                                       PEXP   34
      ENTRY PRES                                                        PEXP   35
      Z=R-EARTHR                                                        PEXP   36
      EX=-Z/HSCALE                                                      PEXP   37
      P=P0*EXP(EX)                                                      PEXP   38
      PPT=0.0                                                           PEXP   39
      PPR=P*EX/Z                                                        PEXP   40
      PPTH=0.0                                                          PEXP   41
```

```
      PPPH=0.0                                                          PEXP   42
      CALL PPRES                                                        PEXP   43
      RETURN                                                            PEXP   44
      END                                                               PEXP   45
```

```
      SUBROUTINE NPPRES                                                 NPPRES 8
C     DO-NOTHING PRESSURE PERTURBATION MODEL                            NPPRES 9
C     COMMON DECK "WW" INSERTED HERE                                    CWW     2
      PARAMETER (NWARSZ=1000)                                           CWW1    3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1    4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2    2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2    3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2    4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2    5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2    6
     8 (RCVRH,W(20)),                                                   CWW2    7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2    8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2    9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2   10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2   11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2   12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2   13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2   14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2   15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2   16
      REAL MMODEL,MFORM,MID                                             CWW3    2
C                                                                       CWW3    3
C     WIND         100-124                                              CWW3    4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)           CWW3    5
C                                                                       CWW3    6
C     DELTA WIND         125-149                                        CWW3    7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)        CWW3    8
C                                                                       CWW3    9
C     SOUND SPEED 150-174                                               CWW3   10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)           CWW3   11
      EQUIVALENCE (W(153),REFC)                                         CWW3   12
C                                                                       CWW3   13
C     DELTA SOUND SPEED 175-199                                         CWW3   14
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)        CWW3   15
C                                                                       CWW3   16
C     TEMPERATURE    200-224                                            CWW3   17
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)           CWW3   18
C                                                                       CWW3   19
C     DELTA TEMPERATURE    225-249                                      CWW3   20
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)        CWW3   21
C                                                                       CWW3   22
C     MOLECULAR   250-274                                               CWW3   23
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)           CWW3   24
C                                                                       CWW3   25
C     RECEIVER HEIGHT   275-299                                         CWW3   26
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)           CWW3   27
```

```
C
C     TOPOGRAPHY   300-324
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)
C
C     DELTA TOPOGRAPHY   325-349
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)
C
C     UPPER SURFACE TOPOGRAPHY   350-374
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)
C     PLOT ENHANCEMENTS CONTROL PARAMETERS
C
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)
C     ABSORPTION          500-524
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)
C
C     DELTA ABSORPTION          525-549
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)
C
C     PRESSURE         550-574
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)
C
C     DELTA PRESSURE          575-599
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)
C
C     COMMON DECK "PP" INSERTED HERE
      REAL MODP
      COMMON/PP/MODP(4),P,PPT,PPR,PPTH,PPPH
C
C     COMMON DECK "CB20" INSERTED HERE
      INTEGER DPMX,DPNTBL,DPITBL,DPFRMTB,IDSDP(10)
      COMMON/B20/DPMX,DPNTBL(10),DPITBL(10),DPFRMTB(10),DPGP(11)
      EQUIVALENCE (DPGP,IDSDP),(ANDP,DPGP(11))
C
      DATA DPMX/1/
      DATA DPNTBL/1,11,8*0/
      DATA DPITBL/1,9*0/
      DATA DPFRMTB/1,9*0/
C
      DATA RECOGDP/0.0/
C
      ENTRY IPPRES
      IF(RECOGDP .NE. DPMODEL)
     1    CALL RERROR('DPPRES ','WRNG MODEL',RECOGDP)
      MODP(3)=6HNPPRES
      MODP(4)=DPID
      RETURN
C
      ENTRY PPRES
      RETURN
      END
```

```
      SUBROUTINE RHORIZ                                                 RHORIZ 8
C     RECEIVER MODEL USING FIXED OFFSET TO EARTHR'S SURFACE.            RHORIZ 9
C     COMMON DECK "RR" INSERTED HERE                                    CRR    2
      REAL MODREC                                                       CRR    4
      COMMON/RR/ MODREC(4)                                              CRR    5
      COMMON/RR/F,PFR,PFRR,PFRTH,PFRPH                                  CRR    6
      COMMON/RR/PFTH,PFPH,PFTHTH,PFPHPH,PFTHPH,FSELECT,FTIME            CRR    7
C     COMMON DECK "WW" INSERTED HERE                                    CWW    2
      PARAMETER (NWARSZ=1000)                                           CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
     8 (RCVRH,W(20)),                                                   CWW2   7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2   9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2  10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2  11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2  12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2  13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2  15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2  16
      REAL MMODEL,MFORM,MID                                             CWW3   2
C                                                                       CWW3   3
C     WIND          100-124                                             CWW3   4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)           CWW3   5
C                                                                       CWW3   6
C     DELTA WIND         125-149                                        CWW3   7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)        CWW3   8
C                                                                       CWW3   9
C     SOUND SPEED 150-174                                               CWW3  10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)           CWW3  11
      EQUIVALENCE (W(153),REFC)                                         CWW3  12
C                                                                       CWW3  13
C     DELTA SOUND SPEED 175-199                                         CWW3  14
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)        CWW3  15
C                                                                       CWW3  16
C     TEMPERATURE    200-224                                            CWW3  17
      EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)           CWW3  18
C                                                                       CWW3  19
C     DELTA TEMPERATURE     225-249                                     CWW3  20
      EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)        CWW3  21
C                                                                       CWW3  22
C     MOLECULAR   250-274                                               CWW3  23
      EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)           CWW3  24
C                                                                       CWW3  25
C     RECEIVER HEIGHT   275-299                                         CWW3  26
      EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)           CWW3  27
C                                                                       CWW3  28
C     TOPOGRAPHY   300-324                                              CWW3  29
      EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)           CWW3  30
C                                                                       CWW3  31
```

```
C     DELTA TOPOGRAPHY    325-349                                        CWW3   32
      EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)         CWW3   33
C                                                                        CWW3   34
C     UPPER SURFACE TOPOGRAPHY   350-374                                 CWW3   35
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)            CWW3   36
C     PLOT ENHANCEMENTS CONTROL PARAMETERS                               CWW3   37
C                                                                        CWW3   38
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)                        CWW3   39
C     ABSORPTION           500-524                                       CWW3   40
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)            CWW3   41
C                                                                        CWW3   42
C     DELTA ABSORPTION          525-549                                  CWW3   43
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)         CWW3   44
C                                                                        CWW3   45
C     PRESSURE          550-574                                          CWW3   46
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)            CWW3   47
C                                                                        CWW3   48
C     DELTA PRESSURE          575-599                                    CWW3   49
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)         CWW3   50
C                                                                        CWW3   51
C     COMMON DECK "RKAM" INSERTED HERE                                   RKAMCOM2
      REAL KR,KTH,KPH                                                    RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)        RKAMCOM5
C                                                                        RHORIZ13
C     COMMON DECK "B8" INSERTED HERE                                     CB10    2
      INTEGER RMX,RNTBL,RITBL,RFRMTBL,IDSR(10)                           CB10    4
      COMMON/B8/RMX,RNTBL(10),RITBL(10),RFRMTBL(10),RGP(10)              CB10    5
      EQUIVALENCE (RGP,IDSR)                                             CB10    6
C                                                                        RHORIZ15
      DATA RECORR/1.0/                                                   RHORIZ16
      DATA RMX/1/                                                        RRIZBL  2
      DATA RNTBL/1,11,8*0/                                               RRIZBL  3
      DATA RITBL/1,9*0/                                                  RRIZBL  4
      DATA RFRMTBL/1,9*0/                                                RRIZBL  5
C                                                                        RHORIZ19
      ENTRY IRECVR                                                       RHORIZ20
C                                                                        RHORIZ21
      IF(RECORR .NE. RMODEL)                                             RHORIZ22
     1    CALL RERROR('RECEIVR','WRNG MODEL',RECORR)                     RHORIZ23
C                                                                        RHORIZ24
      MODREC(1)=6HRHORIZ                                                 RHORIZ25
      MODREC(2)=RID                                                      RHORIZ26
      RETURN                                                             RHORIZ27
      ENTRY RECEVER                                                      RHORIZ28
      F=R-W(1)-W(20)                                                     RHORIZ29
      PFR=1.0                                                            RHORIZ30
      CALL CLEAR(PFRR,8)                                                 RHORIZ31
      END                                                                RHORIZ32


      SUBROUTINE RTERR                                                   RTERR   8
C     RECEIVER MODEL USING FIXED OFFSET TO TERRAIN HEIGHT                RTERR   9
```

```
C       COMMON DECK "RKAM" INSERTED HERE                                    RKAMCOM2
        REAL KR,KTH,KPH                                                     RKAMCOM4
        COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)         RKAMCOM5
C       COMMON DECK "WW" INSERTED HERE                                      CWW     2
        PARAMETER (NWARSZ=1000)                                             CWW1    3
        COMMON/WW/ID(10),MAXW,W(NWARSZ)                                     CWW1    4
        REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                  CWW2    2
        EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),      CWW2    3
      1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),         CWW2    4
      2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),             CWW2    5
      3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),            CWW2    6
      8 (RCVRH,W(20)),                                                      CWW2    7
      4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2    8
      5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                         CWW2    9
      6 (HMIN,W(27)),(RGMAX,W(28)),                                         CWW2   10
      8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                        CWW2   11
      6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),          CWW2   12
      7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))           CWW2   13
      9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),           CWW2   14
      1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))                 CWW2   15
      2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                      CWW2   16
        REAL MMODEL,MFORM,MID                                               CWW3    2
C                                                                           CWW3    3
C       WIND          100-124                                               CWW3    4
        EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)             CWW3    5
C                                                                           CWW3    6
C       DELTA WIND          125-149                                         CWW3    7
        EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)          CWW3    8
C                                                                           CWW3    9
C       SOUND SPEED 150-174                                                 CWW3   10
        EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)             CWW3   11
        EQUIVALENCE (W(153),REFC)                                           CWW3   12
C                                                                           CWW3   13
C       DELTA SOUND SPEED 175-199                                           CWW3   14
        EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)          CWW3   15
C                                                                           CWW3   16
C       TEMPERATURE     200-224                                             CWW3   17
        EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)             CWW3   18
C                                                                           CWW3   19
C       DELTA TEMPERATURE     225-249                                       CWW3   20
        EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)          CWW3   21
C                                                                           CWW3   22
C       MOLECULAR   250-274                                                 CWW3   23
        EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)             CWW3   24
C                                                                           CWW3   25
C       RECEIVER HEIGHT   275-299                                           CWW3   26
        EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)             CWW3   27
C                                                                           CWW3   28
C       TOPOGRAPHY   300-324                                                CWW3   29
        EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)             CWW3   30
C                                                                           CWW3   31
C       DELTA TOPOGRAPHY   325-349                                          CWW3   32
        EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)          CWW3   33
C                                                                           CWW3   34
C       UPPER SURFACE TOPOGRAPHY   350-374                                  CWW3   35
```

```
      EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)          CWW3  36
C     PLOT ENHANCEMENTS CONTROL PARAMETERS                             CWW3  37
C                                                                      CWW3  38
      EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)                      CWW3  39
C     ABSORPTION           500-524                                     CWW3  40
      EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)          CWW3  41
C                                                                      CWW3  42
C     DELTA ABSORPTION           525-549                               CWW3  43
      EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)       CWW3  44
C                                                                      CWW3  45
C     PRESSURE           550-574                                       CWW3  46
      EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)          CWW3  47
C                                                                      CWW3  48
C     DELTA PRESSURE           575-599                                 CWW3  49
      EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)       CWW3  50
C                                                                      CWW3  51
C     COMMON DECK "RR" INSERTED HERE                                   CRR    2
      REAL MODREC                                                      CRR    4
      COMMON/RR/ MODREC(4)                                             CRR    5
      COMMON/RR/F,PFR,PFRR,PFRTH,PFRPH                                 CRR    6
      COMMON/RR/PFTH,PFPH,PFTHTH,PFPHPH,PFTHPH,FSELECT,FTIME           CRR    7
C     COMMON DECK "GG" INSERTED HERE                                   CGG    2
      REAL MODG                                                        CGG    4
      COMMON/GG/MODG(4)                                                CGG    5
      COMMON/GG/G,PGR,PGRR,PGRTH,PGRPH                                 CGG    6
      COMMON/GG/PGTH,PGPH,PGTHTH,PGPHPH,PGTHPH,GSELECT,GTIME           CGG    7
C                                                                      RTERR 14
C     COMMON DECK "B8" INSERTED HERE                                   CB10   2
      INTEGER RMX,RNTBL,RITBL,RFRMTBL,IDSR(10)                         CB10   4
      COMMON/B8/RMX,RNTBL(10),RITBL(10),RFRMTBL(10),RGP(10)            CB10   5
      EQUIVALENCE (RGP,IDSR)                                           CB10   6
C                                                                      RTERR 16
      DATA RMX/1/                                                      RRBL   2
      DATA RNTBL/1,11,8*0/                                             RRBL   3
      DATA RITBL/1,9*0/                                                RRBL   4
      DATA RFRMTBL/1,9*0/                                              RRBL   5
C                                                                      RTERR 19
      DATA RECORR/2.0/                                                 RTERR 20
C                                                                      RTERR 21
      ENTRY IRECVR                                                     RTERR 22
C                                                                      RTERR 23
      IF(RECORR .NE. RMODEL)                                           RTERR 24
     1    CALL RERROR('RECEIVR','WRNG MODEL',RECORR)                   RTERR 25
      MODREC(1)=5HRTERR                                                RTERR 26
      MODREC(2)=RID                                                    RTERR 27
      RETURN                                                           RTERR 28
C                                                                      RTERR 29
      ENTRY RECEVER                                                    RTERR 30
C     GET CURRENT TERRAIN HEIGHT(MUST USE GET1 TO AVOID RECURSION      RTERR 31
C     SINCE WE ARE PROBABLY BEING CALLED BY GET RIGHT NOW)             RTERR 32
      F=GET1(G)-W(20)                                                  RTERR 33
      CALL RMOVE(PFR,PGR,9)                                            RTERR 34
      END                                                              RTERR 35
```

```
      SUBROUTINE RVERT                                                  RVERT   8
C     VERTICAL(CONICAL) RECEIVER SURFACE AT A FIXED RADIUS FROM         RVERT   9
C     A SPECIFIED ORIGIN.                                               RVERT  10
C                                                                       RVERT  11
C     COMMON DECK "RKAM" INSERTED HERE                                  RKAMCOM2
      REAL KR,KTH,KPH                                                   RKAMCOM4
      COMMON//R,TH,PH,KR,KTH,KPH,RKVARS(14),TPULSE,CSTEP,DRDT(20)       RKAMCOM5
C     COMMON DECK "RR" INSERTED HERE                                    CRR     2
      REAL MODREC                                                       CRR     4
      COMMON/RR/ MODREC(4)                                              CRR     5
      COMMON/RR/F,PFR,PFRR,PFRTH,PFRPH                                  CRR     6
      COMMON/RR/PFTH,PFPH,PFTHTH,PFPHPH,PFTHPH,FSELECT,FTIME            CRR     7
C     COMMON DECK "B8" INSERTED HERE                                    CB10    2
      INTEGER RMX,RNTBL,RITBL,RFRMTBL,IDSR(10)                          CB10    4
      COMMON/B8/RMX,RNTBL(10),RITBL(10),RFRMTBL(10),RGP(10)             CB10    5
      EQUIVALENCE (RGP,IDSR)                                            CB10    6
C     COMMON DECK "GAMANG" INSERTED HERE                                CGAMANG2
      COMMON/SPHGAM/SINLM0,COSLM0,GPH0,COSPHD,SINTH,COSTH               CGAMANG4
      COMMON/SPHGAM/GAMFUN,PGMTH,PGMPH,PGMTHTH,PGMPHPH,PGMTHPH          CGAMANG5
C     COMMON DECK "WW" INSERTED HERE                                    CWW     2
      PARAMETER (NWARSZ=1000)                                           CWW1    3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1    4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2    2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2    3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2    4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2    5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2    6
     8 (RCVRH,W(20)),                                                   CWW2    7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2    8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2    9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2   10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2   11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2   12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2   13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2   14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2   15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2   16
      REAL MMODEL,MFORM,MID                                             CWW3    2
C                                                                       CWW3    3
C     WIND         100-124                                              CWW3    4
      EQUIVALENCE (W(100),UMODEL),(W(101),UFORM),(W(102),UID)           CWW3    5
C                                                                       CWW3    6
C     DELTA WIND         125-149                                        CWW3    7
      EQUIVALENCE (W(125),DUMODEL),(W(126),DUFORM),(W(127),DUID)        CWW3    8
C                                                                       CWW3    9
C     SOUND SPEED 150-174                                               CWW3   10
      EQUIVALENCE (W(150),CMODEL),(W(151),CFORM),(W(152),CID)           CWW3   11
      EQUIVALENCE (W(153),REFC)                                         CWW3   12
C                                                                       CWW3   13
C     DELTA SOUND SPEED 175-199                                         CWW3   14
      EQUIVALENCE (W(175),DCMODEL),(W(176),DCFORM),(W(177),DCID)        CWW3   15
C                                                                       CWW3   16
```

```
C      TEMPERATURE     200-224                                              CWW3   17
       EQUIVALENCE (W(200),TMODEL),(W(201),TFORM),(W(202),TID)              CWW3   18
C                                                                           CWW3   19
C      DELTA TEMPERATURE     225-249                                        CWW3   20
       EQUIVALENCE (W(225),DTMODEL),(W(226),DTFORM),(W(227),DTID)           CWW3   21
C                                                                           CWW3   22
C      MOLECULAR    250-274                                                 CWW3   23
       EQUIVALENCE (W(250),MMODEL),(W(251),MFORM),(W(252),MID)              CWW3   24
C                                                                           CWW3   25
C      RECEIVER HEIGHT    275-299                                           CWW3   26
       EQUIVALENCE (W(275),RMODEL),(W(276),RFORM),(W(277),RID)              CWW3   27
C                                                                           CWW3   28
C      TOPOGRAPHY    300-324                                                CWW3   29
       EQUIVALENCE (W(300),GMODEL),(W(301),GFORM),(W(302),GID)              CWW3   30
C                                                                           CWW3   31
C      DELTA TOPOGRAPHY    325-349                                          CWW3   32
       EQUIVALENCE (W(325),GUMODEL),(W(326),GUFORM),(W(327),GUID)           CWW3   33
C                                                                           CWW3   34
C      UPPER SURFACE TOPOGRAPHY    350-374                                  CWW3   35
       EQUIVALENCE (W(350),SMODEL),(W(351),SFORM),(W(352),SID)              CWW3   36
C      PLOT ENHANCEMENTS CONTROL PARAMETERS                                 CWW3   37
C                                                                           CWW3   38
       EQUIVALENCE (W(490),XFQMDL),(W(491),YFQMDL)                          CWW3   39
C      ABSORPTION          500-524                                          CWW3   40
       EQUIVALENCE (W(500),AMODEL),(W(501),AFORM),(W(502),AID)              CWW3   41
C                                                                           CWW3   42
C      DELTA ABSORPTION          525-549                                    CWW3   43
       EQUIVALENCE (W(525),DAMODEL),(W(526),DAFORM),(W(527),DAID)           CWW3   44
C                                                                           CWW3   45
C      PRESSURE          550-574                                            CWW3   46
       EQUIVALENCE (W(550),PMODEL),(W(551),PFORM),(W(552),PID)              CWW3   47
C                                                                           CWW3   48
C      DELTA PRESSURE          575-599                                      CWW3   49
       EQUIVALENCE (W(575),DPMODEL),(W(576),DPFORM),(W(577),DPID)           CWW3   50
C                                                                           CWW3   51
       EQUIVALENCE (RVALPHO,W(278)) ,(RVLAMZ,W(279)), (RVPHO,W(280))        RVERT  17
C                                                                           RVERT  18
       DATA RMX/1/                                                          RRTBL   2
       DATA RNTBL/1,11,8*0/                                                 RRTBL   3
       DATA RITBL/1,9*0/                                                    RRTBL   4
       DATA RFRMTBL/1,9*0/                                                  RRTBL   5
       DATA RECORR/1.0/                                                     RVERT  21
C                 3.0                                                       RVERT  22
C                                                                           RVERT  23
       ENTRY IRECVR                                                         RVERT  24
C                                                                           RVERT  25
       IF(RECORR .NE. RMODEL)                                               RVERT  26
     1    CALL RERROR(7HRECEIVR   ,10HWRNG MODEL ,RECORR)                   RVERT  27
       MODREC(1)=7HRVERT                                                    RVERT  28
       MODREC(2)=RID                                                        RVERT  29
C                                                                           RVERT  30
       SINLMZ=SIN(RVLAMZ)                                                   RVERT  31
       COSLMZ=COS(RVLAMZ)                                                   RVERT  32
       COSALP=COS(RVALPHO)                                                  RVERT  33
C                                                                           RVERT  34
```

```
      RETURN                                                            RVERT 35
C                                                                       RVERT 36
      ENTRY RECEVER                                                     RVERT 37
C                                                                       RVERT 38
      SINLMO=SINLMZ                                                     RVERT 39
      COSLMO=COSLMZ                                                     RVERT 40
      GPHO=RVPHO                                                        RVERT 41
C                                                                       RVERT 42
      CALL GAMANG(TH,PH)                                                RVERT 43
C                                                                       RVERT 44
      F=GAMFUN-COSALP                                                   RVERT 45
      PFR=0.0                                                           RVERT 46
      CALL RMOVE(PFTH,PGMTH,5)                                          RVERT 47
C                                                                       RVERT 48
      RETURN                                                            RVERT 49
      END                                                               RVERT 50
```

```
      SUBROUTINE SMPANN                                                 SMPANN 9
C     ANNOTATION MODEL FOR MINIMUM GRAPHICS SUPPORT                     SMPANN10
C                                                                       SMPANN11
      CHARACTER*(*) S,C                                                 SMPANN12
C                                                                       SMPANN13
C     INITIALIZES PLOT IN DRAFT MODE(DOES NOT REQUIRE DISSPLA)          SMPANN14
C                                                                       SMPANN15
C     COMMON DECK "WWR" INSERTED HERE                                   CWWR   2
      PARAMETER (NWARSZ=1000)                                           CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
     8 (RCVRH,W(20)),                                                   CWW2   7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2   9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2  10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2  11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2  12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2  13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2  15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2  16
C     COMMON DECK "ANNOT" INSERTED HERE                                 ANNOT  2
      CHARACTER*10 ANOTES,HNOTES                                        ANNOT  4
      COMMON/ANNCTL/LENA(4),LENHA(3)                                    ANNOT  5
      COMMON/ANNCTC/ANOTES(2,4),HNOTES(4,3)                             ANNOT  6
C                                                                       SMPANN18
      DATA LENA,ANOTES                                                  SANNBL 2
     1  /2*1,2*2,'DEPTH (M)',' ','DEPTH (KM)',' ',                      SANNBL 3
     2 'HEIGHT (M)',' ','HEIGHT (KM',')'/                               SANNBL 4
C                                                                       SANNBL 5
```

```
      DATA LENHA,HNOTES                                                 SANNBL 6
     1  /3,2,3, 'RANGE AT S','EA LEVEL (','KM)',' '                     SANNBL 7
     2         ,'RANGE (DEG',')',2*' '                                  SANNBL 8
     3         ,'CROSS RANG','E AT SEA L','EVEL (KM)',' '/              SANNBL 9
C                                                                       SMPANN21
      ENTRY SETANN                                                      SMPANN22
C                                                                       SMPANN23
      RETURN                                                            SMPANN24
C                                                                       SMPANN25
      ENTRY ANNFIL(S,C)                                                 SMPANN26
      CALL SFILTR(S,C,'#!')                                             SMPANN27
      END                                                               SMPANN28
```

```
      SUBROUTINE FULANN                                                 FULANN 9
C     ANNOTATION MODEL SUITED FOR PUBLICATION QUALITY LETTERING         FULANN10
C                                                                       FULANN11
      CHARACTER*(*) S,C                                                 FULANN12
C                                                                       FULANN13
C     INITIALIZES PLOT IN PUBLICATION-QUALITY MODE(REQUIRES DISSPLA)    FULANN14
C                                                                       FULANN15
C     COMMON DECK "WWR" INSERTED HERE                                   CWWR   2
      PARAMETER (NWARSZ=1000)                                           CWW1   3
      COMMON/WW/ID(10),MAXW,W(NWARSZ)                                   CWW1   4
      REAL MAXSTP,MAXERR,INTYP,LLAT,LLON                                CWW2   2
      EQUIVALENCE (EARTHR,W(1)),(RAY,W(2)),(XMTRH,W(3)),(TLAT,W(4)),    CWW2   3
     1 (TLON,W(5)),(OW,W(6)),(FBEG,W(7)),(FEND,W(8)),(FSTEP,W(9)),      CWW2   4
     2 (AZ1,W(10)),(AZBEG,W(11)),(AZEND,W(12)),(AZSTEP,W(13)),          CWW2   5
     3 (BETA,W(14)),(ELBEG,W(15)),(ELEND,W(16)),(ELSTEP,W(17)),         CWW2   6
     8 (RCVRH,W(20)),                                                   CWW2   7
     4 (ONLY,W(21)),(HOP,W(22)),(MAXSTP,W(23)),(PLAT,W(24)),(PLON,W(25))CWW2   8
     5,(HMAX,W(26)),(RAYFNC,W(29)),(EXTINC,W(33)),                      CWW2   9
     6 (HMIN,W(27)),(RGMAX,W(28)),                                      CWW2  10
     8 (INTYP,W(41)),(MAXERR,W(42)),(ERATIO,W(43)),                     CWW2  11
     6 (STEP1,W(44)),(STPMAX,W(45)),(STPMIN,W(46)),(FACTR,W(47)),       CWW2  12
     7 (SKIP,W(71)),(RAYSET,W(72)),(PRTSRP,W(74)),(HITLET,W(75))        CWW2  13
     9 ,(BINRAY,W(76)),(PAGLN,W(77)),(PLT,W(81)),(PFACTR,W(82)),        CWW2  14
     1 (LLAT,W(83)),(LLON,W(84)),(RLAT,W(85)),(RLON,W(86))              CWW2  15
     2,(TIC,W(87)),(HB,W(88)),(HT,W(89)),(TICV,W(96))                   CWW2  16
C     COMMON DECK "ANNOT" INSERTED HERE                                 ANNOT  2
      CHARACTER*10 ANOTES,HNOTES                                        ANNOT  4
      COMMON/ANNCTL/LENA(4),LENHA(3)                                    ANNOT  5
      COMMON/ANNCTC/ANOTES(2,4),HNOTES(4,3)                             ANNOT  6
C                                                                       FULANN18
      DATA LENA,ANOTES                                                  FANNBL 2
     1  /4*2,'DEPTH (#M!',')','DEPTH (#KM','!)',                        FANNBL 3
     2'HEIGHT (#M','!)','HEIGHT (#K','M!)'/                             FANNBL 4
      DATA LENHA,HNOTES                                                 FANNBL 5
     1  /3,2,4, 'RANGE AT S','EA LEVEL (','#KM!)',' '                   FANNBL 6
     2         ,'RANGE (#DE','G!)',2*' '                                FANNBL 7
     3         ,'CROSS RANG','E AT SEA L','EVEL (#KM!',')'/             FANNBL 8
      ENTRY SETANN                                                      FULANN21
```

```
C                                                                 FULANN22
      PRINT *,'FULL ANNOTATION MODEL'                             FULANN23
C                                                                 FULANN24
      CALL SCMPLX                                                 FULANN25
      CALL MX1ALF('STAND','!')                                    FULANN26
      CALL MX2ALF('L/CSTD','#')                                   FULANN27
      CALL HEIGHT(HITLET)                                         FULANN28
      RETURN                                                      FULANN29
C                                                                 FULANN30
      ENTRY ANNFIL(S,C)                                           FULANN31
      C=S                                                         FULANN32
      END                                                         FULANN33
```

DDSPLA -- Programs for reading Graphics Output File (Tape File 6)

```
      PROGRAM DDSPLA(TAPE5,INPUT,OUTPUT,TAPE9)
      COMMON/DD/IN,IOR,IT,IS,IC,ICC,IX,IY
      COMMON/SUPNEG/IDEL,NMBS
C
      PARAMETER (LIMPTS=700)
      REAL XV(LIMPTS),YV(LIMPTS)
      INTEGER A(4),C(4),E(4)
      LOGICAL COMPCL
      CHARACTER LINE*72,TEXT*80,S*10
      EQUIVALENCE (IX,XXX),(IY,YYY)
      DATA COMPCL/.TRUE./
      DATA KNT/0/
C
      REWIND 5
10    READ(5,END=100,ERR=100) IT,IX,IY
C      IF(IT.GT.29) PRINT *,IT,IX,IY,XXX,YYY
      IF(IT.GT.20) THEN
         IF(COMPCL) THEN
            WRITE(LINE,'(A,3(I4,1X),2G13.6)')
     1         'NO CALL TO ''COMPRS'' BEFORE---',IT,IX,IY,XXX,YYY
            CALL SYSTEM(52,LINE)
            ENDIF
      ENDIF
      IF(IT.LT.-2 .OR. IT.GT.38) STOP 'CODE>38'
      KNT=KNT+1
      IF(IT.EQ.-1) THEN
         CALL DDEND
      ELSEIF(IT.EQ.-2) THEN
         CALL DDFR
      ELSEIF(IT.EQ.0) THEN
         READ(5) N,M,(TEXT(I:I),I=1,M)
         CALL DDINIT(N,TEXT)
      ELSEIF(IT.EQ.1) THEN
         CALL DDBP
      ELSEIF(IT.EQ.2) THEN
         CALL DDVC
      ELSEIF(IT.EQ.10) THEN
         CALL SCMPLX
      ELSEIF(IT.EQ.11) THEN
         CALL MX1ALF(IX,IY)
      ELSEIF(IT.EQ.12) THEN
         CALL MX2ALF(IX,IY)
      ELSEIF(IT.EQ.13) THEN
         IF(XXX.LE.0.0) THEN
            PRINT *,'HEIGHT OF ZERO!!'
            XXX=.15
         ENDIF
         CALL HEIGHT(XXX)
C
      ELSEIF(IT.EQ.20) THEN
         COMPCL=.FALSE.
         CALL COMPRS
      ELSEIF(IT.EQ.21) THEN
         CALL GRACE(IX,IY)
      ELSEIF(IT.EQ.22) THEN
```

```
         CALL PHYSOR(IX,IY)
      ELSEIF(IT.EQ.23) THEN
         CALL PAGE(IX,IY)
      ELSEIF(IT.EQ.24) THEN
         CALL SCLPIC(IX)
      ELSEIF(IT.EQ.25) THEN
         IF(IX.NE.0 .OR. IY.NE.0) STOP 'ERROR 1'
         READ(5) A,B,C,D,E,F,XAXIS,YAXIS
         CALL XREVTK
         CALL YREVTK
         CALL INTAXS
         CALL TITLE(A,B,C,D,E,F,XAXIS,YAXIS)
      ELSEIF(IT.EQ.26) THEN
         CALL FRAME
      ELSEIF(IT.EQ.27) THEN
         READ(5) W,X,Y,Z
         CALL GRAFB(IX,IY,W,X,Y,Z,XAXIS,YAXIS)
      ELSEIF(IT.EQ.28) THEN
         CALL MARKER(IX)
      ELSEIF(IT.EQ.29) THEN
         IF(IX.GT.LIMPTS) CALL SYSTEM(52,'N>LIMPTS')
         READ(5) (XV(I),I=1,IX),(YV(I),I=1,IX)
         CALL CURVE(XV,YV,IX,IY)
      ELSEIF(IT.EQ.30) THEN
         CALL ENDPL(IX)
      ELSEIF(IT.EQ.31) THEN
         CALL DONEPL
      ELSEIF(IT.EQ.32) THEN
         CALL XTICKS(IX)
      ELSEIF(IT.EQ.33) THEN
         CALL YTICKS(IX)
      ELSEIF(IT.EQ.34) THEN
         CALL MYJACT(IX)
      ELSEIF(IT.EQ.35) THEN
         CALL MYJACT('NUMBERS')
         IDEL=IX
         NMBS=IY
      ELSEIF(IT.EQ.36) THEN
         CALL NOBRDR
      ELSEIF(IT.EQ.37) THEN
         CALL DASH
      ELSEIF(IT.EQ.38) THEN
         READ(5) S
         CALL RESET(S)
      ENDIF
      ENDIF
C
      IF(IT.NE.3) GO TO 10
      READ(5) IOR,N,M,(TEXT(I:I),I=1,M)
      CALL DDTEXT(N,TEXT)
C      PRINT 20,(TEXT(I),I=1,N)
20    FORMAT(8A10)
      GO TO 10
C
100   PRINT *,'NUMBER OF VECTORS=',KNT,IT
```

```
      STOP                                                              DDSPL113
      END                                                               DDSPL114


      SUBROUTINE MYJSUB(IPAR,ITY,IMYJ)                                  DDSPL115
      COMMON/SUPNEG/IDEL,NMBS                                           DDSPL116
      DATA IDEL,NMBS/0,100000/                                          DDSPL117
C                                                                       DDSPL118
C  WE ARE INTERESTED ONLY IN NUMERICAL VALUES CAUSED BY                 DDSPL119
C  A CALL MYJACT('NUMBERS')(IMYJ=5)                                     DDSPL120
C                                                                       DDSPL121
      IF(ITY.GE.0 .OR. IMYJ.NE.5) RETURN                                DDSPL122
C                                                                       DDSPL123
      IF(IDEL.GT.0) IDEL=IDEL-1                                         DDSPL124
C                                                                       DDSPL125
      IF(IDEL.LE.0.AND.NMBS.GT.0) THEN                                  DDSPL126
         NMBS=NMBS-1                                                    DDSPL127
         IPAR=IABS(IPAR)                                                DDSPL128
      ENDIF                                                             DDSPL129
      END                                                               DDSPL130


      SUBROUTINE DDINIT(N,TEXT)                                         DDSPL131
C                                                                       DDSPL132
      COMMON/PLOTCN/NPLOT,INABLE,FX,FY,OFFX,OFFY                        CPLOT   2
      DATA OFFX,OFFY/0.0,0.0/                                           DDSPL134
      DATA NPLOT,INABLE,PLOTSZ,XAXIS,YAXIS/0,0,7.5,11.,8.5/             DDSPL135
C                                                                       DDSPL136
C     NO RE-INITIALIZATIONS BEFORE ENDPL'S                              DDSPL137
      IF(INABLE.GT.0) RETURN                                            DDSPL138
C                                                                       DDSPL139
      IF(NPLOT.GT.0) GO TO 10                                           DDSPL140
C                                                                       DDSPL141
       CALL COMPRS                                                      DDSPL142
      FY=PLOTSZ/1024.                                                   DDSPL143
      FX=FY                                                             DDSPL144
C                                                                       DDSPL145
10    NPLOT=NPLOT+1                                                     DDSPL146
      INABLE=1                                                          DDSPL147
      CALL NOBRDR                                                       DDSPL148
      CALL PAGE(XAXIS,YAXIS)                                            DDSPL149
      CALL PHYSOR(0.0,0.0)                                              DDSPL150
      CALL AREA2D(XAXIS,YAXIS)                                          DDSPL151
      CALL GRACE(0.0)                                                   DDSPL152
      RETURN                                                            DDSPL153
      END                                                               DDSPL154
```

```
      SUBROUTINE DDBP                                                   DDSPL155
      COMMON/PLOTCN/NPLOT,INABLE,FX,FY,OFFX,OFFY                        CPLOT  2
      COMMON/DD/IN,IOR,IT,IS,IC,ICC,IX,IY                               CDDCOM 2
C                                                                       DDSPL158
C     "DDPLOT" DOES NOT REQUIRE RE-INITIALIZATION AFTER EACH FRAME      DDSPL159
C     BUT "DISPLA" DOES SO WE USE THE STATUS OF "INABLE" TO TELL US     DDSPL160
C     WHERE WE ARE.                                                     DDSPL161
      IF(INABLE.EQ.0) CALL DDINIT(-1,0)                                 DDSPL162
      CALL STRTPT(OFFX+IX*FX,OFFY+IY*FY)                                DDSPL163
      RETURN                                                            DDSPL164
      END                                                               DDSPL165


      SUBROUTINE DDVC                                                   DDSPL166
      COMMON/PLOTCN/NPLOT,INABLE,FX,FY,OFFX,OFFY                        CPLOT  2
      COMMON/DD/IN,IOR,IT,IS,IC,ICC,IX,IY                               CDDCOM 2
C                                                                       DDSPL169
C     SEE "DDBP" FOR THE REASON FOR THIS TEST.                          DDSPL170
      IF(INABLE.EQ.0) CALL DDINIT(-1,0)                                 DDSPL171
      CALL CONNPT(OFFX+IX*FX,OFFY+IY*FY)                                DDSPL172
      RETURN                                                            DDSPL173
      END                                                               DDSPL174


      SUBROUTINE DDEND                                                  DDSPL175
      COMMON/PLOTCN/NPLOT,INABLE,FX,FY,OFFX,OFFY                        CPLOT  2
C     CHECK SYNCH, DDFR SHOULD HAVE BEEN CALLED BY NOW                  DDSPL177
      IF(INABLE.GT.0) CALL ENDPL(0)                                     DDSPL178
C                                                                       DDSPL179
      CALL DONEPL                                                       DDSPL180
C                                                                       DDSPL181
      INABLE=0                                                          DDSPL182
      RETURN                                                            DDSPL183
      END                                                               DDSPL184


      SUBROUTINE DDTEXT(N,TEXT)                                         DDSPL185
      COMMON/PLOTCN/NPLOT,INABLE,FX,FY,OFFX,OFFY                        CPLOT  2
      COMMON/DD/IN,IOR,IT,IS,IC,ICC,IX,IY                               CDDCOM 2
C                                                                       DDSPL188
      IF(INABLE.EQ.0) CALL DDINIT(-1,0)                                 DDSPL189
      IF(IOR.EQ.0) CALL ANGLE(0.0)                                      DDSPL190
      IF(IOR.NE.0) CALL ANGLE(90.0)                                     DDSPL191
      CALL MESSAG(TEXT,N*10,OFFX+IX*FX,OFFY+IY*FY)                      DDSPL192
```

```
      RETURN                                                            DDSPL193
      END                                                               DDSPL194
```

```
      SUBROUTINE DDTAB                                                  DDSPL195
      RETURN                                                            DDSPL196
      END                                                               DDSPL197
```

```
      SUBROUTINE DDFR                                                   DDSPL198
      COMMON/PLOTCN/NPLOT,INABLE,FX,FY,OFFX,OFFY                        CPLOT  2
      IF(INABLE.GT.0) CALL ENDPL(0)                                     DDSPL200
      INABLE=0                                                          DDSPL201
      RETURN                                                            DDSPL202
      END                                                               DDSPL203
```

```
      SUBROUTINE SETDDOF(IXF,IYF)                                       DDSPL204
C     SETS RASTOR RANGE TO: -IXF TO 1023-IXF                            DDSPL205
C     AND -IYF TO 1023-IYF.                                             DDSPL206
      COMMON/PLOTCN/NPLOT,INABLE,FX,FY,OFFX,OFFY                        CPLOT  2
      OFFX=IXF*FX                                                       DDSPL208
      OFFY=IYF*FY                                                       DDSPL209
      END                                                               DDSPL210
```

```
      SUBROUTINE GRAFB(XORIG,XSTP,XMAX,YORIG,YSTP,YMAX,XAXIS,YAXIS)     DDSPL211
C                                                                       DDSPL212
      CALL GRAF(XORIG,XSTP,XMAX,YORIG,YSTP,YMAX)                        DDSPL213
      CALL XNONUM                                                       DDSPL214
      CALL YNONUM                                                       DDSPL215
      CALL XGRAXS(XORIG,XSTP,XMAX,XAXIS,' ',-1,0.0,YAXIS)               DDSPL216
      XR=XMAX-XORIG                                                     DDSPL217
      XAX=AINT(XR/XSTP)*XSTP*(XAXIS/XR)                                 DDSPL218
      PRINT *,XORIG,XMAX,XSTP,XAXIS,XAX                                 DDSPL219
      CALL YGRAXS(YORIG,YSTP,YMAX,YAXIS,' ',-1,XAX,0.0)                 DDSPL220
      CALL RESET('XNONUM')                                              DDSPL221
      CALL RESET('YNONUM')                                              DDSPL222
      END                                                               DDSPL223
```

```
      SUBROUTINE TITLEW(A,B,C,D,E,F,G,H)                                 DDSPL224
C     DUMMY ROUTINE ALLOWING SUBSTITUTION OF ALTERNATE TITLE PROGRAMS    DDSPL225
      CALL TITLE(A,B,C,D,E,F,G,H)                                        DDSPL226
      END                                                                DDSPL227
```

DDALT -- Skeleton routines for reading Graphics Output File (Tape File 7)

```
      PROGRAM DDALT(TAPE5,OUTPUT)                                       DDALT
      COMMON/DD/IN,IOR,IT,IS,IC,ICC,IX,IY                               DDALT
      COMMON/SUPNEG/IDEL,NMBS                                           DDALT
C                                                                       DDALT
      PARAMETER (LIMPTS=700)                                            DDALT
      REAL XV(LIMPTS),YV(LIMPTS)                                        DDALT
      INTEGER A(4),C(4),E(4)                                            DDALT
      LOGICAL COMPCL                                                    DDALT
      CHARACTER LINE*72,TEXT*80                                         DDALT
      EQUIVALENCE (IX,XXX),(IY,YYY)                                     DDALT
      DATA COMPCL/.TRUE./                                               DDALT
      DATA KNT/0/                                                       DDALT
C                                                                       DDALT
      REWIND 5                                                          DDALT
 10   READ(5,END=100,ERR=100) IT,IX,IY                                  DDALT
      IF(IT.LT.-2 .OR. IT.GT.24) STOP 'PLOTTING CODE>24'                DDALT
      KNT=KNT+1                                                         DDALT
      IF(IT.EQ.-1) THEN                                                 DDALT
         CALL DDEND                                                     DDALT
      ELSEIF(IT.EQ.-2) THEN                                             DDALT
         CALL DDFR                                                      DDALT
      ELSEIF(IT.EQ.0) THEN                                              DDALT
         READ(5) N,M,(TEXT(I:I),I=1,M)                                  DDALT
         CALL DDINIT(N,TEXT)                                            DDALT
      ELSEIF(IT.EQ.1) THEN                                              DDALT
         CALL DDBP                                                      DDALT
      ELSEIF(IT.EQ.2) THEN                                              DDALT
         CALL DDVC                                                      DDALT
      ELSEIF(IT.EQ.3) THEN                                              DDALT
         READ(5) IOR,N,M,(TEXT(I:I),I=1,M)                              DDALT
         PRINT *,N,M,' ',TEXT(:M)                                       DDALT
         CALL DDTEXT(N,TEXT)                                            DDALT
      ENDIF                                                             DDALT
      GO TO 10                                                          DDALT
C                                                                       DDALT
100   PRINT *,'NUMBER OF VECTORS=',KNT,IT                               DDALT
      STOP                                                              DDALT
      END                                                               DDALT
```

```
      SUBROUTINE DDINIT(N,ID)                                           DDALT
C     INSERT YOUR OWN ROUTINE TO INITIALIZE PLOTTING PROCESS            DDALT
C     ID IS A STRING OF CHARACTERS IDENTIFYING THE PERSON               DDALT
CGETTING THE PLOT, PHONE NUMBER, ETC.                                   DDALT
CN IS THE NUMBER OF CHARACTERS IN THE STRING "ID"                       DDALT
      RETURN                                                            DDALT
      END                                                               DDALT
```

```
      SUBROUTINE DDBP                                                   DDALT
      COMMON/DD/IN,IOR,IT,IS,IC,ICC,IX,IY                               DDALT
C     INSERT YOUR OWN ROUTINE TO DEFINE A VECTOR ORIGIN AT IX,IY        DDALT
      RETURN                                                            DDALT
      END                                                               DDALT
```

```
      SUBROUTINE DDVC                                                   DDALT
      COMMON/DD/IN,IOR,IT,IS,IC,ICC,IX,IY                               DDALT
CINSERT YOUR OWN ROUTINE TO PLOT A STRAIGHT LINE WITH INTENSITY         DDALT
C"IN" FROM THE ORIGIN TO THE END POSITION IX,IY.  A SINGLE CALL         DDALT
CTO DDBP FOLLOWED BY SUCCESSIVE CALLS TO DDVC (CHANGING IX,IY)          DDALT
CPLOTS CONNECTED VECTORS.                                               DDALT
      RETURN                                                            DDALT
      END                                                               DDALT
```

```
      SUBROUTINE DDEND                                                  DDALT
C     CHECK SYNCH, DDFR SHOULD HAVE BEEN CALLED BY NOW                  DDALT
CINSERT YOUR OWN ROUTINE TO EMPTY THE PLOT BUFFER AND RELEASE           DDALT
CTHE PLOTTING COMMAND FILE TO YOUR PLOTTING DEVICE.                     DDALT
      RETURN                                                            DDALT
      END                                                               DDALT
```

```
      SUBROUTINE DDTEXT(N,NT)                                           DDALT
      COMMON/DD/IN,IOR,IT,IS,IC,ICC,IX,IY                               DDALT
CINSERT YOUR OWN ROUTINE TO PLOT A GIVEN ARRAY IN A TABULAR MODE        DDALT
CAFTER INITIALIZING TABULAR PLOTTING WITH DDTAB. NT IS AN ARRAY OF      DDALT
CLENGTH N, CONTAINING "TEXT" FOR TABULAR PLOTTING. SEE APPENDIX C.      DDALT
      RETURN                                                            DDALT
      END                                                               DDALT
```

```
      SUBROUTINE DDTAB                                                  DDALT
      COMMON/DD/IN,IOR,IT,IS,IC,ICC,IX,IY                               DDALT
CINSERT YOUR OWN ROUTINE TO INITIALIZE TABULAR TEXT PLOTTING            DDALT
C        SPECIFY IOR,IS,IX,IY.  TEXT WILL BEGIN AT IX,IY.               DDALT
      RETURN                                                            DDALT
      END                                                               DDALT
```

```
      SUBROUTINE DDFR                                                   DDALT
CINSERT YOUR OWN ROUTINE TO ADVANCE ONE PLOTTING FRAME, WHEN           DDALT
CPLOTTING IS COMPLETED.                                                 DDALT
      RETURN                                                            DDALT
      END                                                               DDALT
```

APPENDIX E. ERRATA FOR NOAA TECH. MEMO. ERL-WPL 103

A versatile three-dimensional Hamiltonian ray-tracing computer program for acoustic waves in the atmosphere.

by R. M. Jones, J. P. Riley, and T. M. Georges

NOAA Tech. Memo. ERL WPL-103

On page 9, the term k_ϕ r sin $d\phi/dP'$ in equation (3.17) should read k_ϕ r $\sin\theta$ $d\phi/dP'$.

On page 10, the denominator in the last fraction in equation (3.18) should be C_{ref} $\partial H/\partial\omega$ instead of C $\partial H/\partial\omega$.

On page 16, $\partial\omega/\partial r$ in (4.27) should be $\partial\Omega/\partial r$.

On page 45, the last line in the caption for Figure 9 should read:
****See Table 29 for details.

INDEX

☆U.S. GOVERNMENT PRINTING OFFICE: 1986 672-748/45294