

NOAA Technical Memorandum ERL WPL-213



USERS' GUIDE TO WPL MICROWAVE RADIATIVE TRANSFER SOFTWARE

J. A. Schroeder
E. R. Westwater

Wave Propagation Laboratory
Boulder, Colorado
October 1991

noaa

NATIONAL OCEANIC AND
ATMOSPHERIC ADMINISTRATION

Environmental Research
Laboratories

NOAA Technical Memorandum ERL WPL-213

USERS' GUIDE TO WPL MICROWAVE RADIATIVE TRANSFER SOFTWARE

J. A. Schroeder
E. R. Westwater

Wave Propagation Laboratory
Boulder, Colorado
October 1991



**UNITED STATES
DEPARTMENT OF COMMERCE**

**Robert A. Mosbacher
Secretary**

**NATIONAL OCEANIC AND
ATMOSPHERIC ADMINISTRATION**

**John A. Knauss
Under Secretary for Oceans
and Atmosphere/Administrator**

**Environmental Research
Laboratories**

**Joseph O. Fletcher
Director**

NOTICE

Mention of a commercial company or product does not constitute an endorsement by NOAA/ERL. Use of information from this publication concerning proprietary products or the tests of such products for publicity or advertising purposes is not authorized.

For sale by the National Technical Information Service, 5285 Port Royal Road
Springfield, VA 22161

CONTENTS

	PAGE
ABSTRACT	1
1. INTRODUCTION	1
2. THEORETICAL BASIS	2
2.1. Defining the Brightness Temperature (T_b)	2
2.2. Defining the Mean Radiating Temperature (T_{mr})	4
2.3. Assumptions and Limitations	5
3. SOFTWARE ALGORITHMS	6
3.1. Modeling a Continuous Atmosphere	8
3.1.1. Profile Input	8
3.1.2. Profile Extrapolation	8
3.1.3. Profile Starting Elevation	9
3.1.4. Profile Level Density	10
3.2. Modeling a Cloudy Atmosphere	10
3.2.1. Cloud Identification and Density Modeling	11
3.2.2. Cloud Density Profile Construction	11
3.3. Modeling the Ray Path	14
3.3.1. Refractivity Profile Calculation	14
3.3.2. Path Length Profile Calculation	15
3.3.3. Path Integral Calculations	15
3.4. Modeling Absorption	16
3.4.1. Clear-Sky Absorption Calculation	16
3.4.2. Cloud Absorption Calculation	17
3.5. Modeling T_b and T_{mr}	18

4.	USING PROGRAM TBMODEL	20
4.1.	Files Accessed by TBMODEL	21
4.1.1.	Input File Containing User Specifications	21
	Output File Title	23
	Profile Output Switch	23
	Frequencies and Elevation Angles	23
	Simulated Antenna Height	23
	Radiosonde Launch Site Latitude	24
	Date and Time Ranges	24
	Short Radiosonde Discard Threshold	24
	Profile Level Density Threshold	24
	Saturation Vapor Pressure Switch	25
	Maximum Number of Cloud Layers	25
	Cloud Detection Threshold	25
	Minimum Cloud Thickness	25
4.1.2.	Input File Containing Radiosonde Data	25
4.1.3.	Text File Containing Output File Header	26
4.1.4.	Output File Containing Results	26
	Header Line	26
	Cloud Boundaries	26
	Profiles	29
	Clear-sky Results	29
	Cloud Model Results	30
4.1.5.	Output File Containing Messages	30
4.2.	Parameters Assigned by Subroutine INITCON	31
4.2.1.	Default Values for Extrapolated Levels	31
4.2.2.	Cloud Water Density Fractions	32
4.2.3.	Safeguard Variables	32
4.2.4.	Physical Constants	32
4.2.5.	Conversion Factors	34
	ACKNOWLEDGMENTS and REFERENCES	36
APPENDIX A:	Nonlinear Path Integration Algorithm	38
APPENDIX B:	Fortran Source Code Listings	40

USERS' GUIDE TO WPL MICROWAVE RADIATIVE TRANSFER SOFTWARE

J.A. Schroeder and E.R. Westwater

NOAA Wave Propagation Laboratory
R/E/WP5, 325 Broadway
Boulder, CO 80303

ABSTRACT. This document describes a computer program that the NOAA Wave Propagation Laboratory (WPL) developed to simulate the brightness temperatures (T_b) observed by an upward-looking microwave radiometer system along a vertical or slant path. FORTRAN source code listings are included as an appendix. WPL uses this software to calibrate radiometers, to simulate hypothetical radiometer systems, and to develop retrieval techniques. The program computes clear-sky T_b from radiosonde profiles of pressure, temperature, and humidity after extrapolating them to 0.1 mb and interpolating between levels where necessary to approximate a continuous atmosphere. Clear-sky attenuation is calculated from the Liebe and Layton water vapor and oxygen absorption model, which is valid for frequencies up to 1 THz. An optional cloud-modeling scheme permits simulations under a variety of cloud configurations. The Rayleigh approximation that underlies the cloud absorption algorithm restricts its validity to nonprecipitating clouds with particle radii less than about 100 μm ; scattering is neglected. Hence, the cloudy T_b simulations are valid only for frequencies less than 100 GHz.

1. INTRODUCTION

The NOAA Wave Propagation Laboratory (WPL) Thermodynamic Profiling Program designs, develops, and operates upward-looking microwave radiometer systems to monitor atmospheric temperature, pressure, humidity, and cloud liquid water content. Retrieving these atmospheric parameters from microwave radiometer measurements is an inverse problem, which cannot be solved without a solution to the forward problem. In this case, the forward problem consists of modeling the brightness temperature (T_b) that an upward-looking microwave radiometer oriented at a specified elevation angle would observe, given a predetermined set of atmospheric parameters. This document describes the algorithms and software that constitute WPL's approach to this forward problem.

WPL scientists have used the T_b computed from these algorithms to develop inversion methods for retrieving atmospheric temperature, water vapor, and cloud liquid (Westwater and Strand, 1968; Westwater, 1972; Decker et al., 1978; Miller and Falls, 1989; Stankov et al., 1991); to calibrate radiometers (Decker and Schroeder, 1991); and to simulate hypothetical radiometer systems (Schroeder et al., 1989).

WPL's approach to the forward problem uses the well-known theory of radiative transfer from a non-scattering atmosphere at microwave frequencies. We use the theory to compute the clear-sky T_b that would be observed, given an atmosphere whose composition is completely defined by radiosonde measurements of pressure, temperature, and humidity, which are taken to be vertical profiles. We extrapolate the radiosonde profiles to 0.1 mb and interpolate between levels where necessary to better approximate a continuous atmosphere. The model of Liebe and Layton (1987) used to compute water vapor and oxygen absorption is valid for frequencies up to 1 THz.

Since conventional radiosondes do not measure cloud water density, we use the model of Decker et al. (1978) to create cloud liquid and ice density profiles consistent with the radiosonde humidity profile. The cloud model permits T_b simulations under a variety of cloud configurations. The Rayleigh approximation that underlies the cloud absorption algorithm restricts its validity to nonprecipitating clouds with particle radii less than about 100 μm ; scattering is neglected. Hence, the cloudy T_b simulations are valid only for frequencies less than 100 GHz.

The purpose of this paper is to document the present algorithms and software to facilitate future improvements by others. We recognize that alternative physical algorithms exist and that the software design is not optimal, especially for a modern UNIX-based computer system. The software design reflects our primary objective, readability, and the constraints imposed by the computer system on which the software evolved (several generations of Control Data Corporation mainframes). The FORTRAN program, TBMODEL, can be obtained from the WPL Thermodynamic Profiling Program.

Section 2 gives the theory and assumptions behind the software algorithms, Section 3 describes the software algorithms themselves, Section 4 gives instructions for program use, and Appendix B contains listings of the FORTRAN source code, where additional documentation resides.

2. THEORETICAL BASIS

This section describes the theory and assumptions behind the radiative transfer software algorithms. The equations presented here will be referenced by number in Section 3, which describes the software algorithms themselves.

2.1. Defining the Brightness Temperature (T_b)

Assuming local thermodynamic equilibrium, the radiative intensity (I), or radiance, at a given frequency for a blackbody radiator is given by the Planck function (Goody and Yung, 1989)

$$B(T) = \left(\frac{2hv^3}{c^2} \right) \left(\frac{1}{e^{\frac{hv}{kT}} - 1} \right), \quad (1)$$

where h = Planck constant
 v = frequency
 c = speed of light
 k = Boltzmann constant
 T = temperature.

I is often expressed as an equivalent blackbody temperature, or brightness temperature, T_b , such that $I = B(T_b)$. A radiometer antenna looking upward toward cold space receives microwave radiation from two sources: the cosmic background and the atmosphere itself. If discrete sources, such as the sun or moon, are outside the radiometer field of view, the cosmic radiation can be treated as a uniform background. At microwave frequencies, atmospheric scattering can be ignored for non-precipitating conditions, so that the propagating radiation is modified by absorption alone. With these two assumptions, we can express the T_b observed at a given frequency as the sum of two terms:

$$B(T_b) = B(T_{bg}) e^{-\tau(0,\infty)} + \int_0^{\infty} B(T(s)) \alpha(s) e^{-\tau(0,s)} ds, \quad (2)$$

where T_b = brightness temperature
 T_{bg} = cosmic background temperature
 s = line-of-sight atmospheric path coordinate
 $T(s)$ = temperature of layer between s and $s + ds$
 $\alpha(s)$ = absorption of layer between s and $s + ds$

and

$$\tau(a,b) = \int_a^b \alpha(s) ds. \quad (3)$$

The factor $e^{-\tau(a,b)}$ in each term represents the exponential decay of the source radiation as it is attenuated by the layer of atmosphere between path coordinates a and b , where 0 represents the antenna position. The first term in (2) represents the cosmic background radiation, attenuated by the entire atmosphere as a single layer. The second term in (2) represents the sum of the radiation contributions from an infinite number of atmospheric layers. Each layer's contribution is attenuated by the layers between it and the antenna.

Note that when we substitute the Planck function defined in (1) back into the three places that it appears in (2), the ratio $2hv^3c^{-2}$ falls out of the equation. Therefore, we define a modified $B(T)$, $\tilde{B}(T)$, to simplify the software (Section 3) as

$$\tilde{B}(T) = \frac{1}{e^{\frac{hv}{kT}} - 1}, \quad (4)$$

where h , ν , k , and T were defined in (1). To extract the temperature from a known $\tilde{B}(T)$, we solve (4) for T as

$$T = \frac{hv}{k \ln \left[1 + \frac{1}{\tilde{B}(T)} \right]}. \quad (5)$$

2.2. Defining the Mean Radiating Temperature (T_{mr})

In general, we can evaluate an integral like the one representing the atmospheric term in (2) by defining a mean radiating temperature (T_{mr}) over the layer defined by the limits on the integral. For example, we can rewrite

$$\int_a^b B(T(s)) \alpha(s) e^{-\tau(0,s)} ds$$

as

$$B(T_{mr}) \int_a^b \alpha(s) e^{-\tau(0,s)} ds. \quad (6)$$

Making the necessary substitutions to integrate (6) as $e^u du$ gives

$$-B(T_{mr}) e^{-\tau(0,s)} \Big|_a^b = B(T_{mr}) (e^{-\tau(0,a)} - e^{-\tau(0,b)}).$$

Therefore, for path coordinate $a < b$, we have

$$\int_a^b B(T(s)) \alpha(s) e^{-\tau(0,s)} ds = B(T_{mr}) e^{-\tau(0,a)} [1 - e^{-\tau(a,b)}]. \quad (7)$$

Solving (7) for $B(T_{mr})$ defines T_{mr} for the layer of atmosphere between levels a and b as

$$B(T_{mr}) = \frac{\int_a^b B(T(s)) \alpha(s) e^{-\tau(0,s)} ds}{e^{-\tau(0,a)} [1 - e^{-\tau(a,b)}]}, \quad (8)$$

where T_{mr} follows directly from (5). Substituting 0 for a and ∞ for b in (8) defines T_{mr} for the entire atmosphere as

$$B(T_{mr}^{\infty}) = \frac{\int_0^{\infty} B(T(s)) \alpha(s) e^{-\tau(0,s)} ds}{1 - e^{-\tau(0,\infty)}} \quad (9)$$

Similarly, the T_{mr} for a cloud layer with base at $a=base$ and top at $b=top$ becomes

$$B(T_{mr}^{cloud}) = \frac{\int_{base}^{top} B(T(s)) \alpha(s) e^{-\tau(0,s)} ds}{e^{-\tau(0,base)} [1 - e^{-\tau(base,top)}]} \quad (10)$$

The formula in (10) holds for a single cloud layer or the lowest one of several cloud layers. The present software does not compute T_{mr} for multiple cloud layers.

2.3. Assumptions and Limitations

The software and algorithms described in this document were designed to calculate thermal emission and absorption in the troposphere. We assume that discrete radiation sources, such as the sun or moon, are outside the radiometer field of view.

The clear-sky atmospheric absorption model (Liebe and Layton, 1987) is valid for frequencies up to 1 THz. However, we only model absorption from water vapor and oxygen, neglecting absorption from minor constituents, such as ozone. Depending on the concentration of a neglected constituent and on the frequency in question, monochromatic calculations may be incorrect. Waters (1976) gives a limited discussion of absorption for selected minor constituents.

For simplicity, the cloud model described here assumes the Rayleigh approximation, under which scattering is negligible relative to absorption, and absorption is independent of cloud particle size distribution. These assumptions restrict the model to nonprecipitating clouds with particle radii less than about 100 μm for frequencies less than 100 GHz. Therefore, the algorithms are not adequate for modeling rain, large droplets, or large ice particles in clouds. The validity of the Rayleigh approximation is discussed by van de Hulst (1981) and Ulaby et al. (1981).

Our calculations of refractivity and refractive index neglect dispersion around the 22.235-GHz water line and in the 60-GHz oxygen complex.

3. SOFTWARE ALGORITHMS

Simulations of the brightness temperature (T_b) observed by an upward-looking microwave radiometer require a model of the atmospheric physics and knowledge of the atmospheric state along the radiometer's line of sight. This section describes the algorithms that program TBMODEL uses to implement the atmospheric physics outlined in Section 2, using radiosonde profiles and (optionally) a cloud model to describe the atmospheric state. Names of variables, arrays, and subroutines are capitalized for emphasis.

The subroutine hierarchy of TBMODEL, shown in Fig. 1, serves as an outline of the program, similar to a flow chart. The subroutine names that begin at the left margin appear in the order that the main program calls them. Similarly, the indented subroutine names listed beneath a subroutine name appear in the order that the parent subroutine calls them. For example, the main program calls PROFILE, PROFILE calls EXTRAP, and EXTRAP calls FIND. Appendix B contains Fortran source code listings for TBMODEL and all of its subroutines. An alphabetical subroutine index for Appendix B is on page 40. All filenames, subroutine names, and variable names have seven characters or less (no extensions) and contain no special characters, in compliance with the operating system under which TBMODEL evolved (several generations of Control Data Corporation mainframes). Variable array dimensions were avoided and use of COMMON was minimized to improve readability, which was the primary software design objective.

Most of TBMODEL is inside a loop that is executed once for each radiosonde profile input (Fig. 1). Before entering that loop, TBMODEL initializes constants (subroutine INITCON), inputs user specifications (subroutine RDSPECS), writes the output file header information (subroutine HEADOUT), and computes quantities that are independent of the atmospheric state. Section 4 discusses the roles of INITCON, RDSPECS, and HEADOUT. TBMODEL computes the set of wavelengths (array WAVE) that correspond to the user-specified channel frequencies (array FRQ) from the relationship

$$\lambda = c / \nu ,$$

where λ = wavelength (cm)
c = speed of light (cm s⁻¹)
 ν = frequency (s⁻¹).

TBMODEL also computes the ratio $h\nu/k$ that appears in (1), (4), and (5) for each frequency and stores the results in array HVK. TBMODEL then uses HVK in (4) to compute a corresponding array of $\tilde{B}(T_{bg})$ values (array BOFTBG). Subroutine CLRABS will use HVK later to compute $\tilde{B}(T)$ for each frequency at each level of the radiosonde temperature profile.

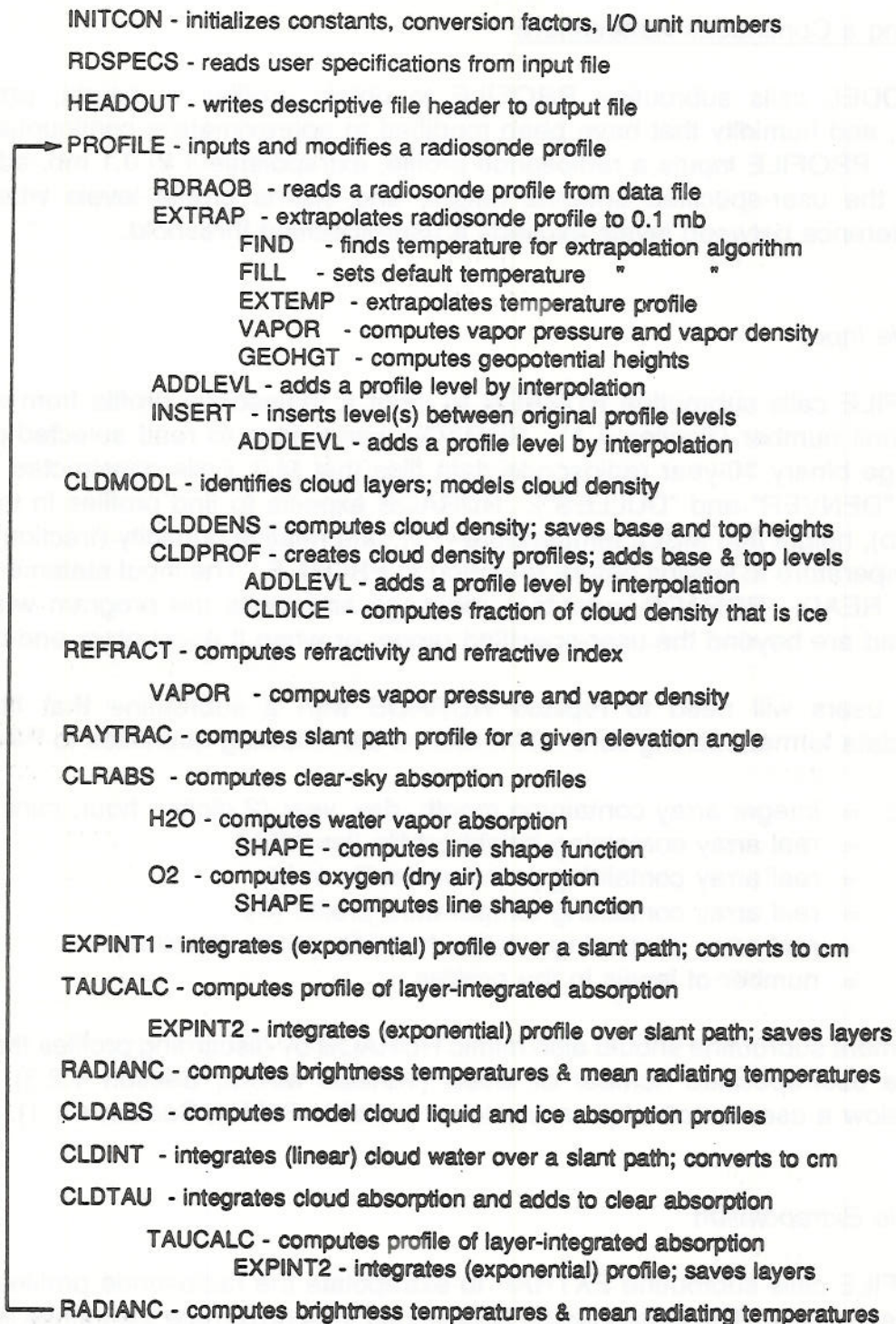


Fig. 1. TBMODEL subroutine hierarchy with nesting indicated by indentation; leftmost subroutines are called by the main program in the order listed. The bracket shows the loop executed for each radiosonde sounding.

3.1. Modeling a Continuous Atmosphere

TBMODEL calls subroutine PROFILE to obtain profiles of height, pressure, temperature, and humidity that have been modified to approximate a continuous, clear atmosphere. PROFILE inputs a radiosonde profile, extrapolates it to 0.1 mb, adjusts it to begin at the user-specified antenna height, and inserts profile levels where the pressure difference between levels exceeds a user-specified threshold.

3.1.1. Profile Input

PROFILE calls subroutine RDRAOB to input a radiosonde profile from a user-designated unit number (Section 4.1). RDRAOB was written to read selected profiles from two large binary 10-year radiosonde data files that M.J. Falls constructed for our group (files "DENVER" and "DULLES"). RDRAOB expects to find profiles in terms of pressure (mb), height (km MSL), temperature (°C), and relative humidity (fraction), but it converts temperature to kelvins before returning to PROFILE. The input statement is an unformatted READ. RDRAOB outputs a message and stops the program when the date/time read are beyond the user-specified range, or when it encounters end-of-file.

Most users will need to replace RDRAOB with a subroutine that fits their radiosonde data format, making sure that it returns the following quantities to PROFILE:

DATE = integer array containing month, day, year (2 digits), hour, minute
Z = real array containing height profile (km MSL)
P = real array containing pressure profile (mb)
T = real array containing temperature profile (K)
RH = real array containing relative humidity profile (fraction)
NL = number of levels in the profiles.

The replacement subroutine should also mimic RDRAOB by discarding profiles that have more than a user-specified number of levels (variable MAXL; Section 4.2.3) or that terminate below a user-specified pressure level (variable PMAX; Section 4.1.1).

3.1.2. Profile Extrapolation

PROFILE calls subroutine EXTRAP to extrapolate the radiosonde profiles to the 0.1-mb pressure level. The extrapolation is designed to approximate the infinite limits on the integrals in Section 2. EXTRAP employs the technique of Crosby et al. (1973) to estimate temperatures at 15 pressure levels between 30 and 0.1 mb, inclusive, from temperatures at 6 lower levels (700, 500, 300, 200, 100, and 50 mb), using regression coefficients derived from rocketsonde data.

EXTRAP calls subroutine FIND to find, by interpolation if necessary, temperatures (array T6) at each of the six pressure levels (array P6) required for the temperature extrapolation. If the required pressure level is outside the range of pressure levels in the radiosonde profile, FIND outputs a message and stops the program.

EXTRAP implicitly assumes that the radiosonde does not terminate below 300 mb. For this reason, subroutines RDSPECS and RDRAOB are designed to discard profiles that terminate below 300 mb, regardless of user specifications (Section 4.1.1). When the radiosonde profile terminates below the 200-, 100-, or 50-mb level, EXTRAP calls subroutine FILL to assign a default temperature (arrays T200, T100, or T50, respectively) to the appropriate element of array T6 and add a new level to the top of the radiosonde profile. If the added level makes the number of profile levels exceed a user-specified maximum (variable MAXL; Section 4.2.3), FILL outputs a message and stops the program. The default temperatures are specified in subroutine INITCON and passed through COMMON block DEFAULT (Section 4.2.1). After all six elements of T6 have been determined, EXTRAP calls subroutine EXTEMP to apply the Crosby et al. (1973) extrapolation coefficients to the six temperatures in T6. Three different sets of extrapolation coefficients (three-dimensional array COEF) reside in DATA statements inside EXTEMP. The coefficient set applied depends on the station latitude passed into EXTEMP (variable RAOBLAT; Section 4.1.1).

EXTRAP extrapolates moisture by computing the vapor pressure that is consistent with a specified constant mixing ratio (variable RMIX). Variable PFRAC, passed through COMMON block DEFAULT, is the fraction of ambient pressure that represents a vapor pressure equivalent to the specified mixing ratio (Section 4.2.1). EXTRAP calls subroutine VAPOR to convert the computed vapor pressure at each extrapolated level to a relative humidity consistent with the corresponding extrapolated temperature. VAPOR will compute saturation vapor pressure in one of two ways, depending on the user-specified value of variable ICE (Section 4.1.1).

EXTRAP calls subroutine GEOHGT to compute geopotential heights consistent with the pressure, temperature, and humidity at the extrapolated levels.

3.1.3. Profile Starting Elevation

PROFILE adjusts the radiosonde profile to begin at the user-specified antenna height (variable ZSTART) to permit simulations from a platform above ground, such as an aircraft. If the original profile terminates below ZSTART, PROFILE outputs a message, discards the profile, and inputs another. If ZSTART is below level 1 of the radiosonde profile, PROFILE resets the antenna height used in the calculations (variable Z0, which appears in the header line of the output file) to the level 1 height. Otherwise, Z0 = ZSTART.

If ZSTART is above the height of level 1, PROFILE searches the profile for ZSTART. If level ZSTART did not exist in the original profile, PROFILE calls subroutine ADDLEVL to add a new profile level at ZSTART by interpolation. ADDLEVL outputs a message and stops the program if the added level would exceed the user-specified maximum number of levels (variable MAXL; Section 4.2.3). After ZSTART is found or computed, PROFILE renumbers the profile levels to begin at ZSTART and converts the height profile (array Z) units to kilometers above the antenna height.

3.1.4. Profile Level Density

PROFILE calls subroutine INSERT to determine when profile levels are too far apart to adequately approximate the continuous functions given in Section 2. INSERT uses the user-specified parameter DPMAX (Section 4.1.1) to determine the number of levels to insert. If DPMAX is non-zero, the maximum permissible pressure difference between each pair of adjacent levels throughout the profile is DPMAX.

If DPMAX is zero, INSERT implements an algorithm that allows the pressure difference between levels to increase with altitude, assuming that profile levels nearest the antenna dominate the radiometer response. The algorithm permits a maximum pressure difference of 5 mb between levels when the upper level lies within the first 50 mb above the antenna. The maximum permissible pressure difference increases by 5 mb for each additional 50 mb above the antenna. For example, if the pressures at two adjacent levels differ by more than 15 mb when the upper level lies between 100 and 150 mb above the antenna, levels are inserted until the 15-mb criterion is met.

When the pressure difference between a pair of adjacent profile levels exceeds the maximum value, INSERT divides the layer between the two levels into the smallest number of equally-spaced pressure levels that differ by less than that value. INSERT calls subroutine ADDLEVL to add each level needed by linear interpolation (ADDDLEVL interpolates pressure in terms of $\ln(\text{pressure})$). If adding the level will make the number of profile levels exceed a user-specified number (variable MAXL; Section 4.2.3), ADDLEVL outputs a message and stops the program. Note that additional levels may be inserted later by the cloud model if clouds are identified in the profile (Section 3.2.2).

3.2. Modeling a Cloudy Atmosphere

TBMODEL offers simulations of T_b under cloudy conditions as an option (Section 4.1.1). Cloudy simulations require a cloud model, because radiosondes do not measure cloud water density. TBMODEL calls subroutine CLDMODL to identify profile levels that meet user-specified cloud criteria and to construct three different models of cloud water density for each profile that contains cloud (Decker et al., 1978). Section 2.3 gave the underlying assumptions and limitations for the cloud model. TBMODEL simulates

radiometer measurements for each of the three cloud models (loop 77). The user controls the number of cloud layers modeled (variable MAXCLD), the relative humidity threshold for cloud detection (variable RHCLD), and the minimum cloud thickness (variable CLDMIN) through values specified in the input file (Section 4.1.1).

3.2.1. Cloud Identification and Density Modeling

Subroutine CLDMODL identifies clouds by searching the relative humidity profile (array RH) from the bottom upward for values that exceed RHCLD. If $RH_i \geq RHCLD$ at the simulated antenna height, we define that height as cloud base (variable BASE). Otherwise, at the first level i where $RH_i \geq RHCLD$, CLDMODL finds the height corresponding to RHCLD, by interpolation if necessary (statement function YINTERP), and assigns that height to BASE. After finding a cloud base, the search progresses upward through the profile, looking for the level at which RH decreases below RHCLD again. The exact height corresponding to RHCLD, found by interpolation if necessary (statement function YINTERP), defines the cloud top (variable TOP).

After finding TOP, CLDMODL calls subroutine CLDDENS to determine the cloud water density. If the cloud is thinner than the user-specified value for CLDMIN, CLDDENS discards the cloud. Otherwise, CLDDENS increments the number of cloud layers in the profile (variable NLAY), computes the cloud water density, and stores the cloud density, base height, and top height in arrays CLDRHO, CLDBASE, and CLDTOP, respectively, indexed by the cloud layer number. CLDDENS computes the cloud water density from the cloud thickness, using the relationship between cloud thickness and density shown by the rightmost of the three curves in Fig. 2. Cloud water densities consistent with the other two curves are obtained later by multiplying the density that CLDDENS computes by the fractions 0.5 and 0.25 (Sections 3.4.2 and 4.2.2).

CLDMODL continues to search for clouds until it identifies MAXCLD cloud layers or reaches the top of the profile. If MAXCLD = 0, no cloud search occurs. The cloud base and top heights that CLDMODL identifies are reported in pairs on the line directly beneath the date and time in the output file (Section 4.1.4).

3.2.2. Cloud Density Profile Construction

CLDMODL calls subroutine CLDPROF to construct a cloud liquid density profile (array DENLIQ) and a cloud ice density profile (array DENICE) with the same vertical coordinates as the modified radiosonde profiles, using the information contained in arrays CLDRHO, CLDBASE, and CLDTOP. If no cloud layers are present, CLDPROF zeroes the cloud water density at all profile levels and zeroes the number of cloud models (variable NMODLS). Otherwise, CLDPROF sets NMODLS to three, referring to the three curves in Fig. 2, and proceeds as outlined below.

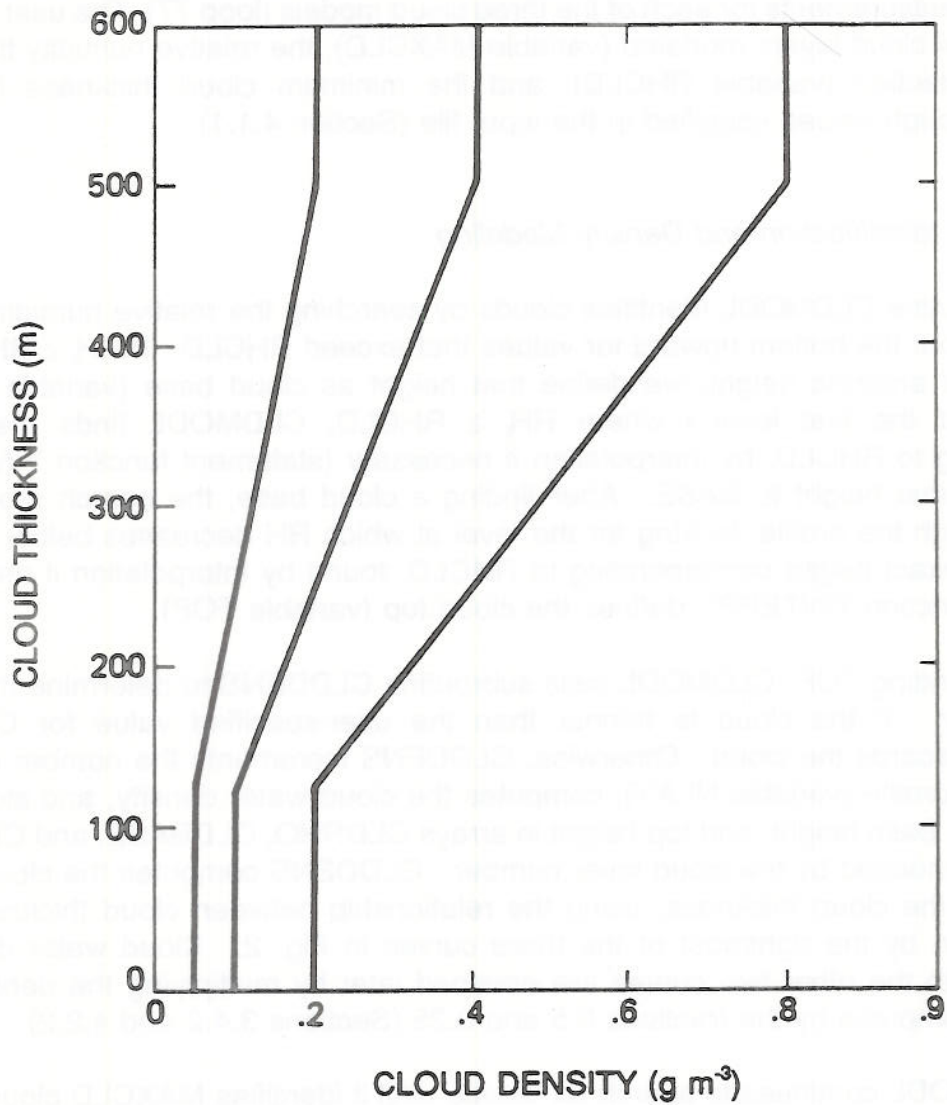


Fig. 2. Relationship between cloud thickness and density on which cloud models are based (from Decker et al., 1978). From left to right, the curves correspond to the so-called cloud density fractions of 0.25, 0.50, and 1.00.

CLDPROF finds the profile level corresponding to each cloud base and top, saving the level numbers as limits for cloud layer integrations performed later. If there is no level precisely at a cloud base or top, CLDPROF calls ADDLEVEL to add a level (by interpolation) at the cloud boundary to the height, pressure, temperature, and relative humidity profiles. ADDLEVEL will output a message stop the program if the resulting number of profile levels exceeds the user-specified maximum (variable MAXL; Section 4.2.3).

For each identified cloud layer (variable LAYER), CLDPROF assigns the cloud water density contained in CLDRHO(LAYER) to the profile levels at heights CLDBASE(LAYER), CLDTOP(LAYER), and the levels between them. CLDMODL uses function CLDICE to determine, for each profile level, the fraction of cloud water density that should be treated as ice, given the level's temperature. Figure 3 depicts the relationship between ice fraction and temperature that CLDICE assumes.

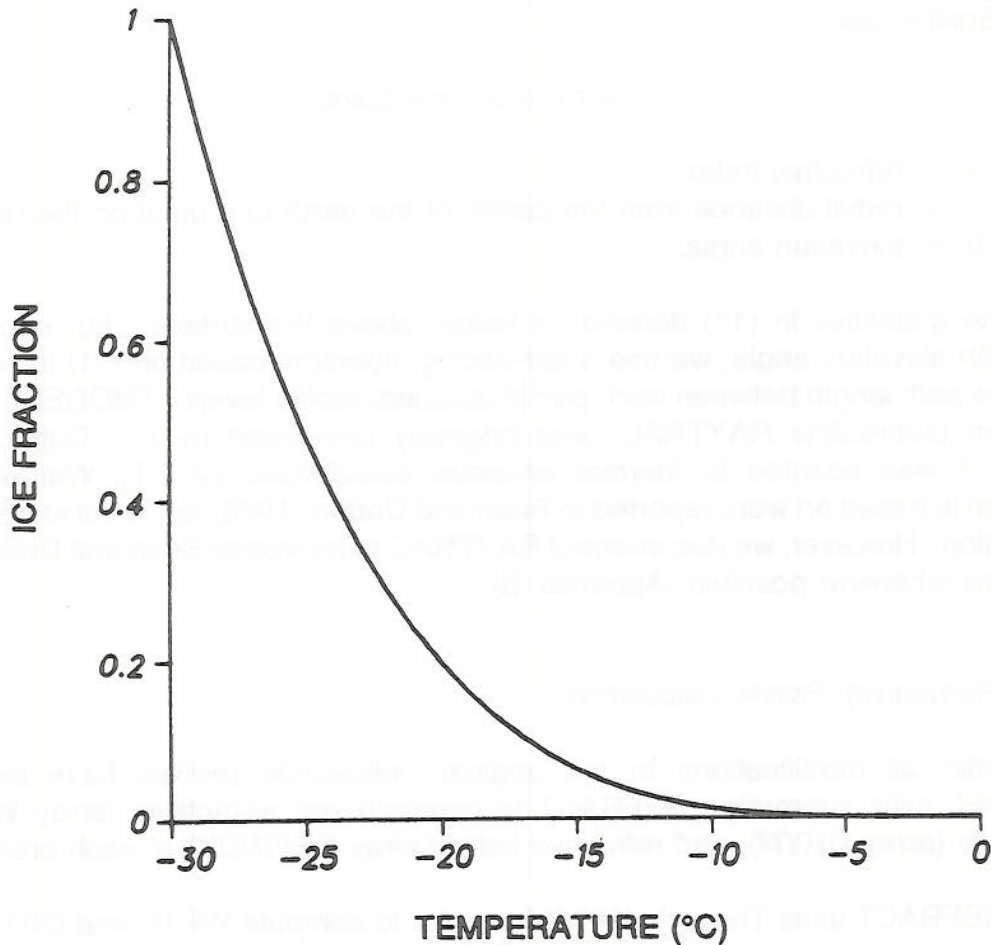


Fig. 3. Function of temperature that determines the fraction of modeled cloud water density that is treated as ice density.

The resulting liquid and ice density profiles, which represent the cloud model depicted by the rightmost curve in Fig. 2, will be written to the output file with the height, pressure, temperature, and relative humidity profiles if profile output is selected (variable IPROF; Section 4.1.1). Liquid and ice density profiles corresponding to the other two curves will be created later (subroutine CLDABS) by multiplying DENLIQ and DENICE by the fractions 0.5 and 0.25 (array DENFRAC), respectively.

3.3. Modeling the Ray Path

When simulating T_b for a zenith-pointing radiometer, we assume that the radiosonde height profile represents the vertical line-of-sight ray path coordinate, s , defined in (2). For nonzenith elevation angles, the ray path must be adjusted to account for the bending of radio waves by atmospheric refraction. To model the ray path behavior, we assume a spherically stratified atmosphere, for which the radio wave path obeys Snell's Law:

$$n r \cos \theta = \text{constant}, \quad (11)$$

where n = refractive index

r = radial distance from the center of the earth to a point on the ray path

θ = elevation angle.

All of the quantities in (11) depend on height above the surface. For each specified nonzenith elevation angle, we use a ray-tracing algorithm based on (11) to compute the refracted path length between each pair of adjacent profile levels. TMODEL's ray-tracing algorithm (subroutine RAYTRAC) was originally developed by E.J. Dutton and G.D. Thayer; it was adapted to thermal emission calculations by E.R. Westwater. The algorithm is based on work reported in Bean and Dutton (1966) but is not identical to their formulation. However, we documented RAYTRAC to reference Bean and Dutton's (1966) equations wherever possible (Appendix B).

3.3.1. Refractivity Profile Calculation

After all modifications to the original radiosonde profiles have been made, TBMODEL calls subroutine REFRACT to compute wet refractivity (array WETN), dry refractivity (array DRYN), and refractive index (array REFINDX) at each profile level.

REFRACT uses Thayer's (1974) formulas to compute WETN and DRYN from the radiosonde profiles of temperature (array TK), relative humidity (array RH), and pressure (array P), even though he states that his formulas were intended for frequencies below 20 GHz. As a result, we neglect dispersion around the 22.235-GHz water line and in the 60-GHz oxygen complex (Liebe and Layton, 1987). REFRACT computes the refractive index profile from the refractivity profiles as

$$n = 1 + (N_w + N_d) \times 10^{-6},$$

where n = refractive index

N_w = wet refractivity

N_d = dry refractivity.

In addition, REFRACT calls subroutine VAPOR to compute profiles of water vapor pressure (array E) and water vapor density (array RHO) from TK and RH, using the Goff-Gratch formulas for saturation vapor pressure (e_s) over water and ice (List, 1963). A user-specified switch (variable ICE; Section 4.1.1) determines whether e_s over water is used throughout the profile or e_s over ice is used when the temperature is $\leq -10^\circ\text{C}$.

3.3.2. Path Length Profile Calculation

TBMODEL calls subroutine RAYTRAC to compute, for each specified elevation angle, the path length between each pair of adjacent profile levels. RAYTRAC stores the resulting path length profile in two-dimensional array DS, whose columns correspond to the array of specified elevation angles (ANGLES). Row I of DS contains the path length between profile levels I and I-1, so Row 1 of DS contains zeroes by definition. Array IRAY, whose elements also correspond to those of ANGLES, contains flags that tell TBMODEL whether the ray path for each elevation angle is valid (1) or invalid (0).

If an elevation angle is within 1° of zenith, RAYTRAC stores the radiosonde height difference profile in the corresponding column of DS. Path length calculations for nonzenith elevation angles require a valid refractive index profile. If RAYTRAC encounters an invalid refractive index value, it outputs a message and zeroes all elements of IRAY that correspond to nonzenith elevation angles. At low elevation angles, refractive ducting may occur, bending the ray back toward the earth. If RAYTRAC encounters ducting conditions for a given angle, it outputs a message and zeroes the corresponding element of IRAY. RAYTRAC counts the number of elevation angles with valid ray paths and stores the result in variable NANGOK, which is written to the header line of the output file (Section 4.1.4).

3.3.3. Path Integral Calculations

TBMODEL employs three different subroutines to numerically integrate various quantities over the ray path: EXPINT1, EXPINT2, and CLDINT.

Subroutine EXPINT1 integrates a profile (array X) along the entire length of the ray path defined by column K of array DS and multiplies the result by an input factor (variable FACTOR) to adjust the units. The algorithm inside EXPINT1 is based on the assumption that the parameter in X decays exponentially with height within the layer defined by a pair of adjacent levels. Appendix A gives a detailed description of the algorithm. EXPINT1 is used to integrate profiles of water vapor density (array RHO), wet refractivity (array WETN), and dry refractivity (array DRYN) to obtain precipitable water vapor (variable SRHO), path delay (excess path length) due to water vapor (variable SWETN), and path delay (excess path length) due to dry air (variable SDRYN), respectively. TBMODEL calls EXPINT1 with FACTOR = 0.1 to obtain these three integrals in centimeters.

EXPINT2 uses the same algorithm as EXPINT1, but it has variable integration limits (variables IBEG and IEND) and operates on column J of a two-dimensional array X. EXPINT2 saves the integrals over each individual layer as an array (XDS), which it returns along with the path integrals. EXPINT2 is used to compute layer-integrated absorption profiles (array TAULAY) for radiative intensity calculations and total integrals of absorption due to water vapor (variable TAUW), dry air (variable TAUD), cloud liquid (variable TAUL), and cloud ice (variable TAUI). EXPINT2's variable integration limits were designed to permit integration between cloud base and top.

Subroutine CLDINT integrates cloud water density with a linear algorithm, adding together the areas of the trapezoids defined by the cloud water density at each profile level between the base (LBASE) and top (LTOP) of each cloud layer. CLDINT multiplies the result by 0.1 so that density in grams per cubic meter integrated over height in kilometers gives the integral in centimeters. TBMODEL calls CLDINT to compute integrated liquid (variable SLIQ) and integrated ice (variable SICE) over the ray path defined by column K of path length array DS.

A subtle difference exists between integrating cloud-dependent quantities between cloud boundaries and integrating them over the entire profile. Even though the cloud-dependent quantities are zero at profile *levels* outside the cloud boundaries, the values associated with the *layers* just below cloud base and just above cloud top will be nonzero, artificially extending the cloud in both directions. An older version of TBMODEL, which integrated cloud water density and cloud absorption over the entire atmosphere, minimized this effect by limiting the thickness of these transition layers to 10 m. To accomplish this, the program inserted levels 10 m below cloud base and 10 m above cloud top, modifying the cloud water density profiles as well as the pressure, height, temperature, and humidity profiles. We decided to abandon this approach and integrate cloud water density and cloud absorption directly from cloud base to cloud top instead.

3.4. Modeling Absorption

As discussed in Section 2.3, we assume that attenuation in the microwave region of interest is due entirely to absorption by dry air (α_{dry}), water vapor (α_{wet}), and cloud water (α_{liq} and α_{ice}). Therefore, the sum [$\alpha_{dry}(s) + \alpha_{wet}(s) + \alpha_{liq}(s) + \alpha_{ice}(s)$] represents the $\alpha(s)$ in Section 2, where the last two terms are zero for clear sky conditions.

3.4.1. Clear-Sky Absorption Calculation

TBMODEL calls subroutine CLRABS to compute profiles of $\alpha_{dry}(s)$ (array ADRY) and $\alpha_{wet}(s)$ (array AWET) for each specified frequency (array FRQ) from the radiosonde profiles of pressure, temperature, and humidity. CLRABS also computes the modified Plank function value defined in (4), $\tilde{B}(T)$ (array BOFT), for each frequency from the

temperature at each profile level. CLRABS calls subroutines H2O and O2 to compute AWET and ADRY, respectively. These subroutines contain the absorption model of Liebe and Layton (1987) with interference coefficients of Rosenkranz (1988). Both H2O and O2 call subroutine SHAPE to compute the imaginary part of the complex line shape function. AWET, ADRY, and BOFT are two-dimensional arrays whose columns correspond to the elements of FRQ.

For each specified elevation angle (loop 44 with index K) with a valid ray path ($IRAY(K) > 0$), TBMODEL calls subroutine TAUCALC to integrate absorption at each frequency along the ray path (loop 33 with index J). TAUCALC integrates column J of AWET and ADRY along the ray path defined by column K of array DS to obtain $\tau_{wet}(0,\infty)$ (variable TAUW) and $\tau_{dry}(0,\infty)$ (variable TAUD), respectively, where $\tau(a,b)$ was defined in (3). TAUCALC calls subroutine EXPINT2 to obtain TAUW, TAUD, and the respective integrals over each profile layer, $\tau_{wet}(i-1,i)$ and $\tau_{dry}(i-1,i)$. TAUCALC adds $\tau_{wet}(i-1,i)$ and $\tau_{dry}(i-1,i)$ together for all i to form a profile of clear-sky, layer-integrated absorption (array TAULAY). If clouds were detected in the profile, TBMODEL stores TAULAY for angle K and frequency J in the three-dimensional array TAUCLR for later use (loop 22).

3.4.2. Cloud Absorption Calculation

If the radiosonde profile contains cloud, as defined by the user-specified relative humidity threshold and minimum thickness (Section 3.2.1), TBMODEL repeats loop 77 (index MODEL) for each cloud density model. The number of cloud models is controlled by variable NMODLS, whose value is assigned in subroutine CLDPROF (Section 3.2.2).

TBMODEL calls subroutine CLDABS to create the cloud liquid and ice density profiles (arrays DENL and DENI) specific to model number MODEL. CLDABS multiplies the cloud density profiles originally constructed in subroutine CLDMODL (arrays DENLIQ and DENICE) by the model-specific fraction, DENFRAC(MODEL) (Sections 3.2.2 and 4.2.2). CLDABS then computes $\alpha_{liq}(s)$ (array ALIQ) and $\alpha_{ice}(s)$ (array AICE) at each profile level for each specified frequency (array FRQ), using the algorithm of Westwater (1974). ALIQ and AICE are two-dimensional arrays, whose columns correspond to the elements of FRQ.

For each specified elevation angle (loop 66 with index K) with a valid ray path ($IRAY(K) > 0$), TBMODEL calls subroutine CLDTAU for each specified frequency (loop 55 with index J) to integrate the cloud liquid and ice absorption over the ray path and construct a profile of layer-integrated absorption (array TAULAY) that is consistent with the modeled cloud density profiles. For each of the NLAY cloud layers identified in the profile (loop 33 with index L), CLDTAU calls subroutine TAUCALC to integrate column J of ALIQ and AICE along the segment of the ray path (column K of array DS) that lies between cloud base (level LBASE(L)) and cloud top (level LTOP(L)). These integrals represent $\tau_{liq}(base,top)$ (variable TL) and $\tau_{ice}(base,top)$ (variable TI), respectively, where

τ was defined in (3). TAUCALC calls subroutine EXPINT2 to perform the integrations (Section 3.3.3). EXPINT2 also returns profiles of layer-integrated liquid and ice absorption, which TAUCALC adds together to form a profile of cloud layer-integrated absorption (array TAUCLD). CLDTAU computes $\tau_{liq}(0,\infty)$ (variable TAUL) and $\tau_{ice}(0,\infty)$ (variable TAU) by adding together the TL and TI values from each cloud layer. CLDTAU adds the values in TAUCLD to the corresponding elements of the clear-sky, layer-integrated absorption profile (array TAULR) to obtain a cloudy-sky, layer-integrated absorption profile (array TAULAY).

3.5. Modeling T_b and T_{mr}

For each elevation angle with a valid ray path, TBMODEL calls subroutine RADIANC to compute the theoretical brightness temperatures (T_b) with and without the cosmic background that an upward-viewing microwave radiometer would observe at a given channel frequency, given the atmospheric conditions defined by the radiosonde profiles with or without the cloud model. RADIANC also computes the mean radiating temperature (T_{mr}) of the atmosphere and, optionally, the T_{mr} of the lowest cloud layer. The equations referenced in this section were derived in Section 2. The discussion below applies to a single combination of elevation angle and frequency.

For a profile containing NL levels, RADIANC expresses the integral that represents the atmospheric term in (2) as the sum of the integrals over each of the NL-1 layers as

$$\int_0^{\infty} B(T(s)) \alpha(s) e^{-\tau(0,s)} ds = \sum_{i=2}^{NL} \int_{s_{i-1}}^{s_i} B(T(s)) \alpha(s) e^{-\tau(0,s)} ds.$$

Using (7), we can express the layer integrals equivalently as

$$\int_{s_{i-1}}^{s_i} B(T(s)) \alpha(s) e^{-\tau(0,s)} ds = B(T_{mr}) e^{-\tau(0,s_{i-1})} [1 - e^{-\tau(s_{i-1},s)}].$$

For a given frequency, RADIANC approximates $B(T_{mr})$ for the atmospheric layer between profile levels i and $i-1$ as

$$BOFTLAY = \frac{BOFT(i-1) + BOFT(i) e^{-TAULAY(i)}}{1 + e^{-TAULAY(i)}},$$

where $BOFT(i) = \hat{B}(T)$ defined in (4) for profile level i
 $TAULAY(i) = \tau(s_{i-1},s)$ defined in (3).

BOFTLAY is a weighted average of the $\tilde{B}(T)$ from the two profile levels that form the layer. The exponential weight, $e^{-\text{TAULAY}(i)}$, represents the attenuation of $\tilde{B}(T)$ over the layer between profile levels i and $i-1$. With this definition, the integral over a single layer becomes

$$\text{BATMLAY} = \text{BOFTLAY} e^{-\text{TAU}(i-1)} (1 - e^{-\text{TAULAY}(i)}),$$

where TAU (i-1) represents $\tau(0, s_{i-1})$ from (3). Note that the attenuation factor $e^{-\text{TAU}(i-1)}$ goes to 1 for the first layer above the antenna. If the absorption is large enough to cause exponential underflow in the above equations (i.e., $\tau > \text{EXPMAX}$; Section 4.2.3), we assume that the intensity contribution from the layer was completely attenuated (BATMLAY = 0).

RADIANC defines the integrated intensity from the antenna to level i as

$$\text{BOFTATM}(i) = \sum_{i=2}^i \text{BATMLAY},$$

so that BOFTATM (NL) represents the total atmospheric term in (2) for a profile with NL levels. Similarly, TAU (NL) represents $\tau(0, \infty)$. Therefore, the total intensity defined in (2), including the cosmic background, becomes

$$\text{BOFTOTL} = \text{BOFTBG} e^{-\text{TAU}(NL)} + \text{BOFTATM}(NL),$$

where BOFTBG represents the cosmic background intensity, $\tilde{B}(T_{bg})$. Then the mean radiative intensity, $\tilde{B}(T_{mr})$, defined in (9) is

$$\text{BOFTMR} = \frac{\text{BOFTATM}(NL)}{1 - e^{-\text{TAU}(NL)}}.$$

Statement function BRIGHT implements (5) to compute temperatures from their respective intensities. RADIANC uses BRIGHT to compute T_b without the cosmic background term (variable TBATM), T_b with the cosmic background term (variable TBTOTL), and the mean radiating temperature of the atmosphere (variable TMR) from BOFTATM (NL), BOFTOTL, and BOFTMR, respectively.

If an argument is set to indicate cloud in the profile (variable ICLD=1), RADIANC uses the integrals in BOFTATM and TAU to compute the T_{mr} of the lowest cloud layer identified (Section 2.2). RADIANC computes the intensity integrated over a single cloud

layer with base and top at profile levels IBASE and ITOP, respectively, from (10) as

$$BOFTCLD = \frac{BATMCLD e^{\tau(IBASE)}}{(1 - e^{-\tau(TAUCLD)})}$$

where $BATMCLD = BOFTATM(ITOP) - BOFTATM(IBASE)$
 $TAUCLD = \tau(ITOP) - \tau(IBASE)$.

If $TAUCLD > EXPMAX$ (Section 4.2.3), RADIANC treats the denominator as unity. Note that the exponential, $e^{-\tau(0,base)}$, that appears in the denominator of (10) is expressed equivalently as $e^{+\tau(0,base)}$ in the numerator above. Again, RADIANC uses BRIGHT to compute T_{mr}^{cloud} (variable TMRCLD) from $\tilde{B}(T_{mr}^{cloud})$ (variable BOFTCLD). If $ICLD = 0$, RADIANC sets TMRCLD to zero.

4. USING PROGRAM TBMODEL

TBMODEL is a Fortran program whose primary purpose is to compute the brightness temperature (T_b) that an upward-looking microwave radiometer would observe, given the atmospheric conditions defined by the pressure, temperature, and humidity profiles measured by radiosonde and an optional model of cloud water density. The user provides the radiosonde data and specifies the height, elevation angle, and frequency of each radiometer channel being simulated. The user also controls the definition of a cloud and the number and thickness of cloud layers modeled (if any). This section is designed to help users run program TBMODEL with parameter values of their choice.

TBMODEL evolved over several decades on several generations of Control Data Corporation (CDC) mainframe computers with minimal documentation. As we deciphered documented, and modified the algorithms, we restructured the Fortran into something resembling Fortran 77. Appendix B contains listings of the Fortran source code in its present form. Figure 1 (p 7) summarized TBMODEL's structure and gave a brief description of each subroutine. Since our primary design objective was readability, we minimized use of COMMON and we did not use variable array dimensions. TBMODEL should run on most computers with little modification, because we developed it under a rather restrictive operating system, by modern standards. For example, the names of all files, variables, and subroutines contain seven characters or less, with no extensions or special characters (e.g., the underscore); there is no directory structure built into the program.

Section 4.1 describes each file that TBMODEL accesses, including the input file, where selected parameters can be changed without modifying the program itself. Section 4.2 describes the parameters assigned by subroutine INITCON, which can only be

changed by modifying INITCON directly. We capitalize the names of variables, subroutines, and files for emphasis.

4.1. Files Accessed by TBMODEL

TBMODEL accesses the five files shown in Table 1 with standard Fortran READ and WRITE statements directed at the respective variable names. The unit numbers shown were assigned to the corresponding variable names in subroutine INITCON (Appendix B) and could be changed there. They are passed through COMMON block IOFILES to subroutines where they are needed. There are no OPEN or CLOSE statements in the program; file assignments are made in the execution procedure. The PROGRAM statement, which is the first statement in the program, is part of the CDC file assignment mechanism. Non-CDC users will want to discard it. CDC users should make sure that the unit numbers assigned to each file in subroutine INITCON match the TAPE numbers in the PROGRAM statement and in their procedure.

Table 1. TBMODEL file assignments. Unit numbers are assigned in subroutine INITCON (Appendix B).

File Name	Description	Variable Name	Unit Number
TBSPECS	Input file containing user specifications	SPECFIL	1
user-supplied	Input file containing radiosonde data	RAOBFIL	2
TBHEADR	Text file containing output file header	HEADFIL	3
user-supplied	Output file containing results from TBMODEL	MODLFIL	4
user-supplied	Output file containing messages from TBMODEL	MESSFIL	5

4.1.1 Input File Containing User Specifications

Figure 4 shows an example of input file TBSPECS. The values specified for the last six parameters are the same values that were hard-wired into the old software. We show these values to enable users to replicate the traditional model as closely as possible. Subroutine RDSPECS reads the input file contents from unit SPECFIL (Table 1). If preferred, RDSPECS could be modified to prompt the user for these same parameters at the terminal, although typing in all the responses could become tedious. Where multiple quantities are requested or selected, they must be separated by a comma, space, or carriage return. Except for the date and time ranges, all of the values specified here will appear in the output file containing the results (Section 4.1.4). This section elaborates on each parameter specified in the input file.

TITLE, FOR OUTPUT WITH FILE DESCRIPTION HEADER (UP TO 70 CHARACTERS):
EXAMPLE FOR DOCUMENT

SPECIFY WHETHER (1) OR NOT (0) TO OUTPUT EXTRAPOLATED, INTERPOLATED
PROFILES (INCLUDING CLOUD DENSITY PROFILES) TO FILE WITH RESULTS:
1

SPECIFY NUMBER OF FREQUENCIES (MAXIMUM OF 11):
2

SPECIFY FREQUENCIES (GHZ):
20.6 31.65

SPECIFY NUMBER OF ELEVATION ANGLES (MAXIMUM OF 10):
2

SPECIFY ELEVATION ANGLES (DEGREES):
90. 19.47122

SPECIFY SIMULATED ANTENNA ELEVATION (KM MSL):
(ANTENNA ELEVATION BELOW RAOB LAUNCH SITE IS RESET TO SITE ELEVATION.)
0.

SPECIFY LATITUDE OF RADIOSONDE LAUNCH SITE (DEGREES):
(REQUIRED FOR EXTRAPOLATION ALGORITHM)
39

SPECIFY RANGE OF 2-DIGIT YEARS TO INPUT FROM RAOB DATA FILE (1ST YR, LAST YR):
74 74

SPECIFY NUMBER OF MONTHS DESIRED FROM EACH YEAR:
1

SPECIFY MONTH NUMBERS DESIRED (ORDER-INDEPENDENT):
1

SPECIFY RANGE OF DAYS DESIRED (1ST DAY, LAST DAY):
10 10

SPECIFY RANGE OF HOURS DESIRED (1ST HOUR, LAST HOUR):
12 12

SPECIFY MAXIMUM ACCEPTABLE PRESSURE AT HIGHEST LEVEL OF INPUT PROFILE (MB):
(MAXIMUM OF 300; PROFILES THAT TERMINATE BELOW THIS LEVEL WILL BE DISCARDED)
300.

SPECIFY MAXIMUM ACCEPTABLE PRESSURE DIFFERENCE BETWEEN ADJACENT LEVELS (MB):
(ZERO SELECTS AN ALGORITHM THAT RELAXES THIS CONSTRAINT WITH HEIGHT.)
0.

SELECT SATURATION VAPOR PRESSURE CALCULATION OVER WATER (0) OR WATER/ICE (1):
0

SPECIFY THE MAXIMUM NUMBER OF CLOUD LAYERS TO MODEL PER PROFILE (4 MAXIMUM):
3

SPECIFY RELATIVE HUMIDITY THRESHOLD FOR CLOUD DETECTION (BETWEEN 0 AND 1):
.95

SPECIFY THE MINIMUM ACCEPTABLE CLOUD LAYER THICKNESS (KM):
0.001

Fig. 4. Example of the input file. The parameters specified here produced the output shown in Fig. 5. The specified values shown for the last six parameters will produce results comparable to older versions of the software.

Output File Title

The title (array TITLE), which will appear on line 1 of the output file, gives the user a place to annotate each run. Note the 70-character limit, designed to restrict the output title line to 80 characters.

Profile Output Switch

The profile output switch (variable IPROF) determines whether (IPROF=1) or not (IPROF=0) the profiles of height, pressure, temperature, relative humidity, cloud liquid density, and cloud ice density that were used in the calculations will be written to the output file along with the results. This option allows the user to see how TBMODEL modified the original radiosonde profiles and how it modeled clouds before making the radiative transfer calculations. Note that these profiles, not the original radiosonde profiles, represent the atmospheric state that produced the results. Therefore, these profiles must be saved with the corresponding results if the user intends to compute retrieval coefficients for any type of profile.

Frequencies and Elevation Angles

Channel frequencies (array FRQ) must be specified in gigahertz, and elevation angles (array ANGLES) must be specified in degrees. The maximum number of frequency channels (variable NFRQ) and elevation angles (variable NANG) shown in Fig. 4 represent the array dimensions currently specified in TBMODEL. Therefore, RDSPECS will input only the first 11 frequencies and the first 10 angles listed in the input file. Changes to these maxima require changes to the corresponding array dimensions throughout the program. These maxima represent a compromise between user needs (e.g., 500 profile levels) and program size constraints imposed by the CDC operating system under which TBMODEL evolved. The maxima differ from each other so that a user can determine immediately from the array dimensions whether a variable is indexed by angle, frequency, or both.

Simulated Antenna Height

The simulated antenna height (variable ZSTART), which must be given in kilometers above sea level (MSL), serves three purposes: 1) the ray-tracing algorithm that computes the slant path for off-zenith elevation angles requires it; 2) it permits calculation of T_p from upward-looking aircraft- as well as ground-based platforms; and 3) it gives all profiles a vertical coordinate of height above the simulated antenna, regardless of the radiosonde launch site elevation used for the simulation. The antenna height actually used in the calculations (variable Z0) is set to ZSTART or the height of the lowest profile level, whichever is higher. Therefore, setting ZSTART to 0.0 sets the simulated antenna height to the first level in each profile, regardless of the individual radiosonde launch site elevation.

Radiosonde Launch Site Latitude

The latitude of the radiosonde launch site (variable RAOBLAT) is required for the radiosonde extrapolation algorithm (Section 3.1.2), whose three sets of regression coefficients correspond to the following three latitude intervals: [0,30],[30,60],[60,90]. If the radiosonde data set contains profiles from a variety of latitudes, subroutines PROFILE and RDRAOB should be modified to overwrite the value specified in the input file.

Date and Time Ranges

Subroutine RDRAOB uses the input range of years (variables YRFROM and YRTHRU), months (array MONTHS), days (variables DAYFROM and DAYTHRU), and hours (variables HRFROM and HRTHRU) to search through a multiyear radiosonde data set (Section 3.1.1). Month numbers can be specified in any order, but RDSPEC will input a maximum of 12 months.

Short Radiosonde Discard Threshold

The maximum acceptable pressure (millibars) at the top of the input profile (variable PMAX) is used to discard radiosonde profiles that terminate below the specified level (Section 3.1.1). Since the extrapolation algorithm requires that profiles reach at least 300 mb, RDSPECS automatically resets this parameter to 300 if a larger value is specified.

Profile Level Density Threshold

The maximum acceptable pressure difference (millibars) between adjacent levels (variable DPMAX) determines the number of profile levels that will be inserted to eliminate excessive distance between original profile levels. Specifying zero for this parameter allows the distance to increase with height, using the algorithm described in Section 3.1.4. Specifying small, nonzero values for this parameter may result in a number of profile levels that exceeds the array dimension safeguard variable (MAXL; Section 4.2.3). If MAXL is exceeded, TBMODEL outputs a message and stops.

The value of DPMAX that is appropriate depends on the purpose of the T_b simulations. For example, when we simulate T_b for radiometer calibration, where the absolute value is more important than computation time, we set DPMAX to 5 mb, producing a very dense vertical grid. However, when we simulate T_b from thousands of radiosonde profiles at once in order to compute retrieval coefficients, the absolute value of T_b is less critical, and computation time becomes important. For this purpose, we set DPMAX to 0, allowing the pressure difference to increase with distance from the simulated antenna, assuming that the lower levels dominate the radiometer response (Section 3.1.4).

Saturation Vapor Pressure Switch

The saturation vapor pressure (e_s) switch (variable ICE) determines whether the Goff-Gratch formula for e_s over water is used throughout the profile (ICE=0) or the formula for e_s over ice is used when the temperature is $\leq -10^\circ\text{C}$ (ICE=1). WPL historically used e_s over water only, regardless of its temperature, to be consistent with the National Weather Service (NWS), who reportedly used e_s over water in their conversion from measured relative humidity to published dewpoint depression values. Now that NWS is reporting relative humidity, the question is reopened.

Maximum Number of Cloud Layers

The maximum number of cloud layers to model per profile (variable MAXCLD) lets the user select single-layer (MAXCLD = 1), multilayer (MAXCLD > 1), or no cloud modeling at all (MAXCLD = 0). Present array dimensions and output file format permit a maximum of four cloud layers per profile, so RDSPECS resets MAXCLD to 4 if a larger value is specified.

Cloud Detection Threshold

The relative humidity (*RH*) threshold for cloud detection (variable RHCLD) must be specified in fractional form ($0 \leq \text{RHCLD} \leq 1$). Clouds will be modeled at profile levels with relative humidity $\geq \text{RHCLD}$, subject to the constraints imposed by the other cloud parameters specified (Section 3.2). WPL's traditional cloud model used RHCLD = 0.95. However, recent changes to radiosonde humidity measurements and/or reporting procedures eliminated *RH* values above this threshold, so the newer data require a lower threshold, such as 0.90.

Minimum Cloud Thickness

The minimum acceptable cloud layer thickness (variable CLDMIN, in kilometers) allows the user to control the thickness of the clouds modeled. Subroutine CLDDENS discards detected cloud layers thinner than CLDMIN.

4.1.2. Input File Containing Radiosonde Data

The user supplies the radiosonde data that will represent the atmospheric state in the simulations. TBMODEL has a subroutine (RDRAOB) that reads radiosonde data from unit RAOBFIL (Table 1). However, since RDRAOB was specifically designed to read the multiyear, binary data sets that M.J. Falls created for our group, most users will need to modify RDRAOB or substitute another subroutine to fit their data file format. Section 3.1.1 contains the information required to make the new subroutine compatible with TBMODEL.

4.1.3. Text File Containing Output File Header

Subroutine HEADOUT writes a descriptive header to the beginning of the output file on unit MODLFIL (Table 1). Figure 5a shows an example of the portion of the output file that HEADOUT wrote, given the values specified in Fig. 4 and those assigned in subroutine INITCON (Section 4.2). First, HEADOUT writes the user-defined title line and a summary of selected parameter values that affect the calculations, are not required to read the results, and do not vary from profile to profile. Then HEADOUT reads the text in file TBHEADR from unit HEADFIL and rewrites it to the output file, MODLFIL.

4.1.4. Output File Containing Results

Figure 5b illustrates the remainder of the output file, which was written to MODLFIL by the main program. Note that the upper profile levels were omitted to save space. This section describes the output file contents in Fig. 5b with more detail than was practical in the file header (Fig. 5a).

Header Line

TBMODEL writes one header line that contains all the information required to identify the profile and read the results. The header line contains the following:

- Date of the radiosonde launch (mm dd yy)
- Time of the radiosonde launch (hh mm)
- Latitude of the radiosonde launch (degrees)
- Simulated antenna height (km MSL)
- Number of cloud layers identified
- Profile output switch (1=yes; 0=no)
- Number of levels in the profiles after all modifications
- Number of cloud models
- Number of elevation angles with valid ray paths
- Number of channel frequencies.

Cloud Boundaries

If at least one cloud layer was modeled in the profile, TBMODEL writes a second line that contains the base and top of each cloud layer (kilometers above the simulated antenna height), in pairs (base top base top ...). The number of pairs is the number of cloud layers given in the *header line*. If no clouds are modeled in the profile, this line does not exist.

TITLE: EXAMPLE FOR DOCUMENT

THE RESULTS IN THIS FILE REFLECT THE FOLLOWING USER SPECIFICATIONS:

DISCARD PROFILES THAT TERMINATE BELOW THIS PRESSURE LEVEL (MB):	300.
INSERT LEVEL IF PRESSURE DIFFERENCE BETWEEN LEVELS EXCEEDS (MB): (ZERO ALLOWS THE PRESSURE DIFFERENCE TO INCREASE WITH HEIGHT.)	0.
CALCULATE SATURATION VAPOR PRESSURE OVER WATER (0) OR WATER/ICE (1):	0
COSMIC BACKGROUND BRIGHTNESS TEMPERATURE (K):	2.75
MAXIMUM NUMBER OF CLOUD LAYERS TO MODEL PER PROFILE:	3
RELATIVE HUMIDITY THRESHOLD FOR CLOUD DETECTION (FRACTION):	.95
MINIMUM ACCEPTABLE CLOUD THICKNESS (KM):	.001

FOR EACH PROFILE INPUT, THIS FILE CONTAINS THE FOLLOWING, IN THE ORDER GIVEN:

HEADER LINE: DATE OF RADIOSONDE LAUNCH (MM DD YY)
TIME OF RADIOSONDE LAUNCH (HH MM)
LATITUDE OF RADIOSONDE LAUNCH (DEGREES)
SIMULATED ANTENNA HEIGHT (KM MSL)
NUMBER OF CLOUD LAYERS IDENTIFIED
PROFILE OUTPUT SWITCH (1=YES; 0=NO)
NUMBER OF LEVELS IN OUTPUT PROFILES
NUMBER OF CLOUD MODEL BLOCKS
NUMBER OF ELEVATION ANGLES WITH VALID RAY PATHS
NUMBER OF CHANNEL FREQUENCIES

OPTIONAL CLOUD BOUNDARY LINE (ABSENT IF NUMBER OF CLOUD LAYERS = 0):
BASE AND TOP OF EACH CLOUD LAYER IDENTIFIED (KM ABOVE ANTENNA)

OPTIONAL PROFILE BLOCK (ABSENT IF PROFILE OUTPUT SWITCH = 0):
1 LINE PER PROFILE LEVEL, CONTAINING:
HEIGHT (KM ABOVE SIMULATED ANTENNA HEIGHT)
PRESSURE (MB)
TEMPERATURE (K)
RELATIVE HUMIDITY (FRACTION)
CLOUD LIQUID DENSITY (G/M3) FOR MODEL WITH DENSITY FRACTION = 1.00
CLOUD ICE DENSITY (G/M3) FOR MODEL WITH DENSITY FRACTION = 1.00

CLEAR-SKY BLOCK (REPEATS FOR EACH ELEVATION ANGLE WITH A VALID RAY PATH):
1 LINE OF FREQUENCY-INDEPENDENT QUANTITIES, CONTAINING:
ELEVATION ANGLE (DEGREES)
PATH-INTEGRATED WATER VAPOR (CM)
EXCESS PATH LENGTH DUE TO WATER VAPOR (CM)
EXCESS PATH LENGTH DUE TO DRY AIR (CM)
1 LINE PER FREQUENCY, CONTAINING:
CHANNEL FREQUENCY (GHZ)
BRIGHTNESS TEMPERATURE WITH COSMIC BACKGROUND TERM (K)
BRIGHTNESS TEMPERATURE WITHOUT COSMIC BACKGROUND TERM (K)
MEAN RADIATING TEMPERATURE OF THE ATMOSPHERE (K)
0. (MEAN RADIATING TEMPERATURE OF NONEXISTENT CLOUD)
PATH-INTEGRATED WATER VAPOR ABSORPTION (NP)
PATH-INTEGRATED DRY AIR ABSORPTION (NP)

OPTIONAL CLOUD MODEL BLOCK (REPEATS FOR EACH CLOUD MODEL AND ELEVATION ANGLE):
1 LINE OF FREQUENCY-INDEPENDENT QUANTITIES, CONTAINING:
ELEVATION ANGLE (DEGREES)
PATH-INTEGRATED CLOUD LIQUID DENSITY (CM)
PATH-INTEGRATED CLOUD ICE DENSITY (CM)
CLOUD DENSITY FRACTION (MULTIPLIER FOR CLOUD DENSITY PROFILES)
1 LINE PER FREQUENCY, CONTAINING:
CHANNEL FREQUENCY (GHZ)
BRIGHTNESS TEMPERATURE WITH COSMIC BACKGROUND TERM (K)
BRIGHTNESS TEMPERATURE WITHOUT COSMIC BACKGROUND TERM (K)
MEAN RADIATING TEMPERATURE OF THE ATMOSPHERE (K)
MEAN RADIATING TEMPERATURE OF LOWEST CLOUD LAYER IDENTIFIED (K)
PATH-INTEGRATED CLOUD LIQUID ABSORPTION (NP)
PATH-INTEGRATED CLOUD ICE ABSORPTION (NP)

Fig. 5a. Beginning of the output file resulting from the user specifications in Fig. 4, showing the title line, user specification summary, and description of file contents.

1	10 74	12 0	39.00	1.611	2	1	86	3	2	2	(1)
.129	1.169	2.565	3.231								(2)
.000	830.60	260.86	.840E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00			
.041	826.20	260.93	.875E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00			
.082	821.80	261.01	.910E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00			
.123	817.40	261.08	.945E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00			
.129	816.74	261.10	.950E+00	.779E+00	.209E-01						
.164	813.00	261.16	.980E+00	.780E+00	.205E-01						
.205	808.67	261.16	.983E+00	.780E+00	.205E-01						
.246	804.33	261.16	.987E+00	.780E+00	.205E-01						
.287	800.00	261.16	.990E+00	.780E+00	.205E-01						
.330	795.50	261.21	.995E+00	.780E+00	.201E-01						
.374	791.00	261.26	1.00E+01	.780E+00	.198E-01						
.454	782.80	261.18	1.00E+01	.780E+00	.203E-01						
.535	774.60	261.10	1.00E+01	.779E+00	.209E-01						
.616	766.40	261.02	1.00E+01	.779E+00	.214E-01						
.699	758.20	260.94	1.00E+01	.778E+00	.220E-01						
.782	750.00	260.86	1.00E+01	.777E+00	.226E-01						
.916	737.00	261.06	1.00E+01	.779E+00	.212E-01						(3)
1.052	724.00	261.26	1.00E+01	.780E+00	.198E-01						
1.169	712.99	260.90	.950E+00	.778E+00	.223E-01						
1.180	712.00	260.86	.945E+00	.000E+00	.000E+00						
1.310	700.00	260.46	.890E+00	.000E+00	.000E+00						
1.447	687.50	259.12	.905E+00	.000E+00	.000E+00						
1.586	675.00	257.76	.920E+00	.000E+00	.000E+00						
1.802	656.00	257.76	.790E+00	.000E+00	.000E+00						
1.871	650.00	257.36	.770E+00	.000E+00	.000E+00						
2.166	625.00	255.16	.680E+00	.000E+00	.000E+00						
2.470	600.00	252.76	.880E+00	.000E+00	.000E+00						
2.565	592.38	251.94	.950E+00	.600E+00	.200E+00						
2.632	587.00	251.36	1.00E+01	.577E+00	.223E+00						
2.866	568.50	249.64	1.00E+01	.498E+00	.302E+00						
3.108	550.00	247.86	1.00E+01	.395E+00	.405E+00						
3.187	544.00	247.16	1.00E+01	.349E+00	.451E+00						
3.231	540.71	246.83	.950E+00	.325E+00	.475E+00						
3.554	517.00	244.36	.580E+00	.000E+00	.000E+00						

< upper profile levels not shown >

20.600	.90000E+02	.50384E+00	.35785E+01	.18877E+03	.20358E-01	.95637E-02					
31.650	10.143	7.923	252.855	.000	.80904E-02	.19662E-01					(4)
20.600	.19471E+02	.15088E+01	.10716E+02	.56179E+03	.60967E-01	.28562E-01					
31.650	24.210	22.120	253.057	.000	.24234E-01	.58722E-01					
20.600	.90000E+02	.28203E-01	.59291E-02	.25000E+00	.31569E-01	.73341E-04					
31.650	17.843	15.694	255.153	261.089	.69281E-01	.11268E-03					(5)
20.600	.19471E+02	.84491E-01	.17734E-01	.25000E+00	.94560E-01	.21937E-03					
31.650	45.305	43.404	255.433	261.090	.20753E+00	.33704E-03					
20.600	.90000E+02	.56406E-01	.11858E-01	.50000E+00	.63137E-01	.14668E-03					
31.650	25.304	23.222	255.919	261.089	.13856E+00	.22536E-03					(6)
20.600	.19471E+02	.16898E+00	.35468E-01	.50000E+00	.18912E+00	.43873E-03					
31.650	64.512	62.783	256.278	261.091	.41506E+00	.67407E-03					
20.600	.90000E+02	.11281E+00	.23717E-01	.10000E+01	.12627E+00	.29337E-03					
31.650	39.542	37.588	256.572	261.090	.27712E+00	.45073E-03					(7)
20.600	.19471E+02	.33796E+00	.70937E-01	.10000E+01	.37824E+00	.87746E-03					
31.650	97.928	96.497	257.085	261.092	.83011E+00	.13481E-02					

Fig. 5b. Continuation of Fig. 5a, showing (1) header line, (2) cloud boundary line, (3) lower portion of profiles where clouds were modeled, (4) clear-sky results, and (5-7) cloud model blocks for the cloud density fractions 0.25, 0.50, and 1.00, respectively.

Profiles

If the *profile output switch* is set (Section 4.1.1), TBMODEL outputs the atmospheric profiles in their final form, one line per profile level (see *header line* for the number of levels). Each line contains

Height (km above simulated antenna height)

Pressure (mb)

Temperature (K)

Relative humidity (fractional form)

Cloud liquid density (g m^{-3})

Cloud ice density (g m^{-3}).

The cloud density profiles correspond to a model with density fraction = 1. (Section 3.2.2).

Clear-Sky Results

TBMODEL treats each radiosonde as a clear profile first, regardless of whether or not the profile contains cloud. If the *header line* contains zero for the number of cloud models, the clear-sky results will be the only results in the output file. The block of output described in this section is repeated for each elevation angle for which a valid ray path was determined (see *header line* for the number of valid angles). TBMODEL writes one line containing the following quantities, which depend on angle but not on frequency:

Elevation angle (degrees)

Integrated water vapor (cm)

Path delay (excess path length) due to water vapor (cm)

Path delay (excess path length) due to dry air (cm).

This line is followed by a series of lines that contain quantities that depend on angle and frequency, one line per frequency (see *header line* for the number of frequencies). Each line contains

Channel frequency (GHz)

Brightness temperature (T_b) with the cosmic background term (K)

T_b without the cosmic background term (K)

Mean radiating temperature (T_{mr} , in K)

0. (T_{mr} of nonexistent cloud layer)

Integrated water vapor absorption (Np)

Integrated dry air absorption (Np).

Cloud Model Results

If the number of cloud models (NMODLS) given in the *header line* is greater than zero, TBMODEL writes NMODLS complete sets of cloud model results to the output file. Each set is analogous to the set described under *Clear-Sky Results* and is written in the same format. This section describes one set.

For each elevation angle with a valid ray path (see *header line* for the number of valid angles), TBMODEL writes one line containing the following quantities, which depend on angle but not on frequency:

- Elevation angle (degrees)
- Integrated cloud liquid density (cm)
- Integrated cloud ice density (cm)
- Cloud density fraction (multiplier for density profiles).

This line is followed by a series of lines containing quantities that depend on both angle and frequency, one line per frequency. Each line contains

- Channel frequency (GHz)
- Brightness temperature (T_b) with the cosmic background term (K)
- T_b without the cosmic background term (K)
- Mean radiating temperature (T_{mr} , in K)
- T_{mr} of lowest cloud layer identified (K)
- Integrated cloud liquid absorption (Np)
- Integrated cloud ice absorption (Np).

4.1.5. Output File Containing Messages

All messages output by program TBMODEL or any of its subroutines are written to unit MESSFIL (Table 1). Each message identifies the subroutine where the message originated and the date and time of the radiosonde profile involved, in the format shown below:

```
FROM RDRAOB: ( 3/13/90 11 0 Z) LAST RAOB BEFORE EOF ON UNIT 2.  NORMAL EXIT.
```

4.2. Parameters Assigned by Subroutine INITCON

TBMODEL assigns default values, cloud density fractions, safeguard variables, physical constants, and conversion factors inside subroutine INITCON (Appendix B). We did not include these parameters in the input file (Section 4.1.1) because users should seldom need to change their values. However, we put them all in one place so that users could find them easily if change became necessary. Most of these parameters are accessed by various subroutines through labeled COMMON, but parameters that are used only by the main program are passed back to it as arguments.

4.2.1. Default Values at Extrapolated Levels

COMMON block DEFAULT contains monthly default temperatures for the 200-mb (array T200), 100-mb (array T100), and 50-mb (array T50) levels, which the profile extrapolation algorithm uses when a radiosonde profile terminates below these levels (Section 3.1.2). Historically, WPL used U.S. Standard Atmosphere temperatures (-56.51, -56.51, and -55.91°C, respectively) as default values at these levels, regardless of season. Recently, we started using monthly mean default temperatures for a specific radiometer location instead. For example, the values shown in the second group of DATA statements in INITCON are monthly mean temperatures for Denver, Colorado, computed from 1970-79 radiosonde data. These values should be replaced if they do not represent the upper atmosphere at the location where the user's radiosonde data were taken. If preferred, users could ensure that the default temperatures are never used at all by setting parameter PMAX to 50 mb (Section 4.1.1). However, this strategy would also eliminate all radiosonde profiles that terminate below the 50-mb level (Section 3.1.1).

DEFAULT also contains variable PFRAC, which determines the default vapor pressure at the extrapolated levels. Using List's (1963) definition of vapor pressure,

$$e = \frac{r}{\epsilon + r} p,$$

where e = vapor pressure

r = mixing ratio

ϵ = ratio of molecular weights (water vapor : dry air)

p = pressure,

we define variable PFRAC as

$$PFRAC = \frac{r}{\epsilon + r}.$$

Assuming a constant mixing ratio at the extrapolated levels (variable RMIX = 10^{-6} g g⁻¹), PFRAC becomes a constant, and e varies directly with pressure.

4.2.2. Cloud Water Density Fractions

The last three lines of INITCON assign three cloud water density fractions to array DENFRAC that correspond to the left, middle, and right curves in Fig. 2, respectively. Subroutine CLDPROF constructs a cloud water density profile consistent with only the rightmost curve. Subroutine CLDABS multiplies this profile by each of the fractions in DENFRAC to obtain cloud water density profiles consistent with each of the three curves in Fig. 2; hence, DENFRAC(3) = 1. Other cloud models could be created easily by changing the fractions in DENFRAC. DENFRAC is passed back to the main program in the argument list.

4.2.3. Safeguard Variables

The variables MAXL and EXPMAX were designed as safeguards to prevent two types of errors from occurring during program execution. Their values must be assigned carefully in order to operate as intended. They are passed to appropriate subroutines through COMMON block CONSTNT.

MAXL, which represents the maximum number of profile levels that array dimensions allow, was designed to alert the user when array dimensions are too small to accommodate a given profile. For example, if the input subroutine RDRAOB encounters a radiosonde profile with more than MAXL levels, it outputs a message, skips over the profile, and inputs another. If adding profile levels by interpolation or extrapolation would result in more than MAXL levels, TBMODEL outputs a message and aborts the program. To fulfill its purpose, **MAXL must be changed whenever profile array dimensions are changed.** In the present software, MAXL = 500 (Appendix B).

EXPMAX is the absolute value of the maximum argument allowed for the Fortran intrinsic exponential function EXP. It is used to prevent underflow during brightness temperature calculations in subroutine RADIANC. When EXPMAX is exceeded, RADIANC executes an alternate path that is consistent with the physics but does not involve EXP (Section 3.5). The appropriate value for EXPMAX may change from compiler to compiler.

4.2.4. Physical Constants

Values for all physical constants used in TBMODEL are assigned or calculated in subroutine INITCON and passed through COMMON block CONSTNT. Table 2 gives the variable names, definitions, values, and units of the physical constants that INITCON assigns directly. The values for m_d and m_v given in Table 2 were computed from the atomic weights and atmospheric composition fractions in Table 3 as

Table 2. Constants assigned in subroutine INITCON. Values and significant figures are from Lide (1990), except for T_{bg} which is from Wilkinson (1986). We truncated π at nine significant figures.

Variable Name	Common Symbol	Definition	Assigned Value	Units
C	c	Speed of light in a vacuum	$2.99792458 \times 10^{10}$	cm s ⁻¹
G	g	Standard acceleration of gravity	9.80665	m s ⁻²
H	h	Planck constant	$6.6260755 \times 10^{-34}$	J s
K	k	Boltzmann constant	1.380658×10^{-23}	J K ⁻¹
MDRY	m_d	Molecular weight of dry air	28.96415	g mol ⁻¹
MVAP	m_v	Molecular weight of water vapor	18.01528	g mol ⁻¹
PI	π	Circle circumference/diameter	3.14159265	
R	R	Universal gas constant	8.314510	J mol ⁻¹ K ⁻¹
RE	R_e	Mean Earth radius	6370.949	km
TBG	T_{bg}	Cosmic background T_b	2.75	K

Table 3. Molecular weights ($m_{species}$) and fractional compositions ($f_{species}$) for gas species involved in molecular weight calculations for dry air (m_d) and water vapor (m_v). Values and significant figures are those of Lide (1990).

m	Gas Species	$m_{species}$ (g mol ⁻¹)	$f_{species}$
m_d	N ₂	2 x 14.00674	0.78084
	O ₂	2 x 15.9994	0.20946
	Ar	39.948	0.00934
	CO ₂	12.011 + (2 x 15.9994)	0.00033
m_v	H ₂ O	(2 x 1.00794) + 15.9994	1.00000

$$m = \sum_1^n m_{species} f_{species}$$

where n = number of gas species involved
 $m_{species}$ = molecular weight of a given species (g mol⁻¹)
 $f_{species}$ = fraction of m occupied by the species.

Table 4 gives the variable names, definitions, formulas, and units for constants that INITCON computes from the physical constants in Table 2. The following dimensional analysis was used to convert R_d and R_v from $J K^{-1} g^{-1}$ to the units shown in Table 4:

$$R_d: \left(\frac{1 \text{ kg } m^2 \text{ s}^{-2}}{1 \text{ J}} \right) \left(\frac{10^3 \text{ g}}{1 \text{ kg}} \right) \left(\frac{1 \text{ km}}{10^3 \text{ m}} \right)$$

$$R_v: \left(\frac{10^7 \text{ erg}}{1 \text{ J}} \right) \left(\frac{1 \text{ dyne cm}}{1 \text{ erg}} \right) \left(\frac{10^{-3} \text{ mb}}{1 \text{ dyne cm}^{-2}} \right) \left(\frac{(10^{-2} \text{ m})^3}{1 \text{ cm}^3} \right).$$

Table 4. Constants computed in subroutine INITCON from constants in Table 1.

Variable Name	Common Symbol	Definition	Formula	Units
EPSILON	ϵ	Molecular weight ratio	m_v / m_d	
PFRAC		Vapor pressure fraction	$r / (\epsilon + r)$	
RDRY	R_d	Gas constant for dry air	R / m_d	$m \text{ s}^{-2} \text{ km } K^{-1}$
RVAP	R_v	Gas constant for water vapor	$.01 \times (R / m_v)$	$\text{mb } m^3 \text{ g}^{-1} K^{-1}$
HOVERK	h/k	Physical constant ratio	h / k	$K \text{ s}$

4.2.5. Conversion Factors

Table 5 gives the variable names and formulas for conversion factors used in TBMODEL. We elaborate on the less familiar conversion from decibels to nepers.

Table 5. Conversion factors assigned in subroutine INITCON.

Variable Name	Conversion		Formula
	From	To	
DB2NP	Decibels	Nepers	$\ln(10) / 10$
DEG2RAD	Degrees	Radians	$\pi / 180$
GHZ2HZ	Gigahertz	Hertz	1×10^9

Decibels (dB) are defined as

$$dB = 10 \log_{10} \left(\frac{I}{I_0} \right),$$

and nepers (Np) are defined as

$$Np = \ln \left(\frac{I}{I_0} \right),$$

where I is intensity and I_0 is some reference intensity. Therefore, converting from decibels to nepers requires the conversion factor

$$\frac{Np}{dB} = \frac{\ln \left(\frac{I}{I_0} \right)}{10 \log_{10} \left(\frac{I}{I_0} \right)}. \quad (12)$$

Using the properties of logarithms, we can express the intensity ratio equivalently as

$$\frac{I}{I_0} = 10^{\log_{10} \left(\frac{I}{I_0} \right)}.$$

Therefore, we can rewrite (12) as

$$\frac{Np}{dB} = \frac{\ln \left[10^{\log_{10} \left(\frac{I}{I_0} \right)} \right]}{10 \log_{10} \left(\frac{I}{I_0} \right)} = \frac{\log_{10} \left(\frac{I}{I_0} \right) \ln(10)}{10 \log_{10} \left(\frac{I}{I_0} \right)} = \frac{\ln(10)}{10},$$

which is the formula for DB2NP shown in Table 5.

ACKNOWLEDGMENTS

We thank the Department of Commerce personnel, past and present, who contributed directly or indirectly to the content of this document. In particular, we thank R.L. Abbott, M.T. Decker, E.J. Dutton, M.J. Falls, F.O. Guiraud, W.B. Sweezy, and G.D. Thayer for algorithms and software. Comments by M.J. Falls clarified the text, suggestions by T.A. Stermitz reorganized Section 4, and A. Weickmann influenced the overall document format.

We especially thank Hans J. Liebe of the National Telecommunications and Information Administration for providing us with regular updates to his millimeter-wave absorption model.

The software revision and document preparation were partially supported by the Department of Energy Atmospheric Radiation Measurements program (ARM), the Winter Icing and Storms Program (WISP) sponsored by the Federal Aviation Administration, and the Jet Propulsion Laboratory NASA Propagation Program.

REFERENCES

- Askne, J.I.H., and E.R. Westwater, 1986: A review of ground-based remote sensing of temperature and moisture by passive microwave radiometers. *IEEE Trans. Geosci. Remote Sensing*, GE-24, 340-352.
- Bean, B.R., and E.J. Dutton, 1966: *Radio Meteorology*. National Bureau of Standards Monograph 92, Central Radio Propagation Laboratory, Boulder, CO, 435 pp.
- Crosby, D.S., H.E. Fleming, and D.Q. Wark, 1973: Covariance matrices and means of atmospheric Planck function profiles for application to temperature sounding from satellite measurements. *J. Atmos. Sci.*, 30, 141-144.
- Decker, M.T., and J.A. Schroeder, 1991: Calibration of ground-based microwave radiometers for atmospheric remote sensing. Tech. Memo. ERL-WPL 197, NOAA/Environmental Research Laboratories, Boulder, CO, 16 pp.
- Decker, M.T., E.R. Westwater, and F.O. Guiraud, 1978: Experimental evaluation of ground-based microwave radiometric remote sensing of atmospheric temperature and water vapor profiles. *J. Applied Meteorol.*, 17, 1788-1795.
- Goody, R.M., and Y.L. Yung, 1989: *Atmospheric Radiation: Theoretical Basis*. Oxford University Press, New York, 519 pp.

- Hulst, H.C. van de, 1981: *Light Scattering by Small Particles*. Dover Publications, New York, 470 pp.
- Lide, D.R. (ed.), 1990: *CRC Handbook of Chemistry and Physics*. 71st ed., CRC Press, Boca Raton, FL.
- Liebe, H.J., and D.H. Layton, 1987: Millimeter-wave properties of the atmosphere: Laboratory studies and propagation modeling. Tech. Rpt. 87-224, U.S. Dept. of Commerce/NTIA, 74 pp.
- List, R.J., 1963: *Smithsonian Meteorological Tables*. Smithsonian Institution, Washington, D.C., 527 pp.
- Miller, P.A., and M.J. Falls, 1989: Pilot study of methods to decrease measurement errors of tropospheric inversions by ground-based microwave radiometry. *J. Atmos. Oceanic Technol.*, 6, 225-234.
- Rosenkranz, P.W., 1988: Interference coefficients for overlapping oxygen lines in air. *J. Quant. Spectrosc. Radiat. Transfer*, 39, 287-297.
- Schroeder, J.A., J.R. Jordan, and M.T. Decker, 1989: Design considerations for a network of thermodynamic profilers. *J. Atmos. Oceanic Technol.*, 6, 840-845.
- Thayer, G.D., 1974: An improved equation for the radio refractive index of air. *Radio Science*, 9, 803-807.
- Ulaby, F.T., R.K. Moore, and A.K. Fung, 1981: *Microwave Remote Sensing, Active and Passive, Volume 1*. Addison-Wesley, Reading, Mass., 456 pp.
- Westwater, E.R., and O.N. Strand, 1968: Statistical information content of radiation measurements used in indirect sensing. *J. Atmos. Sci.*, 25, 750-758.
- Westwater, E.R., 1972: Ground-based determination of low altitude temperature profiles by microwaves. *Monthly Weather Review*, 100, 15-28.
- Westwater, E.R., 1974: Microwave emission from clouds. Tech. Rpt. ERL-219-WPL 18, NOAA/Environmental Research Laboratories, Boulder, CO, 43 pp.
- Wilkinson, D.T., 1986: Anisotropy of the cosmic blackbody radiation. *Science*, 232, 1517-1522.

APPENDIX A: NONLINEAR PATH INTEGRATION ALGORITHM

The algorithm inside EXPINT1 is based on the assumption that the parameter in array X decays exponentially with height within the layer between adjacent levels. We can express this assumption over the layer between level $i-1$ and level i as

$$X(s_i) = X(s_{i-1}) \exp \left[-\frac{(s-s_{i-1})}{H_i} \right], \quad (\text{A1})$$

where H_i is the scale height at level i and $s_{i-1} \leq s \leq s_i$. Letting $s = s_i$, we can solve (A1) for H_i as follows:

$$\frac{X(s_i)}{X(s_{i-1})} = \exp \left[\frac{(s_{i-1} - s_i)}{H_i} \right]$$

$$\ln \left[\frac{X(s_i)}{X(s_{i-1})} \right] = \frac{(s_{i-1} - s_i)}{H_i}$$

$$H_i = \frac{(s_{i-1} - s_i)}{\ln \left[\frac{X(s_i)}{X(s_{i-1})} \right]}. \quad (\text{A2})$$

The integral of X(s) over the layer can be expressed as

$$\int_{s_{i-1}}^{s_i} X(s_{i-1}) \exp \left[-\frac{(s-s_{i-1})}{H_i} \right] ds.$$

Taking constants outside the integral gives

$$X(s_{i-1}) \exp \left(\frac{s_{i-1}}{H_i} \right) \int_{s_{i-1}}^{s_i} \exp \left(-\frac{s}{H_i} \right) ds. \quad (\text{A3})$$

Making the substitutions required to integrate (A3) as $e^u du$ gives

$$-H_i X(s_{i-1}) \exp\left(\frac{s_{i-1}}{H_i}\right) \left[\exp\left(-\frac{s}{H_i}\right) \right]_{i-1}^i,$$

which, after evaluation, becomes

$$-H_i X(s_{i-1}) \exp\left(\frac{s_{i-1}}{H_i}\right) \left[\exp\left(-\frac{s_i}{H_i}\right) - \exp\left(-\frac{s_{i-1}}{H_i}\right) \right].$$

Multiplying the exponentials together gives the result

$$-H_i X(s_{i-1}) \left(\exp\left[\frac{(s_{i-1} - s_i)}{H_i}\right] - 1 \right). \quad (\text{A4})$$

Substituting the definition given in (A2) for H_i in (A4) gives

$$-\frac{(s_{i-1} - s_i) X(s_{i-1})}{\ln\left[\frac{X(s_i)}{X(s_{i-1})}\right]} \left[\exp\left(\frac{(s_{i-1} - s_i)}{\frac{(s_{i-1} - s_i)}{\ln\left[\frac{X(s_i)}{X(s_{i-1})}\right]}}\right) - 1 \right]. \quad (\text{A5})$$

After reducing the exponential term in (A5) to $X(s_i) [X(s_{i-1})]^{-1}$, the final form of the layer integral becomes

$$\int_{i-1}^i X(s_{i-1}) \exp\left[\frac{-(s - s_{i-1})}{H_i}\right] ds = (s_i - s_{i-1}) \left(\frac{[X(s_i) - X(s_{i-1})]}{\ln\left[\frac{X(s_i)}{X(s_{i-1})}\right]} \right), \quad (\text{A6})$$

which is the algorithm used in subroutine EXPINT1. However, two special cases render (A6) unusable. If X_i or X_{i-1} is ≤ 0 , EXPINT1 instead integrates the layer by averaging the two values and multiplying the result by $(s_i - s_{i-1})$. If X_i and X_{i-1} differ by less than a tolerance value (10^{-9}), EXPINT1 integrates the layer as X_i times $(s_i - s_{i-1})$.

APPENDIX B: FORTRAN SOURCE CODE LISTINGS

<u>MODULE</u>	<u>PAGE</u>
MAIN PROGRAM	41
ADDLEVL	61
CLDABS	82
CLDDENS	65
CLDICE	68
CLDINT	83
CLDMODL	64
CLDPROF	66
CLDTAU	84
CLRABS	73
EXPINT1	77
EXPINT2	79
EXTEMP	57
EXTRAP	53
FILL	56
FIND	55
GEOHGT	60
HEADOUT	48
H2O	74
INITCON	44
INSERT	63
O2	75
PROFILE	49
RADIANC	80
RAYTRAC	70
RDRAOB	51
RDSPECS	46
REFRACT	69
SHAPE	75
TAUCALC	78
VAPOR	59

MAIN PROGRAM

PROGRAM TBMODEL (OUTPUT,TAPE1,TAPE2,TAPE3,TAPE4,TAPE5)

```
C   AUGUST 1991 VERSION: GOES WITH USERS GUIDE TO WPL MICROWAVE RADIATIVE
C   TRANSFER SOFTWARE (SCHROEDER AND WESTWATER, 1991).
C
C   COMPUTES BRIGHTNESS TEMPERATURE, MEAN RADIATING TEMPERATURE, & ABSORPTION
C   AT USER-SELECTED ELEVATION ANGLES AND FREQUENCIES FROM RADIOSONDE
C   PROFILES OF PRESSURE, TEMPERATURE, AND RELATIVE HUMIDITY. AUTOMATICALLY
C   MODELS CLOUD WATER DENSITY AFTER DECKER ET AL. (1978) WHEN RELATIVE
C   HUMIDITY EQUALS OR EXCEEDS A USER-SPECIFIED THRESHOLD (UP TO 4 CLOUD
C   LAYERS CAN BE IDENTIFIED). EXTRAPOLATES INPUT PROFILES TO 0.1 MB, USING
C   COEFFICIENTS OF CROSBY ET AL. (1973), ASSUMING PROFILES REACHED 300 MB.
C   CLEAR-SKY CALCULATIONS APPLY TO FREQUENCIES BELOW 1000 GHZ. CLOUD MODEL
C   CALCULATIONS APPLY TO FREQUENCIES BELOW 100 GHZ (RAYLEIGH APPROXIMATION).
C
C   SUBROUTINES CALLED FROM HERE:
C
C   INITCON = INITIALIZES I/O UNIT NOS., DEFAULT TEMPERATURES, & CONSTANTS
C   RDSPECS = READS USER SPECIFICATIONS: ANGLES, FREQUENCIES, ETC.
C   HEADOUT = WRITES DESCRIPTIVE HEADER TO TOP OF OUTPUT FILE
C   PROFILE = INPUTS AND MODIFIES RADIOSONDE PROFILE
C   CLDMODL = IDENTIFIES CLOUDS AND MODELS THEIR LIQUID AND ICE DENSITY
C   REFRACT = COMPUTES REFRACTIVITY, VAPOR PRESSURE, VAPOR DENSITY PROFILES
C   RAYTRAC = RAY-TRACING ROUTINE: COMPUTES SLANT PATH PROFILE ARRAY DS
C   CLRABS  = COMPUTES CLEAR-SKY ABSORPTION & PLANCK FUNCTION PROFILES
C   EXPINT1 = INTEGRATES (EXPONENTIAL DECAY) A PROFILE ALONG A SLANT PATH
C   TAUCALC = COMPUTES PROFILE OF LAYER-INTEGRATED ABSORPTION
C   RADIANC = COMPUTES BRIGHTNESS TEMPERATURES AND MEAN RADIATING TEMPERATURE
C   CLDABS  = COMPUTES ABSORPTION DUE TO MODELED CLOUD LIQUID AND ICE
C   CLDINT  = INTEGRATES (LINEAR) CLOUD WATER ALONG A SLANT PATH
C   CLDTAU = INTEGRATES CLOUD ABSORPTION AND ADDS TO CLEAR ABSORPTION.
C
C   COMMON BLOCKS CONTAIN VALUES INITIALIZED IN SUBROUTINE INITCON:
C
COMMON /IOFILES/ SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE
COMMON /DEFAULT/ T200(12),T100(12),T50(12),PFRAC
COMMON /CONSTNT/ RE,G,EPSILON,RDRY,RVAP,MAXL,EXPMAX,DB2NP,DEG2RAD
REAL Z(500),TK(500),P(500),RH(500),E(500),RHO(500),DRYN(500),
1  ANGLES(10),DS(500,10),REFINDX(500),WETN(500),TAULAY(500),
2  FRQ(11),WAVE(11),HVK(11),BOFTBG(11),BOFT(500,11),
3  AWET(500,11),ADRY(500,11),TAUCLR(500,11,10),
4  ALIQ(500,11),AICE(500,11),CLDBASE(4),CLDTOP(4),
5  DENLIQ(500),DENICE(500),DENL(500),DENI(500),DENFRAC(3)
INTEGER SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE(5),
1  YRFROM,YRTHRU,MONTHS(12),DAYFROM,DAYTHRU,HRFROM,HRTHRU,
2  IRAY(10),LBASE(4),LTOP(4)
CHARACTER TITLE*70
C
C   INITIALIZE I/O UNIT NUMBERS, DEFAULT TEMPERATURES, AND CONSTANTS.
CALL INITCON (TBG,C,HOVERK,GHZ2HZ,DENFRAC)
C
C   READ USER SPECIFICATIONS FROM INPUT FILE (UNIT SPECFIL).
CALL RDSPECS (TITLE,I PROF,NFRQ,FRQ,NANG,ANGLES,ZSTART,RAOBLAT,
1  YRFROM,YRTHRU,MONTHS,NMOS,DAYFROM,DAYTHRU,HRFROM,
2  HRTHRU,PMAX,DPMAX,ICE,MAXCLD,RHCLD,CLDMIN)
C
C   WRITE DESCRIPTIVE FILE HEADER TO OUTPUT FILE (UNIT MODLFIL).
CALL HEADOUT (TITLE,PMAX,DPMAX,ICE,TBG,MAXCLD,RHCLD,CLDMIN)
C
C   COMPUTE WAVELENGTH, RATIO HV/K, AND B(TBG), WHICH DO NOT DEPEND ON RAOB.
DO 11 J = 1,NFRQ
  WAVE(J) = C / (FRQ(J) * GHZ2HZ)
  HVK(J) = (FRQ(J) * GHZ2HZ) * HOVERK
  BOFTBG(J) = 1. / (EXP (HVK(J) / TBG) - 1.)
11 CONTINUE
```

MAIN PROGRAM, continued

```
C *** THE REST OF THE MAIN PROGRAM IS A LOOP THAT EXECUTES ONCE PER RAOB.***
C   INPUT AND EXTRAPOLATE RAOB; ADJUST TO BEGIN AT SIMULATED ANTENNA
C   HEIGHT. INSERT PROFILE LEVELS WHERE LEVELS ARE TOO FAR APART.
20  CALL PROFILE (RAOBLAT, YRFROM, YRTHRU, MONTHS, NMOS, DAYFROM, DAYTHRU,
1    HRFROM, HRTHRU, ZSTART, PMAX, DPMAX, ICE, Z0, Z, P, TK, RH, NL)
C   IDENTIFY CLOUDS; MODEL CLOUD DENSITY; INSERT PROFILE LEVELS AT BOUNDARIES
    CALL CLDMODL (RHCLD, CLDMIN, MAXCLD, Z, P, TK, RH, NL, NLAY, CLDBASE,
1    CLDTOP, LBASE, LTOP, DENLIQ, DENICE, NMODLS)
C   COMPUTE PROFILES OF WET REFRACTIVITY (WETN), DRY REFRACTIVITY (DRYN),
C   REFRACTIVE INDEX (REFINDX), VAPOR PRESSURE (E), VAPOR DENSITY (RHO).
    CALL REFRACT (P, TK, RH, NL, ICE, DRYN, WETN, REFINDX, E, RHO)
C   COMPUTE THE SLANT PATH LENGTH PROFILE FOR EACH ELEVATION ANGLE.
    CALL RAYTRAC (NL, ANGLES, NANG, Z, REFINDX, Z0, DS, IRAY, NANGOK)
C   OUTPUT HEADER LINE, CLOUD BASES AND TOPS, & EXPANDED PROFILES (OPTIONAL).
    WRITE (MODLFIL, 1000) DATE, RAOBLAT, Z0, NLAY, IPROF, NL, NMODLS, NANGOK,
1    NFRQ
    IF (NLAY.GT.0) WRITE (MODLFIL, 2000) (CLDBASE(I), CLDTOP(I), I=1, NLAY)
    IF (IPROF.EQ.1) WRITE (MODLFIL, 3000) (Z(I), P(I), TK(I), RH(I),
1    DENLIQ(I), DENICE(I), I=1, NL)
C   COMPUTE CLEAR ABSORPTION & B(T) PROFILES FOR EACH FREQUENCY.
    CALL CLRABS (P, TK, E, NL, FRQ, NFRQ, HVK, BOFT, AWET, ADRY)
C   REPEAT LOOP 44 FOR EACH ELEVATION ANGLE WITH A VALID RAY PATH (IRAY=1).
    DO 44 K = 1, NANG
      IF (IRAY(K).GT.0) THEN
C       INTEGRATE WATER VAPOR AND REFRACTIVITIES; WRITE RESULTING PWV AND
C       WET AND DRY EXCESS PATH LENGTH VALUES TO THE OUTPUT FILE.
          CALL EXPINT1 (RHO, DS, K, NL, 0.1, SRHO)
          CALL EXPINT1 (WETN, DS, K, NL, 0.1, SWETN)
          CALL EXPINT1 (DRYN, DS, K, NL, 0.1, SDRYN)
          WRITE (MODLFIL, 4000) ANGLES(K), SRHO, SWETN, SDRYN
C       FOR EACH FREQUENCY (LOOP 33), COMPUTE & OUTPUT TBS, TMRs, AND TAUS.
          DO 33 J = 1, NFRQ
            CALL TAUCALC (AWET, ADRY, J, DS, K, 1, NL, TAUW, TAUD, TAULAY)
            CALL RADIANC (NL, TAULAY, BOFT, J, BOFTBG(J), HVK(J), 0, 0, 0, TBATM,
1            TBTOTL, TMR, TMRCLD)
            WRITE (MODLFIL, 5000) FRQ(J), TBTOTL, TBATM, TMR, TMRCLD, TAUW, TAUD
C           IF PROFILE CONTAINS CLOUD(S), SAVE CLEAR-SKY LAYER ABSORPTION.
            IF (NLAY.GT.0) THEN
              DO 22 I = 1, NL
                TAUCLR(I, J, K) = TAULAY(I)
22              CONTINUE
              END IF
33            CONTINUE
            END IF
44          CONTINUE
```


MAIN PROGRAM, continued

```
C *** THE REST OF THE MAIN PROGRAM APPLIES ONLY TO PROFILES WITH CLOUD(S).***
      IF(NLAY.GT.0) THEN
C       REPEAT LOOP 77 FOR EACH CLOUD DENSITY MODEL.
          DO 77 MODEL = 1,NMODLS
C       COMPUTE CLOUD DENSITIES AND CORRESPONDING ABSORPTION FOR THIS MODEL.
          CALL CLDABS (WAVE,NFRQ,NL,TK,DENLIQ,DENICE,DENFRAC(MODEL),
1          DENL,DENI,ALIQ,AICE)
C       REPEAT LOOP 66 FOR EACH ELEVATION ANGLE WITH A VALID RAY PATH (IRAY=1),
          DO 66 K = 1,NANG
              IF (IRAY(K).GT.0) THEN
C               INTEGRATE CLOUD LIQUID AND ICE DENSITY OVER THE RAY PATH.
C               OUTPUT WITH ELEVATION ANGLE AND CLOUD DENSITY FRACTION.
                  CALL CLDINT (DENL,DS,K,NLAY,LBASE,LTOP,SLIQ)
                  CALL CLDINT (DENI,DS,K,NLAY,LBASE,LTOP,SICE)
                  WRITE (MODLFIL,4000) ANGLES(K),SLIQ,SICE,DENFRAC(MODEL)
C               FOR EACH FREQUENCY (LOOP 55), COMPUTE AND OUTPUT TBS, TMRS, TAUS.
                  DO 55 J = 1,NFRQ
                      CALL CLDTAU (NL,TAUCLR,ALIQ,AICE,J,DS,K,NLAY,LBASE,LTOP,
1                      TAU,TAUI,TAULAY)
1                      CALL RADIANC (NL,TAULAY,BOFT,J,BOFTBG(J),HVK(J),NLAY,
1                      LBASE(1),LTOP(1),TBATM,TBTOTL,TMR,TMRCLD)
1                      WRITE (MODLFIL,5000) FRQ(J),TBTOTL,TBATM,TMR,TMRCLD,
1                      TAU,TAUI
55                     CONTINUE
                        END IF
66                     CONTINUE
77                     CONTINUE
                        END IF
                  GO TO 20
1000  FORMAT (/,3X,3I3,3X,2I2,F10.2,F10.3,2I6,17,3I6)
2000  FORMAT (2X,2F8.3,3(4X,2F8.3))
3000  FORMAT (5X,F10.3,2F10.2,3E11.3)
4000  FORMAT (10X,4E13.5)
5000  FORMAT (3X,F8.3,4F10.3,2E13.5)

      END
```

SUBROUTINE INITCON

SUBROUTINE INITCON (TBG,C,HOVERK,GHZ2HZ,DENFRAC)

```
C      INITIALIZES I/O UNIT NUMBERS, DEFAULT TEMPERATURES, AND CONSTANTS.
C      THE PHYSICAL CONSTANTS DEFINED HERE WERE OBTAINED OR DERIVED FROM
C      THE CRC HANDBOOK OF CHEMISTRY AND PHYSICS, 71ST EDITION (LIDE, 1990).
C
C      OUTPUTS PASSED AS ARGUMENTS:
C          TBG      = COSMIC BACKGROUND BRIGHTNESS TEMPERATURE (K)
C          C        = SPEED OF LIGHT IN A VACUUM (CM/S)
C          HOVERK   = [PLANCK CONSTANT (J*S)] / [BOLTZMANN CONSTANT (J/K)]
C          GHZ2HZ  = CONVERSION FACTOR: GHZ TO HZ (1.E9)
C          DENFRAC = CLOUD WATER DENSITY FRACTION ARRAY
C
C      OUTPUTS PASSED THROUGH COMMON:
C          SPECFIL = UNIT NUMBER ASSOCIATED WITH USER-SPECIFICATION INPUT FILE
C          RAOBFIL = UNIT NUMBER ASSOCIATED WITH RADIOSONDE DATA INPUT FILE
C          HEADFIL = UNIT NUMBER ASSOCIATED WITH FILE CONTAINING HEADER TEXT.
C          MODLFIL = UNIT NUMBER ASSOCIATED WITH MODEL CALCULATIONS OUTPUT FILE
C          MESSFIL = UNIT NUMBER ASSOCIATED WITH MESSAGE OUTPUT FILE
C          T200   = ARRAY OF MONTHLY DEFAULT 200-MB TEMPERATURES (C)
C          T100   = ARRAY OF MONTHLY DEFAULT 100-MB TEMPERATURES (C)
C          T50    = ARRAY OF MONTHLY DEFAULT 50-MB TEMPERATURES (C)
C          PFRAC  = PRESSURE FRACTION FOR VAPOR PRESSURE EXTRAPOLATION
C          RE     = MEAN EARTH RADIUS (KM)
C          G      = GENERAL GRAVITATIONAL CONSTANT (M/(S**2)).
C          EPSILON = RATIO OF MOLECULAR WTS.: (WATER VAPOR) / (DRY AIR).
C          RDRY   = GAS CONSTANT FOR DRY AIR (M/(S**2)*KM/K)
C          RVAP   = GAS CONSTANT FOR WATER VAPOR (MB*M**3)/G/K)
C          MAXL   = MAXIMUM NO. OF PROFILE LEVELS THAT FIT ARRAY DIMENSIONS
C          EXPMAX = MAXIMUM ABSOLUTE VALUE FOR EXPONENTIAL FUNCTION ARGUMENT
C                  (COMPUTER/COMPILER-DEPENDENT)
C          DB2NP  = CONVERSION FACTOR: DECIBELS TO NEPERS
C          DEG2RAD = CONVERSION FACTOR: DEGREES TO RADIAN
C
C      COMMON BLOCKS PASS UNIT NOS., DEFAULT TEMPERATURES, & CONSTANTS SET HERE.
C
COMMON /IOFILES/ SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE
COMMON /DEFAULT/ T200(12),T100(12),T50(12),PFRAC
COMMON /CONSTNT/ RE,G,EPSILON,RDRY,RVAP,MAXL,EXPMAX,DB2NP,DEG2RAD
C
REAL K,MDRY,MVAP,DENFRAC(3)
INTEGER SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE(5)
C
C      SET I/O UNIT NUMBERS:
C
DATA SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL / 1,2,3,4,5 /
C
C      SET DEFAULT TEMPERATURES FOR EXTRAPOLATION ALGORITHM (200-,100-,50-MB).
C
DATA T200 / -55.7,-55.5,-55.6,-56.3,-56.7,-56.1,
1          -55.1,-54.6,-55.3,-55.9,-55.8,-55.8 /
DATA T100 / -59.7,-58.5,-58.1,-58.3,-60.1,-63.0,
1          -65.0,-65.5,-64.8,-63.5,-61.8,-60.6 /
DATA T50  / -59.9,-58.8,-58.7,-58.1,-57.5,-56.6,
1          -55.7,-56.2,-57.5,-59.2,-60.6,-60.6 /
C
C      SET LOCAL CONSTANTS:
C          RMIX = DEFAULT MIXING RATIO FOR HUMIDITY EXTRAPOLATION (G/G)
C          R    = UNIVERSAL GAS CONSTANT (J/MOL/K)
C          MDRY = MOLECULAR WEIGHT OF DRY AIR (G/MOL)
C          MVAP = MOLECULAR WEIGHT OF WATER VAPOR (G/MOL)
C          H    = PLANCK CONSTANT (J*S)
C          K    = BOLTZMANN CONSTANT (J/K)
C          PI   = THE CONSTANT PI
C
DATA RMIX, R, MDRY, MVAP / 1.E-6, 8.314510, 28.96415, 18.01528 /
DATA H, K, PI / 6.6260755E-34, 1.380658E-23, 3.14159265 /
```

SUBROUTINE INITCON, continued

C SET OR COMPUTE GLOBAL CONSTANTS (DEFINED ABOVE UNDER OUTPUTS):

TBG = 2.75
C = 2.99792458E10
RE = 6370.949
G = 9.80665
HOVERK = H / K
EPSILON = MVAP / MDRY
RDRY = R / MDRY
RVAP = 0.01 * R / MVAP
PFRAC = RMIX / (EPSILON + RMIX)

C SET LIMITS ON ARRAY SIZE AND EXP ARGUMENT; COMPUTE CONVERSION FACTORS.

MAXL = 500
EXPMAX = 675.
DB2NP = ALOG (10.) * 0.1
DEG2RAD = PI / 180.
GHZ2HZ = 1.E9

C SET CLOUD WATER DENSITY FRACTIONS CORRESPONDING TO THE 3 CLOUD MODELS.

DENFRAC(1) = 0.25
DENFRAC(2) = 0.50
DENFRAC(3) = 1.00

RETURN
END

SUBROUTINE RDSPECS

```

SUBROUTINE RDSPECS (TITLE, IPROF, NFRQ, FRQ, NANG, ANGLES, ZSTART,
1          RAOBLAT, YRFROM, YRTHRU, MONTHS, NMOS, DAYFROM,
2          DAYTHRU, HRFROM, HRTHRU, PMAX, DPMAX, ICE, MAXCLD,
3          RHCLD, CLDMIN)

C   INPUTS USER SPECIFICATIONS FROM UNIT SPECFIL.
C
C   COMMON BLOCK CONTAINS SPECFIL AND MESSAGE FILE UNIT NUMBERS:
COMMON /IOFILES/ SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE

CHARACTER TITLE*70
INTEGER SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE(5),
1          YRFROM,YRTHRU,MONTHS(12),DAYFROM,DAYTHRU,HRFROM,HRTHRU
REAL FRQ(11),ANGLES(10)

C   INPUT TITLE FOR OUTPUT WITH RESULTS (70 CHARACTER MAXIMUM).

READ (SPECFIL,1000,END=999,ERR=999)
READ (SPECFIL,2000,END=999,ERR=999) TITLE

C   INPUT SWITCH TO WRITE EXTRAPOLATED, INTERPOLATED PROFILES (INCLUDING
C   CLOUD DENSITY) TO OUTPUT FILE WITH RESULTS (1=YES; 0=NO).

READ (SPECFIL,1100,END=999,ERR=999)
READ (SPECFIL,*,END=999,ERR=999) IPROF

C   INPUT FREQUENCIES OF INTEREST (GHZ, MAXIMUM OF 11).

READ (SPECFIL,1000,END=999,ERR=999)
READ (SPECFIL,*,END=999,ERR=999) NFRQ
  NFRQ = MIN0 (NFRQ,11)
READ (SPECFIL,1000,END=999,ERR=999)
READ (SPECFIL,*,END=999,ERR=999) (FRQ(I),I=1,NFRQ)

C   INPUT ELEVATION ANGLES OF INTEREST (DEGREES, MAXIMUM OF 10).

READ (SPECFIL,1000,END=999,ERR=999)
READ (SPECFIL,*,END=999,ERR=999) NANG
  NANG = MIN0 (NANG,10)
READ (SPECFIL,1000,END=999,ERR=999)
READ (SPECFIL,*,END=999,ERR=999) (ANGLES(I),I=1,NANG)

C   INPUT SIMULATED ANTENNA ELEVATION (KM MSL); INPUT ANTENNA ELEVATION
C   BELOW RAOB LAUNCH SITE ELEVATION WILL BE RESET TO THE SITE ELEVATION.

READ (SPECFIL,1100,END=999,ERR=999)
READ (SPECFIL,*,END=999,ERR=999) ZSTART

C   INPUT LATITUDE OF RADIOSONDE LAUNCH SITE (DEGREES).
C   (REQUIRED FOR EXTRAPOLATION ALGORITHM)

READ (SPECFIL,1100,END=999,ERR=999)
READ (SPECFIL,*,END=999,ERR=999) RAOBLAT

C   INPUT RANGE OF YEARS TO INPUT FROM RAOB DATA FILE (2 DIGITS).

READ (SPECFIL,1000,END=999,ERR=999)
READ (SPECFIL,*,END=999,ERR=999) YRFROM,YRTHRU

C   INPUT MONTHS OF RAOB DATA DESIRED (12 MAXIMUM, ORDER-INDEPENDENT).

READ (SPECFIL,1000,END=999,ERR=999)
READ (SPECFIL,*,END=999,ERR=999) NMOS
  NMOS = MIN0 (NMOS,12)
READ (SPECFIL,1000,END=999,ERR=999)
READ (SPECFIL,*,END=999,ERR=999) (MONTHS(I),I=1,NMOS)
```

SUBROUTINE RDSPECS, continued

```
C      INPUT DAY AND HOUR RANGES OF RAOB DATA DESIRED.
      READ (SPECFIL,1000,END=999,ERR=999)
      READ (SPECFIL,*,END=999,ERR=999) DAYFROM,DAYTHRU
      READ (SPECFIL,1000,END=999,ERR=999)
      READ (SPECFIL,*,END=999,ERR=999) HRFROM,HRTHRU

C      INPUT MAXIMUM ACCEPTABLE PRESSURE AT TOP OF INPUT PROFILE (MAXIMUM
C      OF 300 MB); PROFILES TERMINATING BELOW THIS LEVEL ARE DISCARDED.
      READ (SPECFIL,1100,END=999,ERR=999)
      READ (SPECFIL,*,END=999,ERR=999) PMAX
      PMAX = AMIN1 (PMAX,300.)

C      INPUT MAXIMUM ACCEPTABLE PRESSURE DIFFERENCE (MB) BETWEEN ADJACENT
C      LEVELS. (DPMAX=0 LETS PRESSURE DIFFERENCE INCREASE WITH HEIGHT.)
      READ (SPECFIL,1100,END=999,ERR=999)
      READ (SPECFIL,*,END=999,ERR=999) DPMAX

C      INPUT SWITCH TO SELECT SATURATION VAPOR PRESSURE CALCULATION OVER WATER
C      ONLY (0) OR WATER AND ICE, DEPENDING ON TEMPERATURE (1).
      READ (SPECFIL,1000,END=999,ERR=999)
      READ (SPECFIL,*,END=999,ERR=999) ICE

C      INPUT MAXIMUM NUMBER OF CLOUD LAYERS TO MODEL PER PROFILE (MAXIMUM OF 4).
      READ (SPECFIL,1000,END=999,ERR=999)
      READ (SPECFIL,*,END=999,ERR=999) MAXCLD
      MAXCLD = MIN0 (MAXCLD,4)

C      INPUT RELATIVE HUMIDITY THRESHOLD FOR CLOUD DETECTION (BETWEEN 0 AND 1).
      READ (SPECFIL,1000,END=999,ERR=999)
      READ (SPECFIL,*,END=999,ERR=999) RHCLD

C      INPUT MINIMUM ACCEPTABLE CLOUD LAYER THICKNESS (KM).
      READ (SPECFIL,1000,END=999,ERR=999)
      READ (SPECFIL,*,END=999,ERR=999) CLDMIN

      RETURN

999  WRITE (MESSFIL,3000) SPECFIL
      STOP

1000 FORMAT (/)
1100 FORMAT (//)
2000 FORMAT (A70)
3000 FORMAT (1X,'FROM RDSPECS: ERROR WHILE READING USER SPECIFICATION',
1      'S FROM UNIT',I2,'. BOMBING...')

      END
```

SUBROUTINE HEADOUT

```
SUBROUTINE HEADOUT (TITLE, PMAX, DPMAX, ICE, TBG, MAXCLD, RHCLD, CLDMIN)

C   READS TEXT FROM UNIT HEADFIL AND WRITES IT TO THE BEGINNING OF THE
C   OUTPUT FILE (UNIT MODLFIL) ALONG WITH OTHER HEADER INFORMATION.
C
C   INPUTS PASSED AS ARGUMENTS:
C   TITLE   = CHARACTER ARRAY CONTAINING USER-SPECIFIED TITLE
C   PMAX    = MAXIMUM ACCEPTABLE PRESSURE AT TOP OF RAOB PROFILE (MB)
C   DPMAX   = MAXIMUM PRESSURE DIFFERENCE BETWEEN PROFILE LEVELS (MB)
C   ICE     = SWITCH FOR SATURATION VAPOR PRESSURE OVER WATER OR WATER/ICE
C   TBG     = COSMIC BACKGROUND BRIGHTNESS TEMPERATURE (K)
C   MAXCLD  = MAXIMUM NUMBER OF CLOUD LAYERS TO MODEL PER PROFILE
C   RHCLD   = RELATIVE HUMIDITY THRESHOLD FOR CLOUD DETECTION (DECIMAL)
C   CLDMIN  = MINIMUM ACCEPTABLE CLOUD THICKNESS FOR MODELING (KM)
C   INPUTS PASSED THROUGH COMMON:
C   HEADFIL = UNIT NO. ASSOCIATED WITH FILE CONTAINING HEADER TEXT.
C   MODLFIL = UNIT NO. ASSOCIATED WITH MODEL CALCULATIONS OUTPUT FILE
C   MESSFIL = UNIT NUMBER ASSOCIATED WITH MESSAGE OUTPUT FILE
C   COMMON BLOCK CONTAINS UNIT NUMBERS SET IN SUBROUTINE INITCON:

COMMON /IOFILES/ SPECFIL, RAOBFIL, HEADFIL, MODLFIL, MESSFIL, DATE

CHARACTER TITLE*70, LINE*80
INTEGER SPECFIL, RAOBFIL, HEADFIL, MODLFIL, MESSFIL, DATE(5)

WRITE (MODLFIL, 1000) TITLE, PMAX, DPMAX, ICE, TBG, MAXCLD, RHCLD, CLDMIN

10  READ (HEADFIL, 2000, END=998, ERR=999) LINE
    WRITE (MODLFIL, 2000) LINE
    GO TO 10

998  RETURN

999  WRITE (MESSFIL, 3000) HEADFIL
    STOP

1000 FORMAT (1X, 'TITLE: ', A70, '//, 1X, 'THE RESULTS IN THIS FILE REFLEC',
1      'T THE FOLLOWING USER SPECIFICATIONS: ', '//, 1X, 'DISCARD PROFILE',
2      'S THAT TERMINATE BELOW THIS PRESSURE LEVEL (MB): ', F13.0, '/', 1X,
3      'INSERT LEVEL IF PRESSURE DIFFERENCE BETWEEN LEVELS EXCEEDS (',
4      'MB): ', F12.0, '/', 1X, '(ZERO ALLOWS THE PRESSURE DIFFERENCE TO IN',
5      'CREASE WITH HEIGHT.) ', '/', 1X, 'CALCULATE SATURATION VAPOR PRESS',
6      'SURE OVER WATER (0) OR WATER/ICE (1): ', I7, '/', 1X, 'COSMIC BACKGR',
7      'OUND BRIGHTNESS TEMPERATURE (K): ', F30.2, '/', 1X, 'MAXIMUM NUMBER',
8      ' OF CLOUD LAYERS TO MODEL PER PROFILE: ', I23, '/', 1X, 'RELATIVE H',
9      'UMIDITY THRESHOLD FOR CLOUD DETECTION (FRACTION): ', F16.2, '/', 1X,
1     'MINIMUM ACCEPTABLE CLOUD THICKNESS (KM): ', F35.3, '/')
2000 FORMAT (A80)
3000 FORMAT (1X, 'FROM HEADOUT: ERROR WHILE READING HEADER TEXT FILE F',
1      'ROM UNIT', I2, '. BOMBING...')

END
```

SUBROUTINE PROFILE

```

SUBROUTINE PROFILE (RAOBLAT,YRFROM,YRTHRU,MONTHS,NMOS,DAYFROM,
1 DAYTHRU,HRFROM,HRTHRU,ZSTART,PMAX,DPMAX,ICE,
2 Z0,Z,P,TK,RH,NL)

C INPUTS 1 RADIOSONDE PROFILE FROM THE SPECIFIED TIME PERIOD; ADJUSTS IT
C TO BEGIN AT THE SPECIFIED ANTENNA HEIGHT (ZSTART) OR THE 1ST PROFILE
C LEVEL, WHICHEVER IS HIGHER; EXTRAPOLATES PROFILE TO 0.1 MB; INSERTS
C LEVEL(S) WHERE RAOB LEVELS WERE TOO FAR APART.
C *** DISCARDS PROFILE IF ZSTART IS ABOVE THE HIGHEST PROFILE LEVEL.***
C
C INPUTS PASSED AS ARGUMENTS:
C RAOBLAT = LATITUDE OF RADIOSONDE LAUNCH SITE (DEGREES)
C YRFROM = BEGINNING YEAR
C YRTHRU = ENDING YEAR (INCLUSIVE)
C MONTHS = ARRAY OF MONTHS DESIRED
C NMOS = NUMBER OF MONTHS DESIRED
C DAYFROM = BEGINNING DAY
C DAYTHRU = ENDING DAY (INCLUSIVE)
C HRFROM = BEGINNING HOUR
C HRTHRU = ENDING HOUR (INCLUSIVE)
C ZSTART = SPECIFIED SIMULATED ANTENNA HEIGHT (KM MSL)
C PMAX = MAXIMUM ACCEPTABLE PRESSURE AT TOP OF INPUT PROFILE (MB)
C DPMAX = MAXIMUM ACCEPTABLE PRESSURE DIFFERENCE BETWEEN LEVELS (MB)
C (ZERO SELECTS ALGORITHM THAT INCREASES DPMAX WITH HEIGHT)
C ICE = SWITCH FOR ES OVER WATER ONLY (0) OR WATER AND ICE (1)
C INPUT PASSED THROUGH COMMON:
C MESSFIL = UNIT NUMBER ASSOCIATED WITH MESSAGE OUTPUT FILE
C OUTPUTS PASSED AS ARGUMENTS:
C Z0 = ACTUAL ANTENNA HEIGHT SIMULATED, >= ZSTART (KM MSL)
C Z = HEIGHT PROFILE (KM ABOVE SIMULATED ANTENNA HEIGHT)
C P = PRESSURE PROFILE (MB)
C TK = TEMPERATURE PROFILE (K)
C RH = RELATIVE HUMIDITY PROFILE (FRACTION)
C NL = NUMBER OF PROFILE LEVELS
C OUTPUT PASSED THRU COMMON:
C DATE = INTEGER ARRAY CONTAINING DATE AND TIME OF CURRENT RAOB
C SUBROUTINES:
C RDRAOB = READS RADIOSONDE PROFILE (DISCARDS SHORT PROFILES)
C EXTRAP = EXTRAPOLATES PROFILES TO 0.1 MB
C ADDLEVL = ADDS A PROFILE AT ANTENNA HEIGHT, IF ABSENT
C INSERT = INSERTS PROFILE LEVELS WHERE THEY ARE TOO FAR APART
C COMMON BLOCK CONTAINS UNIT NUMBER (MESSFIL) SET IN SUBROUTINE INITCON:

COMMON /IOFILES/ SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE

INTEGER YRFROM,YRTHRU,MONTHS(12),DAYFROM,DAYTHRU,HRFROM,HRTHRU,
1 SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE(5)
REAL Z(500),P(500),TK(500),RH(500)

C INPUT A RADIOSONDE PROFILE TAKEN WITHIN THE SPECIFIED DATES AND TIMES.

10 CALL RDRAOB (YRFROM,YRTHRU,MONTHS,NMOS,DAYFROM,DAYTHRU,
1 HRFROM,HRTHRU,PMAX,Z,P,TK,RH,NL)

C IF THE RAOB TERMINATES BELOW THE SPECIFIED ANTENNA HEIGHT (ZSTART),
C OUTPUT MESSAGE, DISCARD PROFILE, AND INPUT ANOTHER.

IF (ZSTART.GT.Z(NL)) THEN
WRITE (MESSFIL,1000) DATE,ZSTART,Z(NL)
GO TO 10
END IF

C EXTRAPOLATE PROFILE TO 0.1 MB, IF NECESSARY.

IF (P(NL).GT.0.1) CALL EXTRAP (RAOBLAT,Z,P,TK,RH,NL,ICE)
```

SUBROUTINE PROFILE, continued

```
C   IF THE SPECIFIED ANTENNA HEIGHT (ZSTART) IS BELOW THE 1ST PROFILE LEVEL,  
C   SET THE ANTENNA HEIGHT (Z0) TO THE 1ST PROFILE LEVEL ELEVATION.  
  
    Z0 = AMAX1 (ZSTART,Z(1))  
  
C   IF Z0 IS ABOVE PROFILE LEVEL 1, FIND THE 1ST LEVEL AT OR ABOVE Z0.  
  
    IF (Z0.GT.Z(1)) THEN  
      DO 11 I = 2,NL  
        IF (Z(I).GE.Z0) GO TO 20  
11     CONTINUE  
  
C   IF THE LEVEL FOUND IS ABOVE Z0, INTERPOLATE TO ADD A LEVEL AT Z0.  
  
20    IF (Z(I).GT.Z0) CALL ADDLEVL (0,Z0,I,Z,P,TK,RH,NL)  
  
C   RENUMBER PROFILE LEVELS TO BEGIN AT Z0. COUNT THE LEVELS (NL).  
  
    NL0 = NL  
    NL = 0  
    DO 22 J = 1,NL0  
      NL = NL + 1  
      P(NL) = P(J)  
      Z(NL) = Z(J)  
      TK(NL) = TK(J)  
      RH(NL) = RH(J)  
22    CONTINUE  
    END IF  
  
C   CONVERT HEIGHT PROFILE TO KM ABOVE Z0.  
  
    DO 33 I = 1,NL  
      Z(I) = Z(I) - Z0  
33    CONTINUE  
  
C   INSERT ADDITIONAL LEVELS WHERE PROFILE LEVELS ARE TOO FAR APART.  
  
    CALL INSERT (DPMAX,Z,P,TK,RH,NL)  
  
    RETURN  
  
1000 FORMAT (1X,'FROM PROFILE: (',2(I2,'/'),',12,13,12,' Z) SIMULATED ',  
1      'ANTENNA HEIGHT (',F6.3,' KM) EXCEEDS',/,33X,'PROFILE TO',  
2      'P (',F6.3,' KM). DISCARDING RAOB.')
```

END

SUBROUTINE RDRAOB

```

SUBROUTINE RDRAOB (YRFROM,YRTHRU,MONTHS,NMOS,DAYFROM,DAYTHRU,
1          HRFROM,HRTHRU,PMAX,Z,P,T,RH,NL)

C      INPUTS RAOB PROFILE FROM UNIT RAOBFIL THAT FALLS WITHIN THE SPECIFIED
C      DATE AND TIME RANGES; CONVERTS TEMPERATURE TO KELVINS. IF RAOB TERMINATES
C      BELOW A SPECIFIED PRESSURE LEVEL (PMAX) OR IT CONTAINS TOO MANY LEVELS
C      FOR SPECIFIED ARRAY DIMENSIONS, RDRAOB WRITES A MESSAGE TO UNIT MESSFIL
C      AND DISCARDS THE RAOB, CONTINUING WITH THE NEXT. RDRAOB STOPS THE PROGRAM
C      WHEN IT ENCOUNTERS EOF OR WHEN DATE/TIME IS BEYOND THE SPECIFIED RANGES.
C *** NOTE: THIS ROUTINE WAS DESIGNED TO READ THE BINARY 10-YR RAOB DATA SETS
C      "DENVER" AND "DULLES", WHICH M.J. FALLS CREATED ON THE CYBER.
C
C      INPUTS PASSED AS ARGUMENTS:
C          YRFROM = BEGINNING YEAR
C          YRTHRU = ENDING YEAR (INCLUSIVE)
C          MONTHS = ARRAY OF MONTHS DESIRED
C          NMOS   = NUMBER OF MONTHS DESIRED
C          DAYFROM = BEGINNING DAY
C          DAYTHRU = ENDING DAY (INCLUSIVE)
C          HRFROM = BEGINNING HOUR
C          HRTHRU = ENDING HOUR (INCLUSIVE)
C          PMAX  = MAXIMUM ACCEPTABLE PRESSURE AT TOP OF INPUT PROFILE (MB)
C
C      INPUTS PASSED THRU COMMON:
C          RAOBFIL = UNIT NUMBER ASSOCIATED WITH RADIOSONDE DATA FILE
C          MESSFIL = UNIT NUMBER ASSOCIATED WITH MESSAGE OUTPUT FILE
C          MAXL   = MAXIMUM NO. OF PROFILE LEVELS THAT FIT ARRAY DIMENSIONS
C
C      OUTPUTS PASSED AS ARGUMENTS:
C          Z      = INPUT HEIGHT PROFILE (KM MSL)
C          P      = INPUT PRESSURE PROFILE (MB)
C          T      = INPUT TEMPERATURE PROFILE, CONVERTED TO KELVINS (K)
C          RH     = INPUT RELATIVE HUMIDITY (FRACTION)
C          NL     = NUMBER OF INPUT PROFILE LEVELS
C
C      OUTPUT PASSED THRU COMMON:
C          DATE = INTEGER ARRAY CONTAINING DATE AND TIME OF CURRENT RAOB
C
C      COMMON BLOCKS CONTAIN UNIT NOS. & CONSTANTS SET IN SUBROUTINE INITCON:
C
C      COMMON /IOFILES/ SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE
C      COMMON /CONSTNT/ RE,G,EPSILON,RDRY,RVAP,MAXL,EXPMAX,DB2NP,DEG2RAD
C
C      INTEGER YRFROM,YRTHRU,MONTHS(12),DAYFROM,DAYTHRU,HRFROM,HRTHRU,
1          SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE(5)
C      REAL Z(500),P(500),T(500),RH(500)
C
C      INPUT DATE, TIME, AND NUMBER OF PROFILE LEVELS.
10  READ (RAOBFIL,END=999,ERR=998) DATE,NL
C
C      IF LEVELS EXCEED ARRAY DIMENSIONS, OUTPUT MESSAGE AND SKIP PROFILE.
C
C      IF (NL.GT.MAXL) THEN
C          WRITE (MESSFIL,1000) DATE,NL,MAXL
C          DO 11 I = 1,NL
C              READ (RAOBFIL,END=998,ERR=998) X1,X2,X3,X4
11          CONTINUE
C          GO TO 10
C      END IF
C
C      INPUT PRESSURE, HEIGHT, TEMPERATURE, AND RELATIVE HUMIDITY AT EACH LEVEL.
C
C      DO 22 I = 1,NL
C          READ (RAOBFIL,END=998,ERR=998) P(I),Z(I),T(I),RH(I)
22  CONTINUE
```

SUBROUTINE RDRAOB, continued

```
C   IF YEAR READ IS BEYOND YEAR(S) OF INTEREST, PRINT MESSAGE AND STOP.
    IF (DATE(3).LE.YRTHRU) THEN
C   OTHERWISE, SUCCESSIVELY CHECK YEAR, MONTH, DAY, AND HOUR AGAINST THE
C   SPECIFIED RANGES. IF THEY FALL OUTSIDE, INPUT ANOTHER PROFILE.
    IF (DATE(3).GE.YRFROM) THEN
      IMO = 0
      DO 33 I = 1,NMOS
        IF (DATE(1).EQ.MONTHS(I)) IMO = 1
33      CONTINUE
      IF (IMO.GT.0) THEN
        IF ((DATE(2).GE.DAYFROM).AND.(DATE(2).LE.DAYTHRU)) THEN
          IF ((DATE(4).GE.HRFROM).AND.(DATE(4).LE.HRTHRU)) THEN
C   IF PROFILE TERMINATES AT OR ABOVE SPECIFIED PRESSURE
C   THRESHOLD (P(NL)), CONVERT T TO KELVIN AND RETURN.
          IF (P(NL).LE.PMAX) THEN
            DO 44 I = 1,NL
              T(I) = T(I) + 273.16
44          CONTINUE
            RETURN
C   OTHERWISE, OUTPUT MESSAGE AND INPUT ANOTHER PROFILE.
          ELSE
            WRITE (MESSFIL,2000) DATE,P(NL)
          END IF
        END IF
      END IF
    END IF
  END IF
  GO TO 10
END IF

C   OUTPUT APPROPRIATE MESSAGES.
  WRITE (MESSFIL,3000) DATE
  STOP
998  WRITE (MESSFIL,4000) DATE,RAOBFIL
  STOP
999  WRITE (MESSFIL,5000) DATE,RAOBFIL
  STOP

1000 FORMAT (1X,'FROM RDRAOB: (',2(I2,'/'),12,13,12,' Z) NUMBER OF ',
1      'RAOB PROFILE LEVELS (',14,' ) EXCEEDS',/,33X,'SPECIFIED M',
2      'AXIMUM (',13,' ). DISCARDING RAOB.')
2000 FORMAT (1X,'FROM RDRAOB: (',2(I2,'/'),12,13,12,' Z) RAOB TERMI',
1      'NATES AT',F7.1,' MB.',/,33X,'DISCARDING RAOB.')
3000 FORMAT (1X,'FROM RDRAOB: (',2(I2,'/'),12,13,12,' Z) DATE/TIME ',
1      'READ IS BEYOND SPECIFIED RANGE.',/,33X,'NORMAL EXIT.')
4000 FORMAT (1X,'FROM RDRAOB: (',2(I2,'/'),12,13,12,' Z) READ ERROR',
1      ' ON UNIT',12,'. BOMBING...')
5000 FORMAT (1X,'FROM RDRAOB: (',2(I2,'/'),12,13,12,' Z) LAST RAOB ',
1      'BEFORE EOF ON UNIT',12,'. NORMAL EXIT.')

  END
```

SUBROUTINE EXTRAP

SUBROUTINE EXTRAP (RAOBLAT,Z,P,TK,RH,NL,ICE)

```
C   EXTRAPOLATES A SET OF PROFILES UP TO 0.1 MB. ESTIMATES TEMPERATURES
C   AT 15 LEVELS ABOVE 50 MB FROM TEMPERATURES AT 700, 500, 300, 200, 100,
C   AND 50 MB, USING THE REGRESSION COEFFICIENTS OF CROSBY ET AL. (1973).
C   ***** ASSUMES ORIGINAL PROFILES TERMINATE AT OR ABOVE 300 MB.*****
C   EXTRAPOLATES HUMIDITY ASSUMING A CONSTANT MIXING RATIO DEFINED IN
C   SUBROUTINE INITCON. COMPUTES GEOPOTENTIAL HEIGHT AT EXTRAPOLATED LEVELS.
C
C   INPUTS PASSED AS ARGUMENTS:
C   → RAOBLAT = LATITUDE OF RADIOSONDE LAUNCH SITE (DEGREES)
C   INPUTS PASSED THRU COMMON:
C       T200 = ARRAY OF MONTHLY DEFAULT 200-MB TEMPERATURES (C)
C       T100 = ARRAY OF MONTHLY DEFAULT 100-MB TEMPERATURES (C)
C       T50  = ARRAY OF MONTHLY DEFAULT 50-MB TEMPERATURES (C)
C       PFRAC = PRESSURE FRACTION FOR VAPOR PRESSURE EXTRAPOLATION
C   INPUTS/OUTPUTS (MODIFIED HERE):
C       Z = ORIGINAL/EXPANDED GEOPOTENTIAL HEIGHT PROFILE (KM)
C       P = ORIGINAL/EXPANDED PRESSURE PROFILE (MB)
C       TK = ORIGINAL/EXPANDED TEMPERATURE PROFILE (K)
C       RH = ORIGINAL/EXPANDED RELATIVE HUMIDITY PROFILE (FRACTION)
C       NL = ORIGINAL/EXPANDED NUMBER OF PROFILE LEVELS
C       ICE = SWITCH FOR ES OVER WATER ONLY (0) OR WATER AND ICE (1)
C   SUBROUTINES:
C       FIND = FINDS TEMPERATURE AT A GIVEN PRESSURE LEVEL.
C       FILL = ASSIGNS DEFAULT TEMPERATURE TO A GIVEN PRESSURE LEVEL.
C       EXTEMP = EXTRAPOLATES TEMPERATURE PROFILE.
C       VAPOR = COMPUTES VAPOR PRESSURE.
C       GEOHGT = COMPUTES GEOPOTENTIAL HEIGHTS.
C
C   COMMON BLOCK CONTAINS DEFAULT VALUES ASSIGNED IN SUBROUTINE INITCON:
C
C   COMMON /DEFAULT/ T200(12),T100(12),T50(12),PFRAC
C
C   REAL Z(500),P(500),TK(500),RH(500),E(500),P6(6),T6(6)
C
C   SET PRESSURE LEVELS REQUIRED BY THE TEMPERATURE EXTRAPOLATION ALGORITHM.
C
C   DATA P6/ 700.,500.,300.,200.,100.,50. /
C
C   SAVE ORIGINAL NUMBER OF PROFILE LEVELS.
C
C   NL0 = NL
C
C   FIND TEMPERATURES (T6) AT THE LOWEST 3 LEVELS OF P6.
C   *** ASSUMING THAT INPUT PROFILE DID NOT TERMINATE BELOW 300 MB.***
C
C   CALL FIND (P,TK,1,NL,P6(1),T6(1),LFOUND)
C   CALL FIND (P,TK,LFOUND,NL,P6(2),T6(2),LFOUND)
C   CALL FIND (P,TK,LFOUND,NL,P6(3),T6(3),LFOUND)
C
C   IF THE PROFILE TERMINATES BELOW 200 MB, ADD A LEVEL WITH THE DEFAULT
C   200-MB TEMPERATURE. OTHERWISE, FIND THE 200-MB TEMPERATURE.
C
C   IF(P(NL).GT.200.) THEN
C       CALL FILL (T200,4,P6,T6,NL,P,TK)
C   ELSE
C       CALL FIND (P,TK,LFOUND,NL,P6(4),T6(4),LFOUND)
C   END IF
C
C   IF THE PROFILE TERMINATES BELOW 100 MB, ADD A LEVEL WITH THE DEFAULT
C   100-MB TEMPERATURE. OTHERWISE, FIND THE 100-MB TEMPERATURE.
C
C   IF(P(NL).GT.100.) THEN
C       CALL FILL (T100,5,P6,T6,NL,P,TK)
C   ELSE
C       CALL FIND (P,TK,LFOUND,NL,P6(5),T6(5),LFOUND)
C   END IF
```

SUBROUTINE EXTRAP, continued

```
C   IF THE PROFILE TERMINATES BELOW 50 MB, ADD A LEVEL WITH THE DEFAULT
C   50-MB TEMPERATURE. OTHERWISE, FIND THE 50-MB TEMPERATURE.

      IF(P(NL).GT.50.) THEN
        CALL FILL (T50,6,P6,T6,NL,P,TK)
      ELSE
        CALL FIND (P,TK,LFOUND,NL,P6(6),T6(6),LFOUND)
      END IF

C   EXTRAPOLATE TEMPERATURE ABOVE 50 MB.

      CALL EXTEMP (RAOBLAT,T6,TK,P,NL)

C   EXTRAPOLATE RELATIVE HUMIDITY PROFILE, ASSUMING CONSTANT MIXING RATIO.

      N1 = NL0 + 1
      DO 22 I = N1,NL
        E(I) = PFRAC * P(I)
        CALL VAPOR (TK(I),1.,ICE,ES,DUMMY)
        RH(I) = E(I) / ES
22    CONTINUE

C   COMPUTE HEIGHT PROFILE AT EXTRAPOLATED LEVELS. (NEED RH IN TERMS OF E)

      CALL VAPOR (TK(NL0),RH(NL0),ICE,E(NL0),DUMMY)
      CALL GEOHGT (P,TK,E,N1,NL,Z)

      RETURN
      END
```

SUBROUTINE FIND

```

SUBROUTINE FIND (PKNOW,TKNOW,LBEG,LEND,PWANT,TWANT,LFOUND)
C   FINDS THE TEMPERATURE AT A GIVEN PRESSURE LEVEL (BY INTERPOLATION, IF
C   NECESSARY), USING A KNOWN PAIR OF PRESSURE AND TEMPERATURE PROFILES.
C   IT RETURNS THE PROFILE LEVEL NUMBER WHERE PWANT WAS FOUND, SO THAT
C   SUBSEQUENT SEARCHES FOR TEMPERATURES AT LOWER PRESSURES CAN BEGIN THERE.
C
C   *** STOPS THE PROGRAM IF THE REQUESTED PRESSURE LEVEL FALLS OUTSIDE
C   THE PRESSURE RANGE DEFINED BY PKNOW(LBEG) AND PKNOW(LEND).
C
C   INPUTS PASSED AS ARGUMENTS:
C       PKNOW = KNOWN PRESSURE PROFILE
C       TKNOW = KNOWN TEMPERATURE PROFILE CORRESPONDING TO PKNOW
C       LBEG  = PROFILE LEVEL AT WHICH TO BEGIN SEARCH THROUGH PKNOW
C       LEND  = PROFILE LEVEL AT WHICH TO END SEARCH THROUGH PKNOW
C       PWANT = PRESSURE AT WHICH TEMPERATURE IS NEEDED
C   INPUTS PASSED THRU COMMON:
C       MESSFIL = UNIT NUMBER ASSOCIATED WITH MESSAGE OUTPUT FILE
C       DATE   = INTEGER ARRAY CONTAINING DATE AND TIME OF CURRENT RAOB
C   OUTPUTS:
C       TWANT  = TEMPERATURE CORRESPONDING TO PRESSURE PWANT
C       LFOUND = PROFILE LEVEL AT OR JUST BELOW PWANT.
C   COMMON BLOCK CONTAINS UNIT NUMBER (MESSFIL) SET IN SUBROUTINE INITCON:
C
COMMON /IOFILES/ SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE
INTEGER SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE(5)
REAL PKNOW(500),TKNOW(500)
C
C   STATEMENT FUNCTION YINTERP INTERPOLATES LINEARLY BETWEEN 2 POINTS:
C
C   YINTERP (X1,X,X2,Y1,Y2) = Y2 + ((Y1-Y2) * (X-X2) / (X1-X2))
C
C   IF REQUESTED PRESSURE LIES OUTSIDE GIVEN PRESSURE RANGE, STOP PROGRAM.
C
C   IF ((PWANT.GT.PKNOW(LBEG)). OR. (PWANT.LT.PKNOW(LEND))) THEN
C       WRITE (MESSFIL,1000) DATE,PWANT,PKNOW(LBEG),PKNOW(LEND)
C       STOP
C   END IF
C
C   FIND THE TEMPERATURE CORRESPONDING TO PWANT, INTERPOLATING IF NECESSARY.
C
DO 11 I = LBEG,LEND
  IF (PWANT.GE.PKNOW(I)) THEN
    IF (PWANT.GT.PKNOW(I)) THEN
      X = ALOG (PWANT)
      X1 = ALOG (PKNOW(I))
      X2 = ALOG (PKNOW(I-1))
      TWANT = YINTERP (X1,X,X2,TKNOW(I),TKNOW(I-1))
      LFOUND = I - 1
    ELSE
      TWANT = TKNOW(I)
      LFOUND = I
    END IF
  RETURN
END IF
11 CONTINUE
1000 FORMAT (1X,'FROM FIND:      (',2(I2,'/'),I2,I3,I2,' Z) PRESSURE L',
1      'EVEL WANTED (',F5.0,' MB) IS OUTSIDE',/,33X,'GIVEN PROFILE',
2      ' PRESSURE RANGE (',F6.1,',',F6.1,' MB).',/,33X,'BOMBING...')
END
```

SUBROUTINE FILL

```
SUBROUTINE FILL (TDEFAULT,I6,P6,T6,NL,P,TK)

C   CONVERTS INPUT DEFAULT TEMPERATURE TO KELVINS, ASSIGNS IT TO T6(I6), AND
C   ADDS A LEVEL TO THE TOP OF PROFILES P AND TK WITH PRESSURE P6(I6) AND
C   TEMPERATURE T6(I6), RESP.
C *** STOPS THE PROGRAM IF ADDED LEVEL EXCEEDS MAXIMUM LEVELS INPUT.***
C
C   INPUTS PASSED AS ARGUMENTS:
C       TDEFAULT = ARRAY OF (MONTHLY) DEFAULT TEMPERATURES (C)
C       I6       = INDEX FOR ARRAYS P6 AND T6
C       P6       = ARRAY OF EXTRAPOLATION COEFFICIENT PRESSURE LEVELS (MB)
C   INPUTS PASSED THRU COMMON:
C       MESSFIL = UNIT NUMBER ASSOCIATED WITH MESSAGE OUTPUT FILE
C       DATE    = INTEGER ARRAY CONTAINING DATE AND TIME OF CURRENT RAOB
C       MAXL    = MAXIMUM NO. OF PROFILE LEVELS THAT FIT ARRAY DIMENSIONS
C   INPUTS/OUTPUTS (MODIFIED HERE):
C       T6 = ARRAY OF TEMPERATURES FOR EXTRAPOLATION ALGORITHM (K)
C       NL = ORIGINAL/EXPANDED NUMBER OF PROFILE LEVELS
C       P  = ORIGINAL/EXPANDED PRESSURE PROFILE (MB)
C       TK = ORIGINAL/EXPANDED TEMPERATURE PROFILE (K)
C
C   COMMON BLOCKS CONTAIN UNIT NOS. & CONSTANTS SET IN SUBROUTINE INITCON:
C
COMMON /IOFILES/ SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE
COMMON /CONSTNT/ RE,G,EPSILON,RDRY,RVAP,MAXL,EXPMAX,DB2NP,DEG2RAD

INTEGER SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE(5)
REAL TDEFAULT(12),P6(6),T6(6),P(500),TK(500)

NL = NL + 1

C   IF ADDING THIS LEVEL EXCEEDS PROFILE ARRAY DIMENSIONS, STOP PROGRAM.

IF (NL.GT.MAXL) THEN
  WRITE (MESSFIL,1000) DATE,P6(I6),MAXL
  STOP
END IF

T6(I6) = TDEFAULT (DATE(1)) + 273.16
P(NL) = P6(I6)
TK(NL) = T6(I6)

RETURN

1000 FORMAT (1X,'FROM FILL:      (',2(I2,'/'),',',I2,I3,I2,' Z) ADDING A L',
1         'EVEL AT ',F4.0,' MB FOR EXTRAPOLATION',/,33X,'ALGORITHM ',
2         'EXCEEDS SPECIFIED MAXIMUM NUMBER OF',/,33X,'LEVELS (MAXL',
3         '= ',I3,')'. STOPPING...')
```

END

SUBROUTINE EXTEMP

SUBROUTINE EXTEMP (RAOBLAT,T6,TK,P,NL)

C ESTIMATES TEMPERATURES AT 15 LEVELS BETWEEN 30 AND 0.1 MB, INCLUSIVE, FROM
 C TEMPERATURES AT 700, 500, 300, 200, 100, AND 50 MB, USING REGRESSION
 C COEFFICIENTS FROM CROSBY ET AL. (1973, J.ATMOS.SCI.,30, 141-144).
 C THERE IS A SEPARATE SET OF COEFFICIENTS FOR EACH OF 3 LATITUDE RANGES.
 C *** STOPS PROGRAM IF EXTRAPOLATED LEVELS EXCEED MAXIMUM NUMBER SPECIFIED.***

C
 C INPUTS PASSED AS ARGUMENTS:
 C RAOBLAT = LATITUDE OF RADIOSONDE LAUNCH SITE (DEGREES)
 C T6 = TEMPERATURES AT 700, 500, 300, 200, 100, 50 MB (K)
 C INPUTS PASSED THRU COMMON:
 C MESSFIL = UNIT NUMBER ASSOCIATED WITH MESSAGE OUTPUT FILE
 C DATE = INTEGER ARRAY CONTAINING DATE AND TIME OF CURRENT RAOB
 C MAXL = MAXIMUM NO. OF PROFILE LEVELS THAT FIT ARRAY DIMENSIONS
 C INPUTS/OUTPUTS (MODIFIED HERE):
 C TK = ORIGINAL/EXPANDED TEMPERATURE PROFILE (K)
 C P = ORIGINAL/EXPANDED PRESSURE PROFILE (MB)
 C NL = ORIGINAL/EXPANDED NUMBER OF PROFILE LEVELS

C
 C COMMON BLOCK CONTAINS CONSTANTS INITIALIZED IN SUBROUTINE INITCON:

COMMON /IOFILES/ SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE
 COMMON /CONSTNT/ RE,G,EPSILON,RDRY,RVAP,MAXL,EXPMAX,DB2NP,DEG2RAD

INTEGER SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE(5)
 REAL T6(6),TK(500),P(500),P15(15),COEF(7,15,3)

C
 C DEFINE THE 15 PRESSURE LEVELS CORRESPONDING TO THE REGRESSION COEFFICIENTS

DATA P15 /30.,25.,20.,15.,10.,7.,5.,4.,3.,2.,1.5,1.,.5.,.2.,.1/

C
 C DEFINE THE REGRESSION COEFFICIENTS FOR LATITUDES WHOSE
 C ABSOLUTE VALUES LIE IN THE INTERVAL (60,90] DEGREES.

DATA ((COEF(I,J,1),I=1,7),J=1,15)/
 1 .0030,-.4122,.2104,-.2353,.0945,.1706,293.0100,
 2 .1869,-.2832,.2494,.1544,-.1125,.2569,154.0889,
 3 .4627,.0462,.2420,.7426,-.2284,.1978,-69.3933,
 4 .6674,.0833,.3362,.6991,-.2295,.4542,-209.0515,
 5 .5543,.2496,.3585,.8005,-.4661,.6072,-234.0684,
 6 .4743,.3667,.3746,.8772,-.6403,.7184,-251.9551,
 7 .4302,.3471,.2671,.6737,-.6346,.8598,-199.2074,
 8 .1574,.4218,.2896,.4731,-.5598,.9468,-155.3909,
 9 -.0298,.4794,.3045,.3378,-.5133,1.0161,-123.7185,
 1 -.225,.5391,.2414,.1667,-.5031,1.2056,-85.1810,
 2 -.1082,.3379,.2342,.0368,-.4720,1.3296,-56.3577,
 3 -.1403,.2063,.1671,.0144,-.3724,1.2859,-30.4098,
 4 -.1142,.1149,.1227,-.0150,-.2910,1.2531,-11.6513,
 5 -.0943,.0792,.0985,-.0047,-.2119,1.2039,-12.3730,
 6 -.0746,.0492,.0765,.0059,-.1443,1.1621,-14.013 /

C
 C DEFINE THE REGRESSION COEFFICIENTS FOR LATITUDES WHOSE
 C ABSOLUTE VALUES LIE IN THE INTERVAL (30,60] DEGREES.

DATA ((COEF(I,J,2),I=1,7),J=1,15)/
 1 .6055,-.5999,.1783,.4776,.2891,-.2045,66.9102,
 2 .5322,-.3325,.2039,.4003,.4351,-.1836,4.4326,
 3 .6016,-.1784,.3375,.2921,.7642,-.3375,-85.7488,
 4 .5830,-.0216,.4294,.3836,.7595,-.3886,-146.2202,
 5 .5871,.0479,.3625,.3444,.7743,-.3687,-152.6067,
 6 .5923,.0970,.3136,.3181,.7845,-.3557,-157.4324,
 7 .5949,-.0149,.2512,.1354,.5849,-.0015,-116.9559,
 8 .5196,-.0341,.2578,.0360,.4254,.1976,-86.1359,
 9 .4595,-.0480,.2654,-.0344,.3000,.3509,-63.4169,
 1 .3372,.0916,.3325,.0032,.2223,.6264,-138.4715,
 2 .2951,.1129,.2677,-.0033,.1056,.7890,-131.8596,
 3 .1905,.1257,.2275,.0112,.0755,.7943,-99.9765,
 4 .1085,.1354,.2097,.0135,.0520,.8018,-76.5450,
 5 .0788,.0770,.2066,-.0032,.0280,.8658,-59.6573,
 6 .0538,.0296,.2050,-.0147,.0077,.9177,-46.2681/

SUBROUTINE EXTEMP, continued

```

C   DEFINE THE REGRESSION COEFFICIENTS FOR LATITUDES WHOSE
C   ABSOLUTE VALUES LIE IN THE INTERVAL [0,30] DEGREES.

DATA ((COEF(I,J,3),I=1,7),J=1,15)/
1      .0557, .0549,-.9796, .8427,-.5035,-.6448,498.3794,
2      .0755,-.0686,-.5784, .4568,-.3291,-.4956,458.0688,
3      -.0236,-.0562,-.1700, .0409,-.1599,-.2423,400.6909,
4      -.0858, .0407, .1097,-.2153,-.0946,-.0222,325.7112,
5      -.1664, .2078, .0294,-.1698,-.2470, .0091,334.6926,
6      -.2107, .2985, .0081,-.1244,-.3584, .0673,326.3660,
7      -.1126, .3427,-.0445,-.0125,-.3688, .3074,218.6801,
8      -.0370, .2349, .0761,-.0088,-.3091, .3636,167.3085,
9      .0236, .1786, .0982, .0235,-.2538, .3837,133.5242,
1     -.0025, .1083, .1871,-.1371,-.2147, .3658,163.7505,
2     .0867,-.0702, .2592,-.1866,-.1037, .3047,165.0714,
3     .1339,-.0777, .2304,-.1618,-.0664, .3609,129.7949,
4     .1633,-.0750, .2239,-.1572,-.0328, .3936,103.3754,
5     .0728,-.0599, .2212,-.1488,-.0047, .4815, 96.1188,
6     -.0126,-.0420, .2265,-.1487, .0275, .5451, 91.4538/

C   SELECT THE APPROPRIATE SET OF REGRESSION COEFFICIENTS BASED ON
C   THE ABSOLUTE VALUE OF STATION LATITUDE.

ABSLAT = ABS (RAOBLAT)
K = 3
  IF (ABSLAT.GT.30.) K = 2
  IF (ABSLAT.GT.60.) K = 1

C   APPLY THE REGRESSION COEFFICIENTS TO ESTIMATE TK AT THE SUBSET OF 15
C   PRESSURE LEVELS ABOVE THE TOP OF THE INPUT PROFILE.

NL0 = NL
DO 22 JJ = 1,15
  IF (P15(JJ).LT.P(NL0)) THEN
    J = 16 - JJ
    SUM = COEF(7,J,K)
    DO 11 I = 1,6
      SUM = SUM + COEF(I,J,K) * T6(I)
11    CONTINUE

C   IF ADDING A LEVEL WILL EXCEED SPECIFIED MAXIMUM NUMBER OF LEVELS,
C   OUTPUT A MESSAGE AND STOP THE PROGRAM.

    NL = NL + 1
    IF (NL.GT.MAXL) THEN
      WRITE (MESSFIL,1000) DATE,MAXL
      STOP
    END IF
    TK(NL) = SUM
    P(NL) = P15(JJ)
  END IF
22  CONTINUE

RETURN

1000 FORMAT (1X,'FROM EXTEMP: (',2(I2,'/'),',12,I3,12,' Z) EXTRAPOLAT',
1         'ION WILL EXCEED THE SPECIFIED MAXIMUM',/,33X,'NUMBER OF ',
2         'PROFILE LEVELS (MAXL=',I3,'). STOPPING.')

END

```


SUBROUTINE VAPOR

```
SUBROUTINE VAPOR (TK,RH,ICE,E,RHO)

C   COMPUTES VAPOR PRESSURE AND VAPOR DENSITY FROM TEMPERATURE AND RELATIVE
C   HUMIDITY.
C
C   INPUTS PASSED AS ARGUMENTS:
C       TK   = TEMPERATURE (K)
C       RH   = RELATIVE HUMIDITY (FRACTION)
C       ICE  = SWITCH TO CALCULATE SATURATION VAPOR PRESSURE OVER
C             WATER ONLY (0) OR WATER AND ICE, DEPENDING ON TK (1)
C   INPUT PASSED THROUGH COMMON:
C       RVAP = GAS CONSTANT FOR WATER VAPOR (MB*(M**3)/G/K)
C   OUTPUTS:
C       E    = VAPOR PRESSURE (MB)
C       RHO  = VAPOR DENSITY (G/M3)
C
C   COMMON BLOCK CONTAINS CONSTANTS INITIALIZED IN SUBROUTINE INITCON:
C   COMMON /CONSTNT/ RE,G,EPSILON,RDRY,RVAP,MAXL,EXPMAX,DB2NP,DEG2RAD

C   COMPUTE SATURATION VAPOR PRESSURE (ES, IN MB) OVER WATER OR ICE AT
C   TEMPERATURE TK (KELVINS), USING THE GOFF-GRATCH FORMULATION (LIST,1963).
C
C   FOR WATER...
      IF ((TK.GT.263.16) .OR. (ICE.EQ.0)) THEN
        Y = 373.16 / TK
        ES = -7.90298 * (Y-1.) + 5.02808 * ALOG10 (Y) -
1         1.3816E-7 * (10 ** (11.344 * (1. - (1./ Y)))) - 1.) +
2         8.1328E-3 * (10.** (-3.49149 * (Y - 1.)) - 1.) +
3         ALOG10 (1013.246)
      ELSE
        Y = 273.16 / TK
        ES = -9.09718 * (Y - 1.) - 3.56654 * ALOG10 (Y) +
1         .876793 * (1.- (1./ Y)) + ALOG10 (6.1071)
      END IF
      ES = 10. ** ES

C   COMPUTE VAPOR PRESSURE AND VAPOR DENSITY.
C   THE VAPOR DENSITY CONVERSION FOLLOWS THE IDEAL GAS LAW:
C   VAPOR PRESSURE = VAPOR DENSITY * RVAPOR * TK

      E = RH * ES
      RHO = E / (RVAP * TK)

      RETURN
      END
```

SUBROUTINE GEOHGT

```
SUBROUTINE GEOHGT (P,TK,E,LBEG,LEND,Z)

C   COMPUTES GEOPOTENTIAL HEIGHT PROFILE AT LEVELS LBEG THRU LEND, INCLUSIVE.
C   *** IF LBEG > 1, INPUT PROFILES MUST CONTAIN VALID VALUES AT LEVEL LBEG-1.***
C   THE ALGORITHM FOLLOWS EQ.(6.5) ON P.79 OF INTRODUCTION TO THEORETICAL
C   METEOROLOGY BY S.L. HESS (1959). MIXING RATIO AND VIRTUAL TEMPERATURE
C   ARE DEFINED ON PP.59 AND 60, RESPECTIVELY.
C
C   INPUTS PASSED AS ARGUMENTS:
C       P       = PRESSURE PROFILE (MB)
C       TK      = TEMPERATURE PROFILE (K)
C       E       = VAPOR PRESSURE PROFILE (MB)
C       LBEG    = BEGINNING LEVEL
C       LEND    = ENDING LEVEL
C   INPUTS PASSED THROUGH COMMON:
C       EPSILON = RATIO OF MOLECULAR WTS.: (WATER VAPOR) / (DRY AIR)
C       G       = STANDARD ACCELERATION OF GRAVITY (M/(S**2))
C       RDRY    = GAS CONSTANT FOR DRY AIR (M/(S**2)*KM/K)
C   INPUT/OUTPUT:
C       Z       = GEOPOTENTIAL HEIGHT PROFILE (KM)
C
C   COMMON BLOCK CONTAINS CONSTANTS INITIALIZED IN SUBROUTINE INITCON:
C
C   COMMON /CONSTNT/ RE,G,EPSILON,RDRY,RVAP,MAXL,EXPMAX,DB2NP,DEG2RAD
C
C   REAL P(500),TK(500),E(500),Z(500)
C
C   IF SPECIFIED BEGINNING LEVEL IS 1, SET CORRESPONDING HEIGHT TO 0.
C
C   IF(LBEG.EQ.1) THEN
C       Z(LBEG) = 0.
C       LBEG = 2
C   END IF
C
C   COMPUTE CONSTANT; COMPUTE MIXING RATIO AND VIRTUAL TEMPERATURE
C   AT THE LOWER BOUNDARY OF THE FIRST LAYER.
C
C   ROVERG = RDRY / G
C   W = EPSILON * E(LBEG-1) / (P(LBEG-1) - E(LBEG-1))
C   TVLOWER = TK(LBEG-1) * (1. + (W / EPSILON)) / (1. + W)
C
C   COMPUTE GEOPOTENTIAL HEIGHT PROFILE.
C
C   DO 11 I = LBEG,LEND
C       W = EPSILON * E(I) / (P(I) - E(I))
C       TVUPPER = TK(I) * (1. + (W / EPSILON)) / (1. + W)
C       TVBAR = SQRT (TVLOWER * TVUPPER)
C       Z(I) = Z(I-1) + (ROVERG * TVBAR * ALOG (P(I-1) / P(I)))
C       TVLOWER = TVUPPER
11  CONTINUE
C
C   RETURN
C   END
```

SUBROUTINE ADDLEVEL

```
SUBROUTINE ADDLEVEL (IPZ,XNEW,LNEW,Z,P,TK,RH,NL)

C   ADDS A LEVEL AT A GIVEN HEIGHT OR PRESSURE TO PROFILES Z,P,TK, AND RH
C   BY BUMPING UP OVERLYING LEVELS AND INTERPOLATING BETWEEN VALUES ABOVE
C   AND BELOW THE NEW LEVEL. IF THE COORDINATE IS PRESSURE, LN(P) IS USED.
C   *** STOPS PROGRAM IF ADDING LEVEL EXCEEDS SPECIFIED MAXIMUM NO.OF LEVELS.**
C
C   INPUTS PASSED AS ARGUMENTS:
C       IPZ   = SWITCH TO IDENTIFY PRESSURE (1) OR HEIGHT (0) COORDINATE
C       XNEW  = HEIGHT OR PRESSURE OF NEW LEVEL (KM OR MB, RESPECTIVELY)
C       LNEW  = PROFILE LEVEL NUMBER CORRESPONDING TO NEW LEVEL
C   INPUTS PASSED THRU COMMON:
C       MESSFIL = UNIT NUMBER ASSOCIATED WITH MESSAGE OUTPUT FILE
C       DATE    = INTEGER ARRAY CONTAINING DATE AND TIME OF CURRENT RAOB
C       MAXL    = MAXIMUM NO. OF PROFILE LEVELS THAT FIT ARRAY DIMENSIONS
C   INPUTS/OUTPUTS (MODIFIED HERE):
C       Z      = HEIGHT PROFILE (KM)
C       P      = PRESSURE PROFILE (MB)
C       TK     = TEMPERATURE PROFILE (K)
C       RH     = RELATIVE HUMIDITY PROFILE (FRACTION)
C       NL     = NUMBER OF LEVELS IN PROFILES
C
C   COMMON BLOCKS CONTAIN UNIT NOS. & CONSTANTS SET IN SUBROUTINE INITCON:
C
C   COMMON /IOFILES/ SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE
C   COMMON /CONSTNT/ RE,G,EPSILON,RDRY,RVAP,MAXL,EXPMAX,DB2NP,DEG2RAD
C
C   INTEGER SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE(5)
C   REAL Z(500),P(500),TK(500),RH(500)
C
C   STATEMENT FUNCTION YINTERP INTERPOLATES LINEARLY BETWEEN 2 POINTS:
C
C   YINTERP (X1,X,X2,Y1,Y2) = Y2 + ((Y1-Y2) * (X-X2) / (X1-X2))
C
C   BUMP UP THE OVERLYING PROFILE LEVELS TO MAKE ROOM FOR THE NEW LEVEL.
C   IF THE RESULTING NUMBER OF LEVELS EXCEEDS THE SPECIFIED MAXIMUM,
C   OUTPUT A MESSAGE AND STOP THE PROGRAM.
C
C   NL = NL + 1
C   IF(NL.GT.MAXL) THEN
C       WRITE (MESSFIL,1000) DATE,MAXL
C       STOP
C   END IF
C
C   NLABOVE = NL - LNEW
C   DO 11 I = 1,NLABOVE
C       L = NL - I
C       Z(L+1) = Z(L)
C       P(L+1) = P(L)
C       TK(L+1) = TK(L)
C       RH(L+1) = RH(L)
11  CONTINUE
C
C   ADD THE NEW LEVEL IN TERMS OF PRESSURE (IPZ=1) OR HEIGHT (IPZ=0).
C
C   PBELOW = ALOG (P(LNEW-1))
C   PABOVE = ALOG (P(LNEW+1))
C
C   IF (IPZ.EQ.1) THEN
C       X1 = PBELOW
C       X = ALOG (XNEW)
C       X2 = PABOVE
C       P(LNEW) = XNEW
C       Z(LNEW) = YINTERP (X1,X,X2,Z(LNEW-1),Z(LNEW+1))
```

SUBROUTINE ADDLEVL, continued

```
ELSE
  X1 = Z(LNEW-1)
  X = XNEW
  X2 = Z(LNEW+1)
  PLOG = YINTERP (X1,X,X2,PBELOW,PABOVE)
  P(LNEW) = EXP (PLOG)
  Z(LNEW) = XNEW
END IF

TK(LNEW) = YINTERP (X1,X,X2,TK(LNEW-1),TK(LNEW+1))
RH(LNEW) = YINTERP (X1,X,X2,RH(LNEW-1),RH(LNEW+1))

RETURN

1000 FORMAT (1X,'FROM ADDLEVL: (',2(I2,'/'),I2,I3,I2,' Z) LEVEL ADDE',
1 'D MAKES NUMBER OF LEVELS EXCEED',/,33X,'SPECIFIED MAXIM',
2 'UM (MAXL=',I3,'). STOPPING...')

END
```

SUBROUTINE INSERT

```
SUBROUTINE INSERT (DPMAX,Z,P,TK,RH,NL)

C   INSERTS ADDITIONAL PROFILE LEVELS WHERE LEVELS ARE TOO FAR APART,
C   AS DETERMINED BY THE INPUT CRITERION, DPMAX.
C
C   INPUT: DPMAX = MAXIMUM ACCEPTABLE PRESSURE DIFFERENCE (MB) BETWEEN
C           ADJACENT PROFILE LEVELS. SETTING DPMAX=0 SELECTS AN
C           ALGORITHM THAT RELAXES THE PRESSURE DIFFERENCE CONSTRAINT (DP) WITH
C           HEIGHT. THE ALGORITHM ENFORCES A DP OF 5 MB IN THE 1ST 50 MB ABOVE
C           THE SIMULATED ANTENNA ELEVATION (Z0). DP INCREASES BY 5 MB FOR EVERY
C           50-MB DECREASE IN PRESSURE. FOR EXAMPLE, BETWEEN 100 AND 150 MB ABOVE
C           Z0, THE PRESSURE DIFFERENCE BETWEEN LEVELS CANNOT EXCEED 15 MB.
C           WHEN IT DOES, LEVEL(S) ARE INSERTED UNTIL IT DOES NOT.
C
C   INPUTS/OUTPUTS (MAY BE MODIFIED HERE):
C       Z = ORIGINAL/EXPANDED GEOPOTENTIAL HEIGHT PROFILE (KM)
C       P = ORIGINAL/EXPANDED PRESSURE PROFILE (MB)
C       TK = ORIGINAL/EXPANDED TEMPERATURE PROFILE (K)
C       RH = ORIGINAL/EXPANDED RELATIVE HUMIDITY PROFILE (FRACTION)
C       NL = ORIGINAL/EXPANDED NUMBER OF PROFILE LEVELS
C   SUBROUTINE: ADDLEVL = ADDS A PROFILE LEVEL AT A SPECIFIED PRESSURE

REAL Z(500),P(500),TK(500),RH(500)

C   ASSIGN MAXIMUM PRESSURE DIFFERENCE BETWEEN ADJACENT LEVELS (DIFFMAX).
C   NL0 AND I0 ARE THE ORIGINAL NUMBER OF PROFILE LEVELS AND THEIR INDEX.
C   NL AND I WILL CHANGE AS NEW LEVELS ARE INSERTED.

DIFFMAX = DPMAX
NL0 = NL
I = 1

C   INSERT LEVELS WHEREVER A PAIR OF ADJACENT LEVELS ARE TOO FAR APART.

DO 22 I0 = 2,NL0
  I = I + 1

C   COMPUTE NEW DIFFMAX FOR THIS LEVEL IF VARIABLE DIFFMAX WAS SELECTED.

  IF (DPMAX.EQ.0.) THEN
    DIFFMAX = FLOAT (5 * (INT (.02 * (P(1) - P(I))) + 1))
  END IF

C   DETERMINE HOW MANY LEVELS TO INSERT (NINSERT).

  PDIFF = P(I-1) - P(I)
  IF (PDIFF.GT.DIFFMAX) THEN
    NINSERT = INT (PDIFF / DIFFMAX)
    REM = AMOD (PDIFF,DIFFMAX)
    IF (REM.LT.1.E-6) NINSERT = NINSERT - 1

C   IF LEVELS ARE NEEDED, INSERT THEM BY INTERPOLATION.

    IF (NINSERT.GT.0) THEN
      DP = PDIFF / FLOAT (NINSERT + 1)
      PNEW = P(I-1)
      DO 11 J = 1,NINSERT
        PNEW = PNEW - DP
        CALL ADDLEVL (1,PNEW,I,Z,P,TK,RH,NL)
        I = I + 1
      11 CONTINUE
    END IF
  END IF
22 CONTINUE

RETURN
END
```

SUBROUTINE CLDMODL

```

SUBROUTINE CLDMODL (RHCLD,CLDMIN,MAXCLD,Z,P,TK,RH,NL,NLAY,CLDBASE,
1 CLDTOP,LBASE,LTOP,DENLIQ,DENICE,NMODLS)

C   CLOUD MODEL: FINDS CLOUD LAYERS IN THE PROFILE AND MODELS THEIR DENSITY.
C   C   INSERTS ADDITIONAL PROFILE LEVELS AT CLOUD BOUNDARIES.
C   C   INPUTS:
C   C     RHCLD   = THRESHOLD RH USED TO DETECT A CLOUD LAYER (FRACTION)
C   C     CLDMIN  = MINIMUM CLOUD THICKNESS (KM; THINNER CLOUDS ARE IGNORED)
C   C     MAXCLD  = MAXIMUM NUMBER OF CLOUD LAYERS PERMITTED (ARRAY LIMIT=4)
C   C   INPUTS/OUTPUTS (MAY BE MODIFIED HERE):
C   C     Z       = HEIGHT PROFILE (KM ABOVE SPECIFIED STARTING HEIGHT)
C   C     P       = PRESSURE PROFILE (MB)
C   C     TK      = TEMPERATURE PROFILE (K)
C   C     RH      = RELATIVE HUMIDITY PROFILE (FRACTION)
C   C     NL      = NUMBER OF PROFILE LEVELS
C   C   OUTPUTS:
C   C     NLAY    = NUMBER OF CLOUD LAYERS IDENTIFIED
C   C     CLDBASE = ARRAY OF CLOUD BASE HEIGHTS (KM ABOVE ANTENNA HEIGHT)
C   C     CLDTOP  = ARRAY OF CLOUD TOP HEIGHTS (KM ABOVE ANTENNA HEIGHT)
C   C     LBASE   = PROFILE LEVELS CORRESPONDING TO IDENTIFIED CLOUD BASES
C   C     LTOP    = PROFILE LEVELS CORRESPONDING TO IDENTIFIED CLOUD TOPS
C   C     DENLIQ  = CLOUD LIQUID DENSITY PROFILE (G/M3)
C   C     DENICE  = CLOUD ICE DENSITY PROFILE (G/M3)
C   C     NMODLS  = NUMBER OF CLOUD DENSITY MODELS (CLEAR=0; CLOUDY=3)
C   C   SUBROUTINES:
C   C     CLDDENS = COMPUTES CLOUD THICKNESS AND DENSITY
C   C     CLDPROF = COMPUTES CLOUD LIQUID AND ICE DENSITY PROFILES

REAL Z(500),P(500),TK(500),RH(500),DENLIQ(500),DENICE(500),
1 CLDBASE(4),CLDTOP(4),CLDRHO(4)
INTEGER LBASE(4),LTOP(4)

C   C   STATEMENT FUNCTION YINTERP INTERPOLATES LINEARLY BETWEEN 2 POINTS:
C   C   YINTERP (X1,X,X2,Y1,Y2) = Y2 + ((Y1-Y2) * (X-X2) / (X1-X2))
C   C   INITIALIZE CLOUD LAYER FLAG (LAYER) AND CLOUD LAYER COUNTER (NLAY).
C   C   LAYER = 0
C   C   NLAY = 0

C   C   FIND BASES AND TOPS OF CLOUD LAYERS:
C   C   DO 22 I = 1,NL

C   C   IF THE MAXIMUM NUMBER OF CLOUD LAYERS HAS BEEN FOUND, STOP THE SEARCH.
C   C   IF (NLAY.GE.MAXCLD) GO TO 30

C   C   UPON ENTERING A NEW CLOUD LAYER, COMPUTE CLOUD BASE HEIGHT.
C   C   IF ((RH(I).GE.RHCLD).AND.(LAYER.EQ.0)) THEN
C   C     LAYER = 1
C   C     IF ((I.EQ.1).OR.(RH(I).EQ.RHCLD)) THEN
C   C       BASE = Z(I)
C   C     ELSE
C   C       BASE = YINTERP (RH(I-1),RHCLD,RH(I),Z(I-1),Z(I))
C   C     END IF

C   C   UPON LEAVING THE CLOUD LAYER, COMPUTE CLOUD TOP HEIGHT.
C   C   CALL SUBROUTINE CLDDENS TO COMPUTE CLOUD THICKNESS AND DENSITY.

C   C   ELSE IF ((RH(I).LT.RHCLD).AND.(LAYER.EQ.1)) THEN
C   C     LAYER = 0
C   C     TOP = YINTERP (RH(I-1),RHCLD,RH(I),Z(I-1),Z(I))
C   C     CALL CLDDENS (BASE, TOP, CLDMIN, NLAY, CLDBASE, CLDTOP, CLDRHO)
C   C   END IF
22 CONTINUE
```

SUBROUTINE CLDMODL, continued

```
C   IF A PROFILE ENDS WITHIN A CLOUD LAYER, ASSIGN CLOUD TOP HEIGHT;  
C   CALL SUBROUTINE CLDDENS TO COMPUTE CLOUD THICKNESS AND DENSITY.  
  
   IF(LAYER.GT.0) THEN  
     TOP = Z(NL)  
     CALL CLDDENS (BASE, TOP, CLDMIN, NLAY, CLDBASE, CLDTOP, CLDRHO)  
   END IF  
  
C   ASSIGN A DENSITY TO EACH PROFILE LEVEL, ADDING LEVELS AT CLD BOUNDARIES.  
  
30  CALL CLDPROF (NLAY, CLDBASE, CLDTOP, CLDRHO, Z, P, TK, RH, NL, NMODLS,  
1    LBASE, LTOP, DENLIQ, DENICE)  
  
   RETURN  
   END
```

SUBROUTINE CLDDENS

```
SUBROUTINE CLDDENS (BASE, TOP, CLDMIN, NLAY, CLDBASE, CLDTOP, CLDRHO)  
  
C   IF THE INPUT BASE AND TOP DEFINE A SUFFICIENTLY THICK CLOUD, THIS  
C   ROUTINE SAVES THE BASE AND TOP AND COMPUTES THE CLOUD DENSITY, USING  
C   THE MODEL OF DECKER ET AL. [1978: EXPERIMENTAL EVALUATION OF GROUND-  
C   BASED MICROWAVE RADIOMETRIC SENSING OF ATMOSPHERIC TEMPERATURE AND  
C   WATER VAPOR PROFILES. J.APPL.METEOROL., VOL.17, 1788-95]. THE DENSITY  
C   IS DEFINED BY THE CLOUD THICKNESS AND THE RIGHTMOST CURVE IN FIG. 1  
C   (PAGE 1790). MULTIPLYING THIS DENSITY BY 0.5 OR 0.25 RESULTS IN DENSITY  
C   VALUES CONSISTENT WITH THE OTHER TWO CURVES.  
C  
C   INPUTS:  
C     BASE = CLOUD BASE HEIGHT (KM ABOVE ANTENNA HEIGHT)  
C     TOP  = CLOUD TOP HEIGHT (KM ABOVE ANTENNA HEIGHT)  
C     CLDMIN = MINIMUM ACCEPTABLE CLOUD THICKNESS (USER-SPECIFIED)  
C   INPUTS/OUTPUTS (MAY BE MODIFIED HERE):  
C     NLAY = NUMBER OF CLOUD LAYERS IDENTIFIED SO FAR  
C     CLDBASE = ARRAY OF CLOUD BASE HEIGHTS (KM ABOVE ANTENNA HEIGHT)  
C     CLDTOP  = ARRAY OF CLOUD TOP HEIGHTS (KM ABOVE ANTENNA HEIGHT)  
C     CLDRHO  = ARRAY OF CLOUD WATER DENSITIES (G/M3)  
  
   REAL CLDBASE(4), CLDTOP(4), CLDRHO(4)  
  
C   COMPUTE CLOUD THICKNESS. IF THICK ENOUGH, SAVE THE BASE AND TOP HEIGHTS  
C   AND COMPUTE THE DENSITY ASSOCIATED WITH THE RIGHTMOST CURVE IN FIG.1 OF  
C   DECKER ET AL. THE MULTIPLIER, 1.6, IS THE INVERSE SLOPE OF THAT CURVE.  
  
   THICK = TOP - BASE  
   IF (THICK.GE.CLDMIN) THEN  
     NLAY = NLAY + 1  
     CLDBASE(NLAY) = BASE  
     CLDTOP(NLAY) = TOP  
     CLDRHO(NLAY) = THICK * 1.6  
     CLDRHO(NLAY) = AMAX1 (CLDRHO(NLAY), 0.2)  
     CLDRHO(NLAY) = AMIN1 (CLDRHO(NLAY), 0.8)  
   END IF  
  
   RETURN  
   END
```

SUBROUTINE CLDPROF

```

SUBROUTINE CLDPROF (NLAY,CLDBASE,CLDTOP,CLDRHO,Z,P,TK,RH,NL,
1      NMODLS,LBASE,LTOP,DENLIQ,DENICE)

C      ASSIGNS A CLOUD DENSITY TO EACH PROFILE LEVEL, ADDING LEVELS AT CLOUD
C      BOUNDARIES, IF NECESSARY.  THE DENSITY ASSIGNED IS THE LARGEST OF THE
C      3 MODELS DESCRIBED BY DECKER ET AL., 1978 (SEE SUBROUTINE CLDDENS.).
C
C      INPUTS:
C      NLAY      = NUMBER OF CLOUD LAYERS IN THIS PROFILE
C      CLDBASE   = ARRAY OF CLOUD BASE HEIGHTS (KM ABOVE ANTENNA HEIGHT)
C      CLDTOP    = ARRAY OF CLOUD TOP HEIGHTS (KM ABOVE ANTENNA HEIGHT)
C      CLDRHO    = ARRAY OF CLOUD DENSITY VALUES (G/M3)
C      INPUTS/OUTPUTS (MAY BE MODIFIED HERE):
C      Z         = HEIGHT PROFILE (KM ABOVE ANTENNA HEIGHT)
C      P         = PRESSURE PROFILE (MB)
C      TK        = TEMPERATURE PROFILE (K)
C      RH        = RELATIVE HUMIDITY PROFILE (FRACTION)
C      NL        = NUMBER OF PROFILE LEVELS
C
C      OUTPUTS:
C      NMODLS    = NUMBER OF CLOUD DENSITY MODELS (CLEAR=0; CLOUDY=3)
C      LBASE     = PROFILE LEVELS CORRESPONDING TO IDENTIFIED CLOUD BASES
C      LTOP      = PROFILE LEVELS CORRESPONDING TO IDENTIFIED CLOUD TOPS
C      DENLIQ    = CLOUD LIQUID DENSITY PROFILE (G/M3)
C      DENICE    = CLOUD ICE DENSITY PROFILE (G/M3)
C
C      SUBROUTINES:
C      ADDLEVEL  = ADDS A PROFILE LEVEL AT A SPECIFIED HEIGHT
C      FUNCTION CLDICE = COMPUTES ICE PORTION OF CLOUD DENSITY (G/M3)
C
C      INTEGER LBASE(4),LTOP(4)
C      REAL Z(500),P(500),TK(500),RH(500),DENLIQ(500),DENICE(500),
1      CLDBASE(4),CLDTOP(4),CLDRHO(4)
C
C      SET THE NUMBER OF CLOUD DENSITY MODELS (0=CLEAR; 3=CLOUDY).
C
C      NMODLS = 0
C      IF (NLAY.GT.0) NMODLS = 3
C
C      I0 AND NL0 = LEVEL INDEX AND NUMBER OF LEVELS IN THE ORIGINAL PROFILES.
C      I AND NL WILL BE THE LEVEL INDEX AND NUMBER OF LEVELS IN THE PROFILES AS
C      THEY ARE MODIFIED BY THE CLOUD MODELING. LAYER IS THE CLOUD LAYER INDEX.
C
C      NL0 = NL
C      I = 0
C      LAYER = 1
C
C      ASSIGN AN APPROPRIATE CLOUD DENSITY TO EACH LEVEL IN THE PROFILE, ADDING
C      LEVELS AT CLOUD BOUNDARIES IF NECESSARY.
C
C      DO 22 I0 = 1,NL0
C      I = I + 1
C
C      ASSIGN ZERO CLOUD DENSITY AS DEFAULT VALUE. KEEP FOR CLEAR LEVELS.
C
C      DENLIQ(I) = 0.
C      DENICE(I) = 0.
C      IF (LAYER.LE.NLAY) THEN
C
C      THE REST OF LOOP 22 AFFECTS LEVELS IN OR ADJACENT TO A CLOUD LAYER.
C
C      IF (Z(I).GE.CLDBASE(LAYER)) THEN
C
C      IF CLOUD IS AT SURFACE, SET CLOUD BASE LEVEL TO 1; SET DENSITY.
C
C      IF (I.EQ.1) THEN
C      LBASE(LAYER) = I
C      DENICE(I) = CLDICE (CLDRHO(LAYER),TK(I))
C      DENLIQ(I) = CLDRHO(LAYER) - DENICE(I)

```


SUBROUTINE CLDPROF, continued

```
C      FOR AN ELEVATED CLOUD LAYER:
      ELSE IF (Z(I).LE.CLDTOP(LAYER)) THEN
C      IF THIS IS THE FIRST (LOWEST) PROFILE LEVEL WITHIN THE CLOUD:
      IF (Z(I-1).LT.CLDBASE(LAYER)) THEN
C      SAVE THE PROFILE LEVEL NUMBER ASSOCIATED WITH THE CLOUD BASE.
C      IF THERE IS NO LEVEL PRECISELY AT CLOUD BASE, ADD A LEVEL THERE
C      WITH THE APPROPRIATE CLOUD LIQUID AND ICE DENSITY.
      LBASE(LAYER) = I
      IF (Z(I).NE.CLDBASE(LAYER)) THEN
        CALL ADDLEVL (0,CLDBASE(LAYER),I,Z,P,TK,RH,NL)
        DENICE(I) = CLDICE (CLDRHO(LAYER),TK(I))
        DENLIQ(I) = CLDRHO(LAYER) - DENICE(I)
        I = I + 1
      END IF
      END IF
C      ASSIGN CLOUD LIQUID AND ICE DENSITY TO CURRENT (CLOUDY) LEVEL.
      DENICE(I) = CLDICE (CLDRHO(LAYER),TK(I))
      DENLIQ(I) = CLDRHO(LAYER) - DENICE(I)
C      IF THIS LEVEL IS THE CLOUD TOP, SAVE THE LEVEL NUMBER.
      IF (Z(I).EQ.CLDTOP(LAYER)) LTOP(LAYER) = I
C      IF THIS IS THE FIRST (LOWEST) LEVEL ABOVE CLOUD TOP:
      ELSE IF (Z(I-1).LE.CLDTOP(LAYER)) THEN
C      IF THERE WAS NO PROFILE LEVEL PRECISELY AT CLOUD TOP, INSERT A
C      LEVEL THERE WITH APPROPRIATE CLOUD DENSITY; SAVE THE LEVEL INDEX.
      IF (Z(I-1).LT.CLDTOP(LAYER)) THEN
        LTOP(LAYER) = I
        CALL ADDLEVL (0,CLDTOP(LAYER),I,Z,P,TK,RH,NL)
        DENICE(I) = CLDICE (CLDRHO(LAYER),TK(I))
        DENLIQ(I) = CLDRHO(LAYER) - DENICE(I)
        I = I + 1
      END IF
C      ASSIGN CLEAR DENSITY TO CURRENT LEVEL; INCREMENT CLOUD LAYER INDEX.
      DENICE(I) = 0.
      DENLIQ(I) = 0.
      LAYER = LAYER + 1
      END IF
      END IF
      END IF
22  CONTINUE
      RETURN
      END
```

FUNCTION CLDICE

FUNCTION CLDICE (RHOCLD,TK)

C COMPUTES ICE PORTION OF CLOUD WATER DENSITY, DEPENDING ON TEMPERATURE.
C THE ICE FRACTION INCREASES EXPONENTIALLY FROM 0.0 AT 0 C TO 1.0 AT -30 C.
C

C INPUTS:

C RHOCLD = TOTAL CLOUD WATER DENSITY (G/M3)

C TK = TEMPERATURE (K)

C OUTPUTS:

C (CLDICE FUNCTION VALUE) = ICE DENSITY (G/M3)

CLDICE = 0.

IF (TK.LT.273.16) THEN

IF (TK.GT.243.16) THEN

CLDICE = RHOCLD * ((TK - 273.16) ** 4) / 810000.

ELSE

CLDICE = RHOCLD

END IF

END IF

RETURN

END

SUBROUTINE REFRACT

```
SUBROUTINE REFRACT (P,TK,RH,NL,ICE,DRYN,WETN,REFINDX,E,RHO)

C   COMPUTES PROFILES OF WET REFRACTIVITY, DRY REFRACTIVITY, REFRACTIVE
C   INDEX, VAPOR PRESSURE, AND VAPOR DENSITY. REFRACTIVITY EQUATIONS
C   WERE TAKEN FROM G.D. THAYER, 1974: AN IMPROVED EQUATION FOR THE
C   RADIO REFRACTIVE INDEX OF AIR. RADIO SCIENCE, VOL.9,NO.10, 803-807.
C   *** THESE EQUATIONS WERE INTENDED FOR FREQUENCIES UNDER 20 GHZ ***
C
C   INPUTS:
C       P   = PRESSURE PROFILE (MB)
C       TK  = TEMPERATURE PROFILE (K)
C       RH  = RELATIVE HUMIDITY PROFILE (FRACTION)
C       NL  = NUMBER OF PROFILE LEVELS
C       ICE = SWITCH FOR ES OVER WATER ONLY (0) OR WATER AND ICE (1)
C
C   OUTPUTS:
C       DRYN   = DRY REFRACTIVITY PROFILE
C       WETN   = WET REFRACTIVITY PROFILE
C       REFINDX = REFRACTIVE INDEX PROFILE
C       E      = VAPOR PRESSURE PROFILE (MB)
C       RHO    = VAPOR DENSITY PROFILE (G/M3)
C   SUBROUTINE: VAPOR = COMPUTES VAPOR PRESSURE AND VAPOR DENSITY
C
REAL P(500),TK(500),RH(500),DRYN(500),WETN(500),REFINDX(500),
1   E(500),RHO(500)

DO 11 I = 1,NL

C   COMPUTE VAPOR PRESSURE (E) AND VAPOR DENSITY (RHO) FOR THIS LEVEL.
C   CALL VAPOR (TK(I),RH(I),ICE,E(I),RHO(I))

C   CALCULATE DRY AIR PRESSURE (PA) AND CELSIUS TEMPERATURE (TC).

PA = P(I) - E(I)
TC = TK(I) - 273.16

C   CALCULATE INVERSE WET (RZW) AND DRY (RZA) COMPRESSIBILITY FACTORS.

TK2 = TK(I) * TK(I)
TC2 = TC * TC
RZA = 1. + PA * (57.90E-8 * (1. + .52/TK(I)) - 9.4611E-4 * TC/TK2)
RZW = 1. + 1650. * (E(I) / (TK(I)*TK2)) * (1. - .01317 * TC +
1   1.75E-4 * TC2 + 1.44E-6 * (TC2 * TC))

C   CALCULATE WET REFRACTIVITY, DRY REFRACTIVITY, AND REFRACTIVE INDEX.

WETN(I) = (64.79 * (E(I)/TK(I)) + (3.776E+5) * (E(I)/TK2)) * RZW
DRYN(I) = 77.6036 * (PA / TK(I)) * RZA
REFINDX(I) = 1. + (DRYN(I) + WETN(I)) * 1.E-6

11 CONTINUE

RETURN
END
```

SUBROUTINE RAYTRAC

```
SUBROUTINE RAYTRAC (NL,ANGLES,NANG,Z,REFINDX,Z0,DS,IRAY,NANGOK)

C RAY-TRACING ALGORITHM OF DUTTON, THAYER, AND WESTWATER, REWRITTEN FOR
C READABILITY & ATTEMPTED DOCUMENTATION. BASED ON THE TECHNIQUE SHOWN IN
C "RADIO METEOROLOGY" BY BEAN AND DUTTON (FIG. 3.20 AND SURROUNDING TEXT).
C
C INPUTS PASSED AS ARGUMENTS:
C   NL      = NUMBER OF LEVELS IN THE PROFILE (> OR = 2)
C   ANGLES  = ELEVATION ANGLE ARRAY (DEGREES)
C   NANG    = NUMBER OF ANGLES IN ARRAY ANGLES
C   Z       = HEIGHT PROFILE (KM ABOVE OBSERVATION HEIGHT, Z0)
C   REFINDX = REFRACTIVE INDEX PROFILE
C   Z0      = OBSERVATION HEIGHT (KM MSL)
C INPUTS PASSED THROUGH COMMON:
C   MESSFIL = UNIT NUMBER ASSOCIATED WITH MESSAGE OUTPUT FILE
C   DATE    = INTEGER ARRAY CONTAINING DATE AND TIME OF CURRENT RAOB
C   DEG2RAD = CONVERSION FACTOR: DEGREES TO RADIAN
C   RE      = MEAN EARTH RADIUS (KM)
C OUTPUTS:
C   DS      = ARRAY CONTAINING SLANT PATH LENGTH PROFILES (KM)
C   IRAY    = FLAG ARRAY INDICATING VALID (1) OR INVALID (0) RAY PATHS
C   NANGOK  = NUMBER OF ANGLES WITH VALID RAY PATHS
C
C COMMON BLOCKS CONTAIN UNIT NOS. & CONSTANTS SET IN SUBROUTINE INITCON:
C
C COMMON /IOFILES/ SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE
C COMMON /CONSTNT/ RE,G,EPSILON,RDRY,RVAP,MAXL,EXPMAX,DB2NP,DEG2RAD
C
C INTEGER SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE(5),IRAY(10)
C REAL REFINDX(500),Z(500),DS(500,10),ANGLES(10)
C
C CHECK FOR REFRACTIVE INDEX VALUES THAT WILL BLOW UP CALCULATIONS.
C
C   IREF = 0
C   DO 11 I = 1,NL
C     IF (REFINDX(I).LT.1.) THEN
C       IREF = 1
C       WRITE (MESSFIL,1000) DATE, Z(I)
C     END IF
11  CONTINUE
C
C LOOP OVER SPECIFIED ELEVATION ANGLES, DETERMINING SLANT PATH IF POSSIBLE.
C
C   NANGOK = 0
C   DO 44 K = 1,NANG
C
C     IF ANGLE IS CLOSE TO 90 DEGREES, MAKE DS A HEIGHT DIFFERENCE PROFILE.
C
C     IF (((ANGLES(K).GE.89.).AND.(ANGLES(K).LE.91.)).OR.
1    ((ANGLES(K).GE.-91.).AND.(ANGLES(K).LE.-89.))) THEN
C       IRAY(K) = 1
C       NANGOK = NANGOK + 1
C       DS(1,K) = 0.
C       DO 22 I = 2,NL
C         DS(I,K) = Z(I) - Z(I-1)
22  CONTINUE
C
C THE REST OF THE SUBROUTINE APPLIES ONLY TO ANGLES OTHER THAN 90 DEGREES.
C SET INVALID RAY PATH FLAG IF ANY REFRACTIVE INDEX VALUE WAS LESS THAN 1.
C
C   ELSE IF (IREF.GT.0) THEN
C     IRAY(K) = 0
```

SUBROUTINE RAYTRAC, continued

```
C OTHERWISE, SET VALID RAY PATH FLAG AND COMPUTE SLANT PATH
ELSE
  IRAY(K) = 1
  NANGOK = NANGOK + 1
C CONVERT ANGLE DEGREES TO RADIAN. INITIALIZE CONSTANT VALUES.
  THETA0 = ANGLES(K) * DEG2RAD
  RS = RE + Z(1) + Z0
  COSTH0 = COS(THETA0)
  SINA = SIN(THETA0 * 0.5)
  A0 = 2. * (SINA**2)
C INITIALIZE LOWER BOUNDARY VALUES FOR 1ST LAYER.
  DS(1,K) = 0.
  PHIL = 0.
  TAUL = 0.
  RL = RE + Z(1) + Z0
  TANTHL = TAN(THETA0)
C CONSTRUCT THE SLANT PATH LENGTH PROFILE.
  DO 33 I = 2,NL
    R = RE + Z(I) + Z0
C COMPUTE LAYER-AVERAGE REFRACTIVE INDEX.
    IF ((REFINDX(I).EQ.REFINDX(I-1)) .OR. (REFINDX(I).EQ.1.)
      1 .OR. (REFINDX(I-1).EQ.1.)) THEN
      REFBAR = (REFINDX(I) + REFINDX(I-1)) * 0.5
    ELSE
      1 REFBAR = 1. + (REFINDX(I-1) - REFINDX(I)) /
        (ALOG ((REFINDX(I-1) - 1.) / (REFINDX(I) - 1.)))
    END IF
    ARGDTH = Z(I) / RS - ((REFINDX(I-1) - REFINDX(I)) *
      1 COSTH0 / REFINDX(I))
    ARGTH = 0.5 * (A0 + ARGDTH) / R
C CHECK FOR DUCTING. IF FOUND, SET INVALID RAY PATH FLAG + SKIP OUT.
    IF (ARGTH.LE.0.) THEN
      IRAY(K) = 0
      NANGOK = NANGOK - 1
      WRITE (MESSFIL,2000) DATE,ANGLES(K)
      GO TO 44
    END IF
C COMPUTE D-THETA FOR THIS LAYER.
    SINT = SQRT (R * ARGTH)
    THETA = 2. * ASIN (SINT)
    IF ((THETA - 2. * THETA0).LE.0.) THEN
      DENDTH = 2. * (SINT + SINA) * COS((THETA + THETA0) * 0.25)
      SIND4 = (0.5 * ARGDTH - Z(I) * ARGTH) / DENDTH
      DTHETA = 4. * ASIN (SIND4)
      THETA = THETA0 + DTHETA
    ELSE
      DTHETA = THETA - THETA0
    END IF
```

SUBROUTINE RAYTRAC, continued

```
C      COMPUTE D-TAU FOR THIS LAYER (EQ.3.71) AND ADD TO INTEGRAL, TAU.

      TANTH = TAN (THETA)
      CTHBAR = ((1. / TANTH) + (1. / TANTHL)) * 0.5
      DTAU = CTHBAR * (REFINDX(I-1) - REFINDX(I)) / REFBAR
      TAU = TAU + DTAU
      PHI = DTHETA + TAU

C      COMPUTE LENGTH OF ARC ACROSS THIS LAYER (SEGMENT QF IN FIG.3.20).

      DS(I,K) = SQRT ((Z(I) - Z(I-1))**2 + 4. * R * RL *
1      ((SIN ((PHI - PHIL) * 0.5))**2))
      IF (DTAU.NE.0.) THEN
          DTAUA = ABS (TAU - TAU)
          DS(I,K) = DS(I,K) * (DTAUA / (2. * SIN (DTAUA * 0.5)))
      END IF

C      MAKE UPPER BOUNDARY INTO LOWER BOUNDARY FOR NEXT LAYER.

      PHIL = PHI
      TAU = TAU
      RL = R
      TANTHL = TANTH
33      CONTINUE
      END IF
44      CONTINUE

      RETURN

1000  FORMAT (1X,'FROM RAYTRAC: (',2(I2,'/'),',I2,I3,I2,' Z) REFRACTIVE',
1      ' INDEX < 1 AT',F6.3,' KM.',/,33X,'SUPPRESSING CALCULATI',
2      'ONS FOR NON-ZENITH ANGLES.')
2000  FORMAT (1X,'FROM RAYTRAC: (',2(I2,'/'),',I2,I3,I2,' Z) SUPPRESSIN',
1      'G CALCULATIONS FOR',F7.3,' DEGREES',/,33X,'ELEVATION AN',
2      'GLE DUE TO DUCTING CONDITIONS.')

      END
```

SUBROUTINE CLRABS

SUBROUTINE CLRABS (P,TK,E,NL,FRQ,NFRQ,HVK,BOFT,AWET,ADRY)

```
C   COMPUTES B(T) PROFILE, WHERE B(T)=PLANCK FUNCTION WITHOUT CONSTANT FACTOR.
C   COMPUTES PROFILES OF WATER VAPOR AND DRY AIR ABSORPTION FOR A GIVEN SET
C   OF FREQUENCIES. SUBROUTINES H2O AND O2 CONTAIN THE ABSORPTION MODEL OF
C   LIEBE AND LAYTON [1987: MILLIMETER-WAVE PROPERTIES OF THE ATMOSPHERE:
C   LABORATORY STUDIES AND PROPAGATION MODELING. NTIA REPORT 87-224, 74PP.]
C   WITH OXYGEN INTERFERENCE COEFFICIENTS FROM ROSENKRANZ
C   [1988: INTERFERENCE COEFFICIENTS FOR OVERLAPPING OXYGEN LINES IN AIR.
C   J. QUANT. SPECTROSC. RADIAT. TRANSFER, 39, 287-97.]
C
C   INPUTS PASSED AS ARGUMENTS:
C   P       = PRESSURE PROFILE (MB)
C   TK      = TEMPERATURE PROFILE (K)
C   E       = VAPOR PRESSURE PROFILE (MB)
C   NL      = NUMBER OF PROFILE LEVELS
C   FRQ     = FREQUENCY ARRAY (GHZ)
C   NFRQ    = NUMBER OF FREQUENCIES IN ARRAY FRQ
C   HVK     = (PLANCK CONSTANT * FREQUENCY(J)) / BOLTZMANN CONST. (K)
C   INPUT PASSED THROUGH COMMON:
C   DB2NP   = CONVERSION FACTOR: DECIBELS TO NEPERS
C
C   OUTPUTS:
C   BOFT    = B(T) PROFILE COMPUTED FROM RAOB TEMPERATURE PROFILE
C   AWET    = WATER VAPOR ABSORPTION PROFILE (NP/KM)
C   ADRY    = DRY AIR ABSORPTION PROFILE (NP/KM)
C
C   SUBROUTINES:
C   H2O     = COMPUTES WATER VAPOR ABSORPTION
C   O2      = COMPUTES OXYGEN (DRY AIR) ABSORPTION
C             (H2O AND O2 BOTH CALL SUBROUTINE SHAPE.)
C
C   COMMON BLOCK CONTAINS CONSTANTS INITIALIZED IN SUBROUTINE INITCON:
C
C   COMMON /CONSTNT/ RE,G,EPSILON,RDRY,RVAP,MAXL,EXPMAX,DB2NP,DEG2RAD
C
C   REAL NPP,NCPP,P(500),TK(500),E(500),AWET(500,11),ADRY(500,11),
1   FRQ(11),HVK(11),BOFT(500,11)
C
C   DO 22 J = 1,NFRQ
C     FACTOR = .182 * FRQ(J)
C     DO 11 I = 1,NL
C
C       COMPUTE B(T) FOR THIS LEVEL AND FREQUENCY.
C
C       BOFT(I,J) = 1. / (EXP (HVK(J) / TK(I)) - 1.)
C
C       COMPUTE INVERSE TEMPERATURE PARAMETER; CONVERT WET AND DRY P TO KPA.
C
C       V = 300. / TK(I)
C       EKPA = E(I) / 10.
C       PDRYKPA = P(I) / 10. - EKPA
C
C       COMPUTE H2O AND O2 ABSORPTION (DB/KM) AND CONVERT TO NP/KM.
C
C       CALL H2O (PDRYKPA,V,EKPA,FRQ(J),NPP,NCPP)
C       AWET(I,J) = (FACTOR * (NPP + NCPP)) * DB2NP
C       CALL O2 (PDRYKPA,V,EKPA,FRQ(J),NPP,NCPP)
C       ADRY(I,J) = (FACTOR * (NPP + NCPP)) * DB2NP
C
11      CONTINUE
22      CONTINUE
C
C   RETURN
C   END
```

SUBROUTINE H2O

SUBROUTINE H2O (P,V,E,FRQ,NPP,NCPP)

C COMPUTES WATER VAPOR LINE ABSORPTION (N"=NPP) AND CONTINUUM ABSORPTION
 C (NCPP). ORIGINALLY, EQUATIONS WERE TAKEN FROM [LIEBE, H.J., 1985: AN UP-
 C DATED MODEL FOR MILLIMETER WAVE PROPAGATION IN MOIST AIR. RADIO SCIENCE,
 C VOL.20, 1069-89]. THIS VERSION INCORPORATES THE CHANGES TO EQUATIONS AND
 C INTERFERENCE COEFFICIENTS RECOMMENDED BY LIEBE THRU SUMMER, 1988.

C *** CHECKED AGAINST AUG., 1987 PAPER BY LIEBE AND LAYTON ON 8/12/87
 C [MILLIMETER-WAVE PROPERTIES OF THE ATMOSPHERE: LABORATORY STUDIES
 C AND PROPAGATION MODELING. NTIA REPORT 87-224, 74PP.].
 C MATCHES, EXCEPT FOR EXCLUSION OF DOPPLER AND ZEEMAN APPROXIMATIONS
 C AND ABSENCE OF HIGH FREQUENCY LINE SHAPE FUNCTION ADJUSTMENT.

C INPUTS: P = PRESSURE (KPA)
 C V = 300./T (T IN KELVINS)
 C E = WATER VAPOR PARTIAL PRESSURE (KPA)
 C FRQ = FREQUENCY (GHZ)

C OUTPUTS: NPP = N"= WATER VAPOR LINE ABSORPTION
 C NCPP = WATER VAPOR CONTINUUM ABSORPTION

C SUBROUTINE: SHAPE = COMPUTES F"= IMAGINARY PART OF LINE SHAPE FUNCTION

REAL NPP,NCPP,F0H2O(30),B(3,30)

DATA F0H2O/ 22.235000, 67.813960,119.995940,183.310117,321.225644,
 + 325.152919,336.187000,380.197372,390.134500,437.346667,
 + 439.150812,443.018295,448.001075,470.888947,474.689127,
 + 488.491133,503.568532,504.482692,556.936002,620.700807,
 + 658.006500,752.033227,841.073593,859.865000,899.407000,
 + 902.555000,906.205524,916.171582,970.315022,987.926764/
 DATA ((B(I,J),I=1,3),J=1,30)/ .1090, 2.1430, 27.8400, .0011,
 + 8.7300, 27.6000, .0007, 8.3470, 27.0000, 2.3000, .6530,
 + 31.6400, .0464, 6.1560, 21.4000, 1.5400, 1.5150, 29.7000,
 + .0010, 9.8020, 26.5000, 11.9000, 1.0180, 30.3600, .0044,
 + 7.3180, 19.0000, .0637, 5.0150, 13.7000, .9210, 3.5610,
 + 16.4000, .1940, 5.0150, 14.4000, 10.6000, 1.3700, 23.8000,
 + .3300, 3.5610, 18.2000, 1.2800, 2.3420, 19.8000, .2530,
 + 2.8140, 24.9000, .0374, 6.6930, 11.5000, .0125, 6.6930,
 + 11.9000,510.0000, .1140, 30.0000, 5.0900, 2.1500, 22.3000,
 + .2740, 7.7670, 30.0000,250.0000, .3360, 28.6000, .0130,
 + 8.1130, 14.1000, .1330, 7.9890, 28.6000, .0550, 7.8450,
 + 28.6000, .0380, 8.3600, 26.4000, .1830, 5.0390, 23.4000,
 + 8.5600, 1.3690, 25.3000, 9.1600, 1.8420, 24.0000,138.0000,
 + .1780, 28.6000/

C USE THE COEFFICIENTS ABOVE TO COMPUTE LINE FACTORS (S,GAMMA,DELTA),
 C F" (FPP), AND N" (NPP) FOR WATER VAPOR.

NPP = 0.
 DO 11 I = 1,30
 S = B(1,I) * E * V ** 3.5 * EXP (B(2,I) * (1.- V))
 GAMMA = B(3,I) * (P * V **.6 + 4.8 * E * V ** 1.1) * 1.E-3
 DELTA = 0.
 CALL SHAPE (FRQ,F0H2O(I),GAMMA,DELTA,FPP)
 NPP = NPP + S * FPP

11 CONTINUE

C COMPUTE WATER VAPOR CONTINUUM ABSORPTION.

BF = 1.13E-6
 BS = 3.57E-5
 NCPP = (BF * E * P * V ** 3. + BS * E * E * V ** 10.8) * FRQ
 RETURN
 END

SUBROUTINE SHAPE

```
SUBROUTINE SHAPE (FRQ,VO,GAMMA,DELTA,FPP)
C   CALCULATES FPP = F'' = IMAGINARY PART OF LINE SHAPE FUNCTION (INVERSE GHZ)
G2 = GAMMA * GAMMA
VMF = VO - FRQ
VPF = VO + FRQ
FPP = FRQ / VO * ((GAMMA - VMF * DELTA) / (VMF ** 2 + G2) +
1      (GAMMA - VPF * DELTA) / (VPF ** 2 + G2))
RETURN
END
```

SUBROUTINE O2

```
SUBROUTINE O2 (P,V,E,FRQ,NPP,NCPP)
C   COMPUTES MOLECULAR OXYGEN LINE ABSORPTION (N''=NPP) AND DRY AIR CONTINUUM
C   ABSORPTION (NCPP). ORIGINALLY, EQUATIONS WERE TAKEN FROM [LIEBE, H.J.,
C   1985: AN UPDATED MODEL FOR MILLIMETER WAVE PROPAGATION IN MOIST AIR.
C   RADIO SCIENCE, VOL.20, 1069-89]. THIS VERSION INCORPORATES THE CHANGES TO
C   EQUATIONS AND INTERFERENCE COEFFICIENTS RECOMMENDED BY LIEBE THRU 4/88.
C
C   *** NONRESONANT ROLLOFF FACTOR REMOVED & GO CONSTANT CHANGED ON 6/9/87.
C
C   *** CHECKED AGAINST AUG., 1987 PAPER BY LIEBE AND LAYTON ON 8/12/87.
C   [MILLIMETER-WAVE PROPERTIES OF THE ATMOSPHERE: LABORATORY STUDIES
C   AND PROPAGATION MODELING. NTIA REPORT 87-224, 74PP.].
C   MATCHED, EXCEPT FOR EXCLUSION OF DOPPLER AND ZEEMAN APPROXIMATIONS
C   AND ABSENCE OF HIGH FREQUENCY LINE SHAPE FUNCTION ADJUSTMENT.
C
C   *** INTERFERENCE COEFFICIENTS A5 AND A6 UPDATED 4/88, AFTER
C   ROSENKRANZ [1988: INTERFERENCE COEFFICIENTS FOR OVERLAPPING OXYGEN LINES
C   IN AIR. J. QUANT. SPECTROSC. RADIAT. TRANSFER, 39, 287-97.].
C
C   INPUTS: P   = PRESSURE (KPA)
C           V   = 300./T   (T IN KELVINS)
C           E   = WATER VAPOR PARTIAL PRESSURE (KPA)
C           FRQ = FREQUENCY (GHZ)
C
C   OUTPUTS: NPP = N'' = MOLECULAR OXYGEN LINE ABSORPTION
C           NCPP = DRY AIR CONTINUUM ABSORPTION
C
C   SUBROUTINE: SHAPE = COMPUTES F''= IMAGINARY PART OF LINE SHAPE FUNCTION
C
C   REAL NPP,NCPP,F002(48),A(6,48)
DATA F002/ 49.452379, 49.962257, 50.474238, 50.987748, 51.503350,
+         52.021409, 52.542393, 53.066906, 53.595748, 54.129999,
+         54.671157, 55.221365, 55.783800, 56.264777, 56.363387,
+         56.968180, 57.612481, 58.323874, 58.446589, 59.164204,
+         59.590982, 60.306057, 60.434775, 61.150558, 61.800152,
+         62.411212, 62.486253, 62.997974, 63.568515, 64.127764,
+         64.678900, 65.224067, 65.764769, 66.302088, 66.836827,
+         67.369595, 67.900862, 68.431001, 68.960306, 69.489021,
+         70.017342,118.750341,368.498350,424.763120,487.249370,
+         715.393150,773.838730,834.145330/
```

SUBROUTINE O2, continued

```

DATA ((A(I,J),I=1,6),J=1,16)/ .120, 11.830, 8.400, 6.700,
+ 1.700, .000, .340, 10.720, 8.500, 6.700, 1.700,
+ .000, .940, 9.690, 8.600, 6.700, 1.700, .000,
+ 2.460, 8.690, 8.700, 6.700, 1.700, .000, 6.080,
+ 7.740, 8.900, 6.685, 1.700, .000, 14.140, 6.840,
+ 9.200, 6.403, 1.700, .000, 31.020, 6.000, 9.400,
+ 6.099, 1.800, .000, 64.100, 5.220, 9.700, 5.772,
+ 1.800, .000, 124.700, 4.480, 10.000, 5.454, 1.900,
+ .000, 228.000, 3.810, 10.200, 5.206, 1.900, .000,
+ 391.800, 3.190, 10.500, 4.814, 1.900, .000, 631.600,
+ 2.620, 10.790, 4.363, 2.000, .000, 953.500, 2.115,
+ 11.100, 4.105, 2.000, .000, 548.900, .010, 16.460,
+ 2.812, .800, .000, 1344.000, 1.655, 11.440, 4.154,
+ 2.000, .000, 1763.000, 1.255, 11.810, 3.548, 2.100,
+ .000/
DATA ((A(I,J),I=1,6),J=17,32)/2141.000, .910, 12.210, 1.122,
+ 3.400, .000, 2386.000, .621, 12.660, .051, 6.000,
+ .000, 1457.000, .079, 14.490, 6.201, .600, .000,
+ 2404.000, .386, 13.190, -4.174, .100, .000, 2112.000,
+ .207, 13.600, 6.853, .400, .000, 2124.000, .207,
+ 13.820, -6.254, .400, .000, 2461.000, .386, 12.970,
+ 3.341, 0.100, .000, 2504.000, .621, 12.480, -1.308,
+ 2.700, .000, 2298.000, .910, 12.070, -2.059, 2.300,
+ .000, 1933.000, 1.255, 11.710, -4.325, 2.000, .000,
+ 1517.000, .078, 14.680, -4.052, .600, .000, 1503.000,
+ 1.660, 11.390, -4.726, 2.000, .000, 1087.000, 2.110,
+ 11.080, -4.538, 2.000, .000, 733.500, 2.620, 10.780,
+ -4.711, 2.000, .000, 463.500, 3.190, 10.500, -5.105,
+ 1.900, .000, 274.800, 3.810, 10.200, -5.451, 1.900,
+ .000/
DATA ((A(I,J),I=1,6),J=33,48)/ 153.000, 4.480, 10.000, -5.663,
+ 1.800, .000, 80.090, 5.220, 9.700, -5.948, 1.800,
+ .000, 39.460, 6.000, 9.400, -6.246, 1.700, .000,
+ 18.320, 6.840, 9.200, -6.522, 1.700, .000, 8.010,
+ 7.740, 8.900, -6.775, 1.700, .000, 3.300, 8.690,
+ 8.700, -6.700, 1.700, .000, 1.280, 9.690, 8.600,
+ -6.700, 1.700, .000, .470, 10.720, 8.500, -6.700,
+ 1.700, .000, .160, 11.830, 8.400, -6.700, 1.700,
+ .000, 945.000, .000, 16.300, -.244, .800, .000,
+ 67.900, .020, 19.200, .000, 1.000, .600, 638.000,
+ .011, 19.160, .000, 1.000, .600, 235.000, .011,
+ 19.200, .000, 1.000, .600, 99.600, .089, 18.100,
+ .000, 1.000, .600, 671.000, .079, 18.100, .000,
+ 1.000, .600, 180.000, .079, 18.100, .000, 1.000,
+ .600/

```

C USE THE COEFFICIENTS ABOVE TO COMPUTE LINE FACTORS (S,GAMMA,DELTA),
C F" (FPP), AND N" (NPP) FOR OXYGEN.

```

NPP = 0.
DO 11 I = 1,48
  S = A(1,I) * P * V ** 3 * EXP (A(2,I) * (1.- V)) * 1.E-6
  GAMMA = A(3,I) * (P * V ** (.8 - A(6,I)) + 1.1 * E * V) * 1.E-3
  DELTA = A(4,I) * P * V ** A(5,I) * 1.E-3
  CALL SHAPE (FRQ,F002(I),GAMMA,DELTA,FPP)
  NPP = NPP + S * FPP

```

11 CONTINUE
NPP = AMAX1 (NPP,0.)

C COMPUTE DRY AIR CONTINUUM ABSORPTION.

```

AO = 6.14E-4
GO = 4.8E-3 * (P + 1.1 * E) * V ** .8
FSQ = FRQ * FRQ
TRM1 = AO / (GO * (1. + (FSQ / (GO * GO))))
TRM2 = 1.4E-10 * (1. - (1.2E-5 * (FRQ ** 1.5))) * P * (V ** 1.5)
NCP = (TRM1 + TRM2) * FRQ * P * V * V

```

C RETURN
END

SUBROUTINE EXPINT1

```
SUBROUTINE EXPINT1 (X,DS,K,NL,FACTOR,SXDS)

C   INTEGRATES ARRAY X ALONG COLUMN K OF DS, ASSUMING EXPONENTIAL DECAY
C   OVER EACH LAYER.  THE RESULT IS MULTIPLIED BY FACTOR.
C   NOTE:
C       X IN G/M3 AND DS IN KM PRODUCES SXDS IN MM.  PASSING IN A
C       FACTOR VALUE OF 0.1 WILL RETURN THE RESULT IN CM.
C   INPUTS:
C       X       = ARRAY TO BE INTEGRATED
C       DS      = PATH OF INTEGRATION (LEVELS I + I-1 ARE SEPARATED BY DS(I,K))
C       K       = COLUMN INDEX OF ARRAY DS
C       NL      = NUMBER OF LEVELS IN X AND DS
C       FACTOR  = FACTOR BY WHICH RESULT IS MULTIPLIED (E.G., UNIT CHANGE)
C   OUTPUT:
C       SXDS   = NUMERICAL INTEGRAL OF X*DS OVER THE NL LEVELS.

REAL X(500),DS(500,10)

SXDS = 0.0

DO 11 I = 2,NL
  IF (ABS (X(I) - X(I-1)) .LT. 1.E-9) THEN
    XDS = DS(I,K) * X(I)
  ELSE IF ((X(I-1).LE.0.).OR.(X(I).LE.0.)) THEN
    XDS = DS(I,K) * ((X(I) + X(I-1)) * .5)
  ELSE
    XDS = DS(I,K) * ((X(I) - X(I-1)) / ALOG (X(I) / X(I-1)))
  END IF
  SXDS = SXDS + XDS
11 CONTINUE

SXDS = FACTOR * SXDS

RETURN
END
```

SUBROUTINE TAUCALC

```

SUBROUTINE TAUCALC (ABS1,ABS2,J,DS,K,LBEG,LEND,TAU1,TAU2,TAULAY)

C   INTEGRATES ABSORPTION (TAU) OVER EACH LAYER BETWEEN LEVELS LBEG AND LEND
C   OF PROFILES ABS1 AND ABS2 AND ADDS THEM TOGETHER. RETURNS THE SUM OF THE
C   LAYER INTEGRALS AND THE TOTAL INTEGRAL FOR EACH ABSORPTION PROFILE.
C
C   INPUTS:
C   ABS1   = 2-DIMENSIONAL ARRAY CONTAINING ABSORPTION PROFILES (NP/KM)
C   ABS2   = 2-DIMENSIONAL ARRAY CONTAINING ABSORPTION PROFILES (NP/KM)
C   J      = COLUMN (FREQUENCY) INDEX FOR ABS1 AND ABS2
C   DS     = 2-DIMENSIONAL ARRAY CONTAINING LAYER DEPTHS (KM)
C   K      = COLUMN (ANGLE) INDEX FOR DS
C   LBEG   = PROFILE LEVEL WHERE INTEGRATION BEGINS
C   LEND   = PROFILE LEVEL WHERE INTEGRATION ENDS
C   OUTPUTS:
C   TAU1   = INTEGRATED ABSORPTION OVER ABS1 LEVELS LBEG TO LEND (NP)
C   TAU2   = INTEGRATED ABSORPTION OVER ABS2 LEVELS LBEG TO LEND (NP)
C   TAULAY = ARRAY CONTAINING LAYER INTEGRALS: ABS1 PLUS ABS2 (NP)
C   SUBROUTINE:
C   EXPINT2 = INTEGRATES (EXPONENTIAL) PROFILE, SAVING LAYER INTEGRALS

REAL ABS1(500,11),ABS2(500,11),DS(500,10),
1  TAULAY1(500),TAULAY2(500),TAULAY(500)

C   COMPUTE AND INTEGRATE LAYER ABSORPTION PROFILES FROM ABS1 AND ABS2.

CALL EXPINT2 (ABS1,J,DS,K,LBEG,LEND,TAULAY1,TAU1)
CALL EXPINT2 (ABS2,J,DS,K,LBEG,LEND,TAULAY2,TAU2)

C   ADD THE 2 LAYER ABSORPTION PROFILES TOGETHER.

DO 11 L = LBEG,LEND
  TAULAY(L) = TAULAY1(L) +TAULAY2(L)
11 CONTINUE

RETURN
END
```

SUBROUTINE EXPINT2

```

SUBROUTINE EXPINT2 (X,J,DS,K,IBEG,IEND,XDS,SXDS)

C   INTEGRATES THE PROFILE IN COLUMN J OF ARRAY X OVER THE LAYERS
C   DEFINED BY COLUMN K OF ARRAY DS, SAVING THE INTEGRALS OVER EACH LAYER.
C   THE ALGORITHM ASSUMES THAT X DECAYS EXPONENTIALLY OVER EACH LAYER.
C   *** OUTPUTS MESSAGE AND STOPS PROGRAM IF X CONTAINS A NEGATIVE VALUE.***
C
C   INPUTS PASSED AS ARGUMENTS:
C     X       = 2-DIMENSIONAL ARRAY WHOSE COLUMNS CONTAIN PROFILES
C     J       = COLUMN OF ARRAY X CONTAINING PROFILE TO BE INTEGRATED
C     DS      = 2-DIMENSIONAL ARRAY OF LAYER DEPTHS (KM)
C     K       = COLUMN (ANGLE) INDEX OF ARRAY DS
C     IBEG    = LOWER INTEGRATION LIMIT (PROFILE LEVEL NUMBER)
C     IEND    = UPPER INTEGRATION LIMIT (PROFILE LEVEL NUMBER)
C   INPUT PASSED THRU COMMON:
C     DATE    = INTEGER ARRAY CONTAINING DATE AND TIME OF CURRENT RAOB
C   OUTPUTS:
C     XDS     = ARRAY CONTAINING INTEGRALS OVER EACH LAYER DS
C     SXDS    = INTEGRAL OF X*DS OVER LEVELS IBEG TO IEND
C   COMMON BLOCK CONTAINS MESSAGE FILE UNIT NUMBER AND DATE:
C
COMMON /IOFILES/ SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE

INTEGER SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE(5)
REAL X(500,11),DS(500,10),XDS(500)

SXDS = 0.
XDS(IBEG) = 0.

DO 11 I = IBEG+1,IEND

C   CHECK FOR NEGATIVE X VALUE. IF FOUND, OUTPUT MESSAGE AND STOP.
      IF ((X(I-1,J).LT.0.).OR.(X(I,J).LT.0.)) THEN
        WRITE (MESSFIL,1000) DATE,J
        STOP

C   FIND A LAYER VALUE FOR X IN CASES WHERE INTEGRATION ALGORITHM FAILS.
      ELSE IF (ABS (X(I,J) - X(I-1,J)) .LT. 1.E-9) THEN
        XLAYER = X(I,J)
      ELSE IF ((X(I-1,J).EQ.0.) .OR. (X(I,J).EQ.0.)) THEN
        XLAYER = (X(I,J) + X(I-1,J)) * 0.5

C   FIND A LAYER VALUE FOR X ASSUMING EXPONENTIAL DECAY OVER THE LAYER.
      ELSE
        XLAYER = (X(I,J) - X(I-1,J)) / ALOG (X(I,J) / X(I-1,J))
      END IF

C   INTEGRATE X OVER THE LAYER AND SAVE THE RESULT IN XDS.
      XDS(I) = XLAYER * DS(I,K)
      SXDS = SXDS + XDS(I)
11  CONTINUE

      RETURN

1000  FORMAT (1X,'FROM EXPINT2: (',2(12,'/',),12,13,12,' Z) ABSORPTION',
1      ' NEGATIVE FOR CHANNEL NO.',13,'.',/,33X,'STOPPING...')

      END

```

SUBROUTINE RADIANC

```

SUBROUTINE RADIANC (NL,TAULAY,BOFT,J,BOFTBG,HVK,ICLD,IBASE,ITOP,
1      TBATM,TBTOTL,TMR,TMRCLD)

C      COMPUTES BRIGHTNESS TEMPERATURE (TB) & MEAN RADIATING TEMPERATURE (TMR).
C      WE USE THE TERM B(T) FOR THE PLANCK FUNCTION WITHOUT THE CONSTANT FACTOR.
C      *** THE CLOUD TMR IGNORES ALL BUT THE LOWEST CLOUD LAYER.
C
C      INPUTS PASSED AS ARGUMENTS:
C          NL      = NUMBER OF PROFILE LEVELS
C          TAULAY  = PROFILE OF ABSORPTION INTEGRATED OVER EACH LAYER (NP)
C          BOFT   = B(T) PROFILE (DERIVED FROM RAOB TEMPERATURE PROFILE)
C          J      = COLUMN (FREQUENCY) INDEX FOR ARRAYS BOFT AND ALPHA
C          BOFTBG = B(T) FOR COSMIC BACKGROUND RADIATION
C          HVK    = PLANCK CONSTANT * FREQUENCY / BOLTZMANN CONST. (K**-1)
C          ICLD   = SWITCH TO COMPUTE TMR OF LOWEST CLOUD LAYER (1) OR NOT (0)
C          IBASE  = PROFILE LEVEL AT BASE OF LOWEST CLOUD (IGNORED IF ICLD=0)
C          ITOP   = PROFILE LEVEL AT TOP OF LOWEST CLOUD (IGNORED IF ICLD=0)
C      INPUTS PASSED THROUGH COMMON:
C          EXPMAX = MAXIMUM ABSOLUTE VALUE FOR EXPONENTIAL FUNCTION ARGUMENT
C          DATE   = INTEGER ARRAY CONTAINING DATE AND TIME OF CURRENT RAOB
C      OUTPUTS: ← MESSFIL = MESSAGE FILE UNIT NUMBER
C          TBATM  = BRIGHTNESS TEMPERATURE WITHOUT THE COSMIC BACKGROUND (K)
C          TBTOTL = BRIGHTNESS TEMPERATURE WITH THE COSMIC BACKGROUND (K)
C          TMR    = MEAN RADIATING TEMPERATURE OF THE ATMOSPHERE (K)
C          TMRCLD = TMR OF LOWEST CLOUD LAYER (K); ZERO IF ICLD=0.
C
C      COMMON BLOCKS CONTAIN CONSTANT AND UNIT NUMBER FROM SUBROUTINE INITCON:

COMMON /IOFILES/ SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE
COMMON /CONSTNT/ RE,G,EPSILON,RDRY,RVAP,MAXL,EXPMAX,DB2NP,DEG2RAD

INTEGER SPECFIL,RAOBFIL,HEADFIL,MODLFIL,MESSFIL,DATE(5)
REAL TAULAY(500),BOFT(500,11),TAU(500),BOFTATM(500)

C      FUNCTION BRIGHT COMPUTES BRIGHTNESS TEMPERATURE FROM B(TB):

BRIGHT (BOFTB,HVK) = HVK / ALOG (1.+ (1./ BOFTB))

C      ZERO INTEGRALS FOR ABSORPTION (TAU) & ATMOSPHERIC RADIANCE (BOFTATM).

TAU(1) = 0.
BOFTATM(1) = 0.

DO 11 I = 2,NL

C      COMPUTE ATMOSPHERIC RADIANCE FOR THIS LAYER (BATMLAY). IF ABSORPTION IS
C      TOO LARGE TO EXPONENTIATE, ASSUME RADIANCE WAS COMPLETELY ATTENUATED.

IF ((TAULAY(I).LT.EXPMAX).AND.(TAU(I-1).LT.EXPMAX)) THEN
    BOFTLAY = (BOFT(I-1,J) + BOFT(I,J) * EXP (-TAULAY(I))) /
              (1.+ EXP (-TAULAY(I)))
    BATMLAY = BOFTLAY * EXP (-TAU(I-1)) * (1.- EXP (-TAULAY(I)))
ELSE
    BATMLAY = 0.
END IF

C      INCREMENT INTEGRAL ARRAYS WITH VALUES FROM THIS LAYER.

BOFTATM(I) = BOFTATM(I-1) + BATMLAY
TAU(I) = TAU(I-1) + TAULAY(I)
11 CONTINUE

```

SUBROUTINE RADIANC, continued

```
C COMPUTE TB WITH (TBTOTL) AND WITHOUT (TBATM) THE COSMIC BACKGROUND TERM;
C COMPUTE ATMOSPHERIC TMR. IF ABSORPTION IS TOO LARGE TO EXPONENTIATE,
C ASSUME COSMIC BACKGROUND WAS COMPLETELY ATTENUATED AND TMR = TBATM.

TBATM = BRIGHT (BOFTATM(NL),HVK)
IF (TAU(NL).LT.EXPMAX) THEN
  BOFTOTL = (BOFTBG * EXP (-TAU(NL))) + BOFTATM(NL)
  BOFTMR = BOFTATM(NL) / (1.- EXP (-TAU(NL)))
  TBTOTL = BRIGHT (BOFTOTL,HVK)
  TMR = BRIGHT (BOFTMR,HVK)
ELSE
  TBTOTL = TBATM
  TMR = TBATM
END IF

C IF SELECTED (ICLD>0), COMPUTE TMR OF LOWEST CLOUD LAYER (BASE AT LEVEL
C IBASE, TOP AT LEVEL ITOP). OTHERWISE, SET CLOUD TMR TO ZERO.
C *** NOTE: THIS ALGORITHM IS NOT DESIGNED FOR MULTIPLE CLOUD LAYERS.***

IF (ICLD.GT.0) THEN

C OUTPUT MESSAGE AND STOP IF ABSORPTION TOO LARGE TO EXPONENTIATE.

IF (TAU(IBASE).GT.EXPMAX) THEN
  WRITE (MESSFIL,1000) DATE
  STOP
END IF

C COMPUTE RADIANCE (BATMCLD) AND ABSORPTION (TAUCLD) FOR CLOUD LAYER.
C (IF TAUCLD IS TOO LARGE TO EXPONENTIATE, TREAT IT AS INFINITY.)

BATMCLD = BOFTATM(ITOP) - BOFTATM(IBASE)
TAUCLD = TAU(ITOP) - TAU(IBASE)
IF (TAUCLD.GT.EXPMAX) THEN
  BOFTCLD = BATMCLD * EXP (TAU(IBASE))
ELSE
  BOFTCLD = BATMCLD * EXP (TAU(IBASE)) / (1.- EXP (-TAUCLD))
END IF
TMRCLD = BRIGHT (BOFTCLD,HVK)
ELSE
  TMRCLD = 0.
END IF

RETURN

1000 FORMAT (1X,'FROM RADIANC: (',2(12,'/'),12,13,12,' Z) ABSORPTION',
1 ' TOO LARGE TO EXPONENTIATE FOR',/,33X,'TMR OF LOWEST CLOU',
2 'D LAYER. STOPPING...')

END
```

SUBROUTINE CLDABS

```

SUBROUTINE CLDABS (WAVE,NFRQ,NL,TK,DENLIQ,DENICE,DENFRAC,DENL,
1 DENI,ALIQ,AICE)

C MULTIPLIES CLOUD DENSITY PROFILES BY A GIVEN FRACTION AND COMPUTES THE
C CORRESPONDING CLOUD LIQUID AND ICE ABSORPTION PROFILES, USING RAYLEIGH
C APPROXIMATION OF WESTWATER [1972: MICROWAVE EMISSION FROM CLOUDS,13-14].
C
C INPUTS PASSED AS ARGUMENTS:
C WAVE = WAVELENGTH (CM)
C NFRQ = NUMBER OF WAVELENGTHS
C NL = NUMBER OF PROFILE LEVELS
C TK = TEMPERATURE PROFILE (K)
C DENLIQ = ORIGINAL LIQUID DENSITY PROFILE (G/M3)
C DENICE = ORIGINAL ICE DENSITY PROFILE (G/M3)
C DENFRAC = FRACTION OF ORIGINAL DENSITIES TO USE FOR THIS MODEL
C INPUT PASSED THROUGH COMMON:
C DB2NP = CONVERSION FACTOR: DECIBELS TO NEPERS
C
C OUTPUTS:
C DENL = FRACTIONAL LIQUID DENSITY PROFILE (G/M3)
C DENI = FRACTIONAL ICE DENSITY PROFILE (G/M3)
C ALIQ = LIQUID ABSORPTION PROFILE (NP/KM)
C AICE = ICE ABSORPTION PROFILE (NP/KM)
C COMMON BLOCK CONTAINS CONSTANTS INITIALIZED IN SUBROUTINE INITCON:
C
COMMON /CONSTNT/ RE,G,EPSILON,RDRY,RVAP,MAXL,EXPMAX,DB2NP,DEG2RAD
REAL WAVE(11),TK(500),DENLIQ(500),DENICE(500),DENL(500),DENI(500),
1 ALIQ(500,11),AICE(500,11)
COMPLEX B,D,EP2,PKLIQ

DO 22 I = 1,NL

C COMPUTE THE FRACTIONAL CLOUD DENSITIES FOR THIS MODEL AT THIS LEVEL.

DENL(I) = DENFRAC * DENLIQ(I)
DENI(I) = DENFRAC * DENICE(I)

DO 11 J = 1,NFRQ

C COMPUTE LIQUID ABSORPTION (DB/KM); CONVERT NON-ZERO VALUE TO NP/KM.

IF (DENL(I).GT.0.) THEN
1 Y = (EXP ((2.120896162E03 / TK(I)) - 6.680720389) /
WAVE(J)) **.98
DR = 1. + 3.141075969E-02 * Y
DI = 9.995065602E-1 * Y
D = CMPLX (DR,DI)
E1 = (32155.45 / TK(I)) - 34.12
B = CMPLX (E1,0.)
EP2 = (4.5,0.) + B / D
PKLIQ = (EP2 - (1.,0.)) / (EP2 + (2.,0.))
ALIQ(I,J) = (8.18645 / WAVE(J)) * DENL(I) * AIMAG (-PKLIQ)
ALIQ(I,J) = ALIQ(I,J) * DB2NP
ELSE
ALIQ(I,J) = 0.
END IF

C COMPUTE ICE ABSORPTION (DB/KM); CONVERT NON-ZERO VALUE TO NP/KM.

IF (DENI(I).GT.0.) THEN
AICE(I,J) = (8.18645 / WAVE(J)) * DENI(I) * 9.59553E-4
AICE(I,J) = AICE(I,J) * DB2NP
ELSE
AICE(I,J) = 0.
END IF

11 CONTINUE
22 CONTINUE

RETURN
END

```


SUBROUTINE CLDINT

```
SUBROUTINE CLDINT (DENCLD,DS,K,NLAY,LBASE,LTOP,SCLD)
C
C INTEGRATES CLOUD WATER DENSITY OVER PATH DS (LINEAR ALGORITHM).
C
C INPUTS:
C   DENCLD = CLOUD CLOUD WATER DENSITY PROFILE (G/M3)
C   DS     = 2-DIMENSIONAL ARRAY CONTAINING LAYER DEPTH PROFILES (KM)
C   K      = COLUMN (ANGLE) INDEX FOR ARRAY DS
C   NLAY   = NUMBER OF CLOUD LAYERS IN THE PROFILE
C   LBASE  = ARRAY CONTAINING PROFILE LEVELS CORRESPONDING TO CLOUD BAS
C   LTOP   = ARRAY CONTAINING PROFILE LEVELS CORRESPONDING TO CLOUD TOP
C
C OUTPUTS:
C   SCLD   = INTEGRATED CLOUD WATER DENSITY (CM)
C
C   INTEGER LBASE(4),LTOP(4)
C   REAL DENCLD(500),DS(500,10)
C
C   SCLD = 0.
C   DO 22 L = 1,NLAY
C     DO 11 I = LBASE(L)+1, LTOP(L)
C       SCLD = SCLD + DS(I,K) * (0.5 * (DENCLD(I) + DENCLD(I-1)))
11    CONTINUE
22    CONTINUE
C
C CONVERT THE INTEGRATED VALUE TO CM.
C
C   SCLD = SCLD * 0.1
C
C RETURN
C END
```

SUBROUTINE CLDTAU

```

SUBROUTINE CLDTAU (NL,TAUCLR,ALIQ,AICE,J,DS,K,NLAY,LBASE,LTOP,
1      TAU,TAUI,TAULAY)
C      INTEGRATES ABSORPTION IN A PROFILE THAT CONTAINS CLOUD(S).
C
C      INPUTS:
C      NL      = NUMBER OF PROFILE LEVELS
C      TAUCLR  = 3-DIMENSIONAL ARRAY CONTAINING CLEAR LAYER ABSORPTION (NP)
C      ALIQ   = CLOUD LIQUID ABSORPTION PROFILE (NP/KM)
C      AICE   = CLOUD ICE ABSORPTION PROFILE (NP/KM)
C      J      = FREQUENCY INDEX FOR ARRAYS TAUCLR, ALIQ, AND AICE
C      DS     = LAYER DEPTH PROFILE (KM)
C      K      = ANGLE INDEX FOR ARRAYS TAUCLR AND DS
C      NLAY   = NUMBER OF CLOUD LAYERS IN PROFILE
C      LBASE  = ARRAY CONTAINING PROFILE LEVEL OF EACH CLOUD BASE HEIGHT
C      LTOP   = ARRAY CONTAINING PROFILE LEVEL OF EACH CLOUD TOP HEIGHT
C
C      OUTPUTS:
C      TAU     = TOTAL INTEGRATED LIQUID ABSORPTION (NP)
C      TAUI    = TOTAL INTEGRATED ICE ABSORPTION (NP)
C      TAULAY  = LAYER ABSORPTION PROFILE, INCLUDING CLOUDS (NP)
C
C      SUBROUTINE:
C      TAUCALC = COMPUTES PROFILE OF LAYER-INTEGRATED ABSORPTION
C
C      INTEGER LBASE(4),LTOP(4)
C      REAL TAUCLR(500,11,10),ALIQ(500,11),AICE(500,11),DS(500,10),
1      TAUCLD(500),TAULAY(500)
C
C      INITIALIZE LAYER ABSORPTION PROFILE AND INTEGRALS TO CLEAR VALUES.
C
C      DO 11 I = 1,NL
C          TAULAY(I) = TAUCLR(I,J,K)
11      CONTINUE
C      TAU = 0.
C      TAUI = 0.
C
C      FIND THE ABSORPTION IN EACH CLOUD LAYER AND ADD TO CLEAR ABSORPTION.
C
C      DO 33 L = 1,NLAY
C          CALL TAUCALC (ALIQ,AICE,J,DS,K,LBASE(L),LTOP(L),TL,TI,TAUCLD)
C          TAU = TAU + TL
C          TAUI = TAUI + TI
C          DO 22 I = LBASE(L),LTOP(L)
C              TAULAY(I) = TAULAY(I) + TAUCLD(I)
22      CONTINUE
33      CONTINUE
C
C      RETURN
C      END
```