NASA Technical Memorandum 100686

A Computer Code to Calculate Line by Line Atmospheric Transmission Spectra on a Microcomputer

H. G. Safren

JULY 1987

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland 20771

CONTENTS

Page

INTRODUCTION 1

WHAT THE PROGRAM DOES 2

HOW TO USE THE PROGRAM 4

STRUCTURE OF THE PROGRAM 5

MEMORY REQUIREMENTS AND OPERATING SPEED 7

SOME EXAMPLES OF COMPUTED TRANSMISSION SPECTRA 7

TESTING OF PROGRAM 8

PROGRAM LISTINGS 11

- MICTRA Driver Routine and Subroutines 11
- Auxiliary Routines and Command Files 45
- Atmosphere and Aerosol Models 64

REFERENCES 71

INTRODUCTION

To support work being done in the Instrument Electro-Optics Branch in remote sensing a decision was made to set up an in-house capability to compute line by line atmospheric transmission spectra.

An available DEC 11/23 microcomputer was dedicated to this task; its VT100 terminal had been modified by another manufacturer to greatly enhance its graphics capability, so that plots of the computed spectra could be displayed directly on the screen.

The computer program, called MICTRA (for MICrocomputer TRAnsmission calculations) is based on the LASER code published by the Air Force Geophysics Laboratory in 1978 (Reference 1); it uses the 1982 edition of the AFGL Atmospheric Absorption Line Parameters Compilation (Reference 9).

The Air Force code was extensively reworked to allow execution on a desk top microcomputer and to generate transmission spectra rather than tables of extinction coefficients. The program treats absorption due to lines of water vapor, carbon dioxide and ozone throughout the spectrum, from the visible to the microwave region, plus certain "continuum" absorption effects, Rayleigh (molecular) scattering and aerosol absorption and scattering. The program does not include radiance calculations.

The program described in this report, along with another program and a plot package developed by the author (to be published — see References 2 and 3) have proven useful in supporting work being done by the Branch in the development of a submillimeter heterodyne radiometer, in lidar studies in the five micrometer region and in feasibility studies for near infrared laser communications.

The MICTRA program was developed by the author from a previous MICTRA routine developed in the Instrument Electro-Optics Branch; this earlier routine was based on the Air Force LASER routine. For clarity these three routines will be referred to as the "new" MICTRA routine, the "old" MICTRA routine and the Air Force routine, respectively.

The old routine was constructed primarily by deleting unneeded portions of the Air Force routine. The most important deletions were:

1. Restriction of the atmosphere and aerosol models to a maximum altitude of 15 km (as compared to the original 100 km) and deletion of the Voigt profile calculation;

2. Restriction of the aerosol models, as functions of wavelength, to a small wavelength range centered at 5 micrometers;

3. Deletion of four of the seven atmospheric absorbing species handled by the Air Force routine — only water vapor, carbon dioxide and ozone were retained;

4. Deletion of molecular (Rayleigh) scattering;

5. Deletion of the 3.5-4.2 micrometer water vapor continuum and the 4-5 micrometer nitrogen continuum.

These changes were made because MICTRA was needed primarily to support lidar work in the 5 micrometer region of the spectrum, at altitudes not exceeding about 15 km and involving water vapor, carbon dioxide and ozone. Below 15 kilometers, and for wavelengths of 1 micrometer or more, the lines are mainly pressure broadened (so the Voigt profile is not needed for a reasonably accurate calculation), and Rayleigh scattering is significant only in the visible part of the spectrum. In the new version of MICTRA all but the first and third of the above features have been restored.

It should be noted that, for wavelengths shorter than 5 micrometers, the use of the Lorentz line profile in place of the Voigt profile becomes less accurate. At 5 micrometers, the ratio of the Voigt halfwidth to the Lorentz halfwidth (V/L) ranges from about 1.02 at 15 km altitude to 1.0006 at sea level. At 1 micrometer the V/L ratio ranges from 1.38 at 15 km to 1.02 at sea level, and at 0.5 micrometers it ranges from 2.02 at 15 km to 1.06 at sea level. (All of these ratios hold to a good approximation for any of the three species included in the program: water vapor, carbon dioxide or ozone.) Because of this decrease in accuracy of the Lorentz approximation at shorter wavelengths it is planned to include the Voigt profile in the transmission calculation in the near future.

Neither the old routine nor the Air Force routine calculated transmission spectra; rather, they calculated and printed out tables of absorption and extinction coefficients for all of the atmospheric layers and for the various atmosphere and aerosol models.

The author's main purpose in modifying the old MICTRA routine was to make it capable of calculating transmission spectra over some given range of wavelength or frequency.

The basic strategy used to accomplish this was to use the old MICTRA routines as subroutines called by a newly-created executive routine, which directs the calculation of the spectrum. The executive routine was also given extensive capability to prompt the user to type in information at the keyboard which is used to set up the run.

In addition, the full tables of aerosol coefficients as functions of wavelength were restored, because new Branch uses for the program involved other regions of the spectrum than just the 5 micrometer region. It is planned to replace the aerosol models by newer models which take account of the effect of condensation of water vapor on the haze particles, as a function of relative humidity. The full 100 km atmosphere and seven absorbing species may also be restored. As mentioned above, several effects included in the Air Force routine but deleted in the old MICTRA routine were also restored: Rayleigh scattering and the continuum absorption effects for water vapor at 3.5-4.2 micrometers and for nitrogen at 4-5 micrometers.

An auxiliary plotting routine (PLOTSP) was written to display the spectra computed by the new MICTRA; this routine uses a plot package designed and written by the author specifically for the microcomputer configuration used to run MICTRA (see Reference 3).

Numerous other changes connected with the coding of the routines, were made, but these are too detailed to merit description here. (The various line files used in the calculation of the spectrum are described in the section of this report dealing with the structure of the program.)

WHAT THE PROGRAM DOES

The program calculates atmospheric transmission spectra on a line-by-line basis over a variety of paths, spanning an arbitrarily chosen wavelength or frequency range. Radiance effects are not included in the calculation.

The path may be chosen to be horizontal, vertical or slanted at any zenith angle, and may extend between any two altitudes contained in the atmosphere model, which is a layered, flat model.

The range covered by the spectrum may be chosen freely and may be specified in angstroms, micrometers, hertz or wavenumbers, depending on the preference of the user.

The atmosphere models used are three of the models used in the original Air Force program: the U.S. Standard Atmosphere, 1962 and two mid-latitude models for average summer and average winter conditions. The aerosol models used are those devised by Shettle and Fenn in 1976 (Reference 4). (A more recent version of these models is available, which includes the effects of relative humidity on the size of the aerosol particles; it is planned to include these models in the future.) The atmosphere models, which originally extended up to 100 km altitude, were modified to extend only to 15 km altitude, because higher altitude effects either were not needed for our work or were not significant. Also, of the seven main atmospheric absorbing species included in the original program, only three were retained: water vapor, carbon dioxide and ozone, because the other species were not essential for our work.

The absorption line data used by the program are taken from the 1982 edition of the main tape compiled by the Air Force Geophysics Laboratory, which lists all the known absorption lines for the seven most important atmospheric absorbing species, from the visible to the microwave region (References 5-9). Segments of line data needed are read from the Air Force tape, preprocessed into the form used by the program and stored on floppy disks.

MICTRA computes a transmission spectrum by computing the transmission over a net of points spanning the chosen wavelength or frequency range; at each point the extinction coefficients are computed by adding up several effects:

- The absorption due to all atmospheric lines (of water vapor, carbon dioxide and ozone) in a neighborhood of the current wavelength, chosen so that absorption due to the tails of more distant lines is negligible;
- Molecular continuum absorption: the 3.5-4.2 and the 8-14 micrometer water vapor continua and the nitrogen continuum at 4-5 micrometers;
- Rayleigh (molecular) scattering;
- Aerosol absorption and scattering (at present the 1976 models of Shettle and Fenn are used; their newer models, which include the effects of relative humidity, will be incorporated in the near future).

All these effects are computed for each of the atmospheric layers (there are fifteen 1 km-thick layers) which the chosen path traverses, and are combined to form the total extinction for the given wavelength, over the given path. The transmission values thus calculated over the chosen wavelength or frequency range are then stored in a disk file, for later display by the auxiliary plotting routine PLOTSP.

The plotting routine allows the user the option of displaying the computed transmission values vs wavelength in either angstroms or micrometers or vs frequency in either hertz or wavenumbers, depending on the region of the spectrum and on the preference of the user. Further, whatever scale is chosen may be displayed as increasing toward either the right or the left of the plot.

Two types of spectral plots may be computed by MICTRA and displayed by the plotting routine: a "continuous" spectrum which consists of the transmission over an equally-spaced net of points, joined by line segments to give a continuous appearance, or an "overview" spectrum in which the transmission is computed only at the exact center of each line in the chosen range. In the overview type of spectrum the plot shows the continuous background transmission due to aerosol and molecular continuum effects as a straight line (a good approximation over small wavelength ranges), with the line-center values displayed as vertical lines coming downward from the continuum; the value at the bottom of each line is, of course, the correct total transmission at that wavelength. To avoid the problem of the equally-spaced grid of points in a continuous spectrum skipping over the centers of absorption lines, thus not showing the true depth of the line or even missing it entirely, the line-center wavenumbers are intercalated into the equally spaced grid; thus the bottoms of all lines are correctly shown.

The overview type of spectrum is intended to give a picture of the gross properties of the spectrum over a large range (but small enough to contain at most a few hundred lines, and to ensure that the linear approximation to the background continuum is good), while the continuous type of spectrum is intended to show the detailed structure of the spectrum, including the shapes of the lines, over a smaller range.

The program allows the user to interactively control or select (at the terminal keyboard) many of the parameters and data sets used by the program and many of the functions provided by the program. For example, the user may: select any of the three atmosphere models and any of the aerosol models provided; select the path; change the sea-level values of temperature, pressure or other parameters in the program; specify the wavelength or frequency range of the spectrum; specify the fineness of the grid for a continuous spectrum; etc. The program also allows the user to interactively retain or reject any or all of the extinction effects, in order to see clearly how great each effect is, taken by itself. For example, the user may decide to compute a spectrum showing only the absorption due to lines, with no aerosol or molecular extinction considered; to see just how strong each line is by itself, the user could choose the neighborhood of surrounding lines to be so small that only the absorption due to each line by itself is computed. Further, any or all lines in the chosen range may be independently selected or rejected; this is accomplished by commanding the program to display each line on the screen, and telling the program whether to include it. Provision is made to accept or reject not just individual lines, but also all lines of a given species or all lines in a freely chosen wavelength range; thus a spectrum could be computed which shows only ozone lines, for example, or even a single line.

Any number of spectra may be set up and computed in a single run; this is made possible by the BATCH compiler, which functions for the microcomputer used as a kind of simplified job control language. Each spectrum is stored in its own disk file, and may later be displayed by the plotting routine.

HOW TO USE THE PROGRAM

The program was designed to use the interactive capability of microcomputers; the user communicates with the program through the terminal keyboard and the video screen.

The user only needs to execute the command file RNMICB (which stands for "run the batch version of MICTRA"), which then takes over and directs the user to execute other command files, depending on the type of run. Three types of runs are provided: create a BATCH file which can be executed later to compute a spectrum; compute a spectrum by executing a previously created BATCH file; or both create and execute a BATCH file.

If one of the last two run types is selected, the command file will automatically call MICTRA, after instructing the user to mount the proper floppy disk in the proper device; MICTRA will then take over and lead the user step-by-step through the run setup and execution.

All data sets needed by the program are stored internally, with one exception: the line file must be supplied by the user. The line file is a segment of data condensed from the Air Force Geophysics Laboratory main tape, which is a compilation of about 190,000 atmospheric absorption lines spanning the spectrum from the visible to the microwave region. Because MICTRA is implemented on a microcomputer with no tape drive, it is necessary to take the needed segment of data from the Air Force tape and store it on a floppy disk. The segment of data also must be "condensed" before MICTRA can use it; this is accomplished by using the auxiliary routine LNEDIT. The condensing process consists of taking the lines from the tape, which are coded as character strings in ASCII, stripping away certain parameters not needed by MICTRA (primarily quantum identifications), and converting the remaining numbers from character strings to binary form. The resulting file is about one fourth the size of the unprocessed file; this allows rather large segments of line data to be stored on a floppy disk.

Each computed spectrum is stored in a disk file, specified by the user upon prompting by MICTRA. The spectrum may be displayed by using the auxiliary routine PLOTSP, which instructs the user how to use it and prompts for any needed information or choices.

Because MICTRA leads the user through the run setup process, it is not necessary to list the required input here; the information and choices required by MICTRA may be clearly seen by examining the listing, which is well documented with explanatory comments. It might be noted that the sequence of items needed for setting up a run is not fixed; many choices are provided, and different choices will result in different sequences of input items.

To eliminate the possibility that the user might wait for a long run to finish without any knowledge of what is going on, the program types each spectrum point (wavelength or frequency and transmission) on the screen as it is computed; thus the user can abort a run that looks useless, instead of waiting blindly for it to finish.

STRUCTURE OF THE PROGRAM

The basic organization of the program is shown in Figure 1. MICTRA is the driver and main computing module; it leads the user through the run setup process and directs the calculation of the spectrum.

The logical flow of the spectrum calculation involves several nested loops. The outside loop steps through the sequence of frequency points at which the transmission is to be calculated. For a continuous type spectrum this will be an equally-spaced grid of frequency points, spanning the desired frequency range; for an overview (line-center) type of spectrum the set of frequency points will be the exact centers of all the absorption lines contained in the frequency range.

The next inner loop steps through the atmospheric layers contained within the altitude range spanned by the chosen atmospheric path. The innermost loop steps through the absorption lines contained in the frequency range of the spectrum.

Thus, for each frequency the program calculates the total absorption coefficient for all the lines in each atmospheric layer, as well as the continuum absorption coefficients, the Rayleigh scattering coefficient and the aerosol absorption and scattering coefficients for each layer. These are all added up to give the total extinction coefficient for each layer. Then the transmission for each layer is computed using the usual decreasing exponential law for extinction (e**-depth, where depth is the product of the total extinction coefficient and the path length through the layer), and finally the transmission over the entire path is computed by multiplying together the transmissions for the various layers.

The subroutines communicate with one another through the three COMMON blocks, which hold variables needed by more than one routine, as well as by arguments in the call sequences.

The program needs two external data sets: a file containing the atmosphere and aerosol models and a file containing a segment of data from the Air Force line tape. The atmosphere model file is located on a floppy disk along with the program load module; it is automatically read into the program by subroutine SETUP and stored in one of the COMMON blocks. The line file must be supplied by the user; it is created by taking the needed segment of data from the Air Force tape and using the auxiliary routine LNEDIT to condense it into the form used by MICTRA.

After the MICTRA routine has obtained the required information from the user (via the keyboard), it forms the run line file, which consists of the lines needed for the current run, by taking from the line file only those lines selected by the user in the run setup process. The run line file may contain all of the lines in the line file, some of them or none of them; the lines to be included may be selected individually, by species or by sub-ranges of the spectrum frequency range.

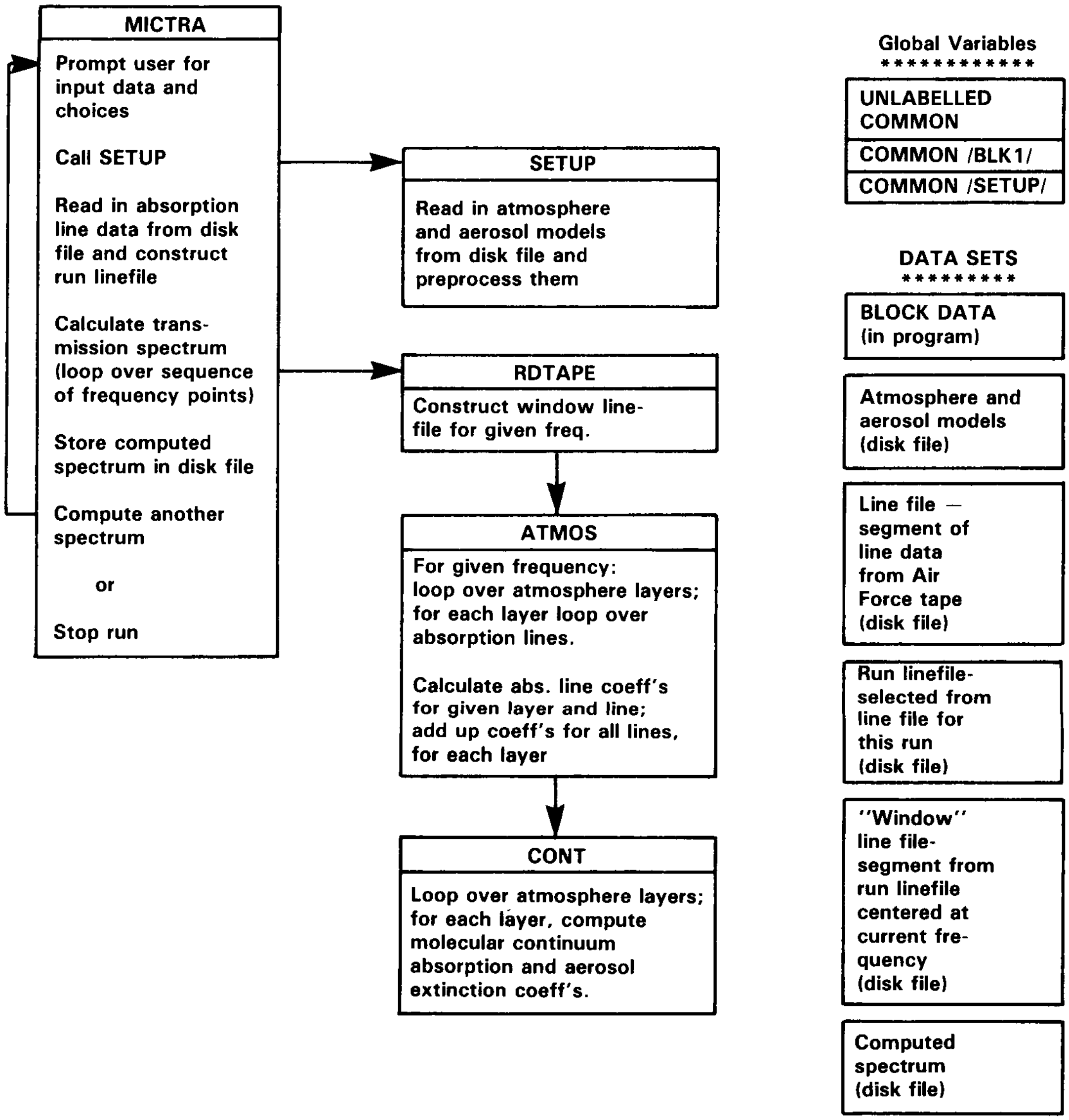

Figure 1. Basic Flow of MICTRA Program

During the calculation of the spectrum subroutine RDTAPE forms a "window" line file for each frequency at which the transmission is to be computed; this consists of all the lines in the run line file which are contained in a window of specified frequency width centered at the current frequency (the width is chosen by the user during the run setup process). The purpose of the window line file is to include, for the calculation of the absorption coefficient at the current frequency, all those lines within some neighborhood of the current frequency which will affect the absorption appreciably. If the user wants to calculate the absorption due to a single line as if it were isolated (even though it may actually have close neighbors which affect it), the window width may be chosen small enough to exclude all nearby lines.

In the implementation of MICTRA used by the author on the DEC 11/23 microcomputer, the line file is located on a floppy disk, while the run file and the window file are located in a part of the computer memory which is treated as a disk (logically) by the system software; thus, the operation of the program is much faster than it would be if these files were located on floppy disks. (In the event that the run file or the window file is too large to fit into that part of the memory, it is automatically put in a file located on a DEC RL02 disk drive, which holds 10.4 megabytes and is much faster than a floppy disk drive.)

MEMORY REQUIREMENTS AND OPERATING SPEED

The entire program (i.e., the load module) requires about 42 kilobytes of memory; this includes all the atmosphere and aerosol models, which are read into the program and stored internally. The line files should be small enough to fit onto a floppy disk. In the DEC computer being used by the author, which handles 8-inch double density floppy disks which hold 512 kilobytes, this allows the line files to hold up to about 20,778 lines, in the condensed form used by MICTRA; this should be large enough for most purposes. The auxiliary programs LNEDIT and PLOTSP, which respectively construct condensed line files and display the spectra, require about 13 and 32 kilobytes of memory, respectively.

The time required to compute a spectrum varies greatly, depending on the type of spectrum (continuous or only line centers) and on other factors such as the wavelength or frequency range covered, the number of absorption lines contained in the range, the width of the window of lines which significantly affect the absorption at the current frequency, the fineness of the frequency grid, etc. In general, the calculation of a line-center plot is much faster than the calculation of a continuous plot; the times may range from a few seconds for a line-center plot over a small interval to several hours for a continuous plot over a large interval with many lines and a wide window.

SOME EXAMPLES OF COMPUTED TRANSMISSION SPECTRA

In this section, several examples of transmission spectra are shown exactly as they appear on the terminal screen when displayed by the auxiliary plotting routine PLOTSP; the plots were copied directly from the screen by a TEKTRONIX hardcopy device.

Figure 2 shows a line-center spectrum in the near infrared, from 8250 to 8300 angstroms. The spectrum was computed for a slanted path extending from the ground to the top of the atmosphere model (at 15 km altitude); the zenith angle was taken to be 51 degrees. The midlatitude summer atmosphere model was used, along with a very hazy aerosol model; many of the lines show complete absorption (all the lines in this region happen to be water vapor lines). The plot gives a very clear overall picture of the absorption by water vapor and aerosols in this region. Figure 3 shows a continuous plot of a part of the region, from 8270 to 8280 angstroms; from this plot it is clear that the lack of line shapes in Figure 2 does not detract much from the qualitative picture of the spectrum, at least in this region. Both of these plots were computed with a window halfwidth of 20 wavenumbers (i.e., all lines within 20 wavenumbers to each side of the current frequency were taken into account in computing the absorption), which gives a very accurate spectrum.

Figure 4 shows a line-center spectrum in the submillimeter region, from about 180 to 200 micrometers. This spectrum was computed for a vertical path extending from 10 to 15 km altitude (if computed from the ground up, the water vapor absorption is so strong and the lines so broadened that the spectrum is virtually flat — almost everything gets absorbed). The midlatitude summer atmosphere model was used; the aerosol model does not extend to the submillimeter region, because the effect is negligible. To allow a very quick look — i.e., a fast (about 10 seconds) calculation of the spectrum — the lines were effectively considered separately by taking the window halfwidth to be only 0.001 wavenumbers; thus each line is shown as it would appear if no other lines were present. In this region both water vapor and ozone lines appear, but only water vapor lines were included in the spectrum. For comparison, the spectrum in Figure 5 was computed. This shows the same spectrum as Figure 4 does, but in a continuous form, to show its actual appearance. Also, the window halfwidth for this spectrum was taken to be 10 wavenumbers, so that it is more accurate. Because of this, and because a fine grid of points was used to span the spectral range, this spectrum took about 2 hours to compute. (It should be recalled that the exact line centers of all lines are intercalated into the grid, so that the line depths are accurately shown — there is no skipping over line bottoms due to grid points missing the line.)

It is interesting to note, from Figure 5, that the water vapor line at about 187 micrometers is extremely strong; the plot clearly shows other, weaker lines extending down from its very broad wing on the high wavelength side; on its low wavelength side there is another very strong line, which is mostly off the plot. Inspection of the line file shows that the strong line at 187 micrometers is in fact unusually strong; its intrinsic line strength is about 45 times greater than the strength of the next strongest line near it (except for the line only partly shown, which is even stronger), and is from two to six orders of magnitude greater than the strength of most of its neighboring lines.

Lastly, Figure 6 shows a spectrum in which only the aerosol effects were computed, over the entire range of the aerosol model from 0.2 to 40 micrometers. This continuous plot shows very clearly how the aerosol absorption and scattering effects decrease markedly outside the visible region of the spectrum. The spectrum was computed over a vertical path extending from the ground to the top of the atmosphere, with a midlatitude summer atmosphere model and rather clear visibility (not much haze). The wavelength grid used was very fine, so that the spectrum shows all details clearly.

TESTING OF PROGRAM

The MICTRA routine was initially tested by comparing its computed transmission values at selected wavelengths with hand calculated points; in all cases the results agreed very closely.

The routine was then tested against an independent program developed by another group at the Goddard Space Flight Center (Reference 10). This program takes account of all lines within 20 wavenumbers of the frequency at which the transmission is being computed, and it uses a Voigt line profile. Transmission spectra were calculated, using both programs, in the 5 micrometer region of the spectrum, from 2045 to 2057 wavenumbers, using the 1982 version of the Air Force absorption line tape. Both programs computed the spectrum with good resolution: every 0.01 wavenumber for the other program and every 0.04 wavenumber for MICTRA. The two calculated spectra were virtually identical; any slight differences in transmission values, at most a few percent, could be attributed to the fact that the two programs were completely independent and thus could not give identical results.

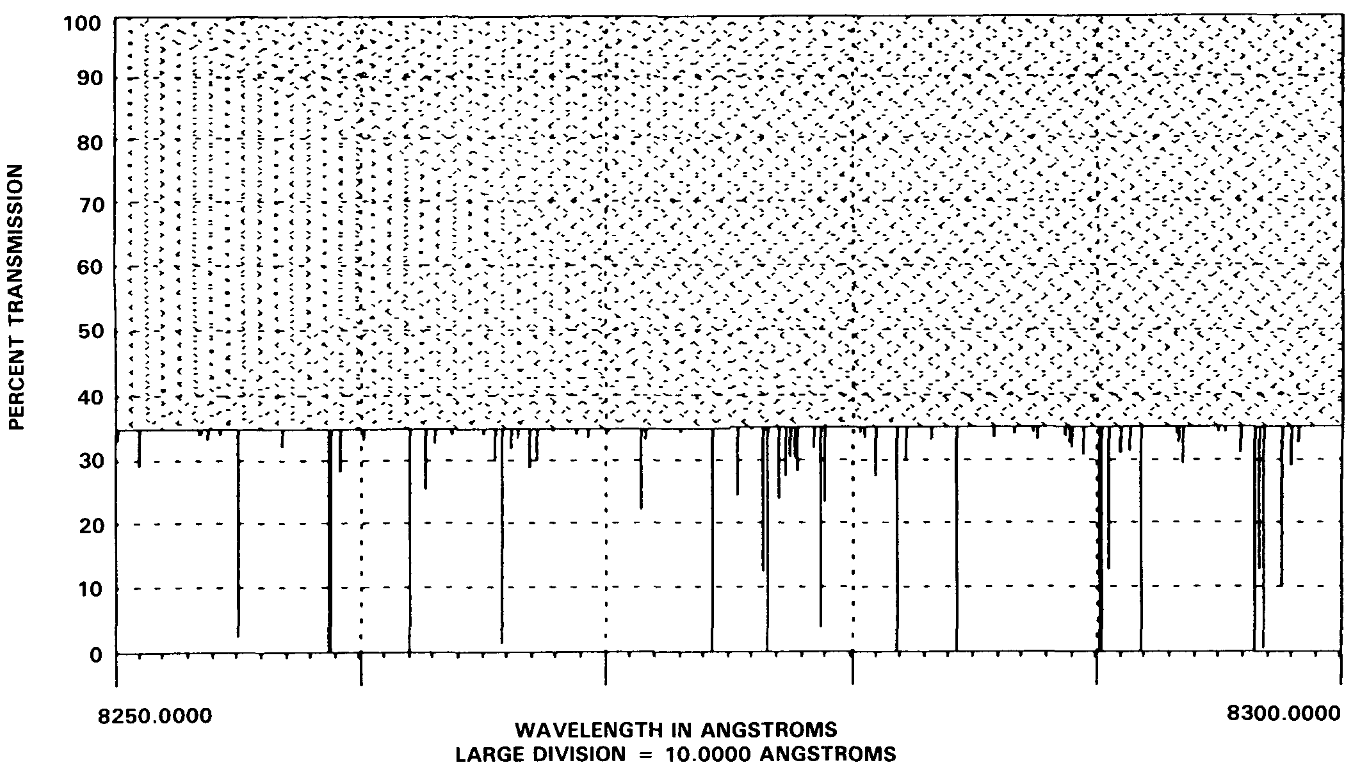

Figure 2. Line Center Spectrum in Near Infrared.

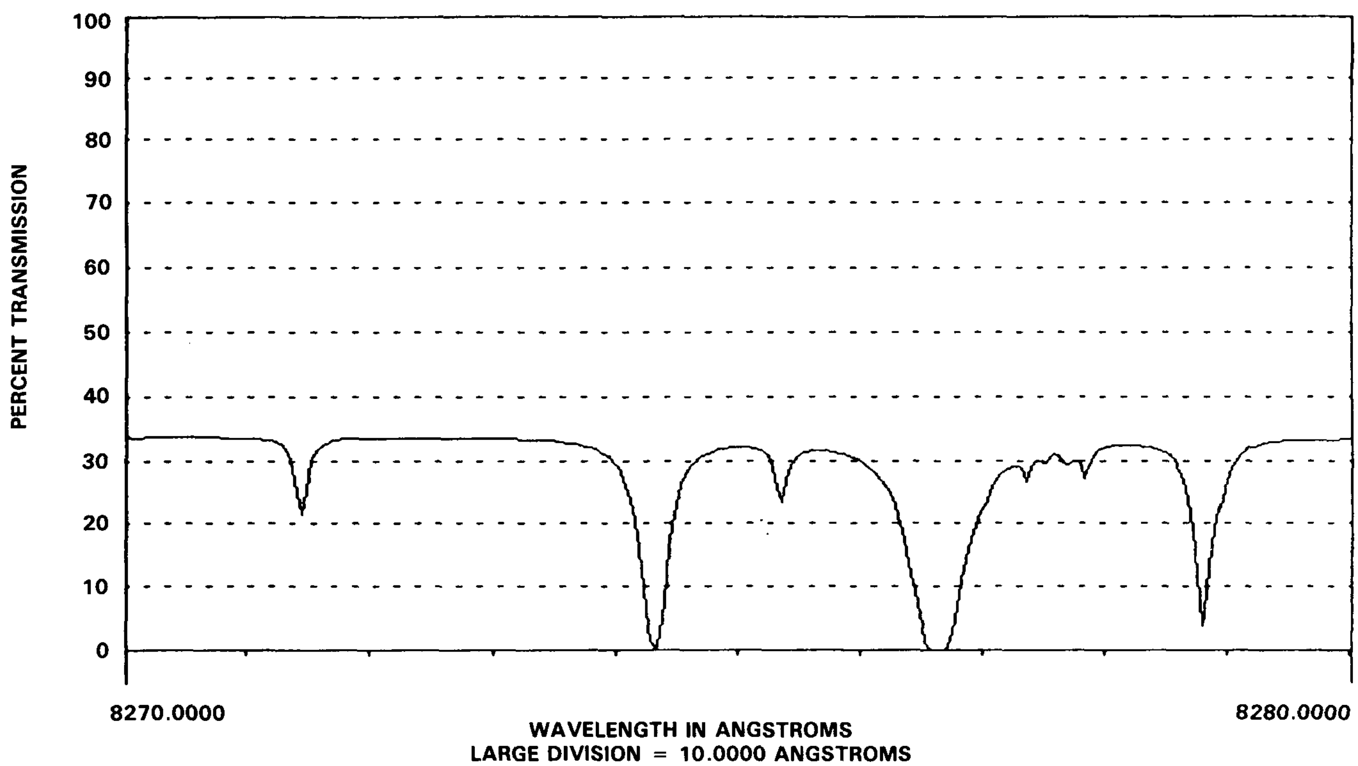

Figure 3. Continuous Plot of Sub-region of Figure 2.

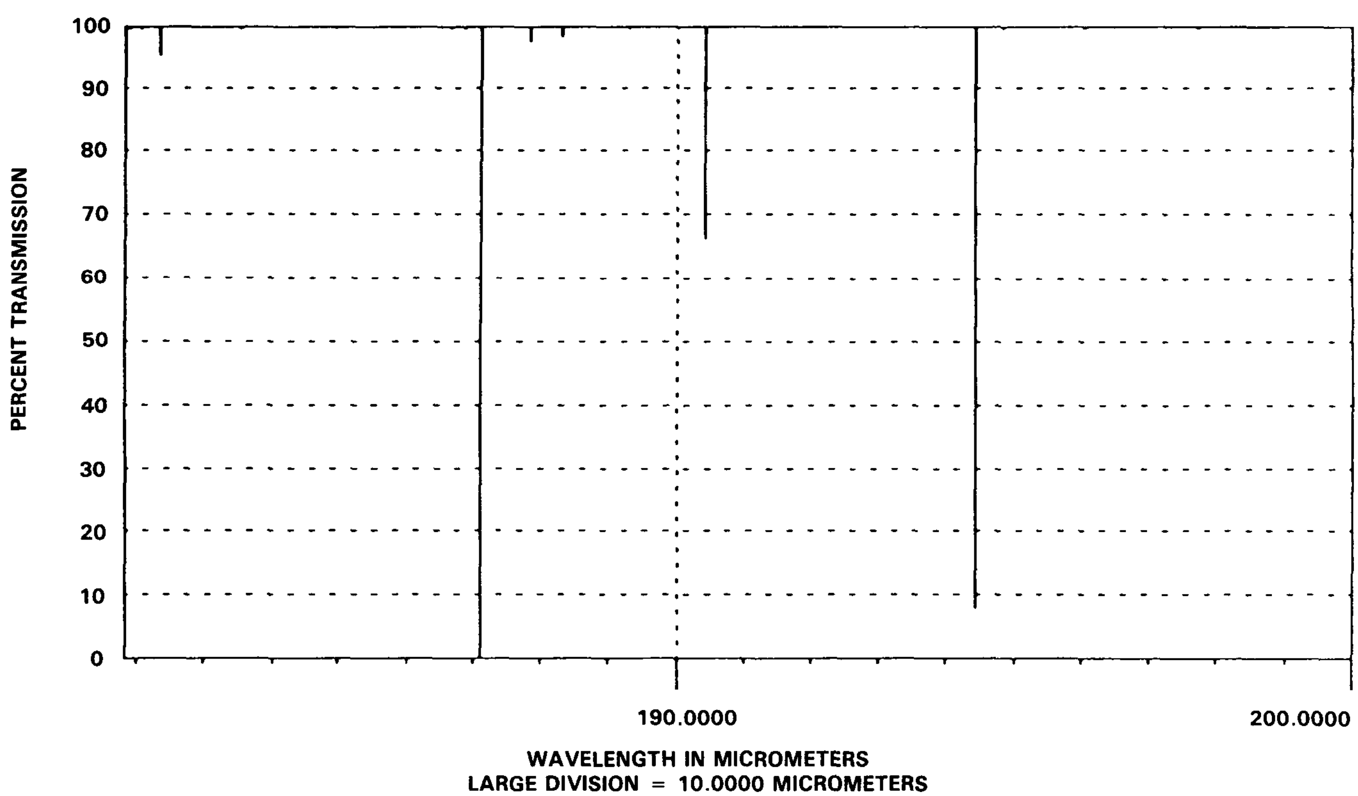

Figure 4. Line Center Spectrum in Submillimeter Region; Water Vapor Lines Only.

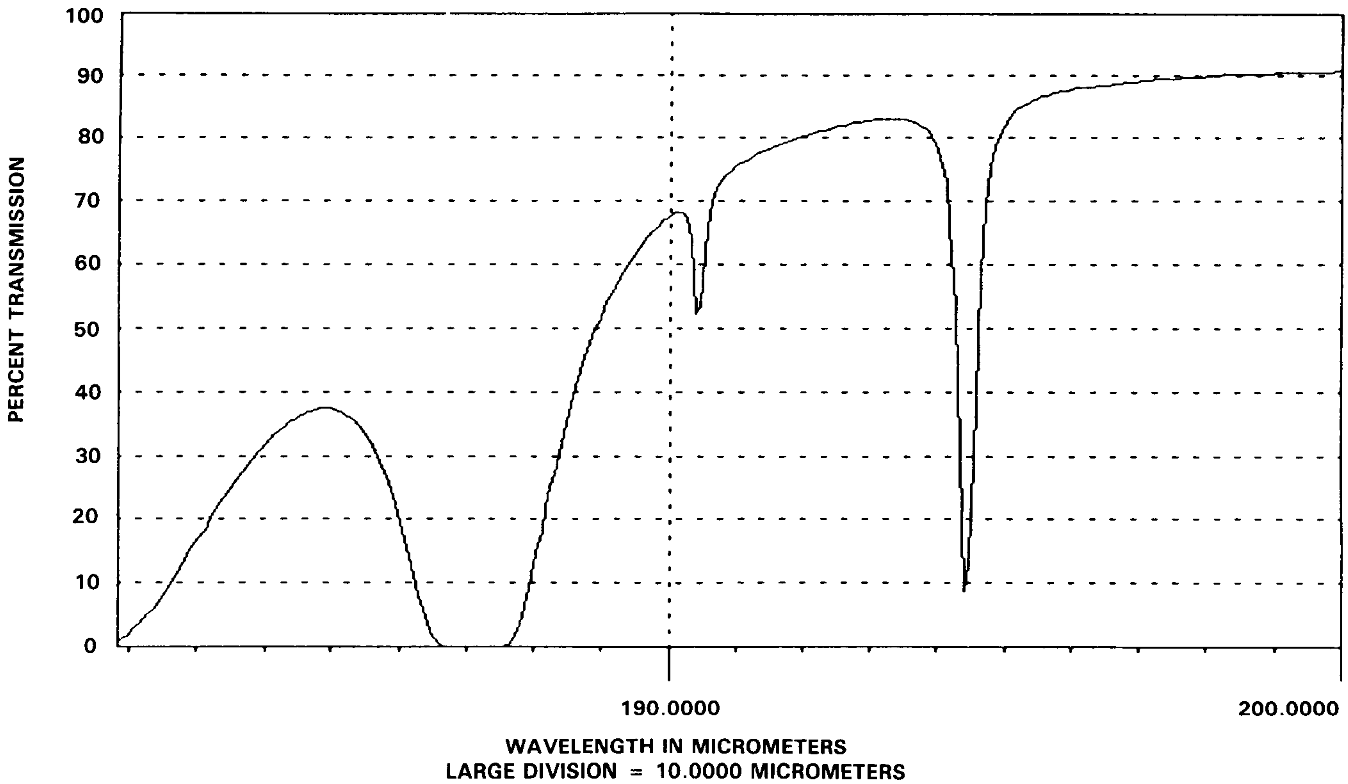

Figure 5. Continuous Plot of the Spectrum in Figure 4.

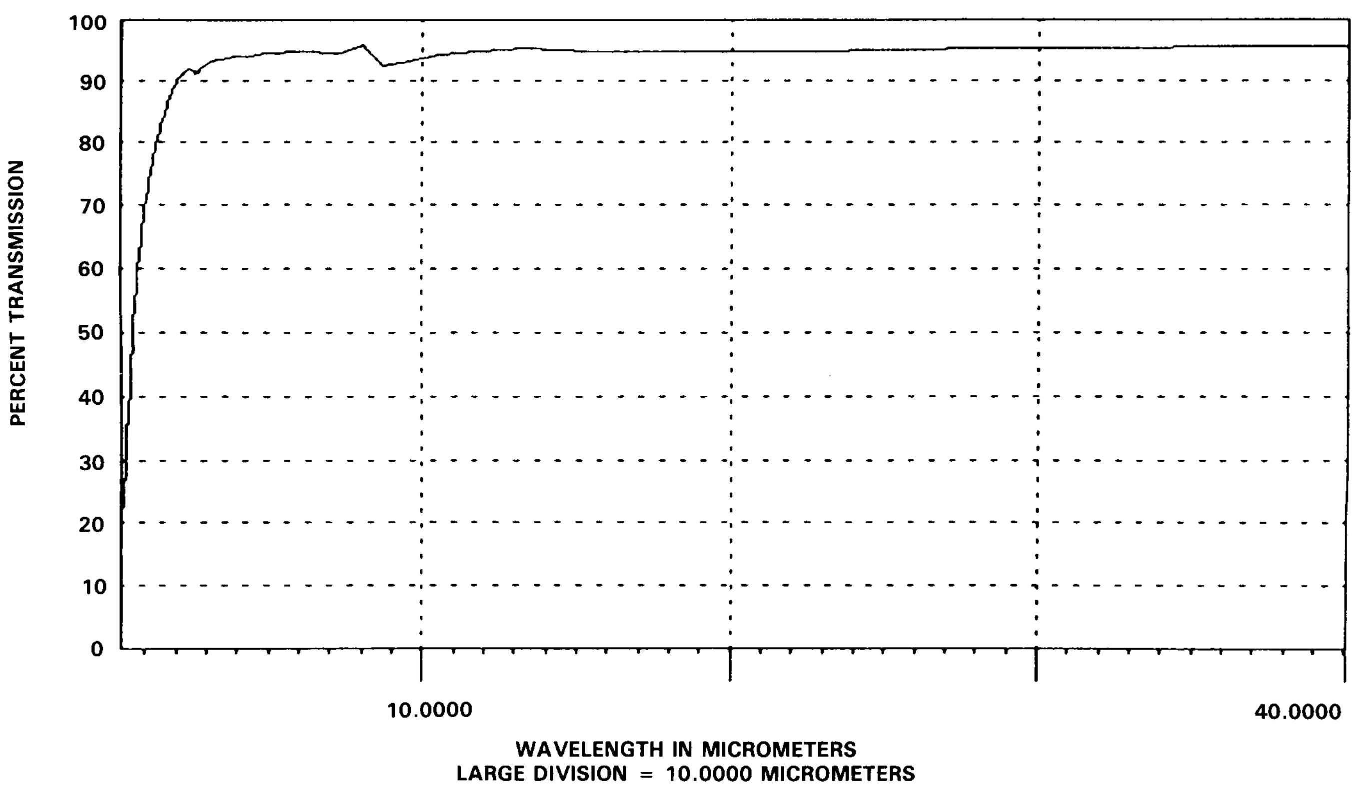

Figure 6. Aerosol Absorption and Scattering Over a Vertical Path Through the Atmosphere, From the Visible to 40 Micrometers.

PROGRAM LISTINGS

```
      PROGRAM MICTRA

C
C     This version of the MICTRA routine is designed to run in the BATCH mode;
C     the purpose is to be able to calculate many transmission spectra with no
C     operator attention.
C
C     Three modes of operation are provided:
C
C             1. Setup mode -    the routine is run interactively to create
C                                a control file for the BATCH program. The
C                                file includes all input to MICTRA which is
C                                normally input interactively at the keyboard,
C                                using ACCEPT statements.
C
C             2. Batch mode -    the routine is run under the control of the
C                                BATCH program to calculate the spectra,
C                                using the control file set up in a previous
C                                run in the setup mode.
C
C             3. Combined mode - the routine creates a BATCH file and executes
C                                it to compute any number of spectra, all in
C                                one run.
C
C     This routine is mainly a driver for the subroutines that actually
C     do the calculations of the transmission spectrum. It contains the
C     following options:
C
C          *  The following data, options and parameters may be specified:
C
C                          *  Atmosphere and aerosol models;
C                          •  Sea-level temperature and pressure;
C                          *  Sea-level concentrations of water vapor, CO2 and
C                             ozone;
C                          *  Absorption line file;
C                          *  Spectral range of the calculated transmission spectrum;
C                          *  "Overview" spectrum (large spectral range) showing
C                             transmission at line centers only, with aerosol and
C                             molecular continuum effects also shown, or "continuous"
C                             spectrum (small spectral range) showing line shapes;
C                          *  Wavenumber increment between calculated points
C                             (may be as small as 0.01 cm-1, or even less);
C                          *  Width of the "window" over which absorption lines are
C                             included in the calculation of the absorption at the
C                             window center.
C
C          *  The spectra are not plotted as they are calculated, because BATCH
C             takes so much space. (The spectra can be later plotted from the
C             disk spectrum files, using PLOTSP.)
C
C          *  The computed spectra are saved on disk files.
C
C          *  A separate routine (PLOTSP) is available to plot the spectra.
C*************************************************************************************
```

```
C                                SPECIFICATION STATEMENTS
C                              ****************************
C       BYTE          reply, linfil, spcfil, edmode, disp, ff,
     *                htunit, pthtyp, batmod, batmd,
     *                grdmod, eqspce, centrs, replys, replyp
        INTEGER       speces, pcount
        REAL*4        lownu, lastwn, molnam
        DIMENSION     linfil(15), spcfil(15), keepsw(9), molnam(3)
C
        COMMON        z(16),p(3,16),
     *                t(3,16),w(3,3,16),cay(3,16),wg(3,3),
     •                ta2(16),fa(6,61),asc(6,61),aab(6,61),
     *                has(16)
C
        COMMON/BLK1/    dnu(16),chi(16),hz1(16),hz2(16),itp,jt,ksam
C
        COMMON/RSETUP/ bound,lownu,highnu,delnu,v,
     *                  iatmod,iarmdg,iarmdt,iarmds,
     *                  numpts,nlines,keepsw,iblayr,itlayr,
     •                  irsize, hzint(16),iaermd
C
        DATA          molnam(1)/'H2O '/, molnam(2)/'CO2 '/, molnam(3)/'O3 '/
        DATA          ff/"014/, linfil/15*"000/, spcfil/15*"000/, batmd/'C'/
        DATA          eqspce/'E'/, centrs/'C'/
        DATA          C/2.997925E10/
C*****************************************************************************
C                                           RUN SETUP
C                                        ***************
C
C     SET UP MODE OF OPERATION: COMMAND FILE SETUP OR CALCULATION UNDER BATCH
C     ///////////////////////////////////////////////////////////////////////
C
        type 4
      4 format(//t2,'Is this run to set up a BATCH command file'/
     •      t2,'or to calculate spectra under BATCH control? (F or C) ',$)
        accept 5, batmod
      5 format(a1)
C
        IF (batmod .eq. 'F') go to 6
        IF (batmod .eq. 'C') go to 9
C
C        Open BATCH command file for input
C        ---------------------------------
C
      6 OPEN (unit=9, name='SD0:BATFIL.BAT', type='NEW', disp='KEEP',
     *         form='FORMATTED', initialsize=200)
C
        write (9,8)
      8 format('$JOB'/'$RUN SD2:MICTRA.SAV'/'$DATA')
        write (9,5) batmd

C      OPEN FILE TO HOLD CALCULATED TRANSMISSION SPECTRUM
C      //////////////////////////////////////////////////
C
      9 type 2
      2 format(//t2,'Type name of file (including device) to store'/
     *           t2,'the computed transmission spectrum:  ',$)
        accept 3, spcfil
      3 format(15a1)
        IF (batmod .eq.  'F')  write(9,3) spcfil
```

```
       IF (BATMOD .EQ.  'C')
     * OPEN (unit=8, name=spcfil, type='NEW', disp='KEEP',
     *       access='SEQUENTIAL', form='UNFORMATTED', recordsize=2,
     *       initialsize=300)
C
C
C    CHOOSE PATH OVER WHICH TRANSMISSION IS TO BE COMPUTED
C    /////////////////////////////////////////////////////////
C
       type 210
 210   format(//t2,'Set up path over which transmission is to',
     *          t44,'be calculated:'//)
C
C       Heights and path length in feet or meters?
C       ---------------------------------------------
C
 214   type 211
 211   format(t10,'Specify heights above sea level and path',
     *          t51,'length'/
     *          t10,' in feet or meters? (F or M): ',$)
       accept 212, htunit
 212   format(a1)
       IF (batmod .eq. 'F') write(9,212) htunit
C
       IF (htunit .eq. 'F') htconv = 3.048E-04    ! convert ft to km
       IF (htunit .eq. 'M') htconv = 1.E-03       ! convert m  to km
C
       IF (htunit .ne. 'F' .and. htunit .ne. 'M') go to 214 ! error corr.
C
C       Is the path horizontal, vertical or slanted?
C       ---------------------------------------------
C
 215   type 213
 213   format(/t10,'Is the path horizontal, vertical or slanted?',
     *          t54,' (H,V or S): ',$)
       accept 212, pthtyp
       IF (batmod .eq. 'F') write(9,212) pthtyp
C
          IF (pthtyp .eq. 'H') go to 220
          IF (pthtyp .eq. 'V' .OR. pthtyp .eq. 'S') go to 240
          go to 215                                          ! error corr.
C      Setup for horizontal path
C      -------------------------
C
  220     type 221
  221     format(/t15,'The path is horizontal;'/
     *              t15,'we need the following information:'//
     *              t20,'Height of path (ft or m, <15 km): ',$)
          accept 222, pathht
  222     format(f10.3)
          IF (batmod .eq. 'F') write(9,222) pathht
          phtkm = htconv * pathht                     ! path height in km
C
          type 223
  223     format(/t20,'Length of path (ft or m): ',$)
          accept 222, pathln
          IF (batmod .eq. 'F') write(9,222) pathln
          plnkm = htconv * pathln                     ! path length in km
C
          call laynum (phtkm, layer, delht)           ! get atm. layer number
          iblayr = layer
          itlayr = layer                              !, (top and bottom same)
          go to 12
```

```
C       Setup for vertical or slant path
C       ---------------------------------
C
    240   type 241
    241   format(/t15,'The path is vertical or slanted;'/
     *            t15,'we need the following information:'//
     *            t20,'Height of lower end of path (ft or m, <15 km): ',$)
          accept 222, phtbot
          IF (batmod .eq. 'F') write(9,222) phtbot
C
          type 242
    242   format(/t20,'Height of upper end of path (ft or m, <15 km): ',$)
          accept 222, phttop
          IF (batmod .eq. 'F') write(9,222) phttop
C
          secant = 1.0                           ! set sec to 1 for vert. path
          IF (pthtyp .eq. 'V') go to 245
C
          type 243                               ! define sec for slant path
    243   format(/t20,'Zenith angle of path, in degrees: ',$)
          accept 244, zenang
    244   format(f7.3)
          IF (batmod .eq. 'F') write(9,244) zenang
          zangrd = (3.141593/180.) * zenang      ! zenith angle in radians
          secant = 1./cos(zangrd)
C
    245   phbkm = htconv * phtbot                ! ht. of bottom of path (km)
          phtkm = htconv * phttop                ! ht. of top    of path (km)
          call laynum (phbkm, layer, delht)      ! bottom layer no. and segment
          iblayr = layer
          botseg = 1. - delht
          IF (delht .eq. 0.) botseg = 0.
C
          call laynum (phtkm, layer, delht)      ! top layer no. and segment
          itlayr = layer
          topseg = delht
          IF (delht .eq. 0.) topseg = 1.
C
C   READ IN ALL ATMOSPHERIC MODELS, INCLUDING AEROSOL MODELS, FROM DATA SET 2
C   ////////////////////////////////////////////////////////////////////////
C
     12   call setup                           ! set up all atmosphere models
C
C   CHOOSE ATMOSPHERE AND AEROSOL MODELS
C    //////////////////////////////////////
C
C      Atmosphere Model
C      ----------------
C
          type 20
    20    format('0',t2,'which of the 3 atmosphere models do you want?'/
     *                  t10,'type: 1 for midlatitude summer model:'/
     *                  t17, '2 for midlatitude winter model;'/
     *                  t17, '3 for U.S. Standard model:  ',$)
          accept 21, iatmod
     21   format(i1)
          IF (batmod .eq. 'F') write(9,21) iatmod
```

```
C       Aerosol Model
C       --------------
C
        TYPE 4000
 4000   FORMAT(//T2,'The sea-level meteorological range for the'/
     *           t2,'aerosol models may be chosen from 5 km'/
     •           t2,'(very hazy) to 50 km (very clear);'/
     •           t2,'type the desired range in km:  ',$)
        ACCEPT 4001, SLVIS
 4001   FORMAT(F6.2)
        IF (BATMOD .EQ. 'F') WRITE(9,4001) SLVIS
C
        DO 4010 I=7,16                              ! construct interpolated
        FINT = (50.-SLVIS)/45.                      ! altitude scaling fac-
        HZINT(I) = HZ1(I) + FINT*(HZ2(I)-HZ1(I))    ! tors for aer. models
 4010   CONTINUE                                    ! (Only below 9 km)
C
C                       Note: HZ1 = clear (50 km) scaling factors
C                             HZ2 = hazy  ( 5 km) scaling factors
C                             (stored in COMMON/BLK1/)
        type 22
   22   format(//t2,'Does any part of the path lie between'/
     *           t2,'0 km (sea-level) and 2 km altitude? (Y or N): ',$)
        accept 23, reply
   23   format(a1)
        IF (batmod .eq. 'F') write(9,23) reply
        IF (reply .eq. 'Y') go to 24
        go to 27
C
   24   type 25
   25   format('0',t2,'which boundary-layer (0-2 km)',
     •             t32,'aerosol model do you want?'/
     *             t10,'type: 1 for rural model;'/
     •             t16,      '2 for tropospheric model;'/
     *             t16,      '3 for maritime model;'/
     *             t16,      '4 for urban model:  ',$)
        accept 26, iarmdg
   26   format(i1)
        IF (batmod .eq. 'F')  write(9,26) iarmdg
C
   27   type 28
   28   format(//t2,'Does any part of the path lie between'/
     *          t2,'2 and 9 km altitude? (Y or N): ',$)
        accept 23, reply
        IF (batmod .eq. 'F') write(9,23) reply
        IF (reply .eq. 'Y') go to 29
        go to 31
C
   29   IARMDT = 2
C
   31   type 32
   32   format(//t2,'Does any part of the path lie above 9 km altitude?',
     •          t53,'(Y or N): ',$)
        accept 23, reply
        IF (batmod .eq. 'F') write(9,23) reply
        IF (reply .eq. 'Y') go to 33
        go to 39
```

```
   33  type 34
   34  format(//t2,'Which stratospheric aerosol model do you want?'/
     *          t10,'type: 5 for clear (background stratospheric)'/
     •          t16,      '6 for hazy (aged volcanic):             ',$)
       accept 26, iarmds
       IF (batmod .eq. 'F')  write(9,26) iarmds
C
       IF (IARMDS .EQ. 5) GO TO 4020
       IF (IARMDS .EQ. 6) GO TO 4030
C
 4020  DO 4025  I=1,6
       HZINT(I) = HZ1(I)
 4025  CONTINUE
       GO TO 39
 4030  DO 4035  I=1,6
       HZINT(I) = HZ2(I)
 4035  CONTINUE
C
C
C    MODIFY ATMOSPHERIC MODEL (AT SEA LEVEL ONLY)
C    ///////////////////////////////////////////
C
   39  type 40
   40  format(//t2,'The following are sea-level values of some parameters'/
     *   t2,'in the atmosphere model chosen; any of them may be changed:'//)
C
C      Modify Sea-level Temperature
C      ----------------------------
       tfahr = (9.0/5.0) * t(iatmod,16) - 459.67    ! convert t to Fahr.
C
       type 41, tfahr
   41  format(t2,'Temperature (deg. F) - ',t30,f7.2,
     *         t45,'change? (Y or N): ',$)
       accept 42, reply
   42  format(a1)
       IF (batmod .eq. 'F') write(9,42) reply
C
       IF (reply .eq. 'N') go to 45
C
       type 43
   43  format(t15,'New value:',t29,' ',$)
       accept 44, tfahr
   44  format(f7.2)
       IF (batmod .eq. 'F') write(9,44) tfahr
C
       t(iatmod,16) = (5./9.)*(tfahr+459.67)    ! convert back to deg. K
C
C     Modify Sea-level Pressure
C     -------------------------
C
   45  type 46, p(iatmod,16)
   46  format(/t2,'Pressure (millibars) - ',t30,f8.3,
     •         t45,'change? (Y or N): ',$)
C      IF (batmod .eq. 'C')  write (8,47)  p(iatmod,16)
       accept 42, reply
       IF (batmod .eq. 'F') write(9,42) reply
C
       IF (reply .eq. 'N') go to 48
C
       type 43
       accept 47, p(iatmod,16)
   47  format(f8.3)
       IF (batmod .eq. 'F') write(9,47) p(iatmod,16)
```

```
C      Modify Sea-level Concentration of Water Vapor
C      --------------------------------------------
C
   48  type 49, wg(iatmod,1)
   49  format(/t2,'Water vapor conc. (molecules/cm**2/km) - ',
     *          t46,e10.2,t61,'change? (Y or N): ',$)
C        IF (batmod .eq. 'C') write (8,51) wg(iatmod,1)
       accept 42, reply
       IF (batmod .eq. 'F') write(9,42) reply
C
       IF (reply .eq. 'N') go to 52
C
       type 50
   50  format(t33,'New value:',t45,' ',$)
       accept 51, wg(iatmod,1)
   51  format(e10.2)
       IF (batmod .eq. 'F') write(9,51) wg(iatmod,1)
C
C      Modify Sea-level Concentration of Carbon Dioxide
C      -----------------------------------------------
C
   52  type 53, wg(iatmod,2)
   53  format(/t2,'Carbon dioxide conc. (molecules/cm**2/km) - ',
     *          t46,e10.2,t61,'change? (Y or N): ',$)
       accept 42, reply
       IF (batmod .eq. 'F') write(9,42) reply
C
       IF (reply .eq. 'N') go to 54
C
       type 50
       accept 51, wg(iatmod,2)
       IF (batmod .eq. 'F') write(9,51) wg(iatmod,2)
C
C      Modify Sea-level Concentration of Ozone
C      ---------------------------------------
C
   54  type 55, wg(iatmod,3)
   55  format(/t2,'Ozone conc. (molecules/cm**2/km) - ',
     *          t46,e10.2,t61,'change? (Y or N): ',$)
       accept 42, reply
       IF (batmod .eq. 'F') write(9,42) reply
C
       IF (reply .eq. 'N') go to 69
C
       type 50
       accept 51, wg(iatmod,3)
       IF (batmod .eq. 'F') write(9,51) wg(iatmod,3)

C     SET UP MASTER LINE FILE FOR THIS RUN (UNformatted, sequential access)
C     //////////////////////////////////////////////////////////////////////
C
C       This master line file should contain ONLY H2O, CO2 and O3 lines (all
C     isotopes) in the region of the spectrum being investigated.
C
C     Each record in the file should contain the following quantities, in
C     UNformatted (i.e., binary) form:
C
C                Wavenumber              real*4
C                Line strength             "
C                Line half-width           "
C                Lower state energy        "
C                Isotope identification  integer (coded)
C                Molecular species       integer (1, 2 or 3)
```

```
C     The exact definitions and units of these quantities are given in the
C     documentation for the Air Force Line Catalogue.
C
C     A line file in this format is called a "condensed" line file, to
C     distinguish it from the original segment of line data taken from the
C     AFGL tapes, which list the lines in ASCII (i.e., formatted) form.
C     A linefile editing routine is available to "condense" desired
C     segments of AFGL line data.
C
   69   type 70
   70   format('0',t2,'Type the name of the line file to be used: ',$)
        accept 71, linfil
   71   format(15A1)
        IF (batmod .eq. 'F') write(9,71) linfil
C
        OPEN (unit=4, name=linfil, type='OLD', form='UNFORMATTED',
     *        disp='KEEP')
C
C     COUNT NUMBER OF LINES IN MASTER LINE FILE,
C     NOTE FIRST AND LAST WAVENUMBERS AND
C     DISPLAY THIS INFORMATION ON SCREEN
C     (in both frequency and wavelength units)
C     ////////////////////////////////////////////
C
        type 710
  710   format(//t2,'Is this LINFIL the same as the one used for the'/
     •         t2,'immediately preceding spectrum? (Y or N; type N'/
     *         t2,'if this is the first spectrum for this run):    ',$)
        accept 711, reply
  711   format(a1)
        IF (batmod .eq. 'F') write(9,711) reply
C
        IF (reply .eq. 'Y') go to 726
        read(4) wavnum, strnth, hwidth, energy, isotop, speces
        frstwn = wavnum
        rewind 4
C
        DO 722  linct=1,30000
  722   read(4, end=725)  wavnum,strnth,hwidth,energy,isotop,speces
C
  725   lastwn = wavnum
        mlines = linct-1
        rewind 4
C
        end1hz = c*lastwn
        end2hz = c*frstwn
        end1a  = 1.e8/lastwn
        end2a  = 1.e8/frstwn
        end1mc = 1.e4/lastwn
        end2mc = 1.e4/frstwn
C
  726   type 727, mlines, lastwn,frstwn, end1hz,end2hz,
     *                  end1a, end2a, end1mc,end2mc
  727   format(//t2,'This line file contains',t34,I5,t40,'lines;'/
     *         t2,'it covers the frequency-wavelength range:'//
     *         t30,'FROM',t60,'TO'//
     *         t5,'wavenumbers:',t25,f14.4,t53,f14.4//
     *         t5,'freq. in hertz:',t24,e16.9,t52,e16.9//
     *         t5,'angstroms:',t24,f15.4,t52,f15.4//
     •         t5,'micrometers:',t26,f12.4,t54,f12.4///)
```

```
C     SPECIFY WAVENUMBER RANGE OVER WHICH TRANSMISSION IS TO BE CALCULATED
C     ///////////////////////////////////////////////////////////////////////
C
          type 730
  730     format(//t2,'Type the endpoints, in either order, of the'/
     *             t2,'spectral range over which the transmission is'/
     *             t2,'to be computed; the endpoints may be expressed'/
     *             t2,'in wavenumbers, hertz, angstroms or micrometers;'/
     *             t2,'use the E format (real).'//
     *             t2,'Type: 1 for endpoints in wavenumbers'/
     *             t2,'      2  "      "       "  hertz'/
     *             t2,'      3  "      "       "  angstroms'/
     *             t2,'      4  "      "       "  micrometers:  ',$)
C
          accept 731, iunit
  731     format(i1)
          IF (batmod .eq. 'F') write(9,731) iunit
C
          type 732
  732     format(//t10,'One endpoint:  ',$)
          accept 733, highnu
  733     FORMAT(E18.6)
          IF (batmod .eq. 'F') write(9,733) highnu
          IF (iunit .eq. 1) highnu = highnu     ! convert to wavenumbers
          IF (iunit .eq. 2) highnu = highnu/c
          IF (iunit .eq. 3) highnu = 1.E8/highnu
          IF (iunit .eq. 4) highnu = 1.E4/highnu
C
          type 734
  734     format(/t10,'Other endpoint:  ',$)
          accept 733, lownu
          IF (batmod .eq. 'F')  write(9,733) lownu
C
          IF (iunit .eq. 1) lownu = lownu      ! convert to wavenumbers
          IF (iunit .eq. 2) lownu = lownu/c
          IF (iunit .eq. 3) lownu = 1.E8/lownu
          IF (iunit .eq. 4) lownu = 1.E4/lownu
C
          IF (lownu .lt. highnu) go to 1999    ! interchange endpoints
          hghnu = highnu                       ! (if necessary)
          highnu = lownu
          lownu = hghnu
C
C
C     CHOOSE EQUALLY SPACED OR LINE CENTER GRID OF WAVELENGTH PTS
C    //////////////////////////////////////////////////////////////
C
  1999    TYPE 2000
  2000    FORMAT(/T2,'Do you want to compute the spectrum:'//
     *       t5,'o   Over an equally spaced grid of wavenumbers (type E)'/
     *       t5,'o   Only at the absorption line centers        (type C)'//
     *       t2,'(If no absorption lines are to be used, you MUST type E): ',$)
          ACCEPT 2001,  GRDMOD
  2001    FORMAT(A1)
          IF (BATMOD .EQ. 'F')  WRITE(9,2001)  GRDMOD
          IF (GRDMOD .EQ. EQSPCE) GO TO 79
          IF (GRDMOD .EQ. CENTRS) GO TO 89
```

```
   79  type 80
   80  format('0',t2,'Type the wavelength increment (in wavenumbers)'/
     *             t2,'to be used in computing the transmission'/
     *             t2,'spectrum:  ',$)
       accept 81, delnu
   81  format(f15.4)
       IF (batmod .eq. 'F') write(9,81) delnu
C
       xnmpts = (highnu-lownu)/delnu + 1.          ! corr. no. of points
       IF (xnmpts .lt. 32000.) numpts = xnmpts
       IF (xnmpts .gt. 32000.) numpts = 32000
       type 82, numpts
    82 format(//t2,'This wavelength increment corresponds to'/
     *             t2,'a total of ', t14, i5, t20, 'points;'/
     •             t2,'is this number of points okay? (Y or N): ',$)
                   accept 83, reply
   83  format(a1)
                   IF (batmod .eq. 'F') write(9,83)  reply
                   IF (reply .eq. 'N')  go to 79
C
C
C    SPECIFY WIDTH OF WAVENUMBER WINDOW IN WHICH LINES ARE CONSIDERED
C    TO CONTRIBUTE TO THE ABSORPTION AT THE WINDOW CENTER
C    /////////////////////////////////////////////////////////////////////
C
   89  type 90
   90  format('0',t2,'Type the HALF-width, in wavenumbers, of the window'/
     *             t2,'over which you want lines to be included in calcu'/
     *             t2,'lating the absorption at the window center:  ',$)
C
       accept 91, bound
   91  format(f12.4)
       IF (batmod .eq. 'F') write(9,91) bound
C
C    EXAMINE WAVENUMBER RANGE AND WINDOW WIDTH;
C    IF RANGE, EXTENDED ABOVE AND BELOW BY WINDOW HALF-WIDTH,
C    IS NOT CONTAINED IN MASTER LINE FILE,
C    GO BACK AND RE-DEFINE THEM (UNLESS this spectrum is to show
C    ONLY aerosol and/or molecular continuum effects, with NO
C    absorption lines included - query user for this information)
C    /////////////////////////////////////////////////////////////////
C
       ibotsw = 0
       itopsw = 0
C
       IF ((lownu-bound) .lt. frstwn)   ibotsw = -1
       IF ((highnu+bound) .gt. lastwn)  itopsw = -1
       isumsw = ibotsw + itopsw
C
       IF (isumsw .eq. 0)   go to 94
C
       IF(ibotsw .eq. -1)   type 92
   92  format(/t2,'lownu-bound falls below the first line'/
     *         t2,'of the master line file - ')
       IF(itopsw .eq. -1)   type 93
   93  format(/t2,'highnu+bound falls above the last line'/
     *         t2,'of the master line file - ')
       type 935
  935  format(/t2,'Is this spectrum to show ONLY aerosol and/or'/
     *         t2,'molecular continuum effects, with NO absorption'/
     •         t2,'lines included? (Y or N): ',$)
       accept 936, reply
  936  format(a1)
```

```
      IF (batmod .eq. 'F') write (9,936) reply
      IF (reply .eq. 'Y') go to 94
      type 931
 931  format(/t2,'re-define lownu, highnu and/or bound:')
      go to 726
  94  IF (GRDMOD .EQ. CENTRS) PLTCOD = 1.
      IF (GRDMOD .EQ. EQSPCE) PLTCOD = 0.
C
      ANGMIC = 0.                              ! output in angstroms or microns
      IF (1.E8/LOWNU .GE. 1.E4)  ANGMIC = 1.
C
      IF (BATMOD .EQ. 'C')       WRITE(8)  PLTCOD,ANGMIC
      IF (BATMOD .EQ. 'C'  .AND.
     *    ANGMIC .EQ. 0.)        WRITE(8)  1.E8/HIGHNU,1.E8/LOWNU !  ang
      IF (BATMOD .EQ. 'C' . AND.
     *    ANGMIC .EQ. 1.)        WRITE(8) 1.E4/HIGHNU,  1.E4/LOWNU ! mic
C
C
C   "NON-PHYSICAL" SELECTION OF LINE, CONTINUUM OR AEROSOL EXTINCTION
C   //////////////////////////////////////////////////////////////////
C

      type 121
 121  format(//t2,'The user may want to arbitrarily select or eliminate'/
     *         t2,'the molecular absorption lines of a given species,'/
     *         t2,'Rayleigh (molecular) scattering, any of the three'/
     *         t2,'molecular continua, aerosol absorption or aerosol'/
     *         t2,'scattering; to allow this, the following options'/
     •         t2,'are provided; provision is made later for more'/
     *         t2,'flexible selection of absorption lines.'//)
               type 122
 122  format(t2,'Eliminate or retain all water lines (E or R): ',$)
      accept 123, reply
 123  format(a1)
      IF (batmod .eq. 'F')  write(9,123) reply
      keepsw(1) = 1
      IF (reply .eq. 'E') keepsw(1)  = 0
C
      type 124
 124  format(t2,
 *       'Eliminate or retain all carbon dioxide lines (E or R): ',$)
      accept 123, reply
      IF (batmod .eq. 'F')  write(9,123) reply
      keepsw(2) = 1
      IF (reply .eq. 'E')  keepsw(2) = 0
C
      type 125
 125  format(t2,'Eliminate or retain all ozone lines (E or R): ',$)
      accept 123, reply
      IF (batmod .eq. 'F')  write(9,123) reply
      keepsw(3) = 1
      IF (reply .eq. 'E')  keepsw(3) = 0
C
      type 126
 126  format(t2,'Eliminate or retain Rayleigh scattering (E or R): ',$)
      accept 123, reply
      IF (batmod .eq. 'F') write(9,123) reply
      keepsw(4) = 1
      IF (reply .eq. 'E') keepsw(4) = 0
```

```
      type 127
127   format(t2,
     *   'Eliminate or retain 3.5-4.2 micron H2O continuum (E or R): ',$)
      accept 123, reply
      IF (batmod .eq. 'F') write(9,123) reply
      keepsw(5) = 1
      IF (reply .eq. 'E') keepsw(5) = 0
C
      type 128
128   format(t2,
     *   'Eliminate or retain 8-14 micron H2O continuum (E or R): ',$)
      accept 123, reply
      IF (batmod .eq. 'F') write(9,123) reply
      keepsw(6) = 1
      IF (reply .eq. 'E') keepsw(6) = 0
C
      type 129
129   format(t2,'Eliminate or retain nitrogen continuum (E or R): ',$)
      accept 123, reply
      IF (batmod .eq. 'F')  write(9,123) reply
      keepsw(7) = 1
      IF (reply .eq. 'E')  keepsw(7) = 0
C
      type 130
130   format(t2,'Eliminate or retain aerosol absorption (E or R): ',$)
      accept 123, reply
      If (batmod .eq. 'F') write(9,123) reply
      keepsw(8) = 1
      IF (reply .eq. 'E') keepsw(8) = 0
C
      type 131
131   format(t2,'Eliminate or retain aerosol scattering (E or R): ',$)
      accept 123, reply
      If (batmod .eq. 'F')  write(9,123) reply
      keepsw(9) = 1
      IF (reply .eq. 'E')  keepsw(9) = 0

C   SET UP THE LINE FILE FOR THIS RUN, BY TAKING THE APPROPRIATE LINES
C   FROM THE MASTER LINE FILE ENTERED ABOVE
C   ////////////////////////////////////////////////////////////////////////
C
C       Open the run line-file (unformatted, sequential access)
C       ----------------------------------------------------------
C
      OPEN (unit=3, name='SCR:RUNFIL.DAT', type='NEW', disp='DELETE',
     *        form='UNFORMATTED', recordsize=6, initialsize=974)
C
C      Set up interactive screening of lines
C      ---------------------------------------
      type 961
961   format(//t2,'Do you want to interactively screen the lines'/
     *          t2,'before they are written to RUNFIL? (Y or N): ',$)
      accept 962, reply
962   format(a1)
      IF (batmod .eq. 'F') write(9,962) reply
C
      iusedt = 0
      IF(reply .eq. 'Y') iusedt = 1
C
C      Initialize line counts
C      -----------------------
      nlines = 0                   ! init. count of lines read to RUNFIL
      iread  = 0                   ! "     "  "   "   "  from LINFIL
```

```
C       Read a line from LINFIL
C       ------------------------
C
  110   READ(4, end=200) wavnum, strnth, hwidth, energy, isotop, speces
C       *****************
C
C       Automatic screening
C       --------------------
        IF (wavnum .lt. (lownu-bound))  go to 110
        IF (wavnum .gt. (highnu+bound)) go to 200
        IF (keepsw(speces) .eq. 0)      go to 110

C       Interactive screening
C       ----------------------
        IF (iusedt .eq. 0) go to 990
C
        iread = iread + 1
        IF (iread .eq. 1) go to 970
        go to 975
C
  970   type 971                                          ! first stop
  971   format(//t2,'Which mode do you want to use up to the next stop?'//
     *          t12,'R - reject  all lines up to next stop'/
     *          t12,'I - include  "    "    "  "   "    "  '/
     *          t12,'E - examine each line "  "    "    "  '//
     *          t2,'Type R, I or E: ',$)
        accept 962, edmode
        IF (batmod .eq. 'F') write(9,962) edmode
C
        type 972
  972   format(/t2,'Type next wavenumber stop: ',$)
        accept 973, wnstop
  973   format(f10.4)
        IF (batmod .eq. 'F') write(9,973) wnstop
C
        go to 980
C
  975   IF (wavnum .le. wnstop)  go to 980
C
        type 971                                          ! succeeding stops
        accept 962, edmode
        IF (batmod .eq. 'F') write(9,962) edmode
        type 972
        accept 973, wnstop
        IF (batmod .eq. 'F') write(9,973) wnstop
C
  980   IF (edmode .eq. 'R')  go to 110
        IF (edmode .eq. 'I')  go to 990
C
        type 385, molnam(speces),isotop,wavnum,strnth,hwidth,energy
         type 981
  981   format(/t2,'Type R to reject this line'/
     *         t2,'Type I to include it:  ',$)
        accept 962, disp
        IF (batmod .eq. 'F') write(9,962) disp
C
        IF (disp .eq. 'R') go to 110
```

```
C       Write this line (which is retained if this point is reached)
C       into RUNFIL
C       --------------------------------------------------------------
   990  write(3) speces, isotop, wavnum, strnth, hwidth, energy
C
C       Keep a running count of the number of lines;
C       go back and read the next line
C
        nlines = nlines + 1
C
        go to 110
C       ***********
C
C       If RUNFIL contains no lines, decide whether to continue the run
C       -----------------------------------------------------------------
C
   200  IF (nlines .eq. 0) go to 201
        go to 380
C
   201  type 202
   202  format(///t2,'No absorption lines in RUNFIL;'/
     *            t2,'continue run? (Y or N): ',$)
        accept 203, reply
   203  format(a1)
        IF (batmod .eq. 'F')  write(9,203) reply
C
        IF (reply .eq. 'N')  STOP
        go to 399
C
C       If RUNFIL contains any lines, do you want to display them
C       on the screen and/or print them?
C       ----------------------------------------------------------
   380  type 381, nlines
   381  format(//t2,'There are',t12,I5,t18,'lines in RUNFIL;'/
     *  t2,'do you want to display them on the screen? (Y or N): ',$)
        accept 382, replys
   382  format(a1)
        IF (batmod .eq. 'F')  write(9,382) replys
        type 386
   386  format(t2,'do you want to print them?'/
     *         t2,'(Turn on the printer!)      (Y or N): ',$)
        accept 382, replyp
        IF (batmod .eq. 'F') write(9,382) replyp
C
        IF (replys .eq. 'N' .AND.
     *  replyp .eq. 'N')       go to 399
C
C       Display and/or print lines in RUNFIL
C       ------------------------------------
        rewind 3
C
        DO 384 I=1,nlines
        read(3) speces,isotop,wavnum,strnth,hwidth,energy
        IF (replys .eq. 'Y'  .and. I .eq. 1)  type 388
        IF (replyp .eq. 'Y'  .and. I .eq. 1)  print 388
        IF (replys .eq. 'Y')
     *  type 385,  molnam(speces),isotop,wavnum,strnth,hwidth,energy
        IF (replyp .eq. 'Y')
     *   print 385, molnam(speces),isotop,wavnum,strnth,hwidth,energy
   384  continue
C
        rewind 3
```

```
  388  format(//t2,'species',t10,'isotope',t18,'wavenumber',
     *          t33,'strength',t48,'halfwidth',t60,'l.s. energy')
  385  format(/t2,a4,t10,i4,t18,f10.4,t33,e10.3,t48,f8.4,t60,f10.3)
c
c      Compute number of blocks required to store RUNFIL
c      -------------------------------------------------
c
  399  irsize = IFIX((FLOAT(nlines)/512.)*24.) + 1 ! no. of blocks
c
c      If RUNFIL size is less than 164 blocks, put RUNFIL in VM
c      --------------------------------------------------------
c
       IF (irsize .GT. 164) go to 1507
c
       OPEN (unit=1, name='VM:RUNFIL.DAT', type='NEW', disp='KEEP',
     *       form='UNFORMATTED', recordsize=6, initialsize=irsize)
c
       IF(nlines .eq. 0)  go to 1509
       rewind 3
c
       DO 1508, II=1,nlines
       read (3) speces, isotop, wavnum, strnth, hwidth, energy
       write(1) speces, isotop, wavnum, strnth, hwidth, energy
 1508  continue
c
 1509  CLOSE(unit=3, disp='DELETE')
       CLOSE(unit=1, disp='KEEP')
       OPEN (unit=3, name='VM:RUNFIL.DAT', type='OLD', disp='DELETE',
     *       form='UNFORMATTED', recordsize=6)
c
c      Close LINFIL
c      ------------
 1507  CLOSE(unit=4,disp='KEEP')
c
c      Comments
c      ---------
c       The line file for this run is now set up; it contains only those
c      lines for which the concentration is non-zero; the total number of
c      lines in RUNFIL is nlines; the wavenumber range contained in the
c      file is lownu-bound to highnu+bound.
c
c************************************************************************
       IF (batmod .eq. 'F') go to 800
c
c                                  CALCULATION
c                                 *************
c
c        To compute the absorption coefficient at each of the net of points
c      of separation delnu spanning the wavenumber range (lownu, highnu),
c      we compute, at each point of wavenumber v, the sum of the contri-
c      butions from all lines which lie in the window (v-bound, v+bound);
c      for an overview spectrum the net of points consists of just the
c      line centers.
c
c
c    COMPUTE AEROSOL-ONLY TRANSMISSION AT HIGHNU AND LOWNU
c    (for use by PLOTSP in drawing the aerosol line
c     for a line-center plot)
c    ////////////////////////////////////////////////////////////
       IAERMD = 1
       ATL = -1.
       ATU = -1.
```

```
        V = HIGHNU
        GO TO 300
C
 2900   IF (ATL .LT. 0.) GO TO 2910
        IF (ATU .LT. 0.) GO TO 2920
C
 2910   ATL = TRANS
        V = LOWNU
        GO TO 300
C
 2920   ATU = TRANS
        WRITE(8) ATL,ATU
        IAERMD = 0
C
C
C     LOOP OVER NET OF POINTS SPANNING THE SPECTRUM
C     ///////////////////////////////////////////////
        IF (GRDMOD .EQ. EQSPCE)  GO TO 3000
        IF (GRDMOD .EQ. CENTRS)  GO TO 3100
C
 3000   v = lownu                              ! start at low end of spectrum
        i = 1                                  ! index for net of points
        itopsw = 0                             ! switch to jump out of loop
        GO TO 300
C
 3100   REWIND 3
        LNINDX = 0
  310   READ(3)  SPECES,ISOTOP,WAVNUM,STRNTH,HWIDTH,ENERGY
        LNINDX = LNINDX + 1
        IF (WAVNUM .LT. LOWNU)  GO TO 310
        FRSTLN = LNINDX
        I = 1
        V = WAVNUM
C
  300   call rdtape                                  ! loop over net of points
C       ************                                   ***********************
C
C      Go to section of code for given path type
C      -----------------------------------------
C
        IF (pthtyp .eq. 'H')  go to 1000
        IF (pthtyp .eq. 'V'   .OR.  pthtyp .eq. 'S')  go to 1100
C
C       Horizontal path
C       ---------------
C
 1000   layer = iblayr
        extcof = cay(iatmod,layer) + aersol(layer)       ! ext. coeff.
        trans = exp(-extcof*plnkm)                       ! transmittance
        waveno = v                                       ! wavenumber
        go to 1400
C
C      Vertical or slant path
C      ----------------------
C
 1100   depth = 0.
C
```

```
      DO 1110 layer=itlayr,iblayr
C     ****************************
      l = layer
      extcof = cay(iatmod,l) + aersol(l)
      pathl = secant
      IF (l .eq. iblayr)  pathl = botseg*secant
      IF (l .eq. itlayr)  pathl = topseg*secant
      depth = depth + extcof*pathl
 1110 continue
C     *********
C
      trans = exp(-depth)
      waveno = v

C     Store this point in spectrum file (formatted, seq. access)
C     ----------------------------------------------------------------
 1400 IF (IAERMD .EQ. 1) GO TO 2900
C
      IF (angmic .eq. 0.)  write(8)  1.e8/waveno, trans  ! conv. to ang.
      IF (angmic .eq. 1.)  write(8)  1.e4/waveno, trans  ! conv. to mic.
C
C     Type this point on the screen to show the user
C     the progress of the calculation
C     ------------------------------------------------
      type 1401,  waveno,trans
 1401 format(t2,e12.5,t18,f8.4)
C
C     Test wavenumber to see if high end of range has been reached
C     ----------------------------------------------------------------
      IF (GRDMOD .EQ. EQSPCE) GO TO 350
C
      REWIND 3
      LNINDX = 0
  351 READ(3,END=800)  SPECES,ISOTOP,WAVNUM,STRNTH,HWIDTH,ENERGY
      LNINDX = LNINDX + 1
      IF (LNINDX .EQ. FRSTLN+I)  GO TO 355
      GO TO 351
  355 I = I+ 1
      V = WAVNUM
      WAVNOP = WAVENO
      TRANSP = TRANS
      IF (V .LE. HIGHNU)  GO TO 300
      I = I - 1
      GO TO 800
C
C
  350 rewind 3
  356 read(3,end=358)  speces,isotop,wavnum,strnth,hwidth,energy
      IF (wavnum .gt. v .and. wavnum .lt. v+delnu) go to 357
      go to 356
  357 v = wavnum
      go to 359
  358 v = v + delnu
  359 i= i + 1
C
      wavnop = waveno            ! store these as "previous" values
      transp = trans
C
      IF (v .gt. highnu)  go to 3355
      go to 300
C     *********
C
```

```
 3355   IF (itopsw .eq. 1)  go to 3356
        itopsw = 1
        v = highnu
        go to 300
C       *********
C
 3356   i = i-1                              ! correct the line count
C
C*****************************************************************************
   800  close(unit=3,disp='DELETE')
        IF (batmod .eq. 'C') close(unit=8,disp='KEEP')
C
        type 1560
 1560   format(//t2,'Do you want to compute another spectrum? (Y or N): ',$)
        accept 1561, reply
 1561   format(a1)
C
        IF (batmod .eq. 'F')  write(9,1561) reply
        IF (reply .eq. 'Y')  go to 9
C
        IF (batmod .eq. 'F')  write(9,1562)
 1562   format('$EOD'/'$EOJ')
        IF (batmod .eq. 'F')  CLOSE (unit=9, disp='KEEP')
C
        STOP
        END
```

```
      SUBROUTINE SETUP
C
C                          DICTIONARY OF SYMBOLS
C                          *********************** 
C
C     AAB        AEROSOL, ABSORPTION COEFFICIENT - INPUT DATA
C     ABSOR      8 - 14 MICRON CONTINUUM ABSORPTION COEFFICIENT
C     AKCL       CLEAR AEROSOL ABSORPTION COEFFICIENT INTERPOLATED AT
C                FREQUENCY V
C     AKHZ       HAZY AEROSOL ABSORPTION COEFFICIENT INTERPOLATED AT
C                FREQUENCY V
C     ALFA0      LINE HALF WIDTH - INPUT FROM TAPE
C     ALPHAL     LORENTZ HALF-WIDTH
C     ASHZ       HAZY AEROSOL SCATTERING COEFFICIENT INTERPOLATED AT
C                FREQUENCY V
C     ARSMOD     AEROSOL MODEL - INPUT DATA
C     ASC        AEROSOL SCATTERING COEFFICIENT - INPUT DATA
C     ATMODL     ATMOSPHERIC MODEL - INPUT DATA
C     BOUND      LIMIT OUTSIDE OF WHICH LINE CONTRIBUTIONS ARE NOT
C                CONSIDERED - INPUT DATA
C     CAS        CLEAR AEROSOL SCATTERING COEFFICIENT (50 KM SEA LEVEL
C                VISIBILITY) - OUTPUT DATA
C     CAY        MOLECULAR ABSORPTION COEFFICIENT - OUTPUT DATA
C     CHI        MODIFICATION TO THE LORENTZ LINE SHAPE FOR CO2
C                - INPUT DATA/BLK DATA
C     CNCS       NITROGEN-BROADENED WATER VAPOR CONTINUUM ABSORPTION
C                COEFFICIENT
C     CONST      (R*1.0E-03)/(A*1.0E+05). WHERE R IS GAS CONSTANT AND A
C                IS AVOGADRO'S NUMBER
C     CS2        PARTITION FUNCTION TEMPERATURE CORRECTION
C     DELP       PRESSURE DIFFERENCE BETWEEN ATMOSPHERIC LEVELS
C     DELTAZ     DIFFERENCE BETWEEN TWO ADJACENT LAYERS
C     DNU        FREQUENCY INCREMENT ASSOCIATED WITH LORENTZ MODIFICATION
C                - INPUT DATA
C     EPP        ENERGY OF LOWER STATE OF TRANSITION -INPUT FROM TAPE
C     EVH1       CLEAR VERTICAL SCALING FACTOR
C     EVH2       HAZY VERTICAL SCALING FACTOR
C     FA         AEROSOL MODEL FREQUENCY - INPUT DATA
C     FAC        INTERMEDIATE INTERPOLATING DATA FOR THE AEROSOL
C                FREQUENCY DATA
C     GNU        LINE FREQUENCY - INPUT FROM TAPE
C     HAS        HAZY: AEROSOL SCATTERING COEFFICIENT (5 KM SEA LEVEL
C                VISIBILITY) - OUTPUT DATA
C     HCONV      0.1*3.34E+22 MOLECULES/CM2
C     HZ1        VERTICAL SCALING FACTOR FOR CLEAR AEROSOL MODEL
C                - INPUT DATA /BLK DATA
C     HZ2        VERTICAL SCALING FACTOR FOR HAZY AEROSOL MODEL
C     H2OLAY     WATER VAPOR CONCENTRATION
C     IK1        ATMOSPHERIC MODEL INDEX
C     ITI        MOLECULAR SPECIES IDENTIFICATION (1=H2O, 2=CO2,
C                3=OZONE) - INPUT FROM TAPE
C     ITP        NUMBER OF ATMOSPHERIC LAYERS - INPUT DATA/BLK DATA
C     I1         NUMBER OF ABSORPTION LINES
C     JT         NUMBER OF ELEMENTS IN THE LORENTZ MODIFICATION FACTOR
C                - INPUT DATA/BLK DATA
C     KSAM       NUMBER OF MODEL ATMOSPHERES - INPUT DATA/BLK DATA
C     LC         CLEAR AEROSOL MODEL INDEX
C     LH         HAZY AEROSOL MODEL INDEX
C     MOL        MOLECULE IDENTIFIER (1= H2O, 2= CO2, 3= O3)-INPUT FROM
C                TAPE
C     O3CON      CONVERSION FACTOR FROM GM/M**3 TO MOLECULES/CM**2
C     P          ATMOSPHERIC LEVEL PRESSURE - INPUT DATA
```

```
C      PATH        OPTICAL DEPTH AT LINE CENTER FOR 1 - KM SEA LEVEL PATH
C      PBAR        AVERAGE LAYER PRESSURE
C      PH2O        WATER VAPOR PRESSURE
C      PI          RATIO OF CIRCUMFERENCE OF A CIRCLE TO ITS DIAMETER
C      S           ABSORPTION LINE INTENSITY - INPUT FROM TAPE
C      SEC         SECANT ANGLE - INPUT DATA
C      ST          TEMPERATURE CORRECTED LINE INTENSITY
C      T           ATMOSPHERIC LAYER TEMPERATURE - INPUT DATA
C      TA1         CLEAR: AEROSOL ABSORPTION COEFFICIENT - OUTPUT DATA
C      TA2         HAZY: AEROSOL ABSORPTION COEFFICIENT - OUTPUT DATA
C      TBAR        AVERAGE LAYER TEMPERATURE
C      V           FREQUENCY AT WHICH THE EXTINCTION COEFFICIENT ARE BEING
C                  CALCULATED - INPUT DATA
C      VBOT        LOWER FREQUENCY LIMIT FOR THE LINE CONTRIBUTIONS.
C                  VBOT= V-BOUND
C      VTOP        UPPER FREQUENCY LIMIT FOR THE LINE CONTRIBUTIONS.
C                  VTOP= V+BOUND
C      W           ABSORBER CONCENTRATION
C      WBAR        MEAN WATER VAPOR CONCENTRATION FOR A LAYER
C      WG          SEA LEVEL VALUES OF MOLECULAR ABUNDANCES - INPUT DATA
C      WH          WATER VAPOR CONCENTRATION AT A SPECIFIC LEVEL-INPUT DATA
C      WH1         SCALE HEIGHT ASSOCIATED WITH WATER VAPOR
C      WH3         SCALE HEIGHT ASSOCIATED WITH OZONE
C      WO          OZONE CONCENTRATION AT A SPECIFIC LEVEL - INPUT DATA
C      WO3         MEAN OZONE CONCENTRATION FOR A LAYER
C      WV          WAVELENGTH CORRESPONDING TO FREQUENCY, V
C      W1D         INTERMEDIATE QUANTITY ASSOCIATED WITH COMPUTING
C                  INTEGRATED WATER AMOUNT
C      W3D         INTERMEDIATE QUANTITY ASSOCIATED WITH COMPUTING
C                  INTEGRATED OZONE AMOUNT
C      X1          LORENTZ LINE MODIFICATION VARIABLE
C      Z           ATMOSPHERIC HEIGHT (KM) - INPUT DATA
C      ZDIST       ABSOLUTE DISTANCE FROM THE LINE CENTER FREQUENCY V
C*****************************************************************************
C
C                          SPECIFICATION STATEMENTS
C                         **************************
C
      IMPLICIT    INTEGER*2 (I-N)
      REAL*4      LOWNU
      DIMENSION ATMODL(3,5),ARSMOD(6,5),WH(3,16),WO(3,16),KEEPSW(9)
C
      COMMON      Z(16),
     *            P(3,16),T(3,16),W(3,3,16),CAY(3,16),WG(3,3),
     *            TA2(16),FA(6,61),ASC(6,61),AAB(6,61),
     *            HAS(16)
C
      COMMON/BLK1/    DNU(16),CHI(16),HZ1(16),HZ2(16),ITP,JT,KSAM
      COMMON/RSETUP/ BOUND,LOWNU,HIGHNU,DELNU,V,
     •                IATMOD,IARMDG,IARMDT,IARMDS,
     •                NUMPTS,NLINES,KEEPSW,IBLAYR,ITLAYR,
     *                IRSIZE, HZINT(16),IAERMD
C
      DATA            HCONV,O3CON/3.34E+21,1.255E+21/
C
C*****************************************************************************
C
```

```
C    READ ATMOSPHERIC MODELS FROM DATA SET 2 AND PROCESS THEM
C    ////////////////////////////////////////////////////////
C
C        OPEN DATA SET 2 (formatted, sequential access)
C        ----------------------------------------------
          OPEN (UNIT=2, NAME='SD2:ATMMOD.DAT', TYPE='OLD',
     *         DISP='KEEP', FORM='FORMATTED')
C
C        READ SEA LEVEL VALUES OF MOLECULAR ABUNDANCES FROM DATA SET 2;
C        STORE THEM IN ARRAY WG
C        --------------------------------------------------------------
          DO 1 I=1,3                             ! I = atm. model
          READ (2,19) (WG(I,M),M=1,3)            ! M = molecule (H2O,CO2,O3)
    1   CONTINUE
C
   19   FORMAT(3E10.3)

C       READ THE NAMES OF THE ATMOSPHERIC MODELS AND THE DATA
C       -------------------------------------------------------
        DO 300 J=1,3                                              ! read names
C       ************
        READ (2,21) (ATMODL(J,M),M=1,5)
C
             DO 200 L=1,ITP               ! read data (L=1-16)
C            ************
             K=ITP-L+1                    ! reverse numbering of layers
C                                           (K=16-1)
             READ (2,22) Z(K), P(J,K),T(J,K),WH(J,K),WO(J,K)
  200        CONTINUE
C
  300   CONTINUE
C       ************
C
C                             Z = height in kilometers
C                             P = pressure
C                             T = temperature
C                             WH = water vapor concentration
C                             WO = ozone concentration
   21   FORMAT (5A4)
   22   FORMAT (F6.1,  E10.3,F6.1,2E10.1)
C
C             NOTE: The order of the atmospheric layers is reversed;
C                   i.e., K = 1-16 runs layers from top to ground.
C                   (K = 1 AT 15 KM,..., K = 16 at sea level.)
C
C
C       COMPUTE THE AVERAGE LAYER CONCENTRATIONS OF H2O, CO2 AND O3
C       -----------------------------------------------------------
C             NOTE: Molecular densities are assumed to decrease expo-
C                   nentially between consecutive levels; we compute
C                   the average layer value by integrating the expo-
C                   nential over the layer. If W1D or W3D approach
C                   zero, the log is poorly defined, so in this case
C                   we use just a simple average.
        K1=ITP-1     ! (= 16-1 = 15)
C
```

```
      DO 7 I=1,3              ! atmospheric model loop (3 models)
      DO 7 K=1,K1             ! layer loop - from level 15 at top (K=1)
C     ************                              ! to level 1 at 1 km (K=15)
C
      WBAR=(WH(I,K)+WH(I,K+1))/2.0
      WO3=(WO(I,K)+WO(I,K+1))/2.0
      DELTAZ=Z(K)-Z(K+1)
C
      W1D=WH(I,K)/WH(I,K+1)
      IF (ABS(W1D-1.00).LT.0.1) GO TO 3
      WH1=-DELTAZ/ALOG(W1D)
      W(I,1,K)=WH1*HCONV*(WH(I,K+1)-WH(I,K))      ! water vapor conc.
      GO TO 4
C
    3 W(I,1,K)=WBAR*HCONV*DELTAZ                  !   "     "     "
C
    4 W3D=WO(I,K)/WO(I,K+1)
C
      IF (ABS(W3D-1.00).LT.0.1) GO TO 5
      WH3=-DELTAZ/ALOG(W3D)
      W(I,3,K)=WH3*O3CON*(WO(I,K+1)-WO(I,K))      ! ozone conc.
      GO TO 6
C
    5 W(I,3,K)=WO3*O3CON*DELTAZ                   !   "     "
C
C         NOTE: THE MOLECULAR DENSITY IN A GIVEN LAYER OF CO2,
C               A UNIFORMLY MIXED GAS, IS DIRECTLY RELATED TO
C               THE PRESSURE INCREMENT BETWEEN THE LAYER BOUNDARIES.
C               THE CONSTANT IN THE FOLLOWING EXPRESSION IS THE
C               UNIFORMLY MIXED GAS CONSTANT FOR CO2.
C
    6 W(I,2,K)=((P(I,K+1)-P(I,K))/1013.0)*7.102E+21    ! CO2 conc.
C
    7 CONTINUE
C     **********
C
C    READ AEROSOL MODELS FROM DATA SET 2
C    AND STORE THEM IN COMMON
C    -----------------------------------
C                         Aerosol models:  rural          - 1
C                                          tropospheric   - 2
C                                          maritime       - 3
C                                          urban          - 4
C                                          background
C                                          stratospheric  - 5
C                                          aged volcanic  - 6
C
      DO 10 I=1,6
      READ (2,25) (ARSMOD (I,M),M=1,5)                 ! read model names
      READ (2,26) (FA(I,J),ASC(I,J),AAB(I,J),J=1,61)   ! read data
   10 CONTINUE
C
   25 FORMAT(5A4)
   26 FORMAT (F8.3,2F7.5)
C
      DO 11 I=1,6                                      ! convert from microns
      DO 12 J=1,61                                     ! to wavenumbers
      FA(I,J) = 1.0E+04/FA(I,J)
   12 CONTINUE
   11 CONTINUE
```

```
C     CLOSE ATMOSPHERE MODEL FILE
C     ---------------------------
   50  CLOSE (UNIT=2)
C
       RETURN
       END



      SUBROUTINE RDTAPE
C
C       This routine selects, from RUNFIL, those lines in the range
C
C                    (v-bound, v+bound)
C
C     and stores them in the (unformatted) file VM:WINDOW.DAT.
C
C     The number of retained lines, LINECT, is sent to subroutine ATMOS.
C*****************************************************************************
C
C                         SPECIFICATION STATEMENTS
C                        **************************
C
       IMPLICIT     INTEGER*2 (I-N)
       INTEGER      SPECES, SWITCH
       REAL*4       LOWNU
       DIMENSION    KEEPSW(9)
C
       COMMON       Z(16),
     *              P(3,16),T(3,16),W(3,3,16),CAY(3,16),WG(3,3),
     *              TA2(16),FA(6,61),ASC(6,61),AAB(6,61),
     *              HAS(16)
C
       COMMON/BLK1/    DNU(16),CHI(16),HZ1(16),HZ2(16),ITP,JT,KSAM
       COMMON/RSETUP/  BOUND,LOWNU,HIGHNU,DELNU,V,
     *                 IATMOD,IARMDG,IARMDT,IARMDS,
     *                 NUMPTS,NLINES,KEEPSW,IBLAYR,ITLAYR,
     *                 IRSIZE, HZINT(16),IAERMD
       DATA         SWITCH/0/
C****************************************************************************
C
C     OPEN THE VIRTUAL MEMORY FILE FOR THE WINDOW OF LINES
C     (unformatted, sequential access)
C
       IF(IRSIZE .LE. 164)
     *   OPEN (UNIT=10,NAME='VM:WINDOW.DAT',TYPE='NEW',DISP='DELETE',
     *        FORM='UNFORMATTED',RECORDSIZE=6,INITIALSIZE=IRSIZE)
C
       IF(IRSIZE .GT. 164)
     *   OPEN (UNIT=10,NAME='SD3:WINDOW.DAT',TYPE='NEW',DISP='DELETE',
C    *        FORM='UNFORMATTED',RECORDSIZE=6,INITIALSIZE=650)

C     READ, FROM RUNFIL, ALL LINES IN THE RANGE (V-BOUND, V+BOUND)
C     AND STORE THEM IN VM:WINDOW.DAT
C     ///////////////////////////////////////////////////////////////
C
       LINECT = 0                       ! initialize line count
       IF (NLINES .EQ. 0) GO TO 20      ! no lines, so no window file
       REWIND 3                         ! position pointer to beginning of 3
```

```
    1  READ(3,END=20)  SPECES, ISOTOP, WAVNUM, STRNTH, HWIDTH, ENERGY
C      **************
C
       IF (WAVNUM .LT. V-BOUND)  GO TO 1
       IF (WAVNUM .GT. V+BOUND)  GO TO 20
C
       LINECT = LINECT + 1
C
       WRITE(10) SPECES, ISOTOP, WAVNUM, STRNTH, HWIDTH, ENERGY
C
       GO TO 1
C      ********
C
C*************************************************************************
C
C     SEND THIS ''WINDOW'' OF LINES TO SUBROUTINE ATMOS
C     ////////////////////////////////////////////////
C
   20  CALL ATMOS (LINECT)
C
       RETURN
       END




       SUBROUTINE ATMOS (LINECT)
C
C                         SPECIFICATION STATEMENTS
C                        **************************
C
       IMPLICIT  INTEGER*2 (I-N)
       REAL*4    LOWNU
       DIMENSION CS2(3),KEEPSW(9)
C
C                          NOTE: ST   = temp. corrected line intensity
C                                CS2  = partition func. temp. correction
C
       COMMON    Z(16),
     *           P(3,16),T(3,16),W(3,3,16),CAY(3,16),WG(3,3),
     *           TA2(16),FA(6,61),ASC(6,61),AAB(6,61),
     *           HAS(16)
C
       COMMON/BLK1/    DNU(16),CHI(16),HZ1(16),HZ2(16),ITP,JT,KSAM
       COMMON/RSETUP/  BOUND,LOWNU,HIGHNU,DELNU,V,
     *                 IATMOD,IARMDG,IARMDT,IARMDS,
     *                 NUMPTS,NLINES,KEEPSW,IBLAYR,ITLAYR,
     *                 IRSIZE, HZINT(16),IAERMD
C
       DATA            CONST,PI/1.380258E-24,3.141593/
C
C             NOTE:  CONST= (R*1.0E-03)/(A*1.0E+05)
C                         = (8.3144E+07*1.0E-03)/(6.0238E+23*1.0E+05)
C                    (R = GAS CONST;  A = AVOGADRO'S NUMBER)
C***************************************************************************************
```

```
C      DO 18 IS THE MAJOR COMPUTATIONAL LOOP ON ATMOSPHERIC LAYERS
C      WITHIN WHICH MONOCHROMATIC MOLECULAR ABSORPTION COEFFICIENTS
C      ARE COMPUTED.
C
         IK1=IATMOD                  ! (use only this atm. model)
C
         DO 18 K=ITLAYR,IBLAYR       ! Loop over atm. layers
C        *********************         *********************
          IF (K.EQ.ITP) GO TO 2
C
          PBAR=(P(IK1,K)+P(IK1,K+1))/2.0                ! av. layer pressure
          TBAR=(T(IK1,K)+T(IK1,K+1))/2.0                ! av. layer temp.
          PH2O=CONST*TBAR*W(IK1,1,K)/ABS(Z(K)-Z(K+1))  ! water vapor pressure
          GO TO 3
C
    2     PBAR=P(IK1,ITP)                               ! av. pressure AT GROUND
          TBAR=T(IK1,ITP)                               ! av. temp. at ground
          PH2O=CONST*TBAR*WG(IK1,1)                     ! water vapor press. at ground
C
    3     CONTINUE
C
C        DETERMINE CORRECT TEMPERATURE DEPENDENCE OF ROTATIONAL PARTITION
C        FUNCTION
C
            CS2(2) = 296.0/TBAR
            CS2(1) = CS2(2)**1.5
            CS2(3) = CS2(1)
C
C           DO 17 LOOP CYCLES THRU ALL ABSORPTION LINES
C           AND ADDS THEIR CONTRIBUTIONS TO THE ABSORPTION
C           COEFFICIENT AT THE WAVENUMBER V.
C
            CAY(IK1,K) = 0.0       ! initialize for loop over LINES
            IF (LINECT .EQ. 0)   GO TO 18
            IF (IAERMD .EQ. 1)   GO TO 18
            REWIND 10              ! position pointer to beginning of 10
C
            DO 17 I=1,LINECT       ! Loop over absorption lines in window
C           ****************         ************************************
C
            READ(10)  MOL, ISOTOP, GNU, S, ALFA0, EPP
C
            M = MOL                        ! Molecular species:
C                                                1 - H2O
C                                                2 - CO2
C                                                3 - O3
C
            ST = S*CS2(M)*EXP(-EPP*                  ! temp. corrected line int.
     *           (296.-TBAR)/(296.*TBAR*.6951))      ! (Boltzmann temp.
C                                                      correction factor)
C
C           INSERT THE ISOTOPIC RELATIVE ABUNDANCE FACTOR
C
            IF (M .EQ. 1) GO TO 100
            IF (M .EQ. 2) GO TO 200
            IF (M .EQ. 3) GO TO 300
C
```

```
C          Water Vapor Isotopes
C
  100      RELBUN = 0.
           IF (ISOTOP .EQ. 161) RELBUN = 0.9976
           IF (ISOTOP .EQ. 171) RELBUN = 0.0004
           IF (ISOTOP .EQ. 181) RELBUN = 0.002
           GO TO 400
C
C       Carbon Dioxide Isotopes
C
  200   RELBUN = 0.
        IF (ISOTOP .EQ. 626)  RELBUN = 0.98418
        IF (ISOTOP .EQ. 636)  RELBUN = 0.01103
        IF (ISOTOP .EQ. 627)  RELBUN = 0.00079
        IF (ISOTOP .EQ. 628)  RELBUN = 0.00394
        GO TO 400
C
C       Ozone Isotopes
C
  300   RELBUN = 0.
        IF (ISOTOP .EQ. 666)  RELBUN = 1.
C
C       Modify Molecular Abundance
C
  400   IF (K .EQ. ITP)  CONC = RELBUN*WG(IK1,M)
        IF (K .NE. ITP)  CONC = RELBUN*W(IK1,M,K)
C
C       INSERT IN PBAR THE WATER VAPOR BROADENING FACTOR
C
        PEFF=PBAR
        IF (M. EQ. 1)  PEFF=PBAR + 4.0*PH20
C
C       SQRT(296./TBAR) IS THE HALF-WIDTH TEMPERATURE CORRECTION FACTOR
C
        ALPHAL=ALFA0*PEFF*SQRT(296./TBAR)/1013.0
        ZDIST=ABS(V-GNU)
        X1=1.0
C
        IF (M .NE. 2)        GO TO 15
C
             X1=0.0
             JT1=JT-1
C
             DO 9 L=1,JT1
C            *************
             IF (ZDIST.GE.DNU(L).AND.ZDIST.LE.DNU(L+1))    GO TO 8
             GO TO 9
C
    8        X1=((CHI(L+1)-CHI(L))/(DNU(L+1)-DNU(L)))*(ZDIST-DNU(L))
     *          +CHI(L)
             GO TO 15
C
    9        CONTINUE
C            *************
C
```

```
C              SINCE PBAR .GT. 10 MB, LORENTZ SHAPE APPLIES.
C              NOTE ALSO THAT SINCE WE NEVER CONSIDER PRESSURE LESS THAN 10 MB,
C              NEITHER VOIGT SHAPE NOR DOPPLER BROADENING APPLY.
C
   15          IF (K .EQ. ITP) GO TO 16          ! (ITP=16 AT ground)
C
               CAY(IK1,K) = CAY(IK1,K)
     *                    + ST*ALPHAL*CONC*X1/(PI*(ZDIST**2+ALPHAL**2))
               GO TO 17
C
   16          CAY(IK1,K) = CAY(IK1,K)
     *                    + ST*ALPHAL*CONC*X1/(PI*(ZDIST**2+ALPHAL**2))
C
C                                         IK1 = atmosphere model index
C                                         K   = atmosphere layer index
C                                         M   = molecule (1=H2O, 2=CO2, 3=O3)
C
   17          CONTINUE                          ! End of absorption line loop
C              **************                      ***************************
C                                NOTE: CAY now is the total abs. coeff.
C                                      (ALL lines) for this atm model
C                                      and this layer.
   18    CONTINUE
C        *************************               ! End of layer loop
C                                                ******************
C      CLOSE WINDOW FILE AND DELETE IT
C      /////////////////////////////////
         CLOSE (UNIT=10, DISP='DELETE')
C
C      COMPUTE CONTINUUM ABSORPTION AND AEROSOL EXTINCTION COEFFICIENTS
C      ////////////////////////////////////////////////////////////////
       CALL CONT (V)
C
       RETURN
       END



          SUBROUTINE CONT
C
C                          SPECIFICATION STATEMENTS
C                         ***************************
C
          IMPLICIT INTEGER*2 (I-N)
          REAL*4   LOWNU, CON(20), FN2(91), CN2(91)
          DIMENSION KEEPSW(9)

          COMMON Z(16),
     *    P(3,16),T(3,16),W(3,3,16),CAY(3,16),WG(3,3),TA2(16),
     *    FA(6,61),ASC(6,61),AAB(6,61),HAS(16)
C
          COMMON/BLK1/    DNU(16),CHI(16),HZ1(16),HZ2(16),ITP,JT,KSAM
          COMMON/RSETUP/  BOUND,LOWNU,HIGHNU,DELNU,V,
     *                    IATMOD,IARMDG,IARMDT,IARMDS,
     *                    NUMPTS,NLINES,KEEPSW,IBLAYR,ITLAYR,
     *                    IRSIZE, HZINT(16),IAERMD
C
```

```
      DATA  CON/0.230, 0.187, 0.147, 0.117, 0.097, 0.087,
     *          0.100, 0.120, 0.147, 1.174, 0.200, 0.240,
     *          0.280, 0.330, 6*0.000/
C
      DATA  FN2/2000., 2050., 2075., 2100., 2125., 2150., 2155., 2160.,
     *          2165., 2170., 2175., 2180., 2185., 2190., 2195., 2200.,
     *          2205., 2210., 2215., 2220., 2225., 2230., 2235., 2240.,
     *          2245., 2250., 2255., 2260., 2265., 2270., 2275., 2280.,
     *          2285., 2290., 2295., 2300., 2305., 2310., 2315., 2320.,
     *          2325., 2330., 2335., 2340., 2345., 2350., 2355., 2360.,
     *          2365., 2370., 2375., 2380., 2385., 2390., 2395., 2400.,
     *          2405., 2410., 2415., 2420., 2425., 2430., 2435., 2440.,
     *          2445., 2450., 2455., 2460., 2465., 2470., 2475., 2480.,
     *          2485., 2490., 2495., 2500., 2505., 2510., 2515., 2520.,
     *          2525., 2530., 2535., 2540., 2545., 2550., 2575., 2600.,
     *          2625., 2650., 2800./
C
      DATA  CN2/1.00E-21, 1.20E-07, 1.80E-07, 6.30E-07, 2.00E-06,
     *          9.00E-06, 1.13E-05, 1.36E-05, 1.66E-05, 1.96E-05,
     *          2.16E-05, 2.36E-05, 2.63E-05, 2.90E-05, 3.15E-05,
     *          3.40E-05, 3.66E-05, 3.92E-05, 4.26E-05, 4.60E-05,
     *          4.95E-05, 5.30E-05, 5.65E-05, 6.00E-05, 6.30E-05,
     *          6.60E-05, 6.89E-05, 7.18E-05, 7.39E-05, 7.60E-05,
     *          7.84E-05, 8.08E-05, 8.39E-05, 8.70E-05, 9.13E-05,
     *          9.56E-05, 1.08E-04, 1.20E-04, 1.36E-04, 1.52E-04,
     *          1.60E-04, 1.69E-04, 1.60E-04, 1.51E-04, 1.37E-04,
     *          1.23E-04, 1.19E-04, 1.16E-04, 1.14E-04, 1.12E-04,
     *          1.12E-04, 1.11E-04, 1.11E-04, 1.12E-04, 1.14E-04,
     *          1.14E-04, 1.12E-04, 1.10E-04, 1.07E-04, 1.02E-04,
     *          9.90E-05, 9.50E-05, 9.00E-05, 8.65E-05, 8.20E-05,
     *          7.65E-05, 7.05E-05, 6.50E-05, 6.10E-05, 5.50E-05,
     *          4.95E-05, 4.50E-05, 4.00E-05, 3.75E-05, 3.50E-05,
     *          3.10E-05, 2.65E-05, 2.50E-05, 2.20E-05, 1.95E-05,
     *          1.75E-05, 1.60E-05, 1.40E-05, 1.20E-05, 1.05E-05,
     *          9.50E-06, 6.00E-06, 3.50E-06, 2.00E-06, 1.50E-06,
     *          1.00E-20/
C*************************************************************************

C
C     INITIALIZE AEROSOL ABSORPTION AND SCATTERING COEFFICIENTS
C     ///////////////////////////////////////////////////////////////
C
      DO2 K=1,ITP                       ! loop over 16 atm. layers
C     ***********
      TA2(K) = 0.0
      HAS(K) = 0.0
    2 CONTINUE
C     ***********
C
C     COMPUTE MOLECULAR CONTINUUM ABSORPTION
C     AND RALEIGH (MOLECULAR) SCATTERING,
C     AND ADD TO MOLECULAR LINE ABSORPTION
C     ////////////////////////////////////////
C
      J = IATMOD                        ! (use only this atm. model)
C
C
      DO 17 K=ITLAYR,IBLAYR             ! Loop over atm. layers
C     *********************               *********************
C
      TBAR = T(J,ITP)                   ! Temp. at ground
      PBAR = P(J,ITP)                   ! Press. at ground
      IF (K .EQ. ITP)  GO TO 100
```

```
      PBAR = (P(J,K) + P(J,K+1))/2.0  ! av. layer pressure
      TBAR = (T(J,K) + T(J,K+1))/2.0  ! av. layer temperature
C
C
C     RAYLEIGH SCATTERING
C     -------------------
C
  100 IF (KEEPSW(4) .EQ. 0)  GO TO 200
      IF (V .GE. 2740.) GO TO 110     ! waveno. range for this effect
      GO TO 200
C
  110 IF (K .EQ. ITP) GO TO 120
      IF (K .NE. ITP) GO TO 130
C
  120 EVM = (273./1013.) * P(J,ITP) / T(J,ITP)          ! ground layer
      GO TO 140
C
  130 HM = 1./ALOG((P(J,K+1)*T(J,K))/(P(J,K)*T(J,K+1)))  ! other layers
      EVM = (273./1013.) * HM * ((P(J,K+1)/T(J,K+1))-(P(J,K)/T(J,K)))
C
  140 TM = 9.807E-20 * EVM * V**4.0117
      CAY(J,K) = CAY(J,K) + TM
C
C
C     3.5-4.2 MICRON H2O CONTINUUM
C     ----------------------------
C
  200 IF (KEEPSW(5) .EQ. 0) GO TO 300
      IF (V .GE. 2350. .AND. V .LE. 2800.) GO TO 210   ! range of effect
      GO TO 300
C
  210 XI = (V-2350.)/50. + 1.
      MH = XI + 1.001
      XH = XI - FLOAT(MH)
      TX5 = CON(MH)
      TX5 = TX5 + XH*(CON(MH)-CON(MH-1))
      CCONT = TX5/3.34E22
      TDEP = EXP (4.56*(296./TBAR-1.))
      CNCS = 0.12 * TDEP
C
      IF (K .EQ. ITP) GO TO 220
      IF (K .NE. ITP) GO TO 230
C
  220 H2OLAY = WG(J,1)                                 ! ground layer
      PH2O = 1.38E-24 * H2OLAY * T(J,ITP)
      GO TO 240
C
  230 H2OLAY = W(J,1,K)                                ! other layers
      PH2O = 4.712E-23 * H2OLAY / ALOG(P(J,K+1)/P(J,K))
C
  240 ABSOR = (CCONT * (PH2O*TDEP + CNCS*(PBAR-PH2O))/1013.) * H2OLAY
      CAY(J,K) = CAY(J,K) + ABSOR
C
C
```

```
C        8-14 MICRON H2O CONTINUUM
C        -------------------------
C
  300    IF (KEEPSW(6) .EQ. 0) GO TO 400
         IF (V .GE. 700. .AND. V .LE. 1250.) GO TO 310    ! range of effect
         GO TO 400                                        ! (8-14 microns)
C
  310    TDEP = EXP (6.08*(296.0/TBAR-1.0))
         CNCS = 0.002
         CCONT = (4.18 + 5578.*EXP(-7.87E-3*V))/3.34E+22
C
         IF (K .EQ. ITP)  GO TO 320
         IF (K .NE. ITP)  GO TO 330
C
  320    H2OLAY = WG(J,1)                                 ! ground layer
         PH2O = 1.38E-24 * H2OLAY * T(J,ITP)
         GO TO 340
C
  330    H2OLAY = W(J,1,K)                                ! other layers
         PH2O = 4.712E-23 * H2OLAY / ALOG(P(J,K+1)/P(J,K))
C
  340    ABSOR = (CCONT * (PH2O*TDEP + CNCS*(PBAR-PH2O))/1013.) * H2OLAY
         CAY(J,K) = CAY(J,K) + ABSOR
C
C
C        NITROGEN CONTINUUM
C        ------------------
C
  400    IF (KEEPSW(7) .EQ. 0)  GO TO 17
         IF (V .GE. 2000. .AND. V .LE. 2800.)  GO TO 410   ! range of effect
         GO TO 17
C
  410    DO 413 I=1,90
C        *************
         IF (V .GE. FN2(I) .AND. V .LE. FN2(I+1))  GO TO 412
         GO TO 413
C
  412    DELN2 = (CN2(I+1)-CN2(I))/(FN2(I+1)-FN2(I))
         CCONT = DELN2*(V-FN2(I)) + CN2(I)
         GO TO 414
C
  413    CONTINUE
C        ********
C
  414    IF (K .EQ. ITP)  GO TO 415
C
         DELP = P(J,K+1) - P(J,K)
         TN2 = 0.781 * CCONT * (PBAR/1013.) * 29.24 * TBAR * (DELP/1013.)
         GO TO 416
C
  415    TN2 = 0.781 * CCONT * (PBAR/1013.)**2 * 1.000*296./T(J,ITP)
C
  416    CAY(J,K) = CAY(J,K) + TN2
C
   17    CONTINUE                             ! end of loop over atm. layers
C        **************                         ****************************
C
C
C
```

```
C     COMPUTE AEROSOL ABSORPTION AND SCATTERING COEFFICIENTS
C     ////////////////////////////////////////////////////////
C
         IF (KEEPSW(8) .EQ. 0 .AND. KEEPSW(9) .EQ. 0)   RETURN
C
C
C        TEST WHETHER V LIES WITHIN RANGE OF AEROSOL MODEL
C        ----------------------------------------------------
         IF (V .LT. FA(1,1)  .OR.
     *   V .GT. FA(1,61))        RETURN          ! v out of range of table
C
         DO 19 J=1,60
         IF (V .GE. FA(1,J)  .AND.  V .LE. FA(1,J+1))    GO TO 20
   19    CONTINUE
C
         RETURN                                  ! v above range of table
C
   20    FAC=(V-FA(1,J))/(FA(1,J+1)-FA(1,J))     ! interpolating factor
C                                                  (for v not in table)
C
C        IF IT DOES, COMPUTE THE COEFFICIENTS
C        ------------------------------------
C
         DO 32   K=ITLAYR,IBLAYR                 ! Loop over atm. layers
C        ***********************                   *********************
C
C        Compute Scaling Factors for Clear and Hazy Aerosol Models
C        .........................................................
C
C               HZ1 = vertical scaling factor for clear aerosol model
C               HZ2 =    "         "       "     "  hazy      "      "
C
C                           HZ1 and HZ2 are given in the block data routine,
C                           for each of the 16 atmospheric layers;
C                           EVH1 is just a local variable for this loop.
C
         IF (K .EQ. ITP)   GO TO 24
         IF (HZINT(K) .EQ. HZINT(K+1))   GO TO 21
C
         HA1=1.0/ALOG(HZINT(K+1)/HZINT(K))
         EVH1=HA1*(HZINT(K+1)-HZINT(K))               ! effective value
         GO TO 25
C
   21    EVH1=HZINT(K)
         GO TO 25
C
   24    EVH1=HZINT(ITP)
C
C        Aerosol model to be used depends on the height Z
C        ................................................
C                     L = aerosol model index
C
   25    IF (Z(K) .LT. 9.0 .AND. Z(K) .GE. 2.0)   GO TO 28
         IF (Z(K) .LE. 2.0)   GO TO 29
C
C        Above 9 km: Use background stratospheric (clear)
C                    or aged volcanic (hazy)
C
         IF (IARMDS .EQ. 5) L=5
         IF (IARMDS .EQ. 6) L=6
         GO TO 30
```

```
C         Between 2 and 9 km, use tropospheric model
C
   28     L = 2
          GO TO 30
C
C         Below 2 km, use rural, tropospheric, maritime or urban model
C
   29     L = IARMDG
C
C
C         Compute aerosol extinction coefficients for this layer
C         .....................................................
C
C                     AK = aerosol absorption coeff. at v
C                     AS = aerosol scattering coeff. at v
C
C                     (These are local variables, for this loop.)
C
   30     AK = AAB(L,J) + FAC*(AAB(L,J+1)-AAB(L,J))
          AS = ASC(L,J) + FAC*(ASC(L,J+1)-ASC(L,J))
C
C
C         Multiply these coefficients by the proper vertical scaling
C         factors and store them in the proper arrays in COMMON
C         ....................................................
C
C              TA2(atm. layer) - aer. abs. coeff.
C              HAS( "      "  ) - aer. scatt. coeff.
C
   31     TA2(K) = EVH1*AK
          HAS(K) = EVH1*AS
C
          IF (KEEPSW(8) .EQ. 0)  TA2(K) = 0.
          IF (KEEPSW(9) .EQ. 0)  HAS(K) = 0.
C
   32     CONTINUE
C         *************
C
          RETURN
          END



      BLOCK DATA
C
      IMPLICIT   INTEGER*2 (I-N)
C
      COMMON/BLK1/   DNU(16),CHI(16),HZ1(16),HZ2(16),ITP,JT,KSAM
C
      DATA   DNU(1),DNU(2),DNU(3),DNU(4),DNU(5),DNU(6),DNU(7),DNU(8),
     .       DNU(9),DNU(10),DNU(11),DNU(12),DNU(13),DNU(14),DNU(15),
     .       DNU(16)/0.0,.5,.6,.7,.8,.9,1.0,1.2,1.5,2.0,2.5,3.0,5.0,8.0,
     .              10.0,15.0/
C
      DATA   CHI(1),CHI(2),CHI(3),CHI(4),CHI(5),CHI(6),CHI(7),CHI(8),
     .       CHI(9),CHI(10),CHI(11),CHI(12),CHI(13),CHI(14),CHI(15),
     .       CHI(16)/1.00,1.00,.96,.89,.82,.77,.70,.60,.50,.41,.34,.31,
     .              .29,.23,.19,0.00/
```

```
      DATA    HZ1(1),HZ1(2),HZ1(3),HZ1(4),HZ1(5),HZ1(6),HZ1(7),HZ1(8),
     .        HZ1(9),HZ1(10),HZ1(11),HZ1(12),HZ1(13),HZ1(14),HZ1(15),HZ1(16)
     .        /6.43E-04,6.45E-04,6.22E-04,6.63E-04,7.14E-04,
     .        7.87E-04,9.80E-04,1.41E-03,2.30E-03,3.54E-03,
     .        4.85E-03,6.43E-03,8.19E-03,9.70E-03,2.85E-02,6.95E-02/
C
      DATA    HZ2(1),HZ2(2),HZ2(3),HZ2(4),HZ2(5),HZ2(6),HZ2(7),HZ2(8),HZ2(9),
     .        HZ2(10),HZ2(11),HZ2(12),HZ2(13),HZ2(14),HZ2(15),HZ2(16)
     .        /2.92E-03,2.89E-03,2.80E-03,2.45E-03,2.11E-03,
     .        1.85E-03,1.81E-03,3.36E-03,6.22E-03,7.71E-03,
     .        9.30E-03,1.85E-02,3.46E-02,6.21E-02,7.57E-01,7.57E-01/
C
      DATA    ITP,JT,KSAM /16,16,3/
C
      END




        SUBROUTINE LAYNUM (Z, LAYNO, DELHT)
C
C     This subroutine accepts the altitude Z (above sea-level) in km;
C   it returns the atmospheric layer number (layno) and the height
C   (in km) above the bottom of the layer (delht).
C
C     The layers are numbered as follows:
C
C             HEIGHT Z                          LAYER NUMBER
C            **********                        **************
C
C          14 < Z <= 15 km                           1
C          13 < Z <= 14                              2
C               .                                    .
C               .                                    .
C               .                                    .
C           0 < Z <= 1                               15
C               Z = 0                                16
C
C***********************************************************************************
C
        IF  (Z .GT. 15.) Z = 15.
C
        IZ = INT(Z)                     ! height of bottom of layer
        DELHT = Z - FLOAT(IZ)           ! height above bottom of layer
C
        LEVELZ = IZ + 1                 ! layer number, numbering from ground up
        IF (DELHT .EQ. 0.)  LEVELZ = IZ
C
        LAYNO = 16 - LEVELZ             ! layer number, numbering from top down
C
        RETURN
        END
```

```
      FUNCTION AERSOL(LAYNUM)
C
C       This subroutine accepts the atmosphere layer number and returns
C   the total aerosol extinction coefficient for that layer; the
C   extinction coefficient is the sum of the aerosol absorption and
C   scattering coefficients. The layers are defined as follows:
C
C     Height (above sea-level)       Layer Number          Region
C    **************************     **************        ********
C
C     14  <  height  <=  15 km           1            :
C     13  <  height  <=  14              2            :
C               .                        .            : Stratosphere
C               .                        .            :
C               .                        .            :
C      9  <  height  <=  10              6            :
C
C
C      8  <  height  <=   9              7            :
C               .                        .            :
C               .                        .            : Troposphere
C               .                        .            :
C      2  <  height  <=   3             13            :
C
C
C      1  <  height  <=   2             14            :
C      0  <  height  <=   1             15            : Boundary layer
C            height  =    0             16            :
C
C      The aerosol models to be used for the various regions are specified
C    by the integer variables iarmdg, iarmdt and iarmds; these variables
C    are defined interactively by the user and are stored in the common
C    block RSETUP.
C**********************************************************************************
C
      REAL*4      lownu
      DIMENSION   keepsw(9)
C
      COMMON      z(16),p(3,16),t(3,16),w(3,3,16),cay(3,16),wg(3,3),
     *            ta2(16),fa(6,61),asc(6,61),aab(6,61),
                  has(16)
C
      COMMON/RSETUP/  bound,lownu,highnu,delnu,v,
     *                iatmod,iarmdg,iarmdt,iarmds,
     *                numpts,nlines,keepsw,iblayr,itlayr,
     *                irsize, HZINT(16),iaermd
C
      IF (V .LT. FA(1,1)  .OR.
     *    V .GT. FA(1,61))     go to 400    ! out of aerosol model range
      aersol = ta2(laynum) + has(laynum)
      return
C
  400 aersol = 0.
      return
C
      end
```

AUXILIARY ROUTINES AND COMMAND FILES

```
      PROGRAM LNEDIT
C
C        This program reads lines one-by-one from an ''original'' line file
C     (AFGL format), retains only water, carbon dioxide and ozone lines (all
C     isotopes), strips away the quantum identification numbers and writes
C     the processed lines out to a ''master'' line file for use by the
C     MICTRA routine.
C
C        The disk containing this program may be run from any drive;
C     the drives and filenames for the input and output files will be
C     typed in by the user upon prompting by the program, which will halt
C     to allow mounting of the input and output disks.
C ***************************************************************************
C
C                          SPECIFICATION STATEMENTS
C                          ************************
C
      BYTE      qntmid, outfil, infil, reply, go
      DIMENSION qntmid(35), outfil(14), infil(14)
C
C ***************************************************************************
C
C    SPECIFY WAVENUMBER RANGE TO BE INCLUDED IN OUTPUT FILE
C    //////////////////////////////////////////////////////
C
      type 1
    1 format(//t2,'Specify wavenumber range to be included',
     *        t42,'in output file:'/
     *        t10,'type minimum wavenumber: ',$)
      accept 2, wnmin
    2 format(f10.4)
      type 3
    3 format(t10,'type maximum wavenumber: ',$)
      accept 2, wnmax
C
C   OPEN THE OUTPUT LINE FILE (for MICTRA; UNformatted, seq. access)
C   ////////////////////////////////////////////////////////////////
C
      type 10
   10 format(//t2,'Type name of output file in DL0:'/
     *        t2,'(i.e., the master file, unformatted,'/
     *        t2,' to be used by the MICTRA routine): ',$)
      accept 11, outfil
   11 format(14A1)
C
C     type 12
C  12 format(//t2,'Mount disk containing output file,'/
C    *        t2,'then type ''G'' for ''GO'': ',$)
C     accept 13, go
C  13 format(A1)
C
      OPEN (unit=8, name=outfil, type='NEW',
     *     disp='KEEP', form='UNFORMATTED', initialsize=974,
     *     recordsize=6)
C
```

```
C     OPEN THE INPUT LINE FILE (formatted, sequential access)
C     ////////////////////////////////////////////////////////////
C
C        This file should be one of the AFGL files;
C        specify the file name:
C
  100    type 120
C ***
  120    format(//t2,'Type name of input file'/
     *              t2,'(one of the AFGL files): ',$)
         accept 11, infil
C
         type 126
  126    format(//t2,'Mount disk containing input file,'/
     *              t2,'then type ''G'' for ''GO'': ',$)
         accept 121, go
  121    format(A1)
C
         OPEN (unit=4, name=infil, type='OLD',
     *         disp='KEEP', form='FORMATTED')
C
C     OPEN TEMPORARY VM FILE
C     ///////////////////////
C
  199    OPEN (unit=3, name='VM:TEMP.DAT', type='NEW',
     *         disp='DELETE', form='UNFORMATTED',
     *         initialsize=300, recordsize=6)
C
C   LOOP - READ LINES ONE AT A TIME, PROCESS AND WRITE TO OUTPUT FILE
C          (via temporary VM file, in groups of 6000 lines)
C   ////////////////////////////////////////////////////////////////
C
C       Read a line
C       ------------
C
         linect = 0
C
  200    READ(4,230,end=300)  wavnum, strnth, hwidth, energy,
     *                        qntmid, idate,                   ! strip these
     *                        isotop, molcul
C
  230    FORMAT(f10.4, e10.3, f5.4, f10.3, 35a1, i3, i4, i3)
C
C      Reject this line if it is below the minimum wavenumber
C      or if it is other than H2O, CO2 or O3
C      -------------------------------------------------------
C
         IF (wavnum .LT. wnmin)  go to 200
         IF (molcul .GT. 3)      go to 200
         linect = linect + 1
C
C     Write stripped-down line into temporary VM file (unformatted)
C     -------------------------------------------------------------
          WRITE(3) wavnum, strnth, hwidth, energy, isotop, molcul
C
C     Branch to appropriate code for given ''end'' condition
C     ------------------------------------------------------
C
          IF (wavnum .GT. wnmax)  go to 400
          IF (linect .GT. 6000)   go to 500
          go to 200                    ! read another line
```

```
C      End of input file reached
C      -------------------------
C
  300       wnlast = wavnum
            IF (linect .EQ. 0) go to 320
C
            rewind 3
            DO 310 I=1,linect
            READ (3) wavnum, strnth, hwidth, energy, isotop, molcul
            WRITE(8) wavnum, strnth, hwidth, energy, isotop, molcul
  310       continue
C
  320       CLOSE (unit=3, disp='DELETE')
            CLOSE (unit=4, disp='KEEP')
C
            type 330, wnlast,wnmax
  330       format(//t2,'Last wavenumber in this input file is:      ',F10.4/
     *             t2,'Max. wavenumber desired for output file is: ',F10.4/
     *             t2,'Is the output file complete? (Y or N): ',$)
            accept 331, reply
  331       format(A1)
            IF (reply .EQ. 'N') go to 100
            CLOSE (unit=8, disp='KEEP')
            STOP
C
C      Maximum wavenumber reached
C      --------------------------
  400       REWIND 3
            DO 410 I=1,linect
            READ (3) wavnum, strnth, hwidth, energy, isotop, molcul
            WRITE(8) wavnum, strnth, hwidth, energy, isotop, molcul
  410       continue
C
            CLOSE (unit=3, disp='DELETE')
            CLOSE (unit=4, disp='KEEP')
            CLOSE (unit=8, disp='KEEP')
            STOP
C
C      Temporary (VM) file is filled
C      -----------------------------
C
  500       REWIND 3
            DO 510 I=1,linect
            READ (3) wavnum, strnth, hwidth, energy, isotop, molcul
            WRITE(8) wavnum, strnth, hwidth, energy, isotop, molcul
  510       continue
C
            CLOSE (unit=3, disp='DELETE')
            GO TO 199
C
            END
```

```
      PROGRAM PLOTSP
C
C     This routine plots the spectra computed by MICTRA.
C
C     It automatically plots on an angstrom scale up to 10,000
C     angstroms, and on a micrometer scale for wavelengths greater
C     than 10,000 angstroms.
C
C     The user has the option of overriding this default to plot
C     on an angstrom, micrometer, hertz or wavenumber scale, with
C     the scale increasing either to the left or right.
C
C     The routine handles the total range of the Air Force line tapes,
C     from 2,000 angstroms to 3 cm.
C
C************************************************************************************
C
      BYTE        SPCFL1(15),GO,MAPTYP(10),BPARAM(10),BOTTOM(2),
     •            LONG,BRIGHT,ARCLIN,LINTYP(2),SHORT,LEFT(2),
     *            NORMAL(2),YES,NO,PLOT,COMPAR,STOPP,LHEDNG(21),
     •            BHDNGA(23),ZERO(4),TEN(4),TWENTY(4),THIRTY(4),
     *            FORTY(4),FIFTY(4),SIXTY(4),SEVNTY(4),EIGHTY(4),
     •            NINETY(4),HUNDRD(4),MAPTPE(10),STDWIN(2),AXIS,
     *            BASE,USER,LFTNUM(9),RGTNUM(9),BHDNGM(25),
     •            WLUNIT(12),UNTDES(41),ANGST(11),MICRON(11),
     *            REPLY,SCALE,SENSE,BHDNGH(18),BHDNGW(24),
     •            BEGNUM(12),ENDNUM(12),HERTZ(11),WAVENM(11),
     *            A,M,R, FTBTCK,FTETCK,F,E
C
      INTEGER*2   IPARAM(10),PARAM(10),FNX,FNY,FNSPCE,IILNY(11),
     *            HPARAM(10)
C
      REAL*4      LOWERW,RPARAM(10),USRWIN(4),SCRWIN(4),LOWERT,
     *            LOWVAL,LOWW
C
      COMMON/CMAP/  MAPTPE,
     *              UXMIN,UXMAX,UYMIN,UYMAX,
     *              TXMIN,TXMAX,TYMIN,TYMAX,
     *              STDWIN,
     *              SXMIN,SXMAX,SYMIN,SYMAX,
     *              AXIS,BASE
C
      COMMON/COORDS/  UX,UY, TX,TY, SX,SY
C
      DATA  SPCFL1/15*''000/,MAPTYP/'L','I','N',7*' '/,
     *      BPARAM/'D',9*' '/,BOTTOM/'A','B'/,LONG/'L'/,BRIGHT/'B'/,
     *      ARCLIN/'L'/,LINTYP/'U','S'/,SHORT/'S'/,PARAM/2,5,8*0/,
     •      LEFT/'O','L'/,NORMAL/'N','R'/,YES/'Y'/,NO/'N'/,PLOT/'P'/,
     *      COMPAR/'C'/,STOPP/'S'/,
     *      LHEDNG/'P','E','R',' ','C','E','N','T',' ',
     *           'T','R','A','N','S','M','I','S','S','I','O','N'/,
     *      BHDNGA/'W','A','V','E','L','E','N','G','T','H',' ',
     *           'I','N',' ','A','N','G','S','T','R','O','M','S'/,
     *      BHDNGM/'W','A','V','E','L','E','N','G','T','H',' ',
     *           'I','N',' ','M','I','C','R','O','M','E','T',
     •           'E','R','S'/,
     •      BHDNGH/'F','R','E','Q','U','E','N','C','Y',' ',
     •           'I','N',' ','H','E','R','T','Z'/,
     *      BHDNGW/'F','R','E','Q','U','E','N','C','Y',' ','I','N',
     *           ' ','W','A','V','E','N','U','M','B','E','R','S'/,
     *      ZERO/' ',' ',' ','0'/,TEN/' ',' ','1','0'/,
```

```
     •        TWENTY/' ',' ','2','0'/,THIRTY/' ',' ','3','0'/,
     *        FORTY/' ',' ','4','0'/,FIFTY/' ',' ','5','0'/,
     *        SIXTY/' ',' ','6','0'/,SEVNTY/' ',' ','7','0'/,
     *        EIGHTY/' ',' ','8','0'/,NINETY/' ',' ','9','0'/,
     *        HUNDRD/' ','1','0','0'/,USER/'U'/,LCPLT/'L'/,
     *        CONPLT/'C'/,GODARD/'G'/,MTHOPK/'H'/,CLEAR/'C'/,
     *        HAZY/'H'/,HPARAM/1,6,8*0/,
     *        LFTNUM/9*' '/, RGTNUM/9*' '/,
     *        BEGNUM/12*' '/,ENDNUM/12*' '/,
     *        UNTDES/'l','a','r','g','e',' ','d','i','v','i','s','i',
     *             'o','n',' ','=',' ',12*' ',' ',11*' '/,
     *        ANGST/'a','n','g','s','t','r','o','m','s',' ',' '/,
     •        MICRON/'m','i','c','r','o','m','e','t','e','r','s'/,
     *        HERTZ/'H','E','R','T','Z',6*' '/,
     *        WAVENM/'W','A','V','E','N','U','M','B','E','R','S'/,
     *        A/'A'/, M/'M'/, R/'R'/, F/'F'/, E/'E'/
C
      DATA  C/2.997925E10/
C
C*******************************************************************************************
C
C     SET UP PLOT SYSTEM
C     ******************
      CALL BEGPLT

C
C     INSTRUCT USER HOW TO USE THIS ROUTINE
C     *************************************
      CALL TYP100
      TYPE 10
   10 FORMAT(//T5,'The following options are provided:'//
     *       t15,'Type P to plot a spectrum;'//
     *       t15,'Type S to stop this program.'////
     *       t5,'FIRST type G,RETURN to remove this message.'/
     *       t5,'*****'//)
      CALL ECHO(YES)
      ACCEPT 11, GO
   11 FORMAT(A1)
      CALL CLR100
      CALL ECHO(NO)
      CALL ALPMOD(1)

C
C     USE ITTINR TO RECEIVE USER'S INSTRUCTIONS
C     *****************************************
 1000 ICHAR = ITTINR()
C
      IF (ICHAR .EQ. PLOT)     GO TO 2000
      IF (ICHAR .EQ. STOPP)    GO TO 4000
      GO TO 1000

C
C     PLOT MODE
C     *********
 2000 CALL TYP100
      CALL ECHO(YES)
      TYPE 2010
 2010 FORMAT(//T5,'Spectrum file? ',$)
      ACCEPT 2011, SPCFL1
 2011 FORMAT(15A1)
      TYPE 2020
 2020 FORMAT(//T5,'Mount file; when ready, type G: ',$)
      ACCEPT 2021, GO
 2021 FORMAT(A1)
```

```
C       OPEN THE SPECTRUM FILE; READ THE FIRST THREE RECORDS
C       ****************************************************
C
        OPEN (UNIT=1, NAME=SPCFL1, TYPE='OLD', DISP='KEEP',
     *        FORM='UNFORMATTED', RECORDSIZE=2)
C
        READ(1)  PLTCOD, ANGMIC              ! line-center or continuous plot,
C                                              in angstroms or microns
        READ(1)  LOWERW, UPPERW              ! bounding wavelengths
C                                              (in angstroms or microns)
        READ(1)  ATL, ATU                    ! bounding aerosol+continuum
C                                              transmission
C
C       MORE USER'S INSTRUCTIONS
C       ************************
C
        TYPE 2022
 2022   FORMAT(////T2,'The spectrum may be plotted on an angstrom,'/
     •            t2,'micrometer, hertz or wavenumber scale,'/
     *            t2,'increasing either to the right or left;'/)
C
        IF (ANGMIC .EQ. 0.)  TYPE 2023
        IF (ANGMIC .EQ. 1.)  TYPE 2024
C
 2023   FORMAT(/T2,'this spectrum is set up to be plotted on an'/
     •         t2,'angstrom scale, increasing toward the right;'/
     *         t2,'do you want to change this? (Y or N): ',$)
 2024   FORMAT(/T2,'this spectrum is set up to be plotted on a'/
     *         t2,'micrometer scale, increasing toward the right;'/
     *         t2,'do you want to change this? (Y or N): ',$)
C
        ACCEPT 2025, REPLY
 2025   FORMAT(A1)
C
        IF (REPLY .EQ. 'Y') GO TO 2026
        CALL CLR100
        CALL ECHO(NO)
        CALL ALPMOD(1)
        IF (ANGMIC .EQ. 0.) SCALE = A
        IF (ANGMIC .EQ. 1.) SCALE = M
        SENSE = R
        GO TO 2032
C
 2026   TYPE 2027
 2027   FORMAT(//T2,'Angstrom, micrometer, hertz or wavenumber scale?'/
     •          t2,'(A,M,H or W): ',$)
        ACCEPT 2025, SCALE
C
        TYPE 2028
 2028   FORMAT(/T2,'Increasing toward right or left?'/
     *         t2,'(R or L): ',$)
        ACCEPT 2025, SENSE
C
        CALL CLR100
        CALL ECHO(NO)
        CALL ALPMOD(1)
C
```

```
C                                                                      ! unit con-
C                                                                        version
C                                                                        factor
 2032   IF (ANGMIC .EQ. 0.   .AND. SCALE .EQ. 'A')  CNVFAC = 1.
        IF (ANGMIC .EQ. 0.   .AND. SCALE .EQ. 'M')  CNVFAC = 1.E-4
        IF (ANGMIC .EQ. 0.   .AND. SCALE .EQ. 'H')  CNVFAC = -1.E8*C
        IF (ANGMIC .EQ. 0    AND. SCALE .EQ. 'W')   CNVFAC = -1.E8
C
        IF (ANGMIC .EQ. 1    AND. SCALE .EQ. 'A')   CNVFAC = 1.E4
        IF (ANGMIC .EQ. 1    AND. SCALE .EQ. 'M')   CNVFAC = 1.
        IF (ANGMIC .EQ. 1    AND. SCALE .EQ. 'H')   CNVFAC = -1.E4*C
        IF (ANGMIC .EQ. 1    AND. SCALE .EQ. 'W')   CNVFAC = -1.E4
C
        IF (CNVFAC .GT. 0.) GO TO 2029                     ! convert wave-
        IF (CNVFAC .LT. 0.) GO TO 2030                     ! length bounds
C
 2029   LOWERW = CNVFAC * LOWERW
        UPPERW = CNVFAC * UPPERW
        GO TO 2031
C
 2030   LOWERW = -CNVFAC/LOWERW
        UPPERW = -CNVFAC/UPPERW
C
 2031   IF (LOWERW .LT. UPPERW)  GO TO 2035
C
        TEMP   = LOWERW                                    ! interchange if
        LOWERW = UPPERW                                    ! necessary to
        UPPERW = TEMP                                      ! keep order
C
C
C       PLOT THE SPECTRUM (on the file just opened)
C       *******************************************
C
C       Compute appropriate wavelength or frequency unit
C       to use on plot and number of units required to span plot
C       --------------------------------------------------------
 2035   WRANGE = UPPERW - LOWERW
        ORDMAG = ALOG10 (WRANGE)
        IF (ORDMAG .GE. 0.) WUNIT = 10.**(AINT(ORDMAG))
        IF (ORDMAG .LT. 0.) WUNIT = 10.**(AINT(ORDMAG)-1.)
C
        UPPERW = UPPERW/WUNIT                        ! upper and lower wave-
        LOWERW = LOWERW/WUNIT                        ! lengths in wunits
C
        UPPVAL = UPPERW
        LOWVAL = LOWERW
        DIFF = UPPVAL - AINT(UPPVAL)
        UPPVAL = AINT(UPPVAL)
        IF (DIFF .GT. 0.) UPPVAL = UPPVAL + 1.
        LOWVAL = AINT(LOWVAL)
C
C        IUPVAL = INT(UPPVAL)
C        ILWVAL = INT(LOWVAL)
C
C       Set up the mapping
C       ------------------
        USRWIN(1) = LOWERW                           ! define user window
        USRWIN(2) = UPPERW
        USRWIN(3) =   0.
        USRWIN(4) = 100.
```

```
      SCRWIN(1) = 150.                          ! define screen window
      SCRWIN(2) = 1000.
      IF (SENSE .EQ. 'L')  SCRWIN(1) = 1000.    ! scale increasing to left
      IF (SENSE .EQ. 'L')  SCRWIN(2) =  150.
      SCRWIN(3) = 170.
      SCRWIN(4) = 770.
C
      CALL SETMAP (MAPTYP, BPARAM,IPARAM,RPARAM, USRWIN,SCRWIN)
C
C     Draw a box around the screen window
C     -----------------------------------
      CALL BOXWIN
C
C     Draw tick marks and grid lines
C     ------------------------------
      BTICK = 0.
      ANGSTR = LOWVAL- 1.                      ! draw long vertical ticks
 2040 ANGSTR = ANGSTR + 1.                     ! and grid lines every
      IF (ANGSTR .GT. UPPERW) GO TO 2045       ! wunit
      IF (ANGSTR .LT. LOWERW) GO TO 2040
      CALL TICK (BOTTOM,LONG,ANGSTR,BRIGHT)
      IF (BTICK .EQ. 0.) BTICK = ANGSTR
      ETICK = ANGSTR
      IF (ANGSTR .NE. LOWERW .AND. ANGSTR .NE. UPPERW)
     * CALL DRWLIN (ANGSTR,0.,ANGSTR,100.,ARCLIN,BRIGHT,LINTYP,PARAM)
      GO TO 2040
C
 2045 ISWETK = 1
      IF (ETICK .EQ. BTICK) ISWETK = 0         ! eliminate unneeded end tick
C
 2050 ANGSTR = LOWVAL - 0.1                    ! draw short vertical ticks
 2051 ANGSTR = ANGSTR + 0.1                    ! every 1/10 th wunit
      IF (ANGSTR .GT. UPPERW) GO TO 2060
      IF (AMOD(ANGSTR,1.) .EQ. 0.)  GO TO 2051
      IF (ANGSTR .GE. LOWERW) CALL TICK (BOTTOM,SHORT,ANGSTR,BRIGHT)
      GO TO 2051
C
 2060 TRANS = -10.                             ! draw long hor. ticks and
 2061 TRANS = TRANS + 10.                      ! grid lines every 10%
      IF (TRANS .GT. 100.)  GO TO 2070
C      CALL TICK (LEFT,LONG,TRANS,BRIGHT)
      IF (TRANS .GT. 0.  .AND.
     *    TRANS .LT. 100.)
     *    CALL DRWLIN (USRWIN(1),TRANS,USRWIN(2),
     •                 TRANS,ARCLIN,BRIGHT,LINTYP,PARAM)
      GO TO 2061
C
 2070 TRANS = TRANS - 1.                       ! draw short hor. ticks
 2071 TRANS = TRANS + 1.                       ! every 1%
      IF (TRANS .GT. 100.)  GO TO 2200
      IF (AMOD(TRANS,10.) .EQ. 0.)  GO TO 2071
C      CALL TICK (LEFT,SHORT,TRANS,BRIGHT)
      GO TO 2071
C
```

```
C       Plot the spectrum
C       -----------------
 2200   IF (PLTCOD .EQ. 1.)  GO TO 2300
        IF (PLTCOD .EQ. 0.)  GO TO 2400
C
C       The aerosol+continuum extinction is given precisely enough
C       (over a small wavelength band) by a straight line,
C       having the general form:
C
C         AT = AT1 + (AT2-AT1)/(W2-W1) * (W-W1) ,
C
C       where  AT = aerosol+continuum transmission in %
C       and    W  = wavelength in angstroms or microns
C
C       (W1,AT1) and (W2,AT2) are the two endpoints of the range.
C
 2300   W1 = LOWERW                                      ! LINE-CENTER PLOT
        W2 = UPPERW                                      ! (overview)
C
        AT1 = ATL*100.
        AT2 = ATU*100.
C
        LOWERT = AT1 + (AT2-AT1)/(W2-W1) * (LOWERW-W1)   ! draw aer/con line
        UPPERT = AT1 + (AT2-AT1)/(W2-W1) • (UPPERW-W1)
        CALL DRWLIN (LOWERW,LOWERT, UPPERW,UPPERT,
     •             ARCLIN,BRIGHT,NORMAL,PARAM)
C
        WIDTH = UPPERW - LOWERW                 ! draw hatched area above line
        HATCHS = WIDTH/50.
C
        W = LOWERW - WIDTH
        DO 2340 I=1,100
        W = W + HATCHS
        AT = AT1 + (AT2-AT1)/(W2-W1) • (W-W1)
        CALL DRWLIN (W,AT, W+WIDTH,AT+100.,
     *             ARCLIN,BRIGHT,LINTYP,HPARAM)
 2340   CONTINUE
C
        W = UPPERW + WIDTH
        DO 2350   I=1,100
        W = W - HATCHS
        AT = AT1 + (AT2-AT1)/(W2-W1) * (W-W1)
        CALL DRWLIN (W,AT, W-WIDTH,AT+100.,
     *             ARCLIN,BRIGHT,LINTYP,HPARAM)
 2350   CONTINUE
C
        2360 READ (1,END=2100)  WAVLEN, TRANSM        ! draw absorption lines
        IF (CNVFAC .GT. 0.)  WAVLEN = CNVFAC * WAVLEN
        IF (CNVFAC .LT. 0.)  WAVLEN = -CNVFAC/WAVLEN
        WAVLEN = WAVLEN/WUNIT
        IF (WAVLEN .GT. UPPERW)  GO TO 2360
        IF (WAVLEN .LT. LOWERW)  GO TO 2100
        AT = AT1 + (AT2-AT1)/(W2-W1)  * (WAVLEN-W1)
        CALL DRWLIN (WAVLEN,AT, WAVLEN,TRANSM*100.,
     •             ARCLIN,BRIGHT,NORMAL,PARAM)
        GO TO 2360
C
```

```
 2400   IPTCT = 1                                               ! CONTINUOUS PLOT
 2410   READ (1,END=2100)  WAVLEN, TRANSM
        IF (CNVFAC .GT. 0.)  WAVLEN = CNVFAC * WAVLEN
        IF (CNVFAC .LT. 0.)  WAVLEN = -CNVFAC/WAVLEN
        WAVLEN = WAVLEN/WUNIT
        IF (IPTCT .EQ. 1)  GO TO 2420
        CALL DRWLIN (WAVLNP,TRANSP*100., WAVLEN,TRANSM*100.,
     *               ARCLIN,BRIGHT,NORMAL,PARAM)
 2420   WAVLNP = WAVLEN
        TRANSP = TRANSM
        IPTCT = IPTCT + 1
        GO TO 2410
C
C       Label the plot
C       --------------
C2100    CALL MOVSEQ (LHEDNG,21,2, 1,700,-90.,20,              ! left heading
C    *                FNX,FNY,FNANGL,FNSPCE)
C        PRINT 2110,  FNX,FNY,FNSPCE
C2110    FORMAT(/T10,'Left heading:'//
C    *           t15,'X: ',t20,I4//
C    *           t15,'Y: ',t20,I4//
C    *           t15,'spacing:',t25,I3///)
 2100   CALL DISPSQ (LHEDNG,21,BRIGHT,2,
     *               42,742,-90.,33)
C
C       IF (ANGMIC .EQ. 0.)                                ! bottom heading
C    *  CALL MOVSEQ (BHDNGA,23,2, 1,10, 0.,20,
C    *               FNX,FNY,FNANGL,FNSPCE)
C       IF (ANGMIC .EQ. 1.)
C    *  CALL MOVSEQ (BHDNGM,25,2, 1,10, 0.,20,
C    *               FNX,FNY,FNANGL,FNSPCE)
C       PRINT 2120, FNX,FNY,FNSPCE
C2120   FORMAT(/T10,'Bottom heading:'//
C    *               t15,'X: ',t20,I4//
C    *               t15,'Y: ',t20,I4//
C    *               t15,'spacing:',t25,I3///)
        IF (SCALE .EQ. 'A') CALL DISPSQ (BHDNGA,23,BRIGHT,2,
    *                                    239,35,0.,28)
        IF (SCALE .EQ. 'M') CALL DISPSQ (BHDNGM,25,BRIGHT,2,
    *                                    239,35,0.,28)
        IF (SCALE .EQ. 'H') CALL DISPSQ (BHDNGH,18,BRIGHT,2,
    *                                    239,35,0.,28)
        IF (SCALE .EQ. 'W') CALL DISPSQ (BHDNGW,24,BRIGHT,2,
    *                                    239,35,0.,28)
C
C            Note: MOVSEQ and DISPSQ are two routines from the author's
C                  plot package, which he wrote to provide acceptable
C                  plotting capability for the DEC 11/23 microcomputer,
C                  with a DEC VT100 terminal enhanced with another
C                  manufacturer's board to give it better plotting
C                  capability. MOVSEQ allows the user to display any
C                  string of characters on the screen and move the entire
C                  string about at will, including changing the angle at
C                  which the string is displayed, by using the arrow keys
C                  and other keys on the terminal keyboard; when the
C                  desired position and orientation of the string have
C                  been found, MOVSEQ can be commanded to return those
C                  values, which may then be permanently coded into the
C                  plot routine. This gives the user a very easy way of
C                  labelling the plot.
```

```
C
C Comment out after get x-value:
C......................................
C        DSPLFT = (UPPERW-LOWERW)/20.                          ! left numbers
C        UX = LOWERW - DSPLFT
C        UY = 0.
C        CALL MAP(USER)
C        ILNX = INT(SX)
C        ILNY = INT(SY)
C
C        CALL MOVSEQ (ZERO,4,1, ILNX,ILNY, 0.,5,
C    *                FNX,FNY,FNANGL,FNSPCE)
C
C         PRINT 2130, FNX,FNY,FNSPCE
C2130     FORMAT(/T10,'Left numbers:'//
C     *           t15,'X: ',t20,I4//
C     •           t15,'Y: ',t20,I4//
C     *           t15,'spacing:',t25,I3///)
C...........................................
C
C         IILNX = printed x-value for 'left numbers'
          IILNX = 87
C         ISPNG = printed spacing for 'left numbers'
          ISPNG = 15
C
         DO 2150 I=1,11
         UX = LOWERW
         UY = 0. + FLOAT((I-1)*10)
         CALL MAP(USER)
         IIX = INT(SX)
         IILNY(I) = INT(SY)
 2150    CONTINUE
C
         CALL DISPSQ (ZERO,4,BRIGHT,1,IILNX,IILNY(1)-6,0.,ISPNG)
         CALL DISPSQ (TEN,4,BRIGHT,1,IILNX,IILNY(2)-6,0.,ISPNG)
         CALL DISPSQ (TWENTY,4,BRIGHT,1,IILNX,IILNY(3)-6,0.,ISPNG)
         CALL DISPSQ (THIRTY,4,BRIGHT,1,IILNX,IILNY(4)-6,0.,ISPNG)
         CALL DISPSQ (FORTY,4,BRIGHT,1,IILNX,IILNY(5)-6,0.,ISPNG)
         CALL DISPSQ (FIFTY,4,BRIGHT,1,IILNX,IILNY(6)-6,0.,ISPNG)
         CALL DISPSQ (SIXTY,4,BRIGHT,1,IILNX,IILNY(7)-6,0.,ISPNG)
         CALL DISPSQ (SEVNTY,4,BRIGHT,1,IILNX,IILNY(8)-6,0.,ISPNG)
         CALL DISPSQ (EIGHTY,4,BRIGHT,1,IILNX,IILNY(9)-6,0.,ISPNG)
         CALL DISPSQ (NINETY,4,BRIGHT,1,IILNX,IILNY(10)-6,0.,ISPNG)
         CALL DISPSQ (HUNDRD,4,BRIGHT,1,IILNX,IILNY(11)-6,0.,ISPNG)
C
C
         UX = BTICK                                        ! map bottom numbers
         UY = 0.                                           ! to screen
         CALL MAP(USER)
         IXBTCK = INT(SX)
         IYBTCK = INT(SY)
C
         UX = ETICK
         UY = 0.
         CALL MAP(USER)
         IXETCK = INT(SX)
         IYETCK = INT(SY)
C
```

```
      BTICK = BTICK * WUNIT                          ! encode bottom numbers
      ETICK = ETICK • WUNIT
C
      FTBTCK = E
      IF (BTICK .GE. 1.E-4  .AND.  BTICK .LE. 99999.)  FTBTCK = F
      FTETCK = E
      IF (ETICK .GE. 1.E-4  .AND.  ETICK .LE. 99999.)  FTETCK = F
C
      IF (FTBTCK .EQ. F)  ENCODE (12,2162,BEGNUM) BTICK
      IF (FTBTCK .EQ. E)  ENCODE (12,2161,BEGNUM) BTICK
      IF (FTETCK .EQ. F)  ENCODE (12,2162,ENDNUM) ETICK
      IF (FTETCK .EQ. E)  ENCODE (12,2161,ENDNUM) ETICK
 2161 FORMAT(E12.5)
 2162 FORMAT(F12.4)
C
      IF (SENSE .EQ. 'R')  GO TO 2500                ! adjust positions
      IF (SENSE .EQ. 'L')  GO TO 2600                ! of bottom no's
C
 2500 IF (FTBTCK .EQ. F)  IXBTCK = MAXO(IXBTCK-7*15,85)
      IF (FTBTCK .EQ. E)  IXBTCK = MAXO(IXBTCK-6*15,85)
      IF (FTETCK .EQ. F)  IXETCK = MINO(IXETCK-7*15,1024-12*15)
      IF (FTETCK .EQ. E)  IXETCK = MINO(IXETCK-6*15,1024-12*15)
      GO TO 2610
C
 2600 IF (FTETCK .EQ. F)  IXETCK = MAXO(IXETCK-7*15,85)
      IF (FTETCK .EQ. E)  IXETCK = MAXO(IXETCK-6*15,85)
      IF (FTBTCK .EQ. F)  IXBTCK = MINO(IXBTCK-7*15,1024-12*15)
      IF (FTBTCK .EQ. E)  IXBTCK = MINO(IXBTCK-6*15,1024-12*15)
C
 2610 CALL DISPSQ (BEGNUM,12,BRIGHT,1,IXBTCK,100,0.,15)   ! display bottom
      IF (ISWETK .EQ. 1)                                  ! numbers
     • CALL DISPSQ (ENDNUM,12,BRIGHT,1,IXETCK,100,0.,15)
C
      IF (WUNIT .LT. 1.E-4 .OR. WUNIT .GT. 99999.)        ! encode scale
     * ENCODE (12,2161,WLUNIT)  WUNIT                     ! unit
      IF (WUNIT .GE. 1.E-4  .AND.  WUNIT .LE. 99999.)
     * ENCODE (12,2162,WLUNIT)  WUNIT
C
      DO 2165 I=1,12
      UNTDES(I+17) = WLUNIT(I)
 2165 CONTINUE
C
      DO 2166 I=1,11
      IF (SCALE .EQ. 'A')  UNTDES(I+30) = ANGST(I)
      IF (SCALE .EQ. 'M')  UNTDES(I+30) = MICRON(I)
      IF (SCALE .EQ. 'H')  UNTDES(I+30) = HERTZ(I)
      IF (SCALE .EQ. 'W')  UNTDES(I+30) = WAVENM(I)
 2166 CONTINUE
C
      IF (SCALE .EQ. 'A') IXUNT = 250                   ! display scale
      IF (SCALE .EQ. 'M') IXUNT = 265
      IF (SCALE .EQ. 'H') IXUNT = 230
      IF (SCALE .EQ. 'W') IXUNT = 260
      CALL DISPSQ (UNTDES,41,BRIGHT,1,IXUNT,1,0.,15)
```

```
C       CLOSE THE SPECTRUM FILE, CLEAR THE SCREEN
C       AND TAKE NEXT INSTRUCTION AT USER'S CUE
C       ****************************************
        CLOSE (UNIT=1,DISP='KEEP')
        CALL ECHO(YES)
        ACCEPT 2180, GO
 2180   FORMAT(A1)
        CALL ECHO(NO)
        CALL CLR640
        GO TO 1000
C
C
C       STOP THE PROGRAM
C       ****************
 4000   CALL ENDPLT
        STOP
        END
```

```
!EDMICB.COM
!                    COMMAND FILE TO EDIT, DEVELOP AND DEBUG MICTRA
!                   ***********************************************
!
!  This command file directs the editing, compilation and linking of
!  the MICTRA routines. Any or all of the routines may be edited;
!  the edited routines are then compiled and linked with the unedited
!  routines to form an updated load module. The user is prompted to
!  mount any needed floppy disks, and housekeeping chores (deleting
!  unneeded files, squeezing disks, etc) are done automatically.
!
!            LOAD FLOPPIES: OPERATING   DISK IN DRIVE 0
!                           DEVELOPMENT DISK IN DRIVE 1
!           *********************************************
!                          (Scratch disk is SD3)
RUN COM:EDMICB.SV1             ! Tell user to save files
INIT/NOQ SD3:
RUN COM:EDMICB.SV2             ! Mount floppies in DY0,
                               DY1
!-----------------------------------------------------------
!               HOUSEKEEPING
!              **************
COPY/NOQ DY1:*.LST SD3:
DELETE/NOQ DY1:*.LST
SQUEEZE/NOQUERY DY1:
!-----------------------------------------------------------
!            EDIT AND RECOMPILE MICDRB
!           ***************************
EDIT DY1:MICDRB.FOR
COPY DY1:MICDRB.BAK SD3:MICDRB.BAK
DELETE/NOQUERY DY1:MICDRB.BAK
SQUEEZE/NOQUERY DY1:
FORT/OBJ:DY1:MICDRB.OBJ/CODE:THR/EXT/U:10/WA/LIST:SD3:MICDRB     DY1:MICDRB.FOR
!-----------------------------------------------------------
!            EDIT AND RECOMPILE SETUP
!           **************************
EDIT DY1:SETUP.FOR
COPY DY1:SETUP.BAK SD3:SETUP.BAK
DELETE/NOQUERY DY1:SETUP.BAK
SQUEEZE/NOQUERY DY1:
FORT/OBJ:DY1:SETUP.OBJ/CODE:THR/EXT/U:10/WA/LIST:SD3:SETUP       DY1:SETUP.FOR
!-----------------------------------------------------------
!            EDIT AND RECOMPILE RDTAPE
!           ***************************
EDIT DY1:RDTAPE.FOR
COPY DY1:RDTAPE.BAK SD3:RDTAPE.BAK
DELETE/NOQUERY DY1:RDTAPE.BAK
SQUEEZE/NOQUERY DY1:
FORT/OBJ:DY1:RDTAPE.OBJ/CODE:THR/EXT/U:10/WA/LIST:SD3:RDTAPE     DY1:RDTAPE.FOR
!-----------------------------------------------------------
!           EDIT AND RECOMPILE ATMOS
!          **************************
EDIT DY1:ATMOS.FOR
COPY DY1:ATMOS.BAK SD3:ATMOS.BAK
DELETE/NOQUERY DY1:ATMOS.BAK
SQUEEZE/NOQUERY DY1:
FORT/OBJ:DY1:ATMOS.OBJ/CODE:THR/EXT/U:10/WA/LIST:SD3:ATMOS    DY1:ATMOS.FOR
!------------------------------------------------------------
```

```
!            EDIT AND RECOMPILE CONT
!           ************************
EDIT DY1:CONT.FOR
COPY DY1:CONT.BAK SD3:CONT.BAK
DELETE/NOQUERY DY1:CONT.BAK
SQUEEZE/NOQUERY DY1:
FORT/OBJ:DY1:CONT.OBJ/CODE:THR/EXT/U:10/WA/LIST:SD3:CONT      DY1:CONT.FOR
!------------------------------------------------------------
!            EDIT AND RECOMPILE BLOCK
!           *************************
EDIT DY1:BLOCK.FOR
COPY DY1:BLOCK.BAK SD3:BLOCK.BAK
DELETE/NOQUERY DY1:BLOCK.BAK
SQUEEZE/NOQUERY DY1:
FORT/OBJ:DY1:BLOCK.OBJ/CODE:THR/EXT/U:10/WA/LIST:SD3:BLOCK    DY1:BLOCK.FOR
!------------------------------------------------------------
!            EDIT AND RECOMPILE LAYNUM
!           **************************
EDIT DY1:LAYNUM.FOR
COPY DY1:LAYNUM.BAK SD3:LAYNUM.BAK
DELETE/NOQUERY DY1:LAYNUM.BAK
SQUEEZE/NOQUERY DY1:
FORT/OBJ:DY1:LAYNUM.OBJ/CODE:THR/EXT/U:10/WA/LIST:SD3:LAYNUM  DY1:LAYNUM.FOR
!------------------------------------------------------------
!            EDIT AND RECOMPILE AERSOL
!           **************************
EDIT DY1:AERSOL.FOR
COPY DY1:AERSOL.BAK SD3:AERSOL.BAK
DELETE/NOQUERY DY1:AERSOL.BAK
SQUEEZE/NOQUERY DY1:
FORT/OBJ:DY1:AERSOL.OBJ/CODE:THR/EXT/U:10/WA/LIST:SD3:AERSOL  DY1:AERSOL.FOR
!------------------------------------------------------------
!                HOUSEKEEPING
!               **************
SQUEEZE/NOQUERY DY1:
!------------------------------------------------------------
!      LINK ENTIRE PROGRAM, THEN DELETE OBJECT MODULES
!              (to save space on disk)
!      ***********************************************
LINK/EXECUTE:DY1:MICTRA.SAV/PROMPT DY1:MICDRB,DY1:SETUP
DY1:RDTAPE,DY1:ATMOS,DY1:CONT,DY1:BLOCK,DY1:LAYNUM
DY1:AERSOL,SY:FORLIB//
!
DELETE/NOQUERY DY1:(MICDRB.OBJ,SETUP.OBJ,RDTAPE.OBJ)
DELETE/NOQUERY DY1:(ATMOS.OBJ,CONT.OBJ,BLOCK.OBJ)
DELETE/NOQUERY DY1:(LAYNUM.OBJ,AERSOL.OBJ)
SQUEEZE/NOQUERY DY1:
!------------------------------------------------------------
!            COPY MICTRA TO OPERATING DISK (DY0)
!                AND SHOW BOTH DIRECTORIES
!           ************************************
COPY DY0:MICTRA.SAV SD3:MICBBK.SAV
DELETE/NOQ DY0:MICTRA.SAV
SQUEEZE/NOQ DY0:
COPY DY1:MICTRA.SAV DY0:MICTRA.SAV
DELETE/NOQ DY1:MICTRA.SAV
COPY/NOQ SD3:*.LST DY1:
DIR/ORDER/FULL DY0:
DIR/ORDER/FULL DY1:
```

```
!RNMICB.COM
!        This command file, to run the batch version of MICTRA,
!     does nothing but run a FORTRAN program which types on the screen
!     instructions to the user on which further command files to run
!     in order to run the MICTRA batch routine in its various modes.
!
RUN COM:RNMICB.SV1



!RMICB1.COM
!         COMMAND FILE TO RUN MICTRA FROM THE RL02
!         IN THE MODE WHICH GENERATES A BATCH FILE
!        ******************************************
!
! Initialize the RL02 simulated disks
! ************************************
!
RUN COM:RNMICB.LF        ! Make sure user has a condensed linefile on hand.
RUN COM:RNMICB.SV0       ! Tell user to save files before initializing disks.
INIT/NOQ SD0:
INIT/NOQ SD1:
INIT/NOQ SD2:
INIT/NOQ SD3:
INIT/NOQ SCR:
!
! Instruct user to load operating MICTRA floppy on DY0:
! and a floppy containing condensed line files on DY1:
! *****************************************************
RUN COM:RNMICB.SV2
!
! Copy appropriate files from floppy to simulated disk on RL02
! *************************************************************
COPY DY0:MICTRA.SAV SD2:MICTRA.SAV
COPY DY0:ATMMOD.DAT SD2:ATMMOD.DAT
COPY DY1:*.* SD1:
!
! Instruct user how to run MICTRA
! ******************************
RUN COM:RNMICB.SV3
!
! Run the batch MICTRA routine from the RL02
! *****************************************
RUN SD2:MICTRA.SAV
!RMICB2.COM
!        COMMAND FILE TO RUN MICTRA FROM THE RL02
!        IN THE MODE WHICH EXECUTES A BATCH FILE
!       ******************************************
!
! Initialize the RL02 simulated disks
! ************************************
!
RUN COM:RNMICB.LF      ! Make sure the user has a condensed linefile on hand.
RUN COM:RNMICB.SOO     ! Tell user to save files on disks to be initialized.
INIT/NOQ SD1:
INIT/NOQ SD2:
INIT/NOQ SD3:
INIT/NOQ SCR:
!
```

```
! Instruct user to load operating MICTRA floppy on DY0:
! and a floppy containing condensed line files on DY1:
! *******************************************************
RUN COM:RNMICB.SV2
!
! Copy appropriate files from floppy to simulated disk on RL02
! **************************************************************
COPY DY0:MICTRA.SAV SD2:MICTRA.SAV
COPY DY0:ATMMOD.DAT SD2:ATMMOD.DAT
COPY DY1:*.* SD1:
!
! See if batch file is on SD0:
! ****************************
RUN COM:RNMICB.SV4
DIR/ORDER/FULL/VOLUMEID SD0:
!
! If not, load a floppy with the batch file into DY0:
! ***************************************************
RUN COM:RNMICB.SV5
!
! Run the BATCH compiler
! **********************
@COM:BATCH.COM
```

```
!RMICB3.COM
!         COMMAND FILE TO RUN MICTRA FROM THE RL02
!         IN THE MODE WHICH GENERATES A BATCH FILE
!         AND THEN EXECUTES IT
!        ***************************************
!
! Initialize the RL02 simulated disks
! ***********************************
RUN COM:RNMICB.LF     ! Make sure the user has a condensed linefile on hand.
RUN COM:RNMICB.SV0    ! Tell user to save files on disks to be initialized.
INIT/NOQ SD0:
INIT/NOQ SD1:
INIT/NOQ SD2:
INIT/NOQ SD3:
INIT/NOQ SCR:
!
! Instruct user to load operating MICTRA floppy on DY0:
!  and a floppy containing condensed line files on DY1:
! *****************************************************
RUN COM:RNMICB.SV2
!
! Copy appropriate files from floppy to simulated disk on RL02
! ************************************************************
COPY DY0:MICTRA.SAV SD2:MICTRA.SAV
COPY DY0:ATMMOD.DAT SD2:ATMMOD.DAT
COPY DY1:*.* SD1:
!
! Instruct user how to run MICTRA
! *****************************
RUN COM:RNMICB.SV3
!
```

```
! Run the batch MICTRA routine from the RL02,
! to create a batch file to be executed
! ***************************************
RUN SD2:MICTRA.SAV
!
! Instruct the user how to respond to BATCH's prompt
! for a file to be executed (the file is SD0:BATFIL.BAT)
! ************************************************
RUN COM:RNMICB.SV6
!
! Run the BATCH compiler
! *********************
@COM:BATCH.COM
```

```
! BATCH.COM
! This command file runs the BATCH compiler
LOAD TT
!LOAD LP
LOAD BA
ASSIGN TT LOG  ! (or LP)
ASSIGN TT LST
R BATCH
```

```
!EDLPLT.COM
!  This command file edits, compiles and links the plot routine PLOTSP,
!  which displays spectra computed by MICTRA
!
RUN COM:EDLPLT.SAV          ! tell user to save any important files on SD0
!                             and to mount the MICTRA operating disk in DY0
!                             and the development disk in drive DY1.
INIT/NOQ SD0:
EDIT DY1:PLOTSP.FOR
FORT/OBJ:SD0:PLOTSP.OBJ/CODE:THR/EXT/WA/LIST:SD0:PLOTSP.LST DY1:PLOTSP.FOR
SQUEEZE/NOQ DY1:
LINK/EXE:SD0:PLOTSP.SAV SD0:PLOTSP.OBJ,PLT:PLTLIB,SY:FORLIB
DELETE/NOQ DY0:PLOTSP.SAV
SQUEEZE/NOQ DY0:
COPY SD0:PLOTSP.SAV DY0:PLOTSP.SAV
DIR/ORDER/FULL/VOLUMEID DY0:
DIR/ORDER/FULL/VOLUMEID DY1:
```

```
!RNLPLT.COM
!  This command file runs the auxiliary plotting routine PLOTSP,
!  which displays spectra computed by MICTRA.
!
RUN COM:RNLPLT.SAV          ! Tell user to mount MICTRA operating disk
!                             in drive DY0.
RUN DY0:PLOTSP.SAV
```

```
!EDLLIN.COM
!  This command file edits, compiles and links the line editing routine
!  LNEDIT, which processes an Air Force line file in ASCII characters
!  into the condensed binary form used by MICTRA.
!
RUN COM:EDLLIN.SAV          ! tell user to save any important files on SD0
!                             and to mount the MICTRA operating disk in DY0
!                             and the development disk in drive DY1.
INIT/NOQ SD0:
EDIT DY1:LNEDIT.FOR
FORT/OBJ:SD0:LNEDIT.OBJ/CODE:THR/EXT/WA/LIST:SD0:LNEDIT.LST DY1:LNEDIT.FOR
SQUEEZE/NOQ DY1:
LINK/EXE:SD0:LNEDIT.SAV SD0:LNEDIT.OBJ,PLT:PLTLIB,SY:FORLIB
DELETE/NOQ DY0:LNEDIT.SAV
SQUEEZE/NOQ DY0:
COPY SD0:LNEDIT.SAV DY0:LNEDIT.SAV
DIR/ORDER/FULL/VOLUMEID DY0:
DIR/ORDER/FULL/VOLUMEID DY1:
```

```
!RNLLIN.COM
!  This command file runs the auxiliary line editing routine LNEDIT,
!  which processes Air Force line files in ASCII characters into
!  the condensed binary form used by MICTRA.
!
RUN COM:RNLLIN.SV1          ! Make sure user has an Air Force ASCII
!                             line file at hand
RUN COM:RNLLIN.SAV          ! Tell user to mount MICTRA operating disk
!                             in drive DY0.
RUN DY0:LNEDIT.SAV
```

ATMOSPHERE AND AEROSOL MODELS

(These models are on a disk file, in exactly the form shown below; the first three lines are the sea-level molecular abundances of H2O, CO2 and O3, in molecules/sq cm/km, for the three atmosphere models.)

```
  4.69E+22  8.24E+20   7.53E+16
  1.17E+22  8.95E+20   7.53E+16
  1.97E+22  8.41E+20   6.78E+16
MIDLATITUDE SUMMER
   0.0 1.013E+03 294.0   1.4E+01   6.0E-05
   1.0 9.020E+02 290.0   9.3E+00   6.0E-05
   2.0 8.020E+02 285.0   5.9E+00   6.0E-05
   3.0 7.100E+02 279.0   3.3E+00   6.2E-05
   4.0 6.280E+02 273.0   1.9E+00   6.4E-05
   5.0 5.540E+02 267.0   1.0E+00   6.6E-05
   6.0 4.870E+02 261.0   6.1E-01   6.9E-05
   7.0 4.260E+02 255.0   3.7E-01   7.5E-05
   8.0 3.720E+02 248.0   2.1E-01   7.9E-05
   9.0 3.240E+02 242.0   1.2E-01   8.6E-05
  10.0 2.810E+02 235.0   6.4E-02   9.0E-05
  11.0 2.430E+02 229.0   2.2E-02   1.1E-04
  12.0 2.090E+02 222.0   6.0E-03   1.2E-04
  13.0 1.790E+02 216.0   1.8E-03   1.5E-04
  14.0 1.530E+02 216.0   1.0E-03   1.8E-04
  15.0 1.300E+02 216.0   7.6E-04   1.9E-04
MIDLATITUDE WINTER
   0.0 1.018E+03 272.2   3.5E+00   6.0E-05
   1.0 8.973E+02 268.7   2.5E+00   5.4E-05
   2.0 7.897E+02 265.2   1.8E+00   4.9E-05
   3.0 6.938E+02 261.7   1.2E+00   4.9E-05
   4.0 6.081E+02 255.7   6.6E-01   4.9E-05
   5.0 5.313E+02 249.7   3.8E-01   5.8E-05
   6.0 4.627E+02 243.7   2.1E-01   6.4E-05
   7.0 4.016E+02 237.7   8.5E-02   7.7E-05
   8.0 3.473E+02 231.7   3.5E-02   9.0E-05
   9.0 2.992E+02 225.7   1.6E-02   1.2E-04
  10.0 2.568E+02 219.7   7.5E-03   1.6E-04
  11.0 2.199E+02 219.2   6.9E-03   2.1E-04
  12.0 1.882E+02 218.7   6.0E-03   2.6E-04
  13.0 1.610E+02 218.2   1.8E-03   3.0E-04
  14.0 1.378E+02 217.7   1.0E-03   3.2E-04
  15.0 1.178E+02 217.2   7.6E-04   3.4E-04
U.S.STANDARD
   0.0 1.013E+03 288.1   5.9E+00   5.4E-05
   1.0 8.986E+02 281.6   4.2E+00   5.4E-05
   2.0 7.950E+02 275.1   2.9E+00   5.4E-05
   3.0 7.012E+02 268.7   1.8E+00   5.0E-05
   4.0 6.166E+02 262.2   1.1E+00   4.6E-05
   5.0 5.405E+02 255.7   6.4E-01   4.6E-05
   6.0 4.722E+02 249.2   3.8E-01   4.5E-05
   7.0 4.111E+02 242.7   2.1E-01   4.9E-05
   8.0 3.565E+02 236.2   1.2E-01   5.2E-05
   9.0 3.080E+02 229.7   4.6E-02   7.1E-05
  10.0 2.650E+02 223.2   1.8E-02   9.0E-05
  11.0 2.270E+02 216.8   8.2E-03   1.3E-04
  12.0 1.940E+02 216.6   3.7E-03   1.6E-04
  13.0 1.658E+02 216.6   1.8E-03   1.7E-04
  14.0 1.417E+02 216.6   8.4E-04   1.9E-04
  15.0 1.211E+02 216.6   7.2E-04   2.1E-04
```

Explanation:
****

Atmosphere models
****

Column	Quantity
1	height (km)
2	pressure (mb)
3	temp. (K)
4	density of water vapor (g/cubic m)
5	density of ozone (g/cubic m)

(The density of CO2 is calculated in the program.)

Aerosol models
****

1	wavelength (micrometers)
2	aerosol scattering coeff.
3	aerosol absorption coefficient

The aerosol coefficients are normalized to 1.00 per km at 0.55 micrometers.

RURAL

40.000	.02311	.04972
35.000	.02601	.04885
30.000	.02807	.04817
27.900	.03095	.04713
25.000	.03505	.04737
22.500	.03883	.04856
21.300	.03988	.04959
20.000	.04175	.04857
18.500	.04276	.04286
18.000	.04495	.04387
17.200	.04512	.04857
16.400	.04353	.04407
15.000	.03376	.05278
14.800	.03926	.03855
14.000	.04733	.03407
13.000	.05164	.03310
12.500	.05370	.03269
11.500	.05823	.03384
11.000	.05972	.03607
10.591	.05749	.04301
10.000	.05722	.05123
9.800	.05774	.05266
9.500	.05477	.06358
9.200	.05289	.08719
9.000	.05719	.07649
8.700	.05497	.07125
8.500	.03319	.07937
8.200	.01340	.05110
7.900	.03074	.03904
7.200	.05216	.04713
6.500	.05488	.03354
6.200	.05696	.03113
6.000	.05870	.02921
5.500	.06904	.02455
5.000	.07672	.02033
4.500	.08011	.02219
4.000	.09001	.01465
3.750	.09394	.01357
3.500	.09598	.01583
3.392	.09251	.01979
3.200	.09339	.02046
3.000	.08330	.03696
2.700	.07835	.06559
2.500	.1016	.02932
2.250	.1140	.02894
2.000	.1312	.02763
1.800	.1535	.04432
1.536	.2057	.05907
1.300	.2697	.06247
1.060	.3802	.06775
0.860	.5122	.06694
0.694	.7023	.06129
0.633	.7900	.06159
0.550	.9307	.06928
0.515	1.008	.06748
0.488	1.068	.07050
0.400	1.311	.08337
0.337	1.537	.09620
0.300	1.673	.1335
0.250	1.856	.2317
0.200	1.916	.5029

TROPOSPHERIC

40.000	.00006	.01807
35.000	.00009	.01730
30.000	.00012	.01732
27.900	.00017	.01608
25.000	.00026	.01555
22.500	.00040	.01553
21.300	.00049	.01616
20.000	.00061	.01530
18.500	.00056	.01345
18.000	.00072	.01324
17.200	.00092	.01578
16.400	.00068	.01488
15.000	.00046	.02374
14.800	.00049	.01389
14.000	.00075	.01076
13.000	.00104	.01036
12.500	.00122	.00997
11.500	.00185	.01027
11.000	.00235	.01108
10.591	.00275	.01445
10.000	.00375	.01895
9.800	.00423	.01966
9.500	.00484	.02700
9.200	.00620	.04812
9.000	.00847	.03750
8.700	.00785	.03419
8.500	.00121	.04847
8.200	.00017	.02929
7.900	.00055	.01801
7.200	.00294	.02036
6.500	.00383	.01276
6.200	.00393	.01169
6.000	.00401	.01080
5.500	.00611	.00805
5.000	.00849	.00623
4.500	.01159	.00701
4.000	.01564	.00434
3.750	.01838	.00402
3.500	.02174	.00483
3.392	.02133	.00648
3.200	.02350	.00677
3.000	.02394	.01589
2.700	.02650	.03891
2.500	.03797	.01157
2.250	.05005	.01146
2.000	.06761	.01092
1.800	.09904	.02200
1.536	.1603	.03451
1.300	.2313	.03796
1.060	.3532	.04336
0.860	.4971	.04326
0.694	.7028	.03916
0.633	.7983	.03985
0.550	.9528	.04716
0.515	1.037	.04589
0.488	1.102	.04884
0.400	1.368	.06144
0.337	1.616	.07429
0.300	1.766	.1119
0.250	1.966	.2184
0.200	2.036	.5090

MARITIME		
40.000	.02533	.1336
35.000	.03006	.1119
30.000	.04085	.1123
27.900	.04611	.1198
25.000	.05367	.1299
22.500	.06128	.1433
21.300	.06514	.1509
20.000	.07033	.1606
18.500	.07484	.1752
18.000	.07611	.1795
17.200	.07504	.1858
16.400	.06689	.1876
15.000	.05950	.1866
14.800	.05801	.1832
14.000	.05448	.1789
13.000	.04841	.1629
12.500	.04623	.1490
11.500	.05652	.1052
11.000	.07216	.08415
10.591	.09447	.06893
10.000	.13066	.06245
9.800	.14392	.06222
9.500	.16085	.06397
9.200	.17855	.07218
9.000	.19546	.07085
8.700	.1996	.07051
8.500	.1848	.07119
8.200	.1869	.06368
7.900	.1988	.05840
7.200	.2415	.05965
6.500	.2879	.07387
6.200	.28860	.14480
6.000	.19840	.14880
5.500	.34310	.03207
5.000	.40720	.03660
4.500	.45200	.04247
4.000	.52890	.01985
3.750	.56450	.01718
3.500	.5849	.04164
3.392	.5670	.08004
3.200	.4594	.2208
3.000	.3271	.3328
2.700	.4399	.09134
2.500	.6093	.02125
2.250	.6772	.01062
2.000	.7159	.01380
1.800	.7534	.00918
1.536	.7951	.01207
1.300	.8288	.01178
1.060	.8682	.01210
0.860	.9013	.01084
0.694	.9396	.00979
0.633	.9556	.00996
0.550	.9882	.01179
0.515	.9975	.01147
0.488	1.014	.01221
0.400	1.063	.01536
0.337	1.115	.01858
0.300	1.142	.02800
0.250	1.181	.05471
0.200	1.188	.13000

URBAN

40.000	.02037	.04028
35.000	.02248	.04112
30.000	.02422	.04226
27.900	.02601	.04246
25.000	.02861	.04374
22.500	.03099	.04552
21.300	.03174	.04672
20.000	.03298	.04691
18.500	.03376	.04505
18.000	.03499	.04595
17.200	.03524	.04911
16.400	.03457	.04751
15.000	.02985	.05336
14.800	.03275	.04621
14.000	.03713	.04476
13.000	.03961	.04562
12.500	.04083	.04604
11.500	.04350	.04842
11.000	.04448	.05040
10.591	.04350	.05493
10.000	.04365	.06037
9.800	.04400	.06161
9.500	.04262	.06802
9.200	.04181	.08090
9.000	.04414	.07610
8.700	.04318	.07416
8.500	.03205	.07916
8.200	.02199	.06560
7.900	.03118	.06030
7.200	.04271	.06784
6.500	.04496	.06393
6.200	.04634	.06492
6.000	.04767	.06472
5.500	.05386	.06613
5.000	.05900	.06824
4.500	.06220	.07474
4.000	.06939	.07739
3.750	.07258	.08063
3.500	.07500	.08610
3.392	.07383	.08966
3.200	.07572	.09346
3.000	.07209	.1075
2.700	.07286	.1288
2.500	.08758	.1157
2.250	.09833	.1240
2.000	.1129	.1339
1.800	.1302	.1524
1.536	.1670	.1753
1.300	.2131	.1964
1.060	.2896	.2256
0.860	.3828	.2545
0.694	.5110	.2866
0.633	.5706	.3027
0.550	.6657	.3343
0.515	.7167	.3482
0.488	.7567	.3591
0.400	.9171	.4036
0.337	1.065	.4417
0.300	1.154	.4792
0.250	1.251	.5474
0.200	1.287	.6767

BACKGR	STRATOSPHERIC	
40.000	0.	.00592
35.000	0.	.00580
30.000	.00001	.00632
27.900	.00001	.00680
25.000	.00001	.00836
22.500	.00002	.01530
21.300	.00002	.01617
20.000	.00002	.01343
18.500	.00004	.01710
18.000	.00005	.02321
17.200	.00005	.04147
16.400	.00004	.03661
15.000	.00005	.01948
14.800	.00005	.01890
14.000	.00007	.01860
13.000	.0001	.01930
12.500	.00013	.01962
11.500	.00026	.03549
11.000	.00026	.05710
10.591	.00027	.04041
10.000	.00049	.04971
9.800	.00067	.07256
9.500	.00071	.09987
9.200	.00058	.08722
9.000	.00072	.09217
8.700	.00096	.1264
8.500	.00095	.1448
8.200	.00077	.1463
7.900	.00050	.1182
7.200	.00019	.04437
6.500	.00055	.03283
6.200	.00088	.04304
6.000	.00105	.05263
5.500	.00103	.05704
5.000	.00145	.04132
4.500	.00242	.04522
4.000	.00403	.05391
3.750	.00510	.06019
3.500	.00623	.07917
3.392	.00627	.08300
3.200	.00600	.07671
3.000	.00632	.05878
2.700	.00931	.00403
2.500	.01560	.00289
2.250	.02570	.00157
2.000	.04055	.00128
1.800	.05817	.00064
1.536	.09972	.00020
1.300	.1642	.00002
1.060	.2886	.00000
0.860	.4685	.00000
0.694	.7063	.00000
0.633	.8224	.00000
0.550	1.000	.00000
0.515	1.087	.00000
0.488	1.150	.00000
0.400	1.376	.00000
0.337	1.515	.00000
0.300	1.555	.00000
0.250	1.553	.00000
0.200	1.487	.00000

AGED VOLCANIC

40.000	.00002	.01328
35.000	.00003	.01192
30.000	.00006	.01124
27.900	.00008	.01072
25.000	.00014	.01038
22.500	.00020	.01104
21.300	.00023	.01054
20.000	.00026	.01390
18.500	.00031	.01741
18.000	.00031	.01826
17.200	.00029	.01706
16.400	.00028	.01620
15.000	.00032	.01635
14.800	.00033	.01532
14.000	.00046	.01447
13.000	.00071	.01530
12.500	.00090	.01623
11.500	.00191	.02282
11.000	.00260	.02511
10.591	.00322	.02859
10.000	.00449	.03009
9.800	.00475	.02979
9.500	.00518	.02871
9.200	.00573	.02658
9.000	.00602	.02506
8.700	.00481	.02438
8.500	.00257	.02218
8.200	.00099	.02180
7.900	.00136	.01652
7.200	.00250	.01112
6.500	.00378	.00665
6.200	.00467	.00525
6.000	.00570	.00451
5.500	.00869	.00396
5.000	.01204	.00335
4.500	.01775	.00319
4.000	.02394	.00349
3.750	.02913	.00488
3.500	.03435	.00654
3.392	.03761	.00743
3.200	.04262	.00936
3.000	.05126	.00949
2.700	.06343	.00842
2.500	.07792	.00843
2.250	.1021	.00867
2.000	.1353	.01019
1.800	.1849	.01191
1.536	.2648	.01490
1.300	.3693	.01881
1.060	.5057	.02452
0.860	.6570	.03178
0.694	.8079	.04084
0.633	.8676	.04532
0.550	.9473	.05271
0.515	.9799	.05651
0.488	1.0030	.05962
0.400	1.0680	.07244
0.337	1.0950	.06497
0.300	1.0790	.1126
0.250	.9009	.2818
0.200	.7006	.4482

REFERENCES

1. R.A. McClatchey and A.P. D'Agati, "Atmospheric Transmission of Laser Radiation: Computer Code LASER," AFGL-TR-78-0029, 31 January 1978.

2. H.G. Safren, "A Computer Code to Calculate the Emission or Absorption Spectra of Atmospheric Water Vapor, Carbon Dioxide, Ozone, Nitrous Oxide, Carbon Monoxide, Methane or Oxygen Lines," NASA TM, to be published.

3. H.G. Safren, "A Plot Package for the DEC 11/23 Microcomputer, Using the DEC VT100 Terminal with the Digital Engineering Retro-Graphics Enhancement," NASA TM, to be published.

4. E.P. Shettle and R.W. Fenn, "Models of the Atmospheric Aerosols and their Optical Properties," AGARD Conference Proceedings No. 183, Optical Propagation in the Atmosphere, pp. 2.1-2.16, 1976.

5. R.A. McClatchey, W.S. Benedict, S.A. Clough, D.E. Burch, R.F. Calfee, K. Fox, L.S. Rothman and J.S. Garing, "AFCRL Atmospheric Absorption Line Parameters Compilation," AFCRL-TR-73-0096, 26 January 1973.

6. L.S. Rothman and R.A. McClatchey, "Updating of the AFCRL Atmospheric Absorption Line Parameters Compilation," Applied Optics letter, Vol. 15, No. 11, November 1976.

7. L.S. Rothman, "Update of the AFGL Atmospheric Absorption Line Parameters Compilation," Applied Optics letter, Vol. 17, No. 22, 15 November 1978.

8. L.S. Rothman, "AFGL Atmospheric Absorption Line Parameters Compilation: 1980 Version," Applied Optics, Vol. 20, N0. 5, 1 March 1981, pp. 791-795.

9. L.S. Rothman, R.R. Gamache, A. Barbe, A. Goldman, J.R. Gillis, L.R. Brown, R.A. Toth, J.-M Flaud and C. Camy-Peyret, "AFGL Atmospheric Absorption Line Parameters Compilation: 1982 Edition," Applied Optics, Vol.22, No 15, 1 August 1983, pp. 2247-2256.

10. C. Weng, private communication.

Report Documentation Page

NASA — National Aeronautics and Space Administration

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-100686		

4. Title and Subtitle	5. Report Date
A Computer Code to Calculate Line by Line Atmospheric Transmission Spectra on a Microcomputer	July 1987
	6. Performing Organization Code: 723.0

7. Author(s)	8. Performing Organization Report No.
H. G. Safren	86B0201
	10. Work Unit No.
9. Performing Organization Name and Address: Goddard Space Flight Center, Greenbelt, Maryland 20771	11. Contract or Grant No.
12. Sponsoring Agency Name and Address: National Aeronautics and Space Administration, Washington, D.C. 20546-0001	13. Type of Report and Period Covered: Technical Memorandum
	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A computer program is described which calculates line by line atmospheric transmission spectra on a microcomputer. Radiance calculations are not included. The program is written in Fortran and could be modified to run on a microcomputer other than the one on which it was implemented, except that the plotting routine would have to be replaced. The program is based on the 1978 Air Force Geophysics Laboratory LASER routine, and uses absorption line data from the 1982 update of the AFGL Atmospheric Absorption Line Parameters Compilation; segments of needed line data are taken from the tape, preprocessed into the form used by the program and stored on floppy disks. The program calculates transmission spectra over a variety of paths and spanning an arbitrarily chosen wavelength or frequency range; the path may be chosen to be horizontal, vertical or slanted at any zenith angle, and may extend between any two altitudes between 0 and 15 km. Aerosol effects are included, presently based on the 1976 Shettle and Fenn models, plus some molecular continuum effects and Rayleigh scattering.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Atmospheric Transmission; Computer Program; Microcomputer	Unclassified - Unlimited Subject Category 17

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified		

NASA FORM 1626 OCT 86