OXFORD PLANETARY GROUP

TITLE: RADTRANS

AUTHORS:

....
P.G.J. IRWIN

DOCUMENT NO.: NIMS/OX/PGJI/SW/136

DOCUMENT TITLE
RADTRANS
DOCUMENT NUMBER: NIMS/OX/PGJI/SW/136

DOCUMENT STATUS SHEET

ISSUE	REV	DATE	REASON FOR CHANGE
1	0	12/11/94	New Software Release
2	0	19/12/94	Addition of Scattering
2	1	20/3/95	Addition of further features
2	2	30/7/96	Major text revisions
2	3	23/10/98	Long overdue revision
2	3a	13/1/00	Minor corrections
3	0	1/7/04	Long overdue overhaul
3	1	5/10/09	Another long due overhaul to update the location of various directories.
3	2	30/4/12	Another overhaul to link with svn control

DISTRIBUTION LIST

	NAME			NAME	
1	P. Irwin	Oxford			
1	N. Teanby	Oxford			
1	N. Bowles	Oxford			
1	C. Wilson	Oxford			

Table of Contents

0. Overview

This document describes the radiative transfer model RADTRANS which has been developed over a number of years by the Planetary Group in Oxford for the modelling of radiances in planetary atmospheres.

0.1. Reference Documents

[R1]   Atmospheric Radiation. Theoretical Basis 2nd Ed, R.M. Goody and Y.L.Yung (1989). *Oxford University Press*

[R2]   Integrated Absorption of a Spectral Line with the Voigt Profile, C.D.Rodgers and A.P.Williams (1974). *J.Quant. Spectrosc. Radiat. Transfer* **14** pp 319-323

[R3]   Approximate Methods of Calculating Transmission by Bands of Spectral Lines, C.D.Rodgers (1976). NCAR/TN-116+1A

[R4]   Spectral Parameters of Methane for Remote Sounding of the Jovian Atmosphere, E.K.Strong (1992). *D.Phil.Thesis,* Oxford

[R5]   AFCRL Atmospheric Absorption Line Parameters Compilation, McClatchey *et al* (1973). *AFCRL-TR-73-0096*

[R6]   A Description of the Correlated k Distribution Method for Modelling Nongray Gaseous Absorption, Thermal Emission, and Multiple Scattering in Vertically Inhomogeneous Atmospheres, A.A.Lacis and V.Oinas (1991). *J.G.R.* **96** pp 9027-9063

[R7]   The Correlated-k Method for Radiation Calculations in Nonhomogeneous Atmospheres, R.Goody, R.West, L.Chen and D.Crisp (1989), *J.Quant. Spectrosc. Radiat. Trans.* **42** pp 539-550

[R8]   Calculated Tabulations of H_2O Line Broadening by H_2O, N_2, O_2 and CO_2 at High Temperature, C.Delaye, J-M Hartmann and J.Taine (1989), *Appl. Opt.* **28** pp 5080

[R9]   Estimation of Rotational Line Widths of Carbon Dioxide Bands, G.Yamamoto, M.Tanaka and T.Aoki (1969), *J.Quant. Spectrosc. Radiat. Trans.* **9** pp 371-382

[R10]   Oxford Planetary Group Radiative Transfer Models, P. Irwin (1994), NIMS/OX/PGJI/SW/133.

[R11]   Light Scattering in Planetary Atmospheres, J.E.Hansen and L.D.Travis (1974), Space Sci. Rev **16** 527-610.

[R12]   Modelling of Pressure-Induced FIR Absorption spectra: Molecular Hydrogen Pairs, J.Borysow *et al.* (1985), *Astrophys.J.* **296** pp 644-654

[R13]   Collision-induced Rototranslational Absorption Spectra of H_2_He pairs at temperatures from 40 to 3000K, J.Borysow *et al.* (1988), *Astrophys.J.* **326** pp 509-515

[R14]   Nemesis, P. Irwin (2004), CIRS/OX/TR/1392.

[R15]   Giant Planets of our Solar System, P.Irwin (2003), Springer-Praxis.

[R16]   Radtran Scattering Code, P. Irwin (1997) NIMS/OX/PGJI/SW/267

[R17]   NIMS Scattering Study, P. Irwin (1997), NIMS/OX/PGJI/SW/268.

[R18]   Multiple scattering of polarized light in planetary atmospheres. Part II. Sunlight reflected by terrestrial water clouds, Hansen, J. E (1971), *J. Atm. Sci.,* **28**, pp 1400-1426,

[R19]   Matrix operator theory of radiative transfer. 1: Rayleigh scattering, G.N. Plass et al. (1973), *Appl. Opt.,* **12**, pp 314- 329

0.2. Defined Fonts

As in [R14], in an attempt to make this document easier to read the following fonts are used to denote different objects.

- Executable programs are underlined. e.g. <u>Radtrans</u>
- Suites of codes in their own subdirectories are in copperplate font. e.g. RADTRANS
- Subroutine files are in courier font. e.g. `genrads.f`
- Variables defined within FORTRAN codes are capitalized. e.g. NCONV, IMOD.

1. Introduction

The RADTRANS model calculates the transmission, absorption or thermal emission spectra of gaseous paths using one of three radiative transfer (RT) models: 1) line-by-line, 2) band or 3) correlated-k [R1]. In addition to transmission, absorption or thermal emission, RADTRANS may compute multiple-scattered spectra, and hemispherically-averaged spectra. Depending on the RT model chosen, spectral information is read in either from line databases, averaged band parameter, or k-distribution files. All the sub-models use identical methods for calculating the atmospheric gas paths and dust opacities and have identical outputs to allow easy intercomparison. The atmospheric profiles are read in from '.prf' files which contain all the information required to model the atmosphere of any planet. The models are thus not limited to the Earth's atmosphere like some but are intended to be general and simple to set up for any planet.

RADTRANS is a development from GENLBL, a general purpose line-by-line transmission calculation routine written by Simon Calcutt. RADTRANS combines the LBL functions of GENLBL with faster, though less accurate, band and correlated-k models described in [R10]. The model takes as its input a '.drv' driver file created by the program Path (which takes as its input a '.pat' path definition file) and reads in the required spectral data files together with any additional files required by the particular RT model used (for example dust optical properties and g-intervals). An overview of the processing process is shown in Figure 1.1.
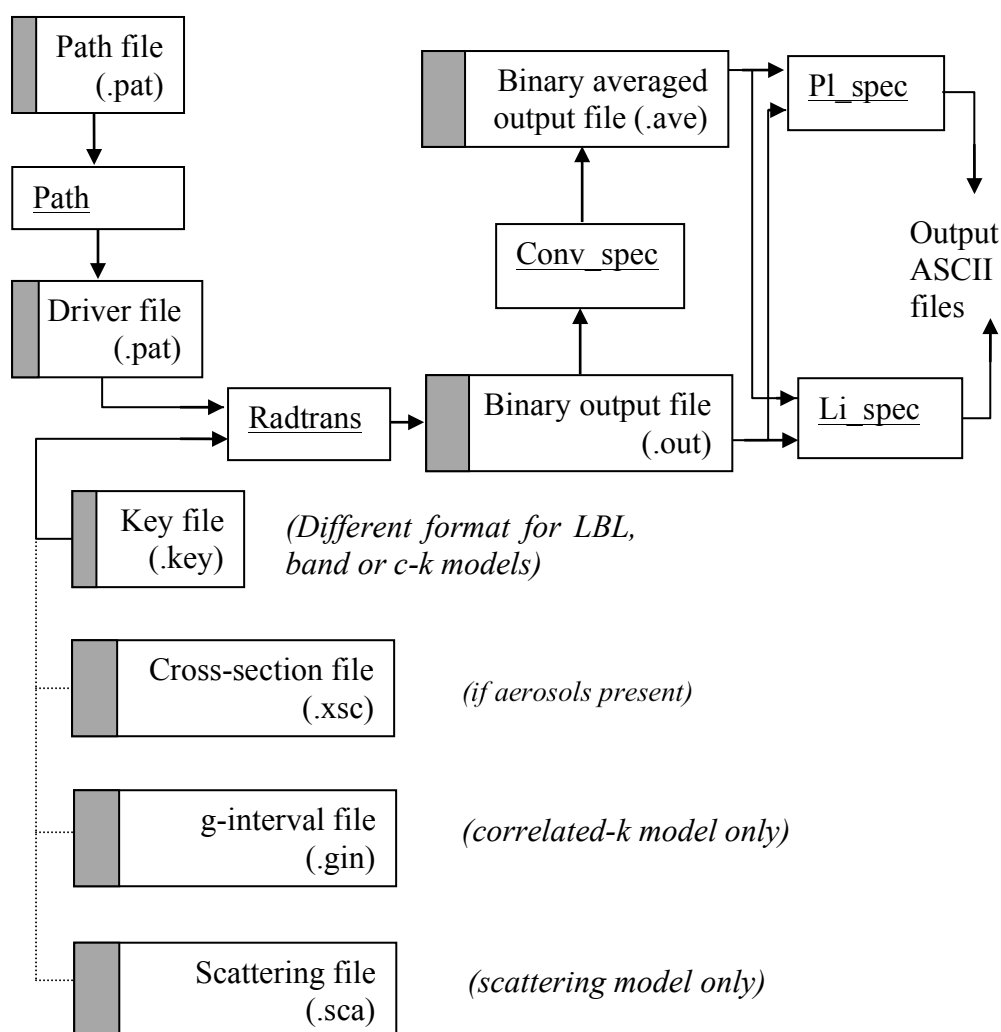
Fig. 1.1 Overview of RADTRANS Processing Software

Spectra are calculated in wavenumber space only and the calculated spectra are written to binary '.out' files. Hence these output files must be processed using one of the three post-processors Conv_spec, Li_spec and Pl_spec which are described later and which may transform the spectra to wavelength space if required and write the final output spectra in ASCII form.

The basic procedure for calculating a spectrum is this:
1) Construct the '.pat' *path file* which defines which atmospheric profiles to use, which spectral information to include and the calculation type required.
2) Run Path to create the '.drv' *driver file*, for which the atmospheric layer properties (pressure, temperature, absorber amount etc.) have been calculated.
3) Run Radtrans to compute the output binary spectra.
4) Process the raw binary '.out' files to the required output using either Conv_spec, Li_spec or Pl_spec (or in some cases Aground, Diskav)

This document is laid out in same the order of these steps, except that a description of the atmospheric '.prf' files is first described. Please note that reference '.prf' files and also spectral databases are already available in the directories listed in section 1.1. New users of RADTRANS should not need to generate reference '.prf' and spectral database files from scratch, only the '.pat' files.

In summary:
Section 2 describes the atmospheric temperature, pressure and vmr '.prf' profiles.
Section 3 describes the dust/aerosol '.prf' profiles.
Section 4 describes the '.pat' *path files* and the program Path.
Section 5 describes the detailed operation of the Radtrans program for the different spectral models available.
Section 6 describes the details of the post-processing programs Conv_spec, Li_spec and Pl_spec.

Finally Section 7 describes how line-by-line, band and correlated-k spectral databases are set up. Normally this will have already been done for you! In addition, Appendix A lists the RADTRANS gas ID and ISO codes.

1.1 Location of Code and Reference Files

All source code is now under SVN management to maintain all versions of the software that might be released. The SVN central repository of RADTRANS (and NEMESIS) is http://scm.physics.ox.ac.uk/svn/radtrancode (please contact P. Irwin for details on how to access the files). The SVN repository contains the following subdirectories:

frecipes/   Numerical recipes Fortran programs. A makefile is included.

raddata/   Reference spectral data for use by Radtrans/Nemesis, including CIA tables, solar/stellar reference spectra, gas continuum spectra, lineshape factors, SVP curves and last, but definitely not least, a list of the current planets defined, together with their mass and radius, etc.

radtran/   Main Radtrans directory, containing a number of sub-directories. Several executables are defined, including Radtrans itself and a number of line data and correlated-k processing programs. The subdirecoties makefiles/ includes shell scipts for making the libraries in one go and for making all the Radtrans executables. The makefiles for each sub-directory are also included.

nemesis/   The Nemesis subroutines and programs. The Nemesis programs are built upon the Radtrans libraries in radtran/ so these must be compiled first.

idl/   Useful IDL programs for reading and plotting Radtrans/Nemesis calculations.

manuals/   The Radtrans and Nemesis manuals.

FOVgreg/   Library of routines for defining MCS Field of View. Only needs
            compiling if you're going to run NemesisMCS or GenerateMCSspx.

All standard line data files are in the directory:
/home/oxpln98/plan/teanby/linedata

Useful planetary profile files (.prf files) are in the directory:
/home/jupiter/plan/plan2/oxpln5/radtran/profiles

Useful band data files (.ban files) are in the directory:
/home/jupiter/plan/plan2/oxpln5/radtran/banddata

Useful k-table files (.kta files) are in the directory:
/home/jupiter/plan/plan2/oxpln5/radtran/ktables

The code is currently using the Intel compiler on the departmental Linux machines
and compiles on both 32 and 64 bit machines using the Intel Fortran Compiler. The
code also compiles on Mac OSX, using gfortran.

2 Atmospheric T,P and VMR Profile Files (.prf)

Atmospheric profiles are read in from standard format ASCII '.prf' files which contain the following information. The old format listed below consisted of 6 column blocks of data each with a descriptive header. Note that if more than three gases were included then these were listed in the next block and so on. **This format has now been superseded by one where all the gas volume mixing ratios for a level are listed on the same line as height, pressure and temperature. Conversion of old format .prf files may be achieved using the program Convert_prf.**

```
AMFORM
IPLANET  LATITUDE  NP  NVMR  MOLWT
ID(1)        ISO(1)
  :            :
  :            :
ID(NVMR)        ISO(NVMR)
"H      P      T      gas1    gas2    gas3"
H(1)   P(1)   T(1)   VMR(1,1) VMR(1,2) VMR(1,3)
  :     :      :      :        :        :
  :     :      :      :        :        :
H(NP) P(NP)  T(NP)  VMR(NP,1) VMR(NP,2) VMR(NP,3)
"Additional gas profiles      4 to      ..."
VMR(1,4) VMR(1,5)  . .....VMR(1,NVMR)
  :       :             :
  :       :             :
VMR(NP,4) VMR(NP,5)  ...VMR(NP,NVMR)
```

AMFORM	Defines the format of the atmospheric profile. Above is the default format with AMFORM=0. If AMFORM=1, then it is assumed that at each level the VMRs add up to 1.0 and hence that the molecular weight can be calculated at each level. As a result of this for AMFORM=1, MOLWT is omitted from the list of parameters in the 2nd line.
IPLANET	Planet ID (3=Earth, 5=Jupiter etc)
LATITUDE	Planetocentric latitude (for calculating centripetal and non-spherical effects on gravity)
NP	Number of points in the profile.
NVMR	Number of gases whose volume mixing ratio (VMR) profiles are included in the file.
MOLWT	Mean molecular weight of the atmosphere in grams.
ID(1)	RADTRANS identification code of gas 1. **SEE APPENDIX 1 FOR DEFINITIONS OF GAS CODES.**
ISO(1)	RADTRANS identification code of gas 1's isotope. 0 = all isotopes in standard terrestrial relative abundance.
H(1)	Height in km of point 1 above reference level (ground for terrestrial planets, usually 1-bar level for gas giants)
P(1)	Pressure in atm at point 1
T(1)	Temperature in K at point 1
VMR(1,2)	Volume mixing ratio of gas 2 at vertical point 1.

Given the diverse sources for generating a planetary profile, it is usually necessary to generate new profiles off-line using bespoke software or an ASCII editor. However it is possible to initiate a new profile using Profile, a flexible manipulation program for the '.prf' files which is now explained.

2.1 Profile

This is a general purpose manipulation program for '.prf' files. At initiation, the program asks whether the user wishes to generate a new file or read in and adapt an existing file. If the user chooses to read in an existing file, the program will prompt for the filename and read in the data. If the user chooses to initiate a new file, the program will prompt for the appropriate data and generate the new profile. This first choice is really entirely up to you. For new profiles with only a few vertical points it is probably easier to enter the points manually using Profile. If there are many vertical points however, the user should probably familiarise himself with the '.prf' format and generate a profile off-line. Should the user choose to enter the points manually however, the program will initially assume that you only know the temperature profile as a function of height. Profile will prompt for the number of vertical points, the planetary mass and radius and the mean molecular weight of the atmosphere. It will then prompt for the altitude and temperature of each point and generate the new profile. It will also compute a pressure profile assuming hydrostatic equilibrium.

Once a profile has been established, manipulation may begin. The program goes into a loop where at each stage the following menu is displayed:

```
----------------------------------------------------
 A - display summary of profiles
 B - add a VMR profile
 C - remove a VMR profile
 D - force VMR to saturation vapour pressure
 E - compute temperature profile from hydrostatic equil.
 F - compute pressure profile from hydrostatic equil.
 K - compute height profile from hydrostatic equil.
 H - output profiles and exit
 I - multiply pressure by const (e.g. scale bar-atm)
 P - output IDL compatable profile data
 M - multiply vmr profile by constant
 G - Map profile on to new pressure grid
W - Set water vmr profile to pr-um
 V - output single gas VMR profile
 Q - quit
----------------------------------------------------
 command?
```

These options are now described:

A - display summary of profiles

This option simply displays a summary of the current profile to the screen. A typical output is:

```
profiles contain  51 points,  planetary nominal radius =  3400.000 km
 5 volume mixing ratio profile(s) included
gas: 1  identifier: 2  isotope:  0
gas: 2  identifier: 1  isotope:  0
gas: 3  identifier: 7  isotope:  0
gas: 4  identifier: 5  isotope:  0
gas: 5  identifier: 3  isotope:  0
```

B - add a VMR profile

This option adds a gas volume mixing ratio to the profile. The new profile may be generated in one of three ways:

1) read in from an external file. The format of the .vmr file is in default ASCII and must contain the following information:

   IDGAS   ISOGAS
   VMRH(1)   VMR(1)
       :   :
   VMRH(NV)   VMR(NV)

   IDGAS and ISOGAS are as before and identify which gas the file contains. The next lines then contain the volume mixing ratio at any number of heights. The heights do not have to correspond to those in the existing temperature/ pressure file and the file does not have to have the same number of points. <u>Profile</u> reads in the data and automatically detects the end of the file. The program then interpolates the points and maps them onto the heights of the stored temperature/pressure file

2) generate the new VMR profile assuming a constant mixing ratio with height which the program prompts for.

3) generate the new VMR profile in the same manner as 1) except reading the data directly from the keyboard and terminating with -1,-1.

C - remove a VMR profile

Simply removes a VMR profile from the stored profile.

D - force VMR to saturation vapour pressure

This option forces an existing VMR profile to be that of a saturated vapour pressure profile. The dependence of saturated vapour pressure (atm) on temperature (K) is assumed to be of the form:

$$svp = \exp\left(a + b/T + cT + dT^2\right)$$

The program prompts for the parameters *a*, *b*, *c* and *d* and then clips the VMR profile where required so that the partial pressure of the gas in question ($PP = VMR(I,J)*P(I)$) is always less than or equal to the saturated vapour pressure. A list of SVP parameters for commonly used gases may be found in …/raddata/SVP.dat

E - compute temperature profile from hydrostatic equil.

Given the height and pressure profile, together with the mean atmospheric molecular weight, the temperature profile is calculated from hydrostatic equilibrium.
F - compute pressure profile from hydrostatic equil.

Program prompts for the pressure level of zero height and then computes all pressures at higher altitudes according to hydrostatic equilibrium using the temperature profile and mean molecular weight of the atmosphere.

K - compute height profile from hydrostatic equil.

Program prompts for the height level of 1atm pressure and then computes all pressures at higher altitudes according to hydrostatic equilibrium using the temperature profile and mean molecular weight of the atmosphere.

H - output profiles and exit

Program prompts for a new profile filename and outputs modified data to a new .prf file.

I - multiply pressure by a const.

Self-explanatory! For example, pressures may be in bars and you want to convert them to atm.

P - output IDL compatable profile data

Outputs plotting data to an ASCII file which is readable by IDL routines such as `plotprf.pro`.

M - multiply vmr profile by constant

Multiply a specified VMR column by a user-defined constant.

G – Map profile on to new pressure grid

Self axplanatory.

W – Set water vmr profile to pr-um

Normalises water vapour VMR profile to the required column amount in units of precipitable microns.

V - output single gas VMR profile

Output new profile to disk containing only the P, T, H information and a single VMR profile.

3. Dust/Aerosol Profiles

Dust/Aerosol profiles are read in from external '.prf' files which have a similar format to temperature/pressure '.prf' files. The dust '.prf' profiles are defined in terms of specific concentration, i.e. number of dust particles per gram of atmosphere. Up to ten different dust particle profiles can be defined at present in a single '.prf' file. The dust profiles may be written by the program Dust_profile or by hand. At present Dust_profile can generate homogeneous, steady state and simplified cloud model profiles. These are defined as follows:

1. Homogeneous	=	Constant specific concentration with height
2. Steady State	=	Developed for Mars, specific concentration is constant at low altitudes and then falls off at a rate depending on updraft/particle size etc.
3. Simple Clouds	=	Define base height, specific concentration at base and scale height of vertically extended cloud.
4. Better Clouds	=	Define base and cloud top pressures, fractional scale height and required integrated optical depth of vertically extended cloud.

Other distributions can be hacked in as necessary or by manually editing an existing dust '.prf' file. Note that to generate the dust profiles, Dust_profile must read the corresponding temperature/pressure profile. The format of a dust '.prf' file is as follows:

```
NAME
NPRO,NDUST
DUST(1,1)  DUST(1,2) . . .  DUST(1,10)
DUST(2,1)  DUST(2,2) . . .  DUST(2,3)
   "          "         . . .    "
   "          "         . . .    "
DUST(NPRO,1)  DUST(NPRO,2) . . .  DUST(NPRO,3)
```

Where:

NAME	CHAR	name of the file
NPRO	INT	number of vertical levels
NDUST	INT	number of dust types
DUST(I,J)	REAL	dust specific concentration of type J at level I.

When Radtrans is run, it looks to see if any dust profiles have been read in and if so prompts the user for the name of the cross-section file. This file contains the extinction cross section of the particles in cm^2 and the single scattering albedo for a number of wavelengths across the spectral interval. The value in each calculation bin

is then determined by quadratic interpolation. The cross-section file may be written by either of the programs Write_xsec or Makephase or by hand. Makephase calculates the phase functions as well as the extinction x-sections of scattering particles. The spectral calculation programs then use the specific concentration profiles and cross-section spectra to calculate the optical thickness of each path due to dust for each spectral bin.

N.B. The default units of RADTRANS are that the aerosol concentrations in the '.prf' files are in units of specific density (number of particles per gram) which is a hangover from Mars radiative transfer days and is not very friendly. The default units of the '.xsc' cross-sections are cm^2. When Path is run, the number density of dust per layer is calculated and converted to units of particles/cm^2. At the end of the day the units are somewhat arbitrary since Radtrans multiplies the number of particles/cm^2 in each layer by the particle cross-section to calculate the optical path and it's only the optical path that really matters! It is thus often (very) convenient to use the fourth model of Dust_profile to normalise the specific density profile of the cloud such that the values in the associated '.xsc' file then equate to total optical depth of the cloud at different wavenumbers rather than the extinction x-section (cm^2) of each particle .

4. Path

Once the atmospheric profiles of the required planet have been defined for a radiative transfer calculation, the atmosphere must be split up into *layers*, so that mean properties can be evaluated. A number of *paths* through these layers must then be defined. The program which performs this function is <u>Path</u> which writes a driver file (.drv) which defines <u>all</u> the details of the calculation for <u>Radtrans</u> and later <u>Pl_spec</u>. <u>Path</u> takes as its input a '.pat' file (described in this section) which contains the details of the atmosphere, layers and paths and computes the absorber amounts, pressures, temperatures etc. in all the layers and then in all the paths. <u>Path</u> also sets the *calculation types* which are discussed in section 4.3. <u>Radtrans</u> calculates the transmission, absorption or emission spectra of all the paths defined in the '.drv' file and outputs these to a binary '.out' file. The *calculation types* are only used by the post processor <u>Pl_spec</u> to control how the path calculations are combined to calculate, for example, the radiance seen in the sideband of a PMR or, as another example the weighting function.

4.1 Layers

To perform radiative transfer calculations, the atmosphere in question must first be split into a number of vertical layers. <u>Path</u> reads the temperature/pressure/vmr profiles and the dust profile (if defined) and then using the subroutine `layers.f` splits the atmosphere into a number of discrete layers. It can do this in a number of ways:
      1) Split by equal changes in pressure through atmosphere
      2) Split by equal changes in log(pressure)
      3) Split by equal changes in height
      4) Split by equal changes in path length along viewing angle

The viewing angle φ is defined as the angle of the line of sight of the calculation with respect to the normal at the bottom of the lowest layer.

The viewing angle is also used when calculating the mean properties of the layers. If the atmosphere is going to be used for a limb calculation, φ would be set at 90° and the layer properties calculated for a limb path to the bottom of the lowest layer. If the atmosphere is going to be used for a nadir calculation, φ would be set at 0°. The path lengths through the layers are then determined in order that the mean properties of the layers may be calculated. This may be done in one of two ways:
      1) The mean temperature, pressures (for each gas) and specific dust concentration q (particles/g) are set at the interpolated values at the middle of the layers as seen along the viewing angle. The total absorber amounts in the layer as seen along the viewing angle is then calculated as:

$$U = \hat{k} \times \frac{P}{T} \times (S_1 - S_2)$$

where P is the mean pressure (atm), T is the mean temperature (K) and $(S_1$-$S_2)$ is the viewing angle path length (km) through the layer. The calculated absorber amount units are in (molecule/cm^2) and hence the *modified Boltzman constant* is calculated as:

$$\hat{k} = 1.013 \times 10^5 \times 10^{-6} \times 10^5 \, / \, k = 7.34339 \times 10^{26} KJ^{-1}$$

The dust amount in the layer along the viewing angle (particles cm^{-2}) for each particle type is calculated as:

$$D = q \times U \times \frac{M}{N}$$

where q is the mean specific dust concentration (particles/g), U is the total atmospheric amount in the vertical layer (molecules/cm^2), M is the mean molecular weight of the atmosphere in grams and N is Avogadro's number.

2) The mean temperature and pressure in the layer are calculated as the absorber amount weighted means along the viewing angle path. The integration is performed using Simpson's rule with the layer split into 101 points along the path. The absorber amount and dust amount are similarly integrated along the line of sight.

The paths through the layers are shown in Fig.4.1.
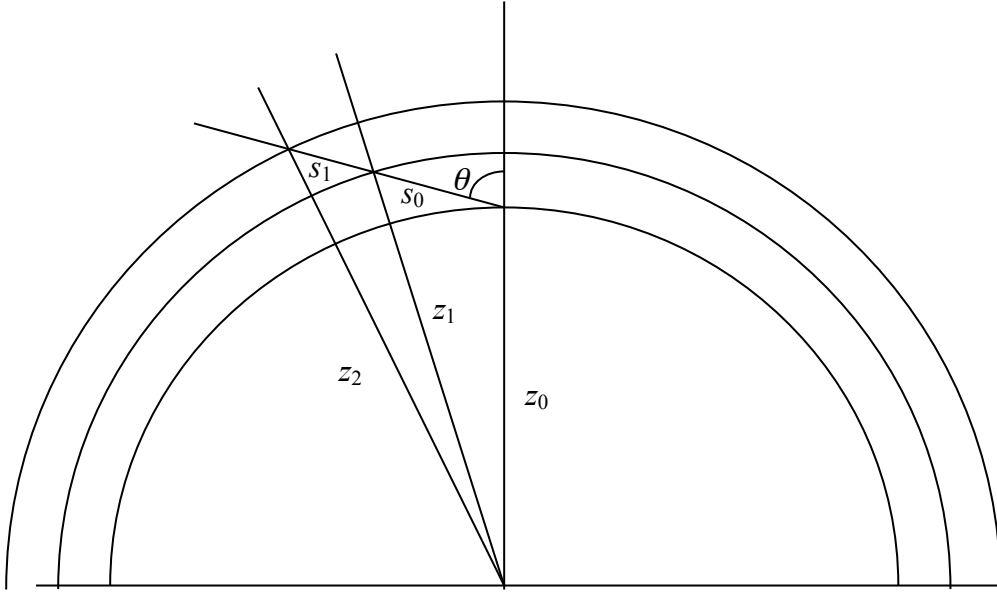
Fig 4.1 Paths, Layers and Scaling Factors

Referring to that diagram, the first path s_0 may be calculated from the cosine rule as:

$$z_1^2 = z_0^2 + s_0^2 + 2z_0 s_0 \cos \theta$$

Solving the quadratic in s_0:

$$s_0 = \sqrt{(z_1^2 - z_0^2 \sin^2 \theta)} - z_0 \cos \theta$$

Other lengths are calculated in the same way. Once the mean pressures and temperatures of the layers, together with the gas absorber and dust absorber amounts have been calculated, the amounts are scaled back using the geometric scaling factor to simulate a vertical layer. For the example above, the scaling factor is:

$$SF = \frac{s_0}{z_1 - z_0}$$

Applying this scaling factor means that the layers can still be used for either nadir or limb calculations though of course it is better to calculate the layers in the manner closest to the majority of paths defined through the atmosphere.

At this stage *cell* 'layers' are also defined which simply contain the mean pressures, temperatures and absorber amounts of any instrument cells (e.g.for simulating PMR or SCR instruments). These 'layers' can be combined with atmospheric layers into *paths*.

4.2 Paths

The primary output from <u>Radtrans</u> are calculated transmission, absorption or emission spectra for a number of *paths* defined in the driver file.

For atmospheric paths, once the layer properties have been calculated, paths are determined by the subroutine `atm.f`. The viewing angle, bottom layer height and number of the lowest layer are defined in the '.pat' file and `atm.f` selects which layers (atmospheric and cell) are to be used in the path (and in what order) and calculates the appropriate scaling factor using the previous equations.

Paths through cells may also be defined and combined as required with atmospheric paths to calculate PMR (Pressure Modulator Radiometer) or SCR (Selective Chopper Radiometer) outputs.

The number of paths in a driver file may vary from one for a simple transmission calculation to many tens of paths for weighting function calculations. However, for each path a model identifier is defined which determines the output of <u>Radtrans</u> for that path. The model identifiers (IMOD) are defined as:

0 = Transmission	(Atmosphere)
1 = Absorption (gives improved accuracy for low absorption)	(Atmosphere)
2 = Emission	(Atmosphere)
3 = Emission but uses source function at centre of current bin (much faster)	(Atmosphere)
4 = The total transmission computed during the last emission calculation (so that surface radiance may be added)	(Atmosphere)
5 = Single cell transmission	(Cell)
6 = PMR sideband transmission	(Cell)
7 = The last *cell* output path multiplied by the last 'Atmosphere' output path N.B. <u>Radtrans</u> won't check to see if last calculation was an absorption or transmission.	(Combined)
8 = The product of two previous outputs paths.	(Combined)
9 = The product of a single layer transmission and the output of the previous path (this is useful for speeding up calculation of transmission weighting functions)	(Atmosphere)

10 = As 2 but assumes transmissions calculated by single
    curtis-godson path   (Atmosphere)
11 = As 10 but uses source function at centre of bin   (Atmosphere)
12 = PMR Wideband transmission   (Cell)
13 = SCR Sideband transmission (1-Tc)   (Cell)
14 = SCR Wideband transmission (1+Tc)/2   (Cell)
15 = Scattering near-nadir calculation   (Atmosphere)
16 = Single-scattering calculation   (Atmosphere)
17 = Thermal emission into hemisphere   (Atmosphere)
18 = Thermal emission into hemisphere but using source
    function at centre of current bin   (Atmosphere)
19 = As 17 but using the Curtis-Godson approximation (Atmosphere)
20 = As 18 but using the Curtis-Godson approximation (Atmosphere)
21 = Net flux calculation (simple thermal emission)   (Atmosphere)
22 = Scattering limb calculation (calculates internal field) (Atmosphere)
23 = Scattering limb calculation (precalc. internal field)   (Atmosphere)
24 = Net flux (up – down) calculation (scattering)   (Atmosphere)
25 = Upwards flux (internal) calculation (scattering)   (Atmosphere)
26 = Upwards flux (top) calculation (scattering)   (Atmosphere)
27 = Downwards flux (bottom) calculation (scattering) (Atmosphere)
28 = Single-scattering calculation (spherical atmosphere) (Atmosphere)

4.3 Calculation Types

The calculation types are defined by <u>Path</u> and are used only by <u>Pl_spec</u> to combine and plot the output of <u>Radtrans</u>. They are not used by the radiative transfer routines at all. Each 'calculation' is encoded in a single integer code ITYPE and passed via the driver file to <u>Pl_spec</u>. In general one bit in ITYPE corresponds to one of the logical variables where possible to simplify(!) evaluation of ITYPE. Calculation codes defined are:

- Atmospheric Codes have bit 7 (128) zero
    bit 6 (64) 1 if limb path
    bit 5 (32) 1 if reflecting path (down then up)
    bit 4 (16) 1 if transmission of single atmospheric path
    bit 3   (8) 1 if Curtis Godson
    bit 2   (4) 1 if weighting function
    bit 1   (2) 1 if emission calculation
    bit 0   (1) 1 if calculate 1 minus transmission

- Non atmosphere codes have bit 7 (128) 1
    128   Single cell transmissions
    129   PMR - two pressure, two temperature approximation
    130   PMR File. Pressures/temperatures defined at a
                number of points in the cycle.
    131   SCR

- Combined cell/atmosphere codes
    160 is a combined cell and atmosphere - produces combined output

161 is a reflecting path and cell - produces combined output

- Reflecting layer atmosphere codes
  200 is a reflecting layer calculation with no cells
  201 is a reflecting layer calculation with cells
  (Both these calculations allow selection of paths in Pl_spec.)

- Scattering Calculation
  256   Full doubling and adding scattering scheme

4.4 Path File Format

Path takes as its input simple ASCII path files (.pat) which contain a series of keywords and parameters. The '.pat' files are block structured, each block starting with a keyword and ending with a blank line. These path files may either be generated directly using an ASCII editor or generated interactively using the program Write_path which prompts for the appropriate inputs.

An example of a typical '.pat' file for Radtrans and an explanation of the possible keywords which may be processed by Path is given below. Note that the 'interval' and 'line data' formats have been slightly modified from [R10].

```
interval
1400 1495 5 5
1 1 5

spec data
~irwin/radtran/linedata/new/tim2

model ~irwin/pmirr_cal/tim2

dust model ~irwin/pmirr_cal/dust

dust spectra testdust

layer
nlay 35
layang 90.
layht 20.
laytyp 1
layint 1

atm
limb 1
therm
nowf
```

```
nocg
noabsorb
binbb
nobroad

reflatm
angles 45.0 30.0 8
nocg
noabsorb

cell
gases 1
1 0 1.0
length 10.
pmr twop
0.015449  292.0
0.020088  308.0
```

Defining the wavenumber interval (essential keyword)

The keyword is **interval**.
The first parameter line is the wavenumber minimum, maximum, output spacing and resolution in wavenumbers.

N.B. The final resolution parameter does not apply to LBL calculations (ICONV=0,1). For the adaptively integrated LBL model (ICONV=1) FWHM is hardwired to DELV as described in section 5.1.

The second parameter line is the spectral model type ICONV followed by two parameters PAR1 and PAR2 which define additional information for the particular spectral model used.

Defining the spectral data (essential keyword)

The keyword is **spec data**. The only parameter is a spectral data key file name (line data, band data, or *k* data) ('.key' extension, though this may be omitted).

The line data files are set up and manipulated by the programs: Makedb, Scan, Check_index, Li_lines, Pl_lines, Select, Cp_lines and Summary. These line data manipulation programs are described in section 7.

The band data files may be set up with the program Makeband.

The *k* data files may be set up with the program Calc_ntable.

Defining an atmospheric model (essential for atmospheric calculations)

The keyword is **model** and again the parameter is just the filename of the atmospheric model to use ('.prf' extension though again this may be omitted). In this case the filename is on the same line as the keyword.

The model files are produced and manipulated with the program Profile and have the extension '.prf'. They contain the pressure, temperature and gas VMR profiles as a function of height. The profile files are described in section 2.

Defining a dust or aerosol model (optional)

The keyword is **dust model** followed by the name of the dust/aerosol profile. The dust/aerosol profiles are explained in section 3.

Defining the dust optical properties (essential if dust is defined)

The keyword is **dust spectra** followed by the name of the dust/aerosol cross section ('.xsc') file. This file has the following form:

```
2
1900.00        1.0E-10        2.4E-10
               0.5            0.1
2100.00        2.0E-10        3.0E-10
               0.5            0.1
```

The first line contains the number of dust types. It must be equal to the dust types defined in the dust model.
Subsequent lines contain the cross-sections (cm^2) for each particle type for as many wavenumbers as desired followed by the single scattering albedoes at the same wavenumber. The subroutine reading in the data automatically detects the end of the file.

Calculating atmospheric layers (essential for atmospheric paths)

The keyword is **layer**. Path splits the atmosphere into a number of layers. This is separate from calculations of the atmospheric paths so that you can have several paths through the same set of layers (e.g. for multiple tangent heights). There are multiple parameters on additional lines each defined by additional keywords. The keywords are:

nlay followed by the number of layers to split the atmosphere into.
layht followed by the height of the base of the lowest layer (km) relative to a reference level defined in the '.prf' file.
layang followed by the angle from the zenith for splitting into layers. For example you'd use 0º for nadir calculations but 90º for limb. The stored layer properties are scaled by the cosine of this angle so that they approximate to the nadir case.
laytyp followed by an integer specifying the layering type.
       0=split by equal changes in pressure over layers
       1=split by equal changes in log pressure over layers
       2=split by equal changes in height over layers
       3=split by equal changes in path length at **layang** over

layers

layint followed by an integer specifying how to integrate absorber amount, pressure and temperature vertically over the layer.

   0=use values at centre of layer

   1=use curtis godson paths for a gas with constant mixing
      ratio.

1 is usually better but 0 is useful for intercomparisons.

In the example above we have defined 35 layers starting from an altitude of 20km by splitting the atmosphere by equal changes in log(pressure). The mean pressure, temperature and absorber amounts of the layers are calculated using the Curtis-Godson approximation assuming that most of the paths will be limb-views to a tangent height of 20km

Calculating atmospheric layers with discrete cloud layers

The keyword is **nlayer** and the option is very similar to **layer**, but tailored for cloudy atmospheres such as that of Jupiter's. Subsequent keywords are:

layint followed by an integer specifying how to integrate absorber amount, pressure and temperature vertically over the layer.

   0=use values at centre of layer

   1=use curtis godson paths for a gas with constant mixing
      ratio.

1 is usually better but 0 is useful for intercomparisons.

layang followed by the angle from the zenith for splitting into layers. For example you'd use zero for nadir calculations but 90 for limb. The stored layers are scaled by the cosine of this angle so that they're always similar to the nadir case. You might use layang=0 for multiple limb paths or for simulating someone elses code.

nlaybot followed by the number of clear gas layers from the base of the profile to the bottom of the first cloud.

nlayg followed by the number of clear gas between cloud layers

nlaytop followed by the number of clear gas layers from the top of the last cloud layer to the top of the profile.

layht followed by the height of the base of the lowest layer (km) relative to a reference level defined in the .prf file.

ncloud followed by the number of main cloud decks

Subsequent line contains the cloud base height (km), cloud top height (km) and number of intermediate cloud layers.

Next lines contain the number of cloud particles per cm^2 for a vertical path for each sub layer.

Sequence then repeats for further cloud decks.

E.g. :

nlayer

```
layint   1
layang   0
nlaybot  8
nlayg    8
nlaytop  8
layht  -90.00
ncloud   3
  -45.00000    -44.00000      1.000000
   1.1200000E+13
  -15.00000    -14.00000      1.000000
   2.9099999E+10
   16.00000     17.00000      1.000000
   7.7999997E+10
```

If bi-modal clouds are required then instead of using **nlayer**, **mlayer** may be used which has an identical format with the exception that two columns now contain the number of cloud particles per cm^2 for a vertical path for each particle type in each sub layer.

Defining an atmospheric path (essential for atmospheric calculations)

The keyword is **atm**. Additional parameters are defined on additional lines each defined by keyword. Most keywords can be negated by preceding them with a **no**. The keywords are

limb specifies a limb calculation and is followed by the number of the bottom layer to use. 1 is the bottom layer defined by **layer** above. Here paths are calculated to the tangent height and then beyond.
nadir specifies a nadir, or near-nadir calculation and is followed by the angle from the zenith and the bottom layer to use. For an instrument looking vertically downwards, the angle is set to 0°. For an instrument on the surface looking upwards the angle is set to 180°. In this case paths stop at the lowest layer. **N.B.** It is possible to add a third integer, IPZEN, on this line to indicate where the zenith angle is defined: 0) at bottom of deepest layer; 1) at the 0km altitude level; or 2) at the very top of the atmosphere.
(no)therm specifies a thermal emission calculation.
(no)hemisphere specifies a hemispherically-integrated thermal emission calculation using exponential integrals.
(no)wf specifies a weighting function calculation. Note that this forces **notherm**.
(no)cg specifies Curtis Godson paths are to be used.
(no)scatter specifies a full doubling and adding scattering calculation. If scattering is defined then when Radtrans is run there must exist a '.sca' file containing details of the scattering calculation. This will be explained more fully later.
(no)single specifies a single-scattering calculation for a plane-parallel atmosphere.
(no)sphsingle specifies a single-scattering calculation for a spherical atmosphere.
(no)netflux specifies if a net flux ($F_{up} - F_{down}$) is to be calculated. This works with scattering on or off. Fluxes are computed at the base of each layer.
(no)upflux specifies if the upward flux only is to be calculated. This only works with scattering on. Fluxes are computed at the base of each layer.

(no)outflux specifies if the upward flux at the very top of the atmosphere is to be calculated. This only works with scattering on.
(no)botflux specifies if the downward flux at the bottom of the atmosphere is to be calculated. This only works with scattering on.

(no)absorb specifies that absorption rather than transmission is to be calculated for non thermal calculations. Needed for small absorptions when adaptive LBL integration (ICONV=1) is selected.
(no)binbb specifies that the Planck function is to be computed at bin centres (see interval above) not each wavenumber which is much faster but less accurate.
(no)broad specifies that an emission calculation is to be performed outside the RT calculation to simulate broad band calculations.

The defaults are limb, nowf, nocg, therm, binbb, noabsorb, noscatter, nonetflux, noupflux, nooutflux, nobotflux.

In the example atmosphere listed above we have defined the view path to be a limb path with a tangent height at the bottom of layer 1. We have specified a thermal radiance calculation where the Planck function is calculated for the centre of the bin only.

Defining a reflected atmospheric path
The keyword is **reflatm**. Additional parameters are defined on additional lines each defined by keyword. Most keywords can be negated by preceding them with a **no**. The keywords are:
angles followed by the incident and reflected angles and the number of the layer from which the reflection occurs (reflection happens at base of layer)
(no)cg Sets Curtis Godson flag as required
(no)absorb Sets absorption flag as required

Defining a gas cell

The keyword is **cell**. Additional keywords follow. These are:

gases followed by the number of gases in the cell followed on subsequent lines by the identifier, isotope and volume mixing ratio of each gas. The identifier is NOT always the same as HITRAN and is listed in Appendix A. Isotope zero includes all isotopes at terrestrial ratios. The RT routines correct line strengths for explicitly specified isotopes as if they were 100%
dop followed by a doppler shift for the cell (cm^{-1})
length followed by the cell length (cm)
sngl specifies single cells followed by the number of single cells.
Subsequent lines hold the pressure (atm) and temperature (K) (in that order) for each cell.
pmr twop specifies a Pressure Modulator Radiometer (PMR) calculation assuming a two pressure approximation
The next line holds the low pressure (atm) and associated temperature (K).
The next line holds the high pressure and associated temperature.
pmr file specifies that a PMR cycle is held in a file

The next line holds the file name and the file contains pressure and temperature for each point in the cycle on separate lines. The format of the files is free-form and <u>Path</u> automatically detects the end of the file.

scr specifies a Selective Chopping Radiometer (SCR) which alternately views the atmosphere directly or through a single cell. The next line holds the pressure and temperature of the cell.

N.B. Cell calculations can only be done on LBL calculations. <u>Radtrans</u> does not explicitly pick this up.

<u>Defining single atmospheric paths</u>

Instead of defining layers and atmospheres, single atmospheric paths may also be defined for various tests. <u>Radtrans</u> treats atmospheric and cell layers differently so this is not the same as defining single cell paths.
1.
The keyword is **sngatm** followed optionally by a file name. If there is no file name the input is read from the subsequent lines.
The first line should contain the keyword **absorb** or **noabsorb** as described earlier.
The following line contains the number of gases, pressure (atm), and temperature (K). Each gas is defined on a subsequent line containing identifier, isotope, volume mixing ratio and amount (molecule cm^{-2}). If the amount is < 1.e10 it is assumed to be a path length in km.
The next line contains the number of dust types NCONT. If NCONT > 0 then the next lines contain the name of the dust cross-section file and the dust amount in the path (number of particles/cm2).

<u>Setting the error limit for adaptively integrated lbl calculations</u>

The keyword is **error** followed by the limit (%) on the same line. This is the limit for terminating adaptive integrations. The default is 1%.

<u>Defining how to process line data for lbl calculations</u>

The keyword is **process** followed by the gas id, isotope id and an integer IPROC to define how to do the processing. At present two options are available:

IPROC=0   Default   Assume Voigt broadening for lines
IPROC=1   Assume Sub-Lorentzian line broadening (CO_2)
IPROC=2   VanVleck-Weisskopf lineshape (far-IR)
IPROC=3   Rosenkrantz-Ben-Reuven lineshape for far-IR NH_3
IPROC=4   Assume Lorentz-broadened lines
IPROC=5   Levy et al. (1994) PH_3 line shape.
IPROC=6   Rosenkrantz-Ben-Reuven NH3 lineshape
                              same as IPROC=3 except default is voigt lineshape
                              instead of sub-Lorentzian
IPROC=7 CO_2 Sub-lorentzian line shape for Venus
IPROC=8   Burch CO_2 Sub-lorentzian line shape
IPROC=81   Burch CO_2 Sub-lorentzian line shape (renormalised)
IPROC=9   Hartmann   Sub-lorentzian   lineshape   for   methane
                              modified for Titan by de Bergh et al.

IPROC=10   Hartmann et al. Sub-lorentzian lineshape for methane in
                  a hydrogen atmosphere

Tidying the driver file

The keyword is **(no)clrlay** and tells Path to remove any unused layers before creating
the driver file. **clrlay** is the default.

Combining atmospheric and cell paths

The keyword is **(no)combine** and tells Path to add additional paths combining all
combinations of atmospheric and cell paths that have been defined. **combine** is the
default. The combination is always at the end so you can't just combine those defined
so far.
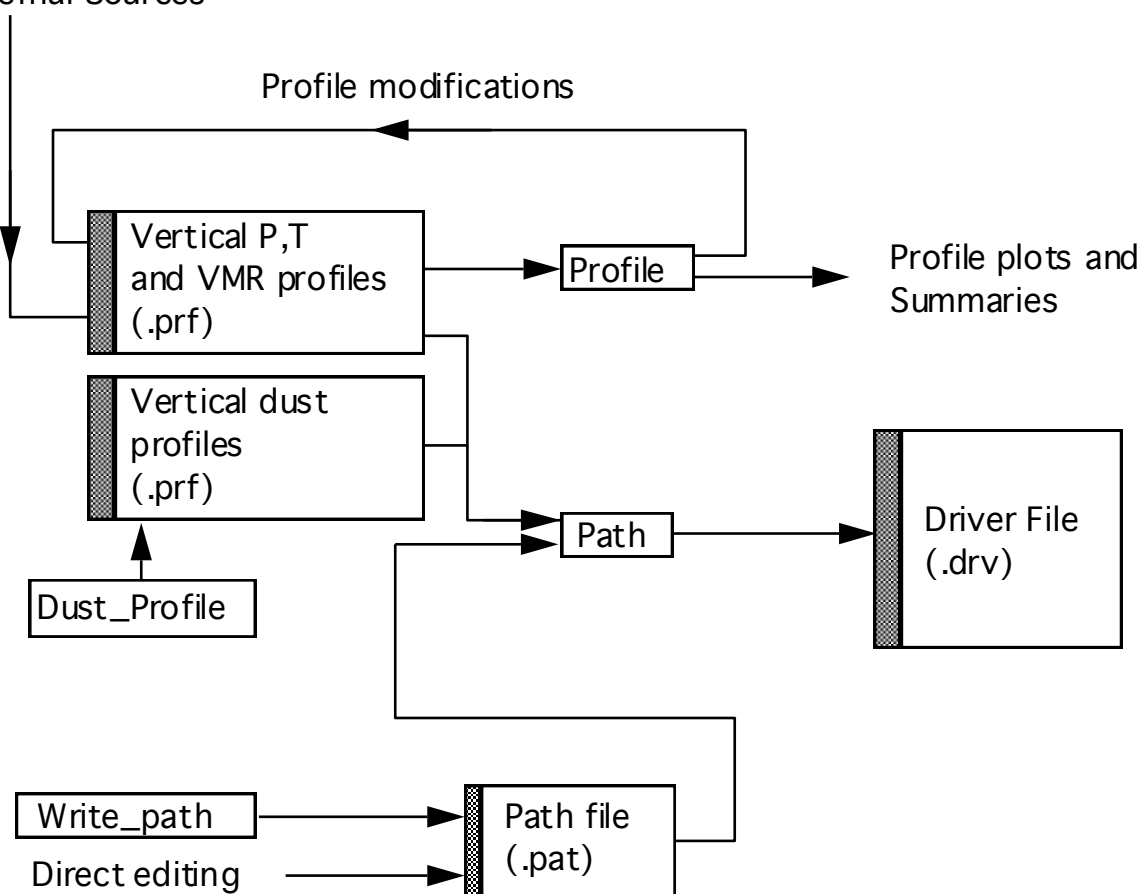
Fig.4.2 Generation of RADTRANS driver files using Path

4.5 Running <u>Path</u>

<u>Path</u> is invoked by simply typing 'Path'. The program prompts for the name of the path (.pat) file and outputs the generated driver (.drv) file together with a name (.nam) file which is useful for running RADTRANS in background. For example if the path file is called surf.pat, Path will generate surf.drv and surf.nam, the latter file is usually just one line long and containing the name of the driver file. However in some case it may also contain extra parameters not easily passed elsewhere such as whether an equilibrium ortho:para H_2 fraction is assumed or a deep 3:1 ratio [R15].

4.6 Changes in driver file format from Genlbl

RADTRANS has been designed to look as much as possible like the previous line-by-line model GENLBL in order that old calculations can be repeated using the same driver files. There are only three ways in which the driver files are different:

1) Since the treatment of dust/aerosols has been changed, NCONT must be set to 0 if there is no dust instead of 1 as in GENLBL.

2) If dust profiles are defined then an extra line follows the line containing ICONV and NCONT which contains the name of the dust extinction cross-section file.

3) Dust amounts in the driver file are now in units of particles/cm2 instead of optical depth.

The lines which contain details of any required filter profiles are still present and RADTRANS will use them in the same way that GENLBL does if required. However it has only been left in to reduce the number of changes between the two systems and it is now recommended that this function should be performed as part of the post-processing in <u>Conv_spec</u>.
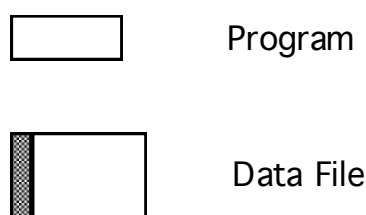
4.7 Non-LTE

User specified NLTE parameters are also included in RADTRANS. To use the facility the driver file must again be manually edited. Either before, after or instead of the hydrogen absorption parameters insert the line:

NLTE 0

The number specified is a code to indicate the type of NLTE required: At present there are two possibilities:

      0   Perfect LTE
      1   Assume NLTE is as that estimated by Glenn Orton for
            Saturn and Titan. Scaling for Planck function is:

$$F = \frac{P(atm) \times 10^6}{1 + P(atm) \times 10^6}$$

5. <u>Radtrans</u>

As outlined earlier in Section 1, <u>Radtrans</u> calculates transmission, absorption or
emission spectra for the paths defined in a '.drv' driver file constructed from a '.pat'
path definition file by the program <u>Path</u>. Spectra are calculated in wavenumber space
only and the calculated spectra are written to binary '.out' files. Hence the output files
must be processed using one of the three post-processors <u>Conv_spec</u>, <u>Li_spec</u> and
<u>Pl_spec</u> which are described later and which may transform the spectra to wavelength
space (if required) and write the outputs in ASCII format. The spectral details of the
calculation are given in the first line of the '.drv' driver file where the wavenumber
minimum, spacing, number of points and bin width are defined. The subsequent line
contains two extra parameters PAR1 and PAR2 whose meaning depends on the
spectral model required which is defined by the integer ICONV. Table 5.1
summarises the model options which are dealt with more fully in the following
sections.

Spectral Model	ICONV	PAR1	PAR2
Line-by-Line (LBL) Calculations			
Line by line model - monochromatic	0	WING	VREL
Line by line model - integrated	1	WING	VREL
Band Model Calculations			
Non-overlapping line model (linedata)	10	EquivW code	-
General random band model (linedata)	11	EW	1
Malkmus-Lorentz band (linedata)	11	EW	2
Goody-Lorentz band (linedata)	11	EW	3
Godson-Lorentz band (linedata)	11	EW	4
Malkmus-Lorentz band (band data)	12	-	-
Goody-Lorentz band (band data)	13	-	-
Godson-Lorentz band (band data)	14	-	-
Goody-Voigt band (band data)	15	-	-
Correlated-*k* Calculations			
Correlated-*k* model (linedata)	20	WING	VREL
Malkmus-Lorentz Corrk (band data)	21	1,2,3	-
Goody-Voigt Corrk (band data)	22	0,1	
Exponential Sum model (not recommended)	23	0,1	-
Correlated-*k* model (look-up table)	24	-	-

Table 5.1 <u>Radtrans</u> Spectral Models
Heavily shaded options not yet completed

5.1 Line by Line Models   **ICONV<10**

Here:
        PAR1=WING
        PAR2=VREL

The spectral region defined in the path and driver files is extended to VMIN-VREL to VMAX+VREL, to cover line wings from lines outside the interval and divided into bins WING wavenumbers wide (Fig 5.2). Unlike GENLN2, RADTRANS ignores all lines outside the range VMIN-VREL to VMAX+VREL. Line data is stored in blocks corresponding to the WING-sized bins. At a particular wavenumber, the absorption by lines from the bin at that wavenumber and the two adjacent bins are treated explicitly. Absorption by lines from other bins is calculated by calculating the contribution to the middle and each end of the bin and fitting a quadratic since at this distance from the line centres it is assumed that we are in the Lorentzian (and thus quadratic) tail of the line shape (Figure 5.1). This greatly increases the calculation speed.
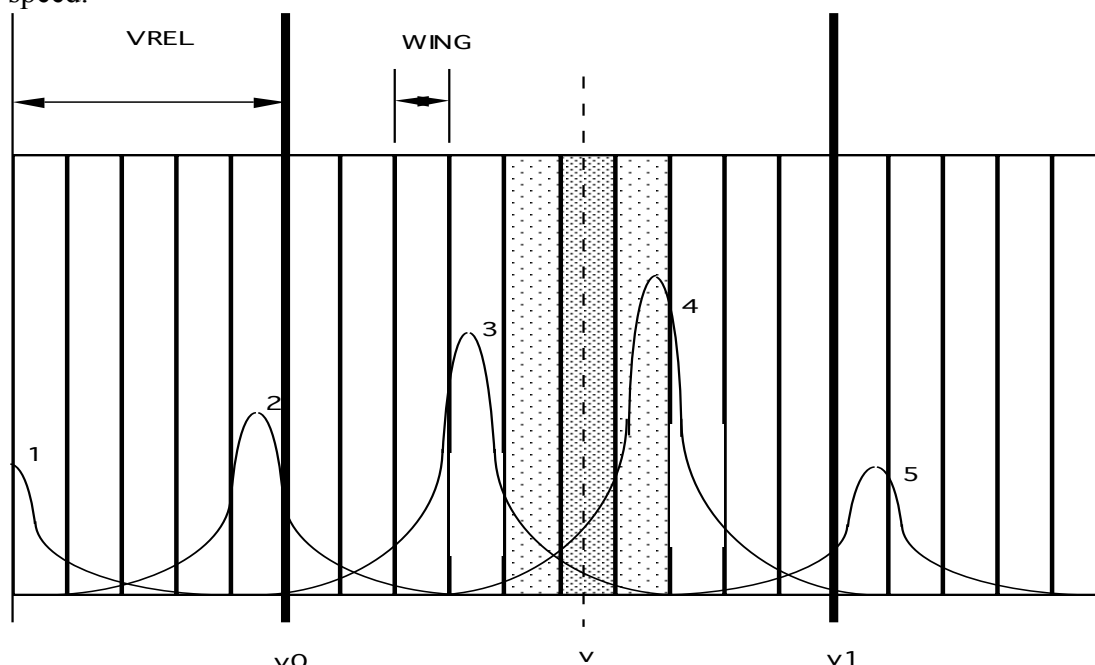
Fig 5.1 Treatment of lines by RADTRANS line-by-line code.

Line-by-line calculations may either be performed monochromatically (ICONV=0) or integrated over square bins of width DELV using a cunning adaptively integrating Simpson's rule scheme (ICONV=1). For the example path file given, Radtrans will integrate the spectrum over 5 cm^{-1} (i.e. DELV) wide regions centred on the required output wavenumber. The first region is thus from 1397.5 to 1402.5 cm^{-1} giving the average transmission or radiance at 1400 cm^{-1}. **[Note - the integration steps and the bin sizes are not related. Hence an integration may cover many bins or just one.]**

The adaptive integration scheme works by first performing a Simpson's rule integration over the square bin of width DELV using 20 points. The number of intervals is then doubled and the integration evaluated again. The scheme tests for convergence by comparing the last two estimates. If they agree to within the

convergence limit (set by **error** in the '.pat' file and defaulting to 1%), the integration stops. Otherwise the number of integration steps is doubled and so on.

5.1.2 Line handling in Radtrans

Previous releases of RADTRANS included a program <u>Radtran</u> which used a clever line-binning scheme for LBL calculations to reduce the amount of data held in memory. However this scheme considerably increased the complexity of the code, making debugging and error analysis difficult. In addition, once the internal memory limit was exceeded a bin-swapping procedure was adopted which greatly increased the computation time. Hence it was decided to no longer support <u>Radtran</u>. Instead <u>Radtrans</u> has been developed which behaves exactly in the same way as <u>Radtran</u>, except that all the lines for all the bins are read in first. If the required calculation covers more than 10000 lines, which is currently the limit of <u>Radtrans</u>, the calculation range must be split up into separate spectral ranges.

5.2 Band Models   9 < ICONV < 20

Band models are useful for low- to medium-resolution spectral modelling. They represent a significant simplification over line-by-line models in that relative position of molecular absorption lines in a spectral interval is ignored. In the models included here it is assumed that the lines are randomly distributed across the interval and thus the equivalent widths of the lines (i.e. integrated absorptions) may simply be added. The models are generally not so accurate as line-by-line models but have the advantage that they are **much** faster and in some cases, where line data is not well known, are the only option available. While dust absorption is easily included, dust scattering is not possible with band models (since they must use the Curtis-Godson approximation), which is why the correlated-*k* models have also been developed. For a full discussion of band and correlated-*k* models, see [R1]. Two generic types of band models are included, those that use line data and those that use band data.

5.2.1 Band Models using Line data (ICONV=10, ICONV=11)

For every central output wavenumber, each line in the integration range of width FWHM is read in and the equivalent width calculated before the mean transmission is calculated using a number of possible approximations.

The equivalent width *W* of each line may be assumed to be that of:
   a) a Lorentz line only W_L.
   b) a Lorentz line combined with a Doppler line using the equation
      $W = \left(W_L^2 + W_D^2 - \left(W_L \times W_D / Sm \right)^2 \right)^{1/2}$ to simulate Voigt lines. (Here *S* is the line strength and *m* is the absorber amount)
   c) interpolated from a Voigt width look-up table [R2].

Both Lorentz and Doppler equivalent widths may be calculated using either the Weak or Strong assumptions or a combined width [R2].

Once the line equivalent widths have been calculated, the path transmissions may be calculated using

a) the non-overlapping line approximation where $T = 1 - \dfrac{\sum\limits_i W_i}{\Delta \tilde{\upsilon}}$

b) the general random band model where $T = \exp\left(\dfrac{-\sum\limits_i W_i}{\Delta \tilde{\upsilon}}\right)$

c) the Malkmus Lorentz band model [R3]. Here

$$T = \exp\left(-\frac{s^2}{2w}\left[\left(1 + \frac{4w^2}{s^2}\right)^{1/2} - 1\right]\right)$$

where

$w = \dfrac{1}{\Delta \tilde{\upsilon}} \sum\limits_i S_i m$ is the weak limit of the optical thickness and

$s = \dfrac{2}{\Delta \tilde{\upsilon}} \sum\limits_i \sqrt{S_i \alpha_i m}$ is the strong limit.

d) the Goody Lorentz band model [R3] where

$$T = \exp\left(-\left(w^{-2} + s^{-2}\right)^{-1/2}\right)$$

e) the Godson Lorentz band model [R3] where:

$$T = \exp\left(-x\left\{e^{-y}I_0(y) + 2ye^{-y}\left[I_0(y) + I_1(y)\right] - 1\right\}\right)$$

where

$$x = 2\pi \frac{s^2}{w} \quad \text{and} \quad y = 2\pi \frac{w^2}{s^2}$$

and I_0 and I_1 are Bessel functions.

For these models, the path file parameter PAR1 contains the code for calculating the equivalent width WIDMOD. WIDMOD is an integer whose three digits when converted to base 3 are the codes COM_MOD, LOR_MOD and DOP_MOD respectively. For example, the number 7 is 021 in base 3 which sets COM_MOD=0, LOR_MOD=2 and DOP_MOD=1. These codes are identified below.

COM_MOD. This code defines whether a voigt line shape is to be assumed and if so how:
    COM_MOD = 0, Equivalent width used is that of a Lorentz line only
    COM_MOD = 1, Equivalent width used is that of a Lorentz line combined with a Doppler line using the equation
$W = \left(W_L^2 + W_D^2 - \left(W_L \times W_D / Sm\right)^2\right)^{1/2}$ to simulate Voigt lines
    COM_MOD=2, Equivalent width used is interpolated from a voigt width look-up table [R2].

LOR_MOD. This code defines how the Lorentz equivalent width is to be calculated.
    LOR_MOD=0, Equivalent Lorentz width is assumed to be Weak.
    LOR_MOD=1, Equivalent Lorentz width is assumed to be Strong
    LOR_MOD=2, Equivalent Lorentz width is assumed to be intermediate between weak and strong as calculated by [R2]

DOP_MOD. This code defines how the Doppler equivalent width is to be calculated.

DOP_MOD=0, Equivalent Doppler width is assumed to be Weak.

DOP_MOD=1, Equivalent Doppler width is assumed to be Strong

DOP_MOD=2, Equivalent Doppler width is assumed to be intermediate between weak and strong as calculated by [R2]

The band models are defined as follows:

Model	ICONV	PAR1	PAR2
Non-overlapping Lines	10	EW code	Undefined
General random band	11	EW code	1
Malkmus-Lorentz	11	EW code	2
Goody-Lorentz	11	EW code	3
Godson-Lorentz	11	EW code	4

5.2.2 Band Models using Band data   (11<ICONV<20)

These models used band data generated off-line either by the program Makeband or from external sources. Radtrans uses these parameters to calculate the spectral transmission or radiance using one of four models [R4] (note that the absorber amount (in units of molecule-cm^{-2}) must first be multiplied by the factor 10^{-20} to cancel the factor in $k_v(T_0)$). The available models are:

Model	ICONV	
Malkmus-Lorentz	12	
Goody-Lorentz	13	
Godson-Lorentz	14	Not completed
Goody-Voigt	15	

The additional parameters PAR1 and PAR2 are not used for these models.

ICONV=12   Malkmus-Lorentz

The Malkmus-Lorentz transmission of a single gas is given by:

$$T = \exp\left(-1/\left[\frac{1}{2A} + \sqrt{\frac{1}{(2A)^2} + \frac{1}{4B}}\right]\right)$$

where:

$A = k_v(T)m$, where m is the absorber amount and

$B = \pi A \alpha_L / \delta$

The Lorentz width α_L is found from the equation

$\alpha_L = \alpha_L^0\left(q + \frac{(1-q)}{SFB}\right)\frac{P}{P_0}\sqrt{\frac{T_0}{T}}$ where q is the volume mixing ratio of the active

gas and the temperature dependence exponent is assumed to be 1/2.

The mean absorption for the conditions of the gas is given by

$$k_v(T) = k_v(T_0)\left(\frac{T_0}{T}\right)^{q_r} \exp\left[1.439 E_l\left(\frac{1}{T_0} - \frac{1}{T}\right)\right]$$ where q_r is the rotational

partition function which is 1.0 for linear molecules and 1.5 for non-linear polyatomic molecules [R5]

If more than one gas contributes to the absorption, the lines are assumed to overlap randomly and thus the equivalent widths are simply added.

Expressing $T = \exp\left(\frac{-W}{\Delta\tilde{v}}\right)$, where W is the equivalent width, then in the weak limit:

$$\text{as } k_v(T)m \to 0 \text{ , } \frac{W}{\Delta\tilde{v}} \to k_v(T)m$$

If we assume randomly oriented lines, then in the weak limit:

$$\frac{W}{\Delta\tilde{v}} = \frac{m}{\Delta\tilde{v}}\sum_i s_i$$

In the strong Lorentz limit:

$$\text{as } k_v(T)m \to \infty \text{ , } \frac{W}{\Delta\tilde{v}} \to \sqrt{\frac{4\pi k_v(T)m\alpha_L}{\delta}}$$

Again assuming randomly oriented Lorentz broadened lines:

$$\frac{W}{\Delta\tilde{v}} = \frac{2}{\Delta\tilde{v}}\sum_i(ms_i\alpha_i)^{1/2}$$

Thus equating the weak limits:

$$k_v(T) = \frac{1}{\Delta\tilde{v}}\sum_i s_i$$

and equating the strong limits:

$$\alpha_L = \frac{\delta}{4\pi k_v(T)}\left(\frac{2}{\Delta\tilde{v}}\sum_i(s_i\alpha_i)^{1/2}\right)^2$$

ICONV=13   Goody-Lorentz

The Goody-Lorentz transmission is given by:

$$T = \exp\left(-1/\sqrt{\frac{1}{A^2} + \frac{1}{B}}\right)$$

where A and B , the temperature dependencies of α_L and the mean absorption are all calculated in the same way as for the Malkmus-Lorentz model. However, here the statistical definition of α_L is different.

Equating the weak limits we again obtain:

$$k_v(T) = \frac{1}{\Delta\tilde{v}}\sum_i s_i$$

but in the strong Lorentz limit:

$$\text{as } k_v(T)m \to \infty \text{ , } \frac{W}{\Delta\tilde{v}} \to \sqrt{\frac{\pi k_v(T)m\alpha_L}{\delta}}$$

which is equivalent to $\frac{W}{\Delta\tilde{v}} = \frac{2}{\Delta\tilde{v}}\sum_i(ms_i\alpha_i)^{1/2}$ for randomly distributed Lorentz lines. Thus equating the limits:

$$\alpha_L = \frac{\delta}{\pi k_v(T)}\left(\frac{2}{\Delta\tilde{v}}\sum_i \left(s_i\alpha_i\right)^{1/2}\right)^2$$

Which is a factor of 4 bigger than for the Malkmus-Lorentz model.

<u>ICONV=14   Godson-Lorentz</u>

Still not completed! (and probably never will be!)

<u>ICONV=15   Goody-Voigt</u>

The Goody-Voigt transmission is given by the integral:

$$T = \exp\left(-2k_v(T)m\int_0^\infty \frac{V(x,y)}{1 + k_v(T)m\delta V(x,y)/\alpha_D^0\sqrt{T}}dx\right)$$

where

$$y = \frac{\alpha_L}{\alpha_D} = \frac{\alpha_L^0}{\alpha_D^0}\frac{P}{P_0}\frac{\sqrt{T_0}}{T}\left[q + \frac{(1-q)}{SFB}\right]$$

This function has to be integrated numerically.

5.2.3 Summary of <u>Radtrans</u> Band Models

The band models available in <u>Radtrans</u> are:

Spectral Model	ICONV
Non-overlapping Lines using line data	10
Random Band Models using line data	11
Malkmus-Lorentz band model (band data)	12
Goody-Lorentz band model (band data)	13
Godson-Lorentz band model (band data)	14 (not complete)
Goody-Voigt band model (band data)	15

5.3 Correlated-K models   **19 < ICONV < 30**

Correlated-*k* models share the same speed advantages of band models but may also be used for scattering calculations ([R1], [R6] and [R7]).

As far as band transmission is concerned [R1], it is of no importance where in a frequency interval Δv, a particular absorption coefficient occurs; it is sufficient to know what fraction of the frequency domain *f(k)dk*, is occupied by absorption coefficients between *k* and *k+dk*. It is thus possible to write the mean transmission function (where the transmission is a function of the absorber amount) for a homogeneous path in either of the two forms:

$$\bar{T}(m) = \frac{1}{\Delta v}\int_{\Delta v}\exp\left(-k_v m\right)dv$$

or

$$\bar{T}(m) = \int_0^\infty f(k)\exp(-km)dk$$

The form of this last equation is a standard Laplace transform and thus we can define the mean transmission function of a path to be the Laplace transform of the absorption coefficient frequency distribution:

$$\overline{T}(m) = L\big[f(k)\big]$$

Similarly, the absorption coefficient frequency distribution may be defined as the inverse Laplace transform of the transmission function:

$$f(k) = L^{-1}\big[\overline{T}(m)\big]$$

Looking at the form of the equations it becomes apparent that the change of variable and integration space need not be restricted to evaluation of the transmission. In general if G is any analytic function where:

$$\overline{G}(m) = \frac{1}{\Delta\upsilon}\int_{\Delta\upsilon} G\big[k_\upsilon, m\big]d\upsilon$$

then the following transformation applies:

$$\overline{G}(m) = \int_0^\infty f(k)G(k,m)\,dk$$

Herein lies the principle advantage of k-distribution models, once the absorption coefficient frequency distribution of the gas absorption is determined, calculations such as multiple scattering may also be performed in k-space, and as we shall see, this space is much more amenable to efficient and fast calculations.

We now define the cumulative frequency distribution $g(k)$:

$$g(k) = \int_0^k f(k)\,dk$$

Since $f(k)dk$ represent fractions of the interval where the absorption coefficient lies between k and $k+dk$, the limit of this integral as k tends to infinity is unity. If we invert this function to express k as a function of g:

$$k(g) = g^{-1}(k)$$

Then the mean transmission may then be written:

$$\overline{T}(m) = \int_0^1 \exp(-k(g)m)\,dg$$

The cumulative frequency distribution $g(k)$ is a smooth monotonic function and we define its inverse to be the k-distribution which also varies monotonically and <u>is thus easily integrated with comparatively rough quadrature</u>. Dividing the space between $g = 0,1$ into N intervals such that the variation of k with g is sufficiently well sampled, the mean transmission of a path may be calculated from:

$$\overline{T}(m) = \sum_i^N \exp(-\overline{k}_i m)\Delta g_i$$

where $\overline{k}_i$ is the mean absorption coefficient in the ith g-interval.

For band models, radiance calculations may only be performed using Curtis-Godson equivalent path approximation, thus precluding scattering. However when k-distribution models the Curtis-Godson approximation is not the only one that may be used. Referring to [R6], for an atmosphere split vertically into a number of layers, for a particular spectral interval there is a very good spectral correlation between the high and low absorption features of different layers. Hence, to a good approximation, the combined transmission of M layers may be calculated as:

$$\overline{T} = \sum_i^N \exp\left(-\sum_j^M \overline{k}_{ij}\Delta m_j\right)\Delta g_i$$

This is the correlated-k approximation and the degree of correlation is in most cases exceptionally good, and usually better than the Curtis-Godson approximation. It should be noted that to use the correlated-k approach, the number and width of g-intervals must be the same for all layers.

5.3.1 Correlated-k model using line data   (ICONV=20)

This option of <u>Radtrans</u> uses line-by-line routines to calculate the absorption coefficient spectrum of each layer and then numerically extracts a high-resolution k-distribution function which is then re-sampled to the required number and widths of the g-intervals.. This process is of a similar speed to a normal line-by-line transmission calculation but has the advantages of accurate determination of $g(k)$ and the amenability of using the k-distributions for fast multiple scattering calculations. [To numerically calculated $g(k)$, the absorption coefficient spectrum is basically analysed to evaluate what fraction of the spectral interval has absorption coefficients in the range k to $k+dk$.]

5.3.2 Malkmus-Lorentz Band Correlated-K   (ICONV=21)

The Malkmus-Lorentz model is particularly suitable for application of k-distribution methods since the k-distribution may be determined analytically from the band model parameters. Expressing the Malkmus-Lorentz band model in the form:

$$\bar{T}(m) = \exp\left\{-\frac{\pi B}{2}\left[\left(1 + \frac{4Sm}{\pi B}\right)^{1/2} - 1\right]\right\}$$

(where $S = k_\nu(T)$ and $B = 4\alpha_L / \delta$) then the absorption coefficient frequency absorption is given by:

$$f(k) = \frac{1}{2}k^{-3/2}(BS)^{1/2}\exp\left[\frac{\pi B}{4}\left(2 - \frac{S}{k} - \frac{k}{S}\right)\right]$$

with a peak value at:

$$k_{max} = \frac{3S}{\pi B}\left[\left(\left(\frac{\pi B}{3}\right)^2 + 1\right)^{1/2} - 1\right]$$

The cumulative frequency distribution may also be expressed analytically as:

$$g(k) = \frac{1}{2}\left[1 - \mathrm{erf}\left(a/\sqrt{k} - b\sqrt{k}\right)\right] + \frac{1}{2}\left[1 - \mathrm{erf}\left(a/\sqrt{k} + b\sqrt{k}\right)\right]e^{\pi B}$$

where $a = \frac{1}{2}(\pi BS)^{1/2}$, $b = \frac{1}{2}(\pi B / S)^{1/2}$ and where erf(z) is the error function:

$$\mathrm{erf}(z) = 2/\sqrt{\pi}\int_0^z e^{-t^2}\,dt$$

Unfortunately, no analytical solution for the k-distribution which is the inverse of $g(k)$ exists and so this must be found by iteration.

As mentioned earlier, once g-intervals have been defined, the mean absorption coefficient is most easily determined as the arithmetic mean. However when this is done the resultant mean transmission is not exactly that of the equivalent band model but depends on the number and width of the g-intervals. However the mean transmission can be forced to be exactly the same as the equivalent band model if we

use the cumulative transmission function $h(k)$ which is dependent on the particular path being calculated:

$$h(k) = \int_0^k f(k') \exp(-k'm) dk'$$

For the Malkmus-Lorentz distribution, this function exists in a closed form:

$$h(k) = \frac{1}{2}\left[1 - \mathrm{erf}\left(a/\sqrt{k} - c\sqrt{k}\right)\right] \bar{T}(m) + \frac{1}{2}\left[1 - \mathrm{erf}\left(a/\sqrt{k} + c\sqrt{k}\right)\right] e^{\pi B}/\bar{T}(m)$$

where $a = \frac{1}{2}(\pi BS)^{1/2}$, $c = \frac{1}{2}(\pi B/S + 4m)^{1/2}$ and where $\bar{T}(m)$ is Malkmus-Lorentz transmission function. Once the limits of k are found for each interval of g, the mean absorption coefficients are defined as transmission weighted averages:

$$e^{-\bar{k}_i m} = \frac{\int_{k_i}^{k_j} f(k) e^{-km} dk}{\int_{k_i}^{k_j} f(k) dk}$$

or

$$\bar{k}_i = \frac{1}{m} \ln(\Delta g_i / \Delta h_i)$$

Since the mean absorption coefficients defined here are weighted averages for a specified absorber amount, the resulting histogram of absorption coefficients is necessarily absorber dependent. Because of this, the transmission computed by

$$\bar{T}(m) = \sum_i^N \exp(-\bar{k}_i m) \Delta g_i$$

remains equal to the Malkmus-Lorentz transmission regardless of the number of g-intervals used. This removes any errors in the calculation due to fitting k-distributions to band models. However the number and width of the g-intervals obviously affects the accuracy of any multiple scattering calculations. In practice it is best to use g-intervals which are unequal in width but which provide approximately equal ratios in absorption coefficient strength. In the case of multiple scattering and vertical inhomogeneity then for a given number of intervals, optimum accuracy is obtained when the g-intervals are more closely spaced near $g = 0$ and the strong absorption end near $g = 1$.

The additional parameter PAR1 here defines further details of how the Malkmus-Lorentz correlated-k model is to be calculated:

PAR1=1   a) The mean absorption of each sub-interval is the simple arithmetic mean of the absorption coefficient in the interval.
              b) Overlapping gases in the intervals are combined using the weak and strong limits.

              This second part may be explained as follows. To determine the B and S Malkmus-Lorentz parameters of the interval when there is more than one active gas, the B and S parameters for each gas are combined in a manner which is consistent with the band model assumption of uncorrelated randomly overlapping absorption lines. Consider, for example, two overlapping gases that are characterised by band model parameters B_a, S_a and absorber amount u_a, and B_b, S_b, and u_b respectively. The statistical parameter definitions of these parameters in the weak and strong limits imply:

$$S = S_a u_a / u + S u_b / u$$

and

$$(BS)^{1/2} = \left(B_a S_a u_a / u\right)^{1/2} + \left(B_b S_b u_b / u\right)^{1/2}$$

PAR1=2   a) The mean absorption of each sub-interval is the absorber amount weighted mean using the cumulative transmission function.
b) Overlapping gases in the intervals are combined using the weak and strong limits.

PAR1=3   a) The mean absorption of each sub-interval is the absorber amount weighted mean using the cumulative transmission function.
b) Overlapping gases in the intervals are combined by summing the optical depths.

Combining the band parameters statistically is reasonable if the absorption of the overlapping gases is comparable with similar degree of saturation. However in the extreme situation where one is strongly absorbing and the other is weakly absorbing, the square root dependence of the product SB can lead to greatly overestimated values of B. In these situations it is better to add the effective Malkmus-Lorentz optical depths of the two gases calculated with:

$$\tau_i = \frac{\pi B_i}{2}\left[\left(1 + 4 S_i u_i / \pi B_i\right)^{1/2} - 1\right]$$

and then use the equation:

$$B = \frac{\tau_i^2}{\pi}\left(Su - \tau\right)^{-1}$$

to determine B. The S parameters, which add linearly, are not adversely affected by the absorber mismatch.

In all three cases, the integral over g is performed by the simple sum solution.

5.3.3 Goody-Voigt Band Correlated-K   (ICONV=22)

For calculations in the Jovian atmosphere, the spectral properties of methane are important. Unfortunately the line data for CH_4 is not very reliable in the near-IR (although is improving steadily) and so band models are often used where the band parameters are fitted to laboratory measurements of methane spectra. [R4] found that the best fit to the laboratory measurements was the Goody-Voigt band and to use this data in a scattering model of Jupiter's atmosphere requires a Goody-Voigt correlated-k model. Unfortunately, $k(g)$ cannot be determined analytically from the Goody-Voigt parameters but must be determined from a numerical Laplace transform of the transmission spectrum. This may be done in one of two ways:
     PAR1 = 0, $k(g)$ is the mean absorption coefficient over the
     interval
     PAR1 = 1, $k(g)$ is weighted by the cumulative transmission
           function to get close agreement between correlated-k
           and band calculations

5.3.4 Exponential Sum Model   (ICONV=23)

Not been updated fro some time – use with caution! Performs least squares fit to the transmission functions calculated with band models to derive the exponential sum *k*-coefficients directly. This is usually better done off-line with the program, Band_ktable and the k-tables then used with the ICONV=24 model below. There are two models depending on PAR1:

        PAR1 = 0,   *k*(*g*) calculated from Malkmus-Lorentz transmission curve
        PAR1 = 1,   *k*(*g*) calculated from Goody-Voigt transmission curve

Model only still exists for historical completeness. Recommend using ICONV=24 instead.

5.3.5 k-Distribution Look-up Table   (ICONV=24)

Identical to ICONV=20, except the *k*-coefficients are calculated for each gas BEFOREHAND over a representative grid of pressures and temperatures using either Calc_ntable for line data, or Band_ktable for band data. Overlapping gases are combined as per [R8]. Look-up tables calculated with Calc_ntable and Band_ktable may be inspected using the program Read_table.

5.4 Running Radtrans

To run the program Radtrans, the atmospheric information in the '.prf' profile files and spectral details must be combined into a driver file (.drv) using the program Path as previously described in Section 4. In addition, the user must supply the '.cia' and '.fla' files to define various model options. These files are described in the Nemesis manual. If the driver file has the name 'surf.drv' then Radtrans is invoked (in background) using the line:

        Radtrans < surf.nam > surf.prc &

The binary output spectra are sent to the file surf.out while any processing information is sent to surf.prc. In addition a log file is written, called in this case surf.log, which contains a full listing of the run details for reference. An overview of the processing software was shown earlier in Fig. 1.1:

If one of the correlated-*k* models has been selected then there must also exist a '.gin' file which contains the desired *g*-ordinates and weights for a Gauss-Legendre integration. This file has the same name as the driver. A typical '.gin' file has the following format:

10	
1.3046736E-02	3.3335671E-02
6.7468315E-02	7.4725673E-02
0.1602952	0.1095432
0.2833023	0.1346334
0.4255628	0.1477621
0.5744371	0.1477621
0.7166977	0.1346334
0.8397048	0.1095432

0.9325317	7.4725673E-02
0.9869533	3.3335671E-02

where the first integer is NG, the number of g-ordinates and the subsequent columns contain the *g*-ordinates and weights respectively.

Finally if a scattering path is defined, then a '.sca' file must also be present whose format is described in Section 8.5.2.

6. Post Processing of RADTRANS Output Spectra

Three main programs are available for processing the binary output data of Radtrans:

> Conv_spec
> Li_spec
> Pl_spec

In addition, the programs Aground and Diskav may also be used to produce more specific output (described below).

6.1 Conv_spec

The program Conv_spec convolves the output spectra with user specified instrument functions . The program is very similar to Conv described in [R10] but slightly more general and manipulable.

The output from Radtrans are binary spectra with interval DELV cm^{-1} where the calculated quantities are either averages over a square bin of width FWHM centred the output wavenumber, or for ICONV=0, pure monochromatic spectra. For example, if ICONV $\neq$ 0, if VMIN = 2000 cm^{-1}, DELV = 5 cm^{-1} and FWHM = 5 cm^{-1} then the first output at 2000 cm^{-1} will be the average from 1997.5 to 2002.5cm^{-1}. Conv_spec, takes these spectra and applies one of the following functions:

1	Square bin convolution
2	Triangular bin convolution
3	Multiplication by a Filter profile followed by square bin convolution
4	Sinc function convolution

The convolutions shapes are of user-specified widths and may be applied in either wavelength or wavenumber space. Conv_spec limits the convolution to wavenumbers where the convolution shape is within the originally defined interval so output '.ave' spectra always contain less points than the original .out spectra. The centres of these convolution shapes is at the same wavenumbers as the output spectra and Conv_spec determines the average of the convolution shape over each bin and calculates the weighted mean. If the convolution shape covers half a bin, then an appropriate fraction of the bin is added. This of course assumes that the absorption is evenly spread over the bin which may or may not be a good idea. In general it is better not to use fractions of bins so if you want to calculate an output spectra with triangular convolution of FWHM=10cm^{-1}, for example, set DELV = 6.666 and FWHM = 6.666 cm^{-1} .

A typical input sequence is shown below:

```
driving file name? :testc1
enter FWHM, wavespace [0=wavenumber, 1=microns]:10,0
Enter ICONV [1=square, 2=triangle, 3=filter, 4=sinc ] :2
```

6.2 <u>Li_spec</u>

The program <u>Li_spec</u> reads in either a direct output or convolved spectrum and writes the data to the screen or an ASCII file. The user is guided interactively through its operation.

6.3 <u>Pl_spec</u>

The program <u>Pl_spec</u> again reads in either a direct output or convolved spectrum and outpots spectra to a an ASCII file suitable for display by the IDL routine `plot_radtran.pro`. After the spectrum has been read in, the program gives the user the following options:

```
A - display calculations
B - display paths
C - display layers
D - plot a spectrum for a path
E - plot results of a calculation
Q - quit
```

To illustrate the effects of the above options, consider the spectrum generated using the following driver file:

```
original name of this file: testc1.drv
  1900.000  5.00000      21  5.000     :Vmin dV Npoints FWHM (BAND)
  0.000  2.000       : Additional codes PAR1 and PAR2
5jupiter5
  12    0           : type of convolution/model & NCONT
   1  1  1          : number of layers, paths and gases
   6                        : identifier for gas 1
    1   0                     : isotope ID and process parameter
format of layer data
 layer baseH delH baseP    baseT totam      pressure   temp  doppler
     absorber amounts and partial pressures
     continuum points if any
  1  0.000   0.000 0.00000E+00   0.000 0.36717E+25 0.10000E+01 296.000  0.0000
      0.36717E+25 0.10000E+01
   1   0 0.10000E-01          : Nlayers, model & error limit, path  1
   1   1  0.000 0.10000E+01  :   layer or path, emision temp, scale
   1                      : number of filter profile points
 0.00000E+00    0.000              : filter profile point   1
testc1.out
  1                       :number of calculations
 128  2  0  0                   :type and # of parameters for calc  1
      1
      1
```

<u>A - display calculations</u>

This option simply outputs to the screen the number of different calculations being done and the number and value of the parameters defining it. The output in this case is:

```
there are  1 calculations

--------------------------------------------------
Calculation :  1
Cell calculation
Single cell calculations
Number of cells =   1
Path:  1 Layer:   1

--------------------------------------------------
```

From the driver file, the first calculation has ITYP=128 which corresponds to a single cell transmission.

B - display paths

This option summarises the paths for which spectra have been calculated. The output in this case is:

```
path: 1  model type: 0,   1layers included
   1
 --------------------------------------------------
```

Thus this driver contains just one path which is a cell path for which the transmission is calculated (IMOD = 0) and comprises just one layer, LAYER=1.

C - display layers

This options summarises the layers used in the spectral calculations and the number of gases used. The output in this case is:

```
there are  1 gases
gas: 1  identifier:  6  isotope id:    1
--------------------------------------------------
layer: 1  pressure: 0.10000E+01bars  temp: 296.00K
gas amounts
 0.36717E+25
--------------------------------------------------
```

D - plot a spectrum for a path

This option plots the calculated spectra for any of the paths. The option has an interactive input to define which path spectrum to output and in what form. A sample session for the '.drv' file above is shown below.

```
absorbtion plot? [1-output] :y
 Plot against wavenumber (1) or microns (2) :1
log plot? [Y/N] :n
   1945.000      1900.000
 wavenumber ( x ) limits are:   1900.000  2000.000
 y limits are:  0.00000E+00 0.88040E+00
 force plotting axes? [y/n] :n
```

Different options appear for different spectral calculations, e.g. brightness temperatures may be plotted for radiance calculations.

E - plot results of a calculation

This option plots the results of calculations defined by the ITYPE integers. In most cases, the calculation output is simply one of the calculated path spectra. However, this option also plots the PMR transmissions/emissions and weighting functions. This is the option which should be most commonly used. An example input session is shown below.

```
there are   1 calculations
 plot which one? :1


 -----------------------------------------
Cell calculation
Single cell calculations
Number of cells =   1
Path:  1 Layer:   1


 -----------------------------------------

Cell calculation
Single cell transmission
There are   1 paths :
       1
 Enter desired path :1
absorbtion plot? [1-output] :y
Plot against wavenumber (1) or microns (2) :1
log plot? [Y/N] :n
  1945.000      1900.000
wavenumber ( x ) limits are:   1900.000  2000.000
y limits are:  0.00000E+00 0.88040E+00
force plotting axes? [y/n] :n
```

As can be seen, the option first prints the details of the calculation in the same way as for option A.

A typical input sequence if the calculation is for reflecting layers is:

```
there are   5 calculations
 plot which one? :5


 -----------------------------------------
 Reflecting Layer Calculation
 Pure atmospheric paths
Number of atmosphere calculations =   1
Number of reflected atmosphere calcs =   3
 Atmospheric calculations are:
       4
```

```
 Reflection Atmosphere calculations are:
       1
       2
       3


 ------------------------------------------

 Cell calculation
 Reflecting layers calculation
 Pure atmosphere paths
Number of atmosphere calculations =   1
Number of reflected atmosphere calcs =   3
 Atmospheric calculations are:
       4
 Reflection Atmosphere calculations are:
       1
       2
       3
 Enter required number of reflecting calcs :3
 Enter reflecting atmosphere calc :1
 Enter albedo of layer :0.9
 Enter reflecting atmosphere calc :2
 Enter albedo of layer :0.5
 Enter reflecting atmosphere calc :3
 Enter albedo of layer :0.1
 Enter required number of atmospheric calcs :1
 Enter atmosphere calc :4
 Enter distance from sun (AU) :5
 Enter surface temperature of Sun (K) :6000
 plot brightness temperature? :y
 Plot against wavenumber (1) or microns (2) :1
 log plot? [Y/N] :n
   1998.850      1700.000
 wavenumber ( x ) limits are:   1700.000  2299.850
 y limits are:  0.00000E+00 0.26876E+03
 force plotting axes? [y/n] :n
```

The reflecting atmosphere calculations correspond to **reflatm** in the path file and the atmosphere calculations correspond to **atm**.

The calculated radiance spectra in this case are determined as:

$$R(\tilde{v}) = R_a(\tilde{v}) + \sum_{i=1}^{N} \tau_i(\tilde{v}) A_i I_0(\tilde{v}) / \pi$$

where:
R_a is the calculated thermal radiance from the atmosphere alone.
t_i is the transmission from the sun to the ith reflecting layer and back to the instrument.
A is the albedo of the reflection layer
I_0 is the incident solar flux:

$$I_0(\tilde{\upsilon}) = B(\tilde{\upsilon},T)\pi\left(\frac{\theta_0}{D}\right)^2$$

where θ_0 is the angular radius of the sun at 1 A.U. and D is the distance from the sun in A.U.

6.4 Aground

Adds ground emission spectrum multiplied by an atmospheric transmission spectrum to an atmospheric emission spectrum. Assumes .out file contains both the atmospheric thermal emission and the transmission spectrum to the ground and combines them appropriately given the surface temperature. This program basically does the same job as the old Addground program without the use of intermediate ASCII files.

6.5 Diskav

Program for numerically simulating disk-averaged spectra. The RADTRANS '.out' file is assumed to contain thermal emission spectra calculated at a number of emission angles from a .pat file written by Write_diskpat. A Guassian Quadrature over emission angle is assumed with the angles and weights defined by `getquad.f`. Diskav simply combines the output spectra (which may also include ground radiances) using the weights specified by `getquad.f` to generate disk-averaged spectra. This program has been somewhat superseded by the introduction of the analytical exponential-integral method of achieving the same result.

7 Setting up Spectral Data

7.1 Line data

A number of programs are available for setting up and manipulating direct access line databases. An overview of the software is shown in Fig.7.1.
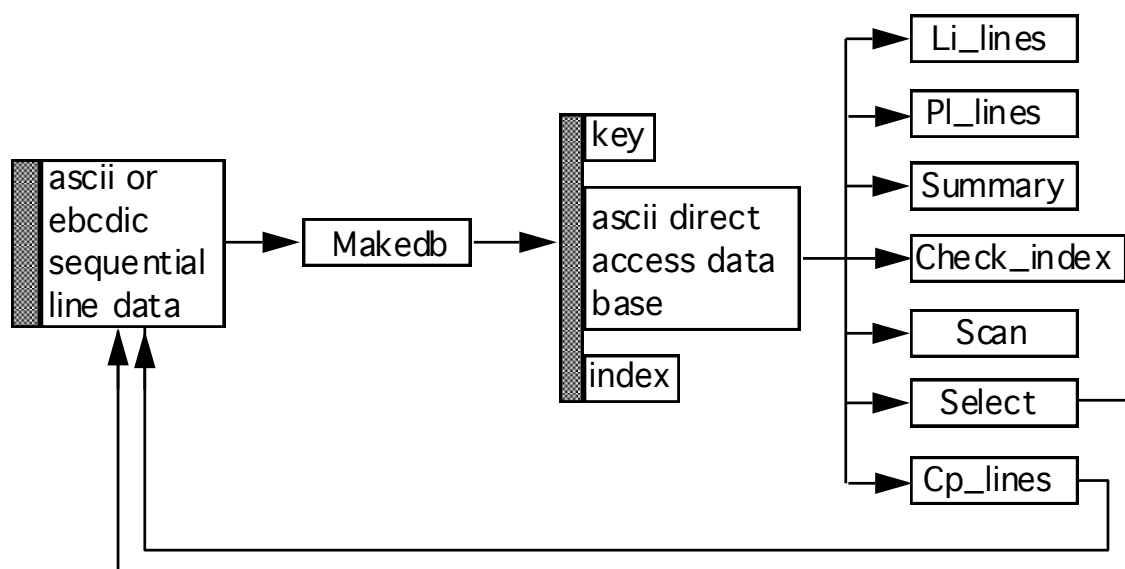
Linedata Routines Overview

n.b. Programs are denoted by:

Data files are denoted by:

Fig. 7.1. Overview of Linedata Manipulation Software

7.1.1 <u>Makedb</u>

This is the central program of the system and transcribes sequential-access line databases (eg HITRAN or GEISA) from files as copied from tape into direct access files containing one transition per record. The records are in FORTRAN character format as in the original database and conversion to ASCII from EBCDIC is allowed. An index file is also produced consisting of one record corresponding to each record (transition) in the database. This record contains a pointer to the database record holding the next transition for this species (of any isotope). The program also produces a key file summarising the following information:

      the name of the database file
      the name of the index file
      the name of the file describing the gas_info file, which lists the gases known
        to the RADTRANS software and also includes a cubic polynomial
        description of the partition function for each isotope (see section 7.1.13),
        together with the relative abundances of these isotopes assumed in line
        database. The line strengths of the different isotopes are already multiplied
        by these relative abundances in the line database and so if you want to see

the line strengths of particular isotopes, the listed strengths need to be divided by the relative abundance.

the name of the file containing the translation between isotope identifiers in the database used and those used by Radtran

Thus only this key file has to be specified to identify the database. Since HITRAN and GEISA use slightly different gas identifiers, Makedb and all the subsequent RADTRANS software uses its own identification system, based on HITRAN so that both data formats can be transcribed. The gas identification system used is shown in Appendix A. The local gas identifier agrees with HITRAN IDs as far as possible (Jan 1988) with extra IDs for gases not in the HITRAN compilation e.g. those in the GEISA compilation. Isotope identifiers also agree as far as possible with HITRAN identifiers. Similar (1,2,3...n) identifiers have been defined for additional gases.

Makedb also converts the sequential files output by Select and Cp_lines into direct access databases.

7.1.2 Scan
This program opens a line database specified by an input key file name and lists the line data records directly to screen. The program prompts for a starting wavenumber and then lists all transitions from that point in the storage format until told to abort.

7.1.3 Check_index
Check_index opens a line database specified by an input key file name and lists the index file records directly to screen. The program prompts for a starting wavenumber and then lists all transitions from that point in the storage format until told to abort.

7.1.4 Li_lines
This program opens a line database specified by an input key file name and lists selected line data directly to screen. The line data is read in by the subroutine `lines.f` which in turn reads in line data records with the subroutine `rdline.f`. To avoid numerical underflow when passing line strengths between these subroutines, the strengths are multiplied by the factor 10^{47} in `rdline.f`. Before listing the strengths therefore, Li_lines multiplies the returned strengths by 10^{-47} so that the strengths are listed in units of $cm^{-1}/(molecule\ cm^{-2})$ as defined in the line databases (see section 2.1). The program prompts for the required gases and the wavenumber range and then lists main line data parameters for these gases.

7.1.5 Pl_lines
Pl_lines plots out line details for a particular spectral region ASCII format, suitable for display using the IDL routine `pl_alines.pro`. The program prompts for line data key, gas identifiers, wavelength range and plotting details and then plots out line strengths versus wavenumber. The same scaling and de-scaling of the line strengths by the factor 10^{47} is applied as in Li_lines.

7.1.6 Select
This a line data manipulation/editing program which opens a line database created by Makedb (e.g. HITRAN or GEISA) specified by an input key file name and copies a specified subset of the line database to a new sequential ASCII file. This file may then be modified and perhaps merged with new spectral lines before being turned back into a new direct access database using Makedb. All lines or only selected lines

between given wavenumber limits are copied. Selection can be performed by gas ID, isotope, and line strength limit. The strength limit for a particular gas or isotope is set so that the sum of strengths of omitted lines is less than n% of the total sum of strengths. This criteria is calculated by finding the number of lines in each decade of strength. Note that the strength limit can be set independently for each isotope or can be set for all of the isotopes to be included. In the latter case terrestrial isotopic abundances are assumed, i.e the strengths are used uncorrected. The logic behind this is that if the isotopic abundance may vary you want to treat them independently anyway. The line data may also be modified to have, for example, different line widths for comparison with other programs or to check details of the line data itself.

Example input sequence:

```
Enter wavenumber limits :1400,1450
Enter bin size :10
Enter error limit [%] :0
Enter new file name :test
set self broadened width to air broadened? :n
multiply H2O air broadened width by 1.3? :n
replace H2O air b. by delaye CO2-broadened data? :n
generate linedata using PMR2 assumptions? :n
generate linedata using Wedads assumptions? :n
generate linedata using Brunos assumptions? :n
generate linedata using all PMIRR assumptions? :n
H2-He broaden PH3 for Jupiter? :n
-------------------------------------------------
enter gas details
enter gas id and isotope [0=all isotopes]
ALL includes all gases, GO starts execution,
? gives summary
ID ISO ? :1,0
ID ISO ? :GO
```

As can be seen, the new sequential file is called 'test' and contains linedata between 1400 and 1450 cm^{-1} for all isotopes of water vapour only. The program may then perform a number of modifications to the line data which are historical in nature. The first three possible operations are self-explanatory, the data for the third operation coming from [R8]. It should also be noted that the third option replaces the transmission probability column with the temperature dependence of the self-broadening since this column is not normally used by RADTRANS (This data is not used in version 1.1). The last three operations are more obscure.

PMR2 Assumptions - This operation generates linedata equivalent to that used by PMR2, a non-overlapping line model developed by J.T.Schofield. Modifications are:

H_2O Multiply air broadening coefficient by 1/1.013246

           Set self-broadening to 6.5 times air broadening

           Set temperature dependancies to 0.5.

           Replace transition probability column with the temperature dependence of the self-broadening

CO$_2$ Set the air and self broadened widths to those calculated by Yamamoto
[R9] multiplied by the empirical factor 1.1/1.013246.
Set the temperature dependancies to those calculated by Yamamoto.
Replace transition probability column with the temperature
dependence of the self-broadening

Wedad Assumptions - This operation generates linedata equivalent to that used by
Wedad Adbou at JPL for some of the PMIRR intercomparison tests. Modifications
are:

H$_2$O   Multiply air broadening coefficient by 1/1.013246
Set self-broadening to 6.5 times air broadening
Set temperature dependancies to 0.5.
Replace transition probability column with the temperature
dependence of the self-broadening

CO$_2$ If the self broadened width is listed as zero, set it to that calculated by
Yamamoto [R9] multiplied by 1/1.013246.
Set the temperature dependancies to those calculated by Yamamoto.
Replace transition probability column with the temperature
dependence of the self-broadening

Bruno Assumptions – Various modifications to linewidths etc. for CIRS Jovian
radiative transfer tests.

PMIRR Assumptions - These are the assumptions that would have been used to
generate the final transmittance tables for PMIRR. Modifications are:

H$_2$O   Replace air broadened width and temperature dependence of water
with Delaye calculated CO2 broadened value
Replace transition probability column by temperature dependence of
self broadening calculated from Delaye
If self broadening is set to zero, substitute Delaye calculated value.

CO$_2$ Leave air broadened CO2 unchanged
If self broadening is listed as zero, set it to that calculated by
Yamamoto/1.013426
Replace air broadened temperature dependence with that calculated by
Yamamoto
Replace transition probability column by temperature dependence of
self broadening calculated from Yamamoto

H2-He broaden PH3 for Jupiter – self-explanatory.

7.1.7 Selecttemp
Identical to Select, except that the line strength selections are made at a temperature
of the user's choice. This is because at different temperatures, different lines become
more important depending on their lower state energies. This program also requires
the user to specify whether IPTF=0 and the default partition functions (defined by the

gasinfo file) are to be used or IPTF=1, in which case the partition functions of those gases listed in the extra partition function file are to be used instead.

7.1.8 Selecttemploop

Identical to Selecttemp, except that the several line selections are made for a range of different temperatures defined by the user. Again, the variable IPTF must be defined by the user.

7.1.9 Cp_lines

Cp_lines opens a line database created by Makedb (eg HITRAN or GEISA) specified by an input key file name and copies a specified subset of the line database to a new sequential ASCII file just like Select. Unlike Select however, it copies **all** lines between given wavenumber limits rather than selecting them by gas, isotope and strength.

7.1.10 Summary

This program opens a line database specified by an input key file name and prompts for a wavelength range and bin width. Summary then producues an ASCII output file called "summary.out" which summaries the content of the datebase for each gas for each bin in the specified range. The first few lines of a typical summary file are listed below:

```
database: hitran92
from DB file: /vol/local/linedata/hitran92.asc
as defined in key: hitran92.key
-------------------------------------------------------
BIN: 1    1400.000 -  1410.000 cm-1
        id iso   smin          smax       nlines
 H2O    1  1 0.47000E-26 0.27800E-19    29
 H2O    1  2 0.28800E-25 0.55200E-22    11
 H2O    1  3 0.52400E-26 0.10300E-22    11
 H2O    1  4 0.10370E-26 0.96180E-23    29
total of    80 H2O    lines
        id iso   smin          smax       nlines
 CO2    2  3 0.94910E-26 0.70040E-25    40
 CO2    2  4 0.50400E-26 0.19980E-25    13
total of    53 CO2    lines
        id iso   smin          smax       nlines
 O3     3  1 0.10200E-24 0.29100E-23   280
total of   280 O3     lines
```

7.1.11 Merge

This program takes two sequential access line database files and then merges them together in order of wavenumber. This program is very useful for adding extra lines to existing line databases.

7.1.12 Line data Strengths and Absorber Amounts Units

This area causes a lot of confusion! When Radtrans reads in line data from one of the databases set up by Makedb, they do so using the subroutine `lines.f` which in turn

uses the subroutine `rdline.f.` The units of line strength in the line databases are the standard ones of cm^{-1}/ (molecule cm^{-2}) and as such have rather small exponents. In order to avoid numerical underflow when passing small strengths between the subroutines, `rdline.f` multiplies the strengths by the factor 10^{47}. Thus when the final calculation routines get hold of the strengths, they must be converted back to their correct units using the factor 10^{-47}. This number is too small to be applied in one go using real*4 numbers and so is applied in two stages. The factor 10^{-27} is applied directly and the factor 10^{-20} is applied to the absorber amounts which are defined in the units of (molecule cm^{-2}).

7.1.13 Partition Functions

The line strengths defined in databases such as HITRAN are for a specific temperature only, $T_0 = 296K$, and must first be scaled to be used at any other temperature. The line strength at temperature, T, is related to that at the reference temperature, T_0, by the equation:

$$S(T) = S(T_0) \times f_Z \times e^{\alpha E} \times f_S$$

where:

> f_Z is a term to scale the strengths by the relative partition functions at the two temperatures $f_Z = \frac{Z(T_0)}{Z(T)}$ where $Z(T)$ is the total partition function at temperature T, and is equal to $Z_{rot}(T) \times Z_{vib}(T)$. At the usual temperatures found in the solar system atmospheres, Z is dominated by the rotational partition function $Z(T) \approx Z_{rot}(T) = T^n$, where $n=1$ for linear molecules and 1.5 for non-linear molecules. At higher temperatures, the vibrational partition function can come into play.

> E is the lower state energy of the line (expressed in cm^{-1}).
> α is a temperature factor $\alpha = 1.439(T - T_0)/(T_0 T)$
> f_S is a stimulated emission term given by $f_S = \frac{1 - \exp(-1.439\tilde{v}/T)}{1 - \exp(-1.439\tilde{v}/T_0)}$, where $\tilde{v}$ is the wavenumber of the line.

For standard line databases covering the temperature ranges found in the Solar System atmospheres, the partition function is fitted with a cubic polynomial and the four coefficients listed in the gas_info file for each isotope of each gas recognised by the software, in addition to their relative abundances. At the calculation temperature, T, the total partition function is then calculated as:

$$Z(T) = c_0 + c_1 T + c_2 T^2 + c_3 T^3$$

where c_{0-3} are the listed coefficients. This system works well over limited temperature ranges, but can quickly diverge at temperatures outside the normal range, even leading to negative partition functions. Modelling extrasolar planets, whose temperatures can far exceed those for which the polynomials are valid can be extremely problematic and in this case, RADTRANS allows the user to override the partition functions calculated with the polynomials for specific gases by values interpolated from a table supplied by the user in …/raddata/partfextra.dat. The format of this file is:

```
# Any number of lines of header preceded by a #
NPARTEXTRA                   ! Number of gases for which data exist in table
IDEXTRA(1),ISOEXTRA(1)! ID, ISO of gases listed. ISOEXTRA(I)=0 is allowed.
. . .
IDEXTRA(NPARTEXTRA),ISOEXTRA(NPARTEXTRA)
# Any number of lines of header preceded by a #
NTAB(1)                      ! Number of temperatures for 1st gas in list
TEMP(1,1) PARTF(1,1)
. . .                        ! Temperatures and partition functions.
TEMP(1,NTAB(1)),PARTF(1,NTAB(1))
# Any number of lines of header preceded by a #
NTAB(2)                      ! Number of temperatures for 2nd gas in list
TEMP(2,1) PARTF(2,1)
. . .                        ! Temperatures and partition functions.
TEMP(2,NTAB(2)),PARTF(2,NTAB(2))
. . . and so on.
```

Please note that PARTF(I,J) = f_Z = $Z(T_0)/Z(T)$.

7.2 Band Data

RADTRANS contains some spectral models which use band data instead of line data. When these routines are used, the spectral data must be read in from a band parameter file which has a '.ban' suffix. This band data file may be generated either by statistical analysis of line data in the HITRAN or GEISA databases or calculated from low resolution laboratory experiments. For every '.ban' file there must also be a .key file which contains the location of the '.ban' file and the gas identifier files

A typical '.ban' file has the following format:

```
-----------------------------------------------------------------------------
Original file name : 5jupiter5.ban
Originator : Pat Irwin
Date : 25/10/94
Data sources :
Gas :  6  1 HITRAN92
Gas : 11  0 "
Gas :  6  3 "
Gas : 28  0 "
Gas : 33  0 "
Gas :  1  0 "
Corrected line treatment in Makeband
****************************************************
VMIN = 1800.00
DELV =    5.00
FWHM =    5.00
NPOINT =   61
NGAS =  6
Gas :  6  1
Gas : 11  0
Gas :  6  3
Gas : 28  0
Gas : 33  0
Gas :  1  0
 ------------------------------------------------------------------------
 Band data for gas :  6  1
```

V0	dV	Kv(T0)	delta/AD0	y0	El	SFB
1800.00	5.00	0.52856E-03	0.88439E+03	0.26271E+03	0.14892E+04	0.13373E+01
1805.00	5.00	0.11291E-02	0.91134E+03	0.27025E+03	0.14911E+04	0.13193E+01
1810.00	5.00	0.12633E-03	0.11360E+04	0.45271E+03	0.16362E+04	0.13533E+01
1815.00	5.00	0.38263E-03	0.12947E+04	0.37365E+03	0.16080E+04	0.13393E+01
1820.00	5.00	0.89474E-03	0.93499E+03	0.25684E+03	0.15506E+04	0.13320E+01

The first half of the header contains information on the data sources, who generated the file and when. The second half contains the spectral details of the file and the number of gases contained.

The data itself contains the following columns for each spectral bin defined for each gas:

$$\tilde{\upsilon} \qquad \Delta\tilde{\upsilon} \qquad k_{\upsilon}(T_0) \quad \delta / \alpha_D^0 \; y_0 \qquad E_l \qquad SFB$$

where:
$\tilde{\upsilon}$ is the central wavenumber of the bin
$\Delta\tilde{\upsilon}$ is the width of the bin in wavenumbers
$k_{\upsilon}(T_0)$ is 10^{20} × the mean absorption coefficient at T_0 = 296K (units of absorption are (molecule cm^{-2})$^{-1}$)
δ / α_D^0 is the mean line spacing δ over the mean Doppler width coefficient α_D^0.
$y_0 = \alpha_L^0 / \alpha_D^0$ is the mean self-broadened Lorentz width at T_0, P_0 (= 1 atm) over the mean doppler width coefficient.
E_l is the mean lower state energy (cm^{-1})
SFB is the mean self-to-foreign broadening ratio.
α_D^0 is the Doppler width coefficient for the bin which depends only on the mean wavenumber of the bin and the molecular mass. The Doppler width at temperature T is then found from: $\alpha_D = \alpha_D^0 \sqrt{T}$

A typical '.key' file has the following format:

5jupiter5.ban
/home/plan/linedata/gas_info.dat

7.2.1 Makeband

The data in the '.ban' files may come either from laboratory experiments or from analysis of line databases. The program which performs this latter function is Makeband which also generates a '.key' file. The band parameters defined above are derived from the linedata as follows:

From [R4], the mean absorption in a spectral interval is given by:

$$k_{\upsilon}(T_0) = \frac{\frac{1}{N}\sum_i s_i}{\delta} = \frac{w}{\Delta\tilde{\upsilon}}$$

where $w = \sum_i s_i$ is the sum of the line strengths and the mean line spacing is defined as $\delta = \frac{\Delta\tilde{\upsilon}}{N}$.

The mean Doppler width coefficient depends on the mean wavenumber and absorber molecular weight (g) only through the equation:

$$\alpha_D^0 = \tilde{\upsilon}\,\frac{4.301 \times 10^{-7}}{\sqrt{M}}$$

In Section 5 we saw that the Goody and Malkmus distributions have different definitions of α_L with the Malkmus value being 4 times smaller. Using the Goody definition the mean self-broadened half-width at T_0, P_0 may be calculated to be:

$$\alpha_L^0 = \frac{\delta s^2}{\pi k_\upsilon(T_0)}$$

where $s = \frac{2}{\Delta\tilde{\upsilon}} \sum_i \left(s_i \alpha_i^s\right)^{1/2}$. When using this table to calculate Malkmus bands, the value of α_L^0 must thus first be divided by 4.

The mean self/foreign broadening ratio and mean lower state energy are simply the mass weighted means:

$$SFB = \frac{\sum_i s_i \alpha_i^s / \alpha_i^f}{w}$$

$$E_l = \frac{\sum_i s_i E_{li}}{w}$$

Makeband is a user-interactive program which is fairly self-explanatory. The input session used to create the band parameter file in the example is shown below. N.B. Gas Ids are RADTRANS IDs.

```
Enter name of lbl key file :/home/oxpln3/plan/linedata/hitran92
38 gases recognised by software
32 gases in database
Enter wavenumber range and spacing.
Enter Vmin, Vmax, DelV, FWHM :1800,2100,5,5
Enter number of gases :6
Gas :          1
Enter Gas ID and ISO :6 1
Gas :          2
Enter Gas ID and ISO :11 0
Gas :          3
Enter Gas ID and ISO :6 3
Gas :          4
Enter Gas ID and ISO :33 0
Gas :          5
Enter Gas ID and ISO :28 0
Gas :          6
Enter Gas ID and ISO :1 0


Enter name of output file :5jupiter5
%writing output
Enter Originator :Pat Irwin
Enter Date :15/10/94
```

```
 Enter source of data for gas   6  1
HITRAN92
 Enter source of data for gas   11 0
"
 Enter source of data for gas   6  3
"
 Enter source of data for gas   33 0
"
 Enter source of data for gas   28 0
"
 Enter source of data for gas   1  0
"
 Enter any additional header information
Corrected line treatment in Makeband
 %calculation complete
```

In this example the spacing of the points and the bin width is the same so that the bins do not overlap. Hence the resultant spectra are under-sampled. Nyquist sampling (at least!) is usually recommended. DELV and FWHM are independent of each other and can be set at will.

7.2.2 Convert_data

The routines written by Kim Strong for fitting band parameters to laboratory measurements produce an output which is similar but not identical to the .ban format used by Radtrans. The output from Kim's routines can be converted using Convert_data.

Unfortunately, Kim's files do not specify which parameter is which so that the correspondence between this output and the '.ban' format has to be hard-coded into Convert_data. In particular, Convert_data is currently set up to convert the fits made of Simon Calcutt's NH_3 data by Kam Sihra which do not include an SFB parameter. Thus the code will have to be modified to cope with any subsequent changes to the output from Kim's routines.

7.3 Correlated-k data

Radtrans may also operate with pre-calculated correlated-k tables. These are created by the program Calc_ntable (line data) and Band_ktable (band data) and may be inspected using the program Read_table.

7.4 Gas Continua and Collision Induced Absorption (CIA)

Collision-induced absorption by Hydrogen-Helium has now been incorporated into RADTRANS [R12,13]. Additional CIA coefficients for N_2, CH_4 have also been incorporated. You must tell Radtrans which CIA table to use (which are all stored in raddata/ directory, by including a .cia file, whose format is described in the NEMESIS manual.

Other sources of continuum opacity such as the H_2O continuum absorption routines used in GENLN2, together with NIR methane absorption coefficients from Karkoschka (1994) and Fink *et al.* (1997) are built into the subroutine gascon.f, which should provide a means of easy expansion. To turn these continuum absorptions on or off, you need to define the appropriate flags in the .fla file, whose format is described in the NEMESIS manual.

8. Scattering

RADTRANS allows the calculation of thermal emission and reflected sunshine in a multiply-scattering atmosphere through the use of the Matrix Operator Method (also known as Doubling and Adding). This section has been compiled from previous technical reports [R16, R17] to describe RADTRANS scattering calculations in one place and also to emphasise how it fits within RADTRANS. [R15] gives a general introduction to scattering in planetary atmospheres.

8.1 Phase Functions and Normalisation

All particles scatter radiation as a function of the scattering angle α called the *Phase Function*. For simple scattering cases such as Rayleigh scattering this is a simple function of $\cos\alpha$. In RADTRANS, the phase function is calculated by the routine phase1.f. To conserve energy in the model we need some sort of normalisation. The one chosen for RADTRANS is that:

$$\int P(\theta,\phi)d\Omega = \int P(\theta,\phi)\sin\theta d\theta d\varphi = 1$$

or

$$\int_0^{2\pi}\int_{-1}^{1} P(\mu,\phi)d\mu d\phi = 1$$

where $\mu = \cos\theta$. Now in RADTRANS, the phase function at a defined incident and emission zenith angle is expressed as a Fourier series and is calculated by calc_pmat6.f :

$$P(\mu_i,\mu_j,\phi) = \sum_{r=0}^{N} P_r(\mu_i,\mu_j)\cos\phi$$

In this case the normalisation requirement reduces to just:

$$2\pi\int_{-1}^{1} P_0(\mu_i,\mu_j)d\mu_i = 2\pi\int_{-1}^{1} P_0(\mu_i,\mu_j)d\mu_j = 1$$

Of course the RADTRANS scattering code uses numerical techniques to evaluate the radiative transfer and in particular the integration over zenith angles above is simplified by using Lobatto quadrature to a simple weighted sum. In addition, instead of writing a total integral between negative and positive values of μ, the phase function is split into two parts, one where the incident and scattered radiation are in the positive μ direction called $P_{++}(\mu_i,\mu_j,\phi)$ and one where the incident and scattered radiation are in opposite directions called $P_{+-}(\mu_i,\mu_j,\phi)$. By symmetry arguments, $P_{++} = P_{--}$ and $P_{+-} = P_{-+}$. The normalisation requirements then become:

$$\sum_{i=1}^{N} P_{++}^0(\mu_i,\mu_j)w_i + \sum_{i=1}^{N} P_{+-}^0(\mu_i,\mu_j)w_i = \sum_{j=1}^{N} P_{++}^0(\mu_i,\mu_j)w_j + \sum_{j=1}^{N} P_{+-}^0(\mu_i,\mu_j)w_j = \frac{1}{2\pi}$$

For any but the simplest phase functions, inherent inaccuracies in the zenith quadrature scheme reveal themselves in the above equation, and if uncorrected lead to highly unphysical results such as negative reflectivities for thick clouds. Hence renormalisation is almost always required and then one most commonly used is that of [R18]. Here, the reverse scattering matrix is left unaltered (since for most cases the inaccuracies arise in the forward part) and instead for forward scattering matrix is modified to ensure that for illumination angle μj, the sum of the reflected and transmitted intensities adds up to one, i.e. $t_j + r_j = 1$ where:

$$r_j = 2\pi \sum_{i=1}^{N} P_{+-}^0(\mu_i, \mu_j) w_i \quad , \quad t_j = 2\pi \sum_{i=1}^{N} P_{++}^0(\mu_i, \mu_j) w_i$$

In the Hansen scheme [R18], each element of the forward matrix for each Fourier component is modified by the correction matrix f_{ij} which is found iteratively for the zeroth order Fourier component and ensures that $\varepsilon_j = |1 - r_j - t_j| \leq 10^{-14}$:

$$P_{++}^m(\mu_i, \mu_j) = f_{ij} P_{++}^m(\mu_i, \mu_j) \tag{8.7}$$

8.1.1 Phase functions used in RADTRANS

The simplest quasi-realistic phase function available is the Henyey-Greenstein phase function:

$$p(\theta) = \frac{1}{4\pi} \frac{1 - g^2}{\left(1 + g^2 - 2g\cos\theta\right)^{3/2}}$$

where θ is the scattering angle (0 = forward), g is the asymmetry factor (1=forward, 0=isotropic and -1 = backward), and the factor of 4π ensures the correct normalisation:

$$\int p d\Omega = \int_{-1}^{1} \int_{0}^{2\pi} p(\phi, \mu) d\mu d\phi = 1$$

This function is easy to use and has only one defining parameter but is actually not always very practical since most scatterers have both a forward- and back-scattering peak. Hence a more general purpose function is a combined Henyey-Greenstein function:

$$p(\theta) = \frac{1}{4\pi}\left[f \frac{1 - g_1^2}{\left(1 + g_1^2 - 2g_1\cos\theta\right)^{3/2}} + (1 - f)\frac{1 - g_2^2}{\left(1 + g_2^2 - 2g_2\cos\theta\right)^{3/2}} \right]$$

where f is some fraction between 0 and 1 and g_1, g_2 are the asymmetry factors of the two contributing functions. This is the form of the H-G phase function used by RADTRANS. To be specific, in RADTRANS:

$$c_1 = f$$
$$c_2 = g_1$$
$$c_3 = g_2$$

This function can quite accurately reproduce most phase function curves.

8.2 Fourier Decomposition

The routine `phase1.f` calculates the phase function at required scattering angles α and this function is Fourier analysed to determine the Fourier coefficients by the

routine `rephase.f` over the range $0 < \alpha < \pi$ in the normal way. The phase function can then of course be written as:

$$p(\alpha) = \sum_{n=0}^{N} a_n \cos(n\alpha) \tag{1}$$

Now in `calc_pmat6.f`, we want to calculate the azimuth angle Fourier components of the phase function at fixed incident and scattered zenith angles, i.e:

$$P_{++}^r(\mu_i, \mu_j, \phi) = \sum_{r=0}^{N} b_r \cos(r\phi) \quad P_{+-}^r(\mu_i, \mu_j, \phi) = \sum_{r=0}^{N} c_r \cos(r\phi)$$

These are both equal to Eq.1 if we use geometry to calculate the phase angle α from zenith and azimuth angles:

$$\cos(\alpha_{++}) = \sin\theta_i \sin\theta_j \cos\phi + \cos\theta_i \cos\theta_j$$

$$\cos(\alpha_{+-}) = \sin\theta_i \sin\theta_j \cos\phi - \cos\theta_i \cos\theta_j$$

The required Fourier components b_r and c_r are then found from the calculated phase angle components a_n from the matrix equations: $\mathbf{b} = \mathbf{A}_+\mathbf{a}$ and $\mathbf{c} = \mathbf{A}_-\mathbf{a}$ where the elements of the $\mathbf{A}$ matrices are:

$$A_+(r,n) = \frac{1}{\pi} \int_0^{2\pi} \cos r\phi \cos n\alpha_{++} \, d\phi$$

$$A_-(r,n) = \frac{1}{\pi} \int_0^{2\pi} \cos r\phi \cos n\alpha_{+-} \, d\phi$$

and these integrations are performed once by `ftable.f` using a simple trapezium rule code and stored.

8.3 Doubling and adding

Consider two homogeneous layers in an atmosphere. For any incident, emission and azimuth angle $(\theta_i, \theta_j, \phi)$ we can define coefficients of reflection and transmission as well as a value for the source function, giving the radiance emanating from within the layer due to thermal emission. Taking the positive μ direction to be downwards, and for now taking

$$I_1^+(\mu,\phi) = \int_0^{2\pi}\int_0^1 I_0^+(\mu',\phi') t_{01}(\mu',\mu,\phi'-\phi)\mu d\mu d\phi + \int_0^{2\pi}\int_0^1 I_1^-(\mu',\phi') r_{10}(\mu',\mu,\phi'-\phi)\mu d\mu d\phi + J_{01}^+(\mu,\phi) \tag{2}$$

where the extra μ factor is due to the illumination angle geometry. similarly

$$I_0^-(\mu,\phi) = \int_0^{2\pi}\int_0^1 I_0^+(\mu',\phi') r_{01}(\mu',\mu,\phi'-\phi)\mu d\mu d\phi + \int_0^{2\pi}\int_0^1 I_1^-(\mu',\phi') t_{10}(\mu',\mu,\phi'-\phi)\mu d\mu d\phi + J_{10}^-(\mu,\phi)$$

First of all, we can see that Fourier decomposition in azimuth angle works! Consider the 1st term on the RHS of Eq. 2. Writing the radiance and transmission as Fourier series we get:

$$\int_0^{2\pi}\left[\sum I_{0r}^+(\mu')\cos r\phi'\left[\sum t_{01s}(\mu',\mu)\cos s(\phi'-\phi)\right]\mu d\mu d\phi = (1+\delta_{0s})\pi\sum\cos s\phi\int_0^1 I_{0s}^+(\mu')t_{01s}(\mu',\mu)\mu d\mu\right.$$

where the integrals over the cross-terms average out leaving just the diagonal elements, forming the required Fourier series. In practice of course, the integral over zenith angle is replaced by the Lobatto quadrature scheme:

$$R = \sum_{s=0}^M \pi(1+\delta_{0s})\cos s\phi\sum_{n=1}^N I_{0sn}^+ t_{01sn}w_n$$

and thus the integral equations are replaced by matrix equations:

$$I_1^+ = t_{01}I_0^+ + r_{10}I_1^- + J_{01}^+$$
$$I_0^- = r_{01}I_0^+ + t_{10}I_1^- + J_{10}^-$$

$$I_2^+ = t_{12}I_1^+ + r_{21}I_2^- + J_{12}^+$$
$$I_1^- = r_{12}I_1^+ + t_{21}I_2^- + J_{21}^-$$

$$I_2^+ = t_{02}I_0^+ + r_{20}I_2^- + J_{02}^+$$
$$I_0^- = r_{02}I_0^+ + t_{20}I_2^- + J_{20}^-$$

which may be rearranged to give:

$$t_{02} = t_{12}\left(E - r_{10}r_{12}\right)^{-1}t_{01}$$
$$r_{20} = r_{21} + t_{12}\left(E - r_{10}r_{12}\right)^{-1}r_{10}t_{21}$$
$$J_{02}^+ = J_{12}^+ + t_{12}\left(E - r_{10}r_{12}\right)^{-1}\left(J_{01}^+ + r_{10}J_{21}^-\right)$$

This is what the subroutine add.f does where the subscripts become:

$$T_{ans} = T_1\left(E - R_{sub}R_1\right)^{-1}T_{sub}$$
$$R_{ans} = R_1 + T_1\left(E - R_{sub}R_1\right)^{-1}R_{sub}T_1$$
$$J_{ans} = J_1 + T_1\left(E - R_{sub}R_1\right)^{-1}\left(J_{sub} + R_{sub}J_1\right)$$

which looks a lot simpler!

The last complication is how do you start off? We need an initial reflection and transmission matrix for one layer from which we'll calculate everything else. The scheme used is to start with a very optically thin layer where $\Delta_{10} = \tau_1 - \tau_0 < 2^{-16}$ then:

$$R^r = \Delta_{10}M^{-1}\omega\pi(1+\delta_{0r})P_{+-}^r C$$
$$T^r = E - \Delta_{10}M^{-1}\left[E - \omega\pi(1+\delta_{0r})P_{++}^r C\right]$$

where

$$
M = \begin{pmatrix} \mu_1 & 0 & . & . & 0 \\ 0 & \mu_2 & . & . & . \\ . & . & . & . & . \\ . & . & . & . & 0 \\ 0 & . & . & 0 & \mu_M \end{pmatrix}, \quad
C = \begin{pmatrix} w_1 & 0 & . & . & 0 \\ 0 & w_2 & . & . & . \\ . & . & . & . & . \\ . & . & . & . & 0 \\ 0 & . & . & 0 & w_M \end{pmatrix}
$$

The various elements are not so obscure as they at first seem! The product of $\Delta_{10} M^{-1}$ gives the optical depth at each zenith angle. The $\pi(1+\delta_{or})$ term deals with the integration over azimuth angle factors (see Eq. 14) and the C matrix contains the weights required for this matrix version of an integral equation.

Consider the case for isotropic scattering where $p=1/4\pi$ for the zeroth order only. Then:

$$
R_0 = \begin{pmatrix}
\Delta_{10}\omega w_1/2\mu_1 & \Delta_{10}\omega w_2/2\mu_1 & . & . & \Delta_{10}\omega w_N/2\mu_1 \\
\Delta_{10}\omega w_1/2\mu_2 & \Delta_{10}\omega w_2/2\mu_2 & . & . & \Delta_{10}\omega w_N/2\mu_2 \\
. & . & . & . & . \\
. & . & . & . & . \\
\Delta_{10}\omega w_1/2\mu_N & \Delta_{10}\omega w_2/2\mu_N & . & . & \Delta_{10}\omega w_N/2\mu_N
\end{pmatrix}
$$

$$
T_0 = \begin{pmatrix}
1-\Delta_{10}(1-\omega w_1/2)/\mu_1 & \Delta_{10}\omega w_2/2\mu_1 & . & . & \Delta_{10}\omega w_N/2\mu_1 \\
\Delta_{10}\omega w_1/2\mu_2 & 1-\Delta_{10}(1-\omega w_2/2)/\mu_2 & . & . & \Delta_{10}\omega w_N/2\mu_2 \\
. & . & . & . & . \\
. & . & . & . & . \\
\Delta_{10}\omega w_1/2\mu_N & \Delta_{10}\omega w_2/2\mu_N & . & . & 1-\Delta_{10}(1-\omega w_N/2)/\mu_N
\end{pmatrix}
$$

The subroutine `double1.f` calculates the required matrices for a homogeneous layer by first defining a very thin layer as above and then doubling up the thickness N times to get the resultant matrices. In this case N is 16.

8.3.1 Quadrature details

In the scattering code it is assumed that for light incident at $\mu=\cos\theta, \phi=0$ and leaving at $\mu'=\cos\theta', \phi=\phi$, where θ relates to zenith angles and ϕ to azimuth angles, the phase function may be expanded in the Fourier series:

$$
p(\theta) = p(\mu,\mu',\phi) = \sum_{l=0}^{N} p_l(\mu,\mu')\cos l\phi
$$

With such an expansion, the coefficients may be found in the usual way:

$$
p_0(\mu,\mu') = \frac{1}{2\pi}\int_0^{2\pi} p(\mu,\mu',\phi)d\phi
$$

for *l=0* and for *l>0* :

$$
p_l(\mu,\mu') = \frac{1}{\pi}\int_0^{2\pi} p(\mu,\mu',\phi)\cos l\phi d\phi
$$

Hence the first check for a sufficiently precise quadrature scheme is that once *N* coefficients have been calculated, that they adequately describe the function to be fitted to within a prescribed error limit. Using this expansion of the phase matrix, it is <u>assumed</u> that a similar expansion applies to the final computed radiances also [R19]. Thus the PM coefficients calculated above may be used in *N+1* independent radiative

transfer equations and the final radiances calculated at the end by combining all the components.

The second quadrature takes place over zenith angle. Checking the fit here is more problematic but a good place to start is to ensure that the integral of the phase function over solid angle is close to unity. i.e.:

$$\iint p(\mu, \mu', \phi) d\mu' d\phi = 1 = \int_0^{2\pi} d\phi \int_{-1}^{1} d\mu' \sum_{l=0}^{N} p_l(\mu, \mu') \cos l\phi$$

$$= \sum_{l=0}^{N} \int_0^{2\pi} \cos l\phi d\phi \int_{-1}^{1} p_l(\mu, \mu') d\mu' = 2\pi \int_{-1}^{1} p_0(\mu, \mu') d\mu'$$

i.e.:

$$\int_{-1}^{1} p_0(\mu, \mu') d\mu' = \frac{1}{2\pi}$$

As can be seen, all integrals for $l > 0$ are identically zero and hence the normalisation check refers to the 0th order only. In practice, the scattering code splits up the phase function into forward and reverse components and, using Guassian-Lobatto quadrature which allows computation of radiances at $\mu = \pm 1$, integrates as:

$$r_j = \sum_{i=1}^{N} P_{ij}^{+-} w_i \quad , \quad t_j = \sum_{i=1}^{N} P_{ij}^{++} w_i$$

Clearly we want to ensure that $t_j + r_j = 1$ and this is exactly what the renormalisation scheme of Hansen [R18] does, where the forward part of the phase function is iteratively modified until $\varepsilon_j = |1 - r_j - t_j| \leq 10^{-14}$. Such an approach is sensible for highly forward scattering particles where the main quadrature error appears in the forward part but is maybe of dubious quality for particles with both sharp forward and back scattering peaks. Whatever the quality of the assumption however, it is probably safe to assume that there are insufficient quadrature points if ε_j is greater than some prescribed value (say 2%) for any value of j and this can be used as a check for the zenith integration.

8.4 Miscellaneous Calculation Details

Final points about averaging and so on are as follows. For a Lambertian reflecting layer, the matrix is defined as, for zeroth order Fourier component only as:

$$R_L = \begin{pmatrix} 2A\mu_1 w_1 & 2A\mu_2 w_2 & . & . & 2A\mu_M w_M \\ 2A\mu_1 w_1 & 2A\mu_2 w_2 & . & . & 2A\mu_2 w_2 \\ . & . & . & . & . \\ . & . & . & . & . \\ 2A\mu_1 w_1 & 2A\mu_2 w_2 & . & . & 2A\mu_2 w_2 \end{pmatrix}$$

And for incident sunlight which is clearly an irradiance from a specific zenith and azimuth angle, the incident radiance vector is defined as:

$$I = \begin{pmatrix} 0 \\ I_s/(2\pi w_i) \\ . \\ 0 \end{pmatrix}$$

where the appearance of the weight w_2 ensures correct treatment in the zenith quadrature of a delta function.

8.5 Calculating scattering properties

If a scattering path is defined in the '.pat' path file, then for the <u>Radtrans</u> program to run, further details of the scattering calculation must be supplied. Extinction cross-sections and single scattering albedos are already supplied in the '.xsc' file discussed previously. However <u>Radtrans</u> also needs to know how to calculate the phase function of the aerosols and how to integrate over the solid angles. All these details are contained in the '.sca' file which is discussed more fully below. For simple cases like dipole and isotropic scattering, the calculation of the phase function may be performed in real time. However for more complicated cases like Mie scattering (which is the analytical solution to the case of scattering of plane electromagnetic waves by dielectric spheres) the calculation must be done off-line using the program <u>Makephase</u>. It should be noted that <u>Makephase</u> also writes the extinction cross-section and single scattering albedo '.xsc' file.

8.5.1 Calculating scattering properties with <u>Makephase</u>

<u>Makephase</u> is an interactive program which prompts the user on how to set up the Mie calculation. When the program is invoked, the user is given the following calculation options (N.B. the term 'mode' below refers to a particular particle type, i.e. composition, radius, size distribution):

```
 Enter number of aerosol modes :
2
 enter wavenumber range:  (start / end / delta)
1900,2100,200
 Enter phase function calculation ID:        1

 1) Mie Scattering. Standard Gamma Distribution
 2) Mie Scattering. Log-Normal distribution
 3) Mie Scattering. Single particle size
 4) Isotropic scattering
 5) Henyey-Greenstein scattering
 6) Dipole scattering
```

As can be seen the user must first define the number of aerosol types and the required wavenumber grid. The user must then select the required model following which certain other constants must be supplied:

<u>1) Mie Scattering. Standard Gamma Distribution</u>

Here the aerosols are assumed to have the following size distribution [R11]:

$$n(r) \propto r^{(1-3b)/b} e^{-r/ab}$$

Here a is the effective mean radius of the distribution and b is the effective variance.

The user must thus supply first both a and b.
The user must then either enter the real and imaginary parts of the refractive index of the aerosol or specify a refractive index table.

The user may then let <u>Makephase</u> select the integration range over the radius distribution to determine the mean phase function or may set it manually

The extinction cross-section and single scattering albedo are calculated automatically and written to a '.xsc' file.

 2) Mie Scattering. Log-Normal distribution

Here the aerosols are assumed to have the following size distribution:

$$n(r) \propto \frac{1}{r} \exp\left[-\left(\ln r - \ln r_g \right)^2 / \left(2\sigma_g^2 \right) \right]$$

where r_g is the mean radius of the particle distribution and σ_g may be thought of as the mean value of $\ln(r)/\ln(r_g)$.

Hence the user must supply r_g (in microns) and σ_g.

After that the user must then again supply the refractive index and the preferred integration range (if required).

The extinction cross-section and single scattering albedo are calculated automatically and written to a '.xsc' file.

3) Mie Scattering. Single particle size

Here the user is simply required to supply the mean particle radius followed by the refractive index.

The extinction cross-section and single scattering albedo are calculated automatically and written to a .xsc file.

4) Isotropic scattering

Here the phase function is set to be uniform in all directions. <u>Makephase</u> prompts the user for the extinction cross-section and single scattering albedo at each wavelength

5) Henyey-Greenstein scattering

The Henyey-Greenstein phase function (a useful parameterised approximation to real phase functions) was described earlier and is defined as:

$$f_2 = c_1$$

$$f_1 = 1 - f_2$$

$$hg_{11} = 1 - c_2^2$$

$$hg_{12} = 2 - hg_{11}$$

$$hg_{21} = 1 - c_3^2$$

$$hg_{22} = 2 - hg_{21}$$

$$p(\cos\alpha) = f_1 \cdot hg_{11} / (hg_{12} - 2c_2 \cos\alpha)^{3/2} + f_2 \cdot hg_{21} / (hg_{22} - 2c_3 \cos\alpha)^{3/2}$$

where the constants c_1, c_2 and c_3 are supplied by the user, and α is the scattering angle.

Makephase again prompts for the extinction cross section and single scattering albedo.

6) Dipole scattering

Here the user must supply the mean particle radius (microns) and the complex refractive index. The phase function is then calculated as:

$$p(\cos\alpha) = \frac{3}{4}\left(1 + \cos^2\alpha\right)$$

$$Q_{sca} = \frac{8}{3}x^4 \left| \frac{n_c^2 - 1}{n_c^2 + 2} \right|$$

$$Q_{abs} = -4x\,\text{Im}\left\{ \frac{n_c^2 - 1}{n_c^2 + 2} \right\}$$

Where x is the size parameter $2\pi r/\lambda$, n_c is the complex refractive index and Q_{sca} is the scattering efficiency factor which is the ratio of the scattering x-section to the geometrical x-section. By conservation of energy:

$$A_{ext} = \pi r^2 \times Q_{ext} = \pi r^2 (Q_{sca} + Q_{abs})$$

$$\omega = \frac{Q_{sca}}{Q_{ext}}$$

Where A_{ext} is the extinction cross section and ω is the single scattering albedo.

In all cases the calculated phase functions are written to the files PHASENN.DAT where the two-digit number NN corresponds to the number of the mode.

8.5.2 Format of '.sca' file.

If a scattering path is defined in the '.pat' file, then a scattering '.sca' file must exist which contains additional details of the calculation. This file also has the same name as the driver and has the following format:

LOBATTO.DAT generated by Testlob   18/11/94

```
21
 3.784971655960363E-002  7.566327910711329E-002
 0.113333349888654      7.522988661327083E-002
 0.188165825654151      7.436558405320763E-002
 0.261921504390139      7.307532206213684E-002
 0.334176927804947      7.136648893356323E-002
 0.404518200290651      6.924887923879364E-002
 0.472542436198248      6.673461588649052E-002
 0.537859990923905      6.383809986245448E-002
 0.600096702575684      6.057592108845711E-002
 0.658896148204803      5.696651339530945E-002
 0.713921568917289      5.303148882011104E-002
 0.764857292175293      4.878216609358788E-002
 0.811413731158588      4.432398443333577E-002
 0.853316624328711      3.944781980769774E-002
 0.890346703886177      3.485291182131400E-002
 0.922246795926299      2.863839588550722E-002
 0.948912561286221      2.528072619188358E-002
 0.970082573462594      1.693047310774074E-002
 0.985752649473397      1.542868909873032E-002
 0.995737073151015      6.230266387428623E-003
 1.00000000000000       1.161440185830430E-003
1                       ! isol
5.2                     ! dist
1                       ! lowbc
1.0                     ! galb
42.0, 0.0               ! sol_ang,emiss_ang
0.0                     ! aphi
0                       ! nf (azimuth orders)
3                       ! ncont
1                       ! iscat(1)
1                       ! norm(1)
0                       ! ncons(1)
1                       ! iscat(2)
1                       ! norm(2)
0                       ! ncons(2)
1                       ! iscat(3)
1                       ! norm(3)
0                       ! ncons(3)
```

The first line contains the header and the second line the number of points N to use in the zenith angle quadrature. The next N lines contain the ordinates of cos(theta) and the weights to use in the Gaussian-Lobatto quadrature. The remaining parameters are defined as follows:

isol	integer	Sunlight switch.
		0=night
		1=day
dist	real	Distance of planet from sun in AU
lowbc	integer	Lower boundary condition identifier:

		0 = thermally emitting boundary
		1 = Lambertian reflecting boundary
galb	real	Albedo of lower boundary
sol_ang	real	Sunlight incident zenith angle
emiss_ang	real	Viewing, or emission, zenith angle
aphi	real	Azimuth angle between incident and viewing directions
nf	integer	Number of orders in azimuth fourier expansion. 0 = azimuthally symmetric
ncont	integer	Number of scattering modes
iscat	integer	Phase function identifier
		0 = dipole
		1 = isotropic scattering
		2 = Henyey-Greenstein - (3 coefficients read into array CONS below)
		3 = Legendre polynomial (ICONS (<= 10) coefficients in read into array CONS below)
		4 = Phase function read in from disk file
		5 = Henyey-Greenstein pretabulated phase functions
norm	integer	Normalization identifier:
		0 = none
		1 = Hansen
		2 = Barkstrom
ncons	integer	Number of constants defined for Phase function
cons	real(10)	List of ncons constants

Note: if iscat > 3 then ncons and cons are replaced by the name of the phase function file (created by Makephase) for the particular dust mode such that the end of the file looks like this:

```
1                    ! isol
5.2                  ! dist
1                    ! lowbc
1.0                  ! galb
42.0, 0.0            ! sol_ang,emiss_ang
0.0                  ! aphi
0                    ! nf (azimuth orders)
3                    ! ncont
4                    ! iscat(1)
1                    ! norm(1)
PHASE1.DAT           ! phase(1)
1                    ! iscat(2)
1                    ! norm(2)
PHASE2.DAT           ! phase(2)
1                    ! iscat(3)
1                    ! norm(3)
PHASE3.DAT           ! phase(3)
```

Appendix A. RADTRANS gas ID codes

Since HITRAN and GEISA use slightly different gas identifiers, RADTRANS uses its own gas identification system, based on HITRAN, so that both data formats can be transcribed. The gas identification system used is shown in Table A.1. The local gas identifier agrees with HITRAN IDs as far as possible (Jan 1988) with extra IDs for gases not in the HITRAN compilation e.g. those in the GEISA compilation. Isotope identifiers also agree as far as possible with HITRAN identifiers. Similar (1,2,3...n) identifiers have been defined for additional gases. For example, ID = 1, ISO =1 refers to the first isotope of H_2O. ID = 6, ISO = 3 is CH_3D. ID = 6, ISO = 0, means use all isotopes of CH_4 with relative isotope abundances equal to their terrestrial values.

Table A.1 Gases Defined for RADTRAN

RADTRAN ID	Gas	Hitran04 ID	Geisa03 ID	Number of Isotopes	Radtran order Isotopes
1	H_2O	1	1	5	161,181,171,162, 182
2	CO_2	2	2	9	626,636,628,627, 638,637,828,728, 838
3	O_3	3	3	5	666,668,686,667, 676
4	N_2O	4	4	5	446,456,546,448, 447
5	CO	5	5	6	26,36,28,27,38,37
6	CH_4	6	6	3	211,311,212
7	O_2	7	7	3	66,68,67
8	NO	8	8	3	46,56,48
9	SO_2	9	9	2	626,646
10	NO_2	10	10	1	646
11	NH_3	11	11	2	411,511
12	HNO_3	12	13	1	146
13	OH	13	14	3	61,81,62
14	HF	14	15	1	19
15	HCL	15	16	2	15,17
16	HBr	16	17	2	19,11
17	HI	17	18	1	17
18	ClO	18	19	2	56,76
19	OCS	19	20	7	622,624,632,822, 623,634,722
20	H_2CO	20	21	3	126,136,128
21	HOCl	21	32	2	165,167
22	N_2	22	33	1	44
23	HCN	23	27	3	124,134,125
24	CH_3CL	24	34	2	215,217
25	H_2O_2	25	35	1	166
26	C_2H_2	26	24	2	221,231
27	C_2H_6	27	22	2	226,236
28	PH_3	28	12	1	131
29	C_2N_2	40	29	1	224
30	C_4H_2	43	30	1	211

NOTES:

48   numbers in red are made up numbers to allow mapping between hitran and geisa formats and vice-versa. If Hitran or Geisa are updated to include more molecules this will cause problems if the made up IDs overlap with the updated IDs.

RADTRAN ID	Gas	Hitran04 ID	Geisa03 ID	Number of Isotopes	Isotopes
31	HC$_3$N	44	31	5	124,144,154,164,125,225
32	C$_2$H$_4$	38	25	2	211,311
33	GeH$_4$	45	26	1	411
34	C$_3$H$_8$	42	28	1	211
35	HCOOH	32	37	1	261
36	H$_2$S	31	36	3	121,131,141
37	COF$_2$	29	38	1	269
38	SF$_6$	30	39	1	29
39	H$_2$	47	43	2	11,12
40	He	n/a	n/a	1	4
41	AsH$_3$	46	45	1	181
42	C$_3$H$_4$	41	40	1	211
43	ClONO$_2$	35	42	2	564,764
44	HO$_2$	33	41	1	166
45	O	34	46	1	6
46	NO+	36	47	1	4
47	CH$_3$OH	39	44	1	211
48	H	n/a	n/a	1	1
49	C$_6$H$_6$	48	48	1	221
50	CH3CN	49	49	1	241
51	CH2NH	50	50	4	241, 341, 251, 242
52	C2H3CN	51	51	6	241, 441, 541, 641, 242, 251
53	HCP	52	52	1	123
54	CS	53	53	1	22
55	HC5N	54	54	1	241
56	HC7N	55	55	1	241
57	C2H5CN	56	56	5	241,441,541,641,251
58	CH3NH2	57	57	1	124
59	HNC	58	58	4	142 143 152 242
60	Na	59	59	1	11
61	K	60	60	1	19
62	TiO	61	61	5	581,582,583,584,585
63	VO	62	62	1	591