

libRadtran

library for radiative transfer calculations
Edition 1.0 for libRadtran version 1.0-beta
September 2002

Arve Kylling and Bernhard Mayer

Copyright © 1997-2001 Arve Kylling, Bernhard Mayer.

This edition of the libRadtran documentation is consistent with version 1.0-beta of libRadtran.

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the same conditions as for modified versions.

Preface

libRadtran is a library of radiative transfer routines and programs. It has evolved from the `uvspec` radiative transfer model. If you are not familiar with `uvspec`, please note the following: The central program of the libRadtran package is an executable called `uvspec` which can be found in the ‘`tools`’ directory. If you are interested in a user-friendly program for radiative transfer calculations, this is the main information you need to know. A description of `uvspec` is provided in the first part of this manual, and examples including various input files for different atmospheric conditions are provided in the examples directory.

The ‘`tools`’ directory also provides related utilities, like e.g. a mie program (`mie`), some utilities for the calculation of the position of the sun (`zenith`, `noon`, `sza2time`), a few tools for interpolation, convolution, and integration (`spline`, `conv`, `integrate`), and some other small tools.

The second part of the manual (which is only available in the developer manual) describes library routines which might be of interest if you want to write your own programs. The documentation is far from complete at present. Available are routines to read ASCII files, to do interpolations, integrations, convolutions, for Mie theory, to calculate the position of the sun, and some other stuff. More might be available somewhen in the future.

Please note that this document is by no means complete. It is under rapid development and major changes will take place.

Acknowledgements

Many people have already contributed to libRadtran’s development. In addition to Arve Kylling (arve.kylling@nilu.no) and Bernhard Mayer (bernhard.mayer@dlr.de), the following people have contributed to libRadtran or helped out in various other ways.

- The `disort` solver was developed by Knut Stamnes, Warren Wiscombe, S.C. Tsay, and K. Jayaweera
- Warren Wiscombe provided the Mie code `MIEV0`, and the routines to calculate the refractive indices of water and ice, `REFWAT` and `ICEWAT`.
- Seiji Kato (kato@aerosol.larc.nasa.gov) provided the correlated-k tables described in Kato et al. (1999).
- Tom Charlock (t.p.charlock@larc.nasa.gov), Quiang Fu (qfu@atm.dal.ca), and Fred Rose (f.g.rose@larc.nasa.gov) provided the most recent version of the Fu and Liou code.
- David Kratz (kratz@aquila.larc.nasa.gov) provided the routines for the simulation of the AVHRR channels described in Kratz (1995).
- Frank Evans (evans@nit.colorado.edu) provided the `polradtran` solver.
- Ola Engelsen provided data and support for different ozone absorption cross sections.
- Albano Gonzales (aglezf@ull.es) included the Yang et al. (2000), Key et al. (2002) ice crystal parameterization.
- Tables for the radiative properties of ice clouds for different particle “habits” were obtained from Jeff Key and Ping Yang, Yang et al. (2000), Key et al. (2002).

- Paul Ricchiazzi (paul@icess.ucsb.edu) and colleagues allowed us to include the complete gas absorption parameterization of their model SBDART into uvspec.
- Many unnamed users helped to improve the code by identifying or fixing bugs in the code.

— The Detailed Node Listing —

1 A Brief Overview of libRadtran

This manual documents how to install and use libRadtran and corresponds to libRadtran version 1.0-beta.

libRadtran is a collection of C and Fortran functions and subroutines useful for radiative transfer calculations in the Earth's atmosphere. In addition, programs in C, Fortran and Perl using the libRadtran functions and subroutines are included to allow the user to develop his/her own programs.

It is expected that the reader is familiar with radiative transfer terminology. In addition, a variety of techniques and parameterizations from various sources are used. For more information about the usefulness and applicability of these methods in certain contexts, the user is referred to the referenced literature.

1.1 Radiative transfer calculations

The central program of the libRadtran package is called **uvspec**. It was originally designed to calculate spectral irradiance in the ultraviolet and visible spectral ranges, hence the name. Over the years, **uvspec** has evolved to become a tool for many applications, including the simulation of instruments, the calculation of the radiation budget of the Earth, or the development of remote sensing techniques. **uvspec** is driven with a human-readable input file which allows the definition of the model input in a user-friendly way. Various commands are available to specify the properties of the atmosphere, including Rayleigh scattering, molecular absorption, aerosols, water and ice clouds, and the surface albedo. A selection of several radiative transfer solvers is available to simulate different aspects of the radiation field, including the **disort** solver and a pseudo-spherical version of this code, a fast twostream code, Frank Evans' **polRadtran**, and the three-dimensional **MYSTIC** code.

Radiative transfer calculations with **uvspec** are straightforward. The input to the radiative transfer solver is specified in the 'input_file'. Output is written to **stdout** and can easily be re-directed into an 'output_file':

```
uvspec < input_file > output_file
```

The syntax of input and output is described in [Section 2.1 \[uvspec\]](#), page 5. Examples of **uvspec** input and output files are found in the 'examples' directory.

1.2 Ozone retrieval from global irradiance measurements

Stamnes et al. (1991) described a method to derive the total ozone column from global irradiance measurements. The method is based on the comparison of measured irradiance ratios at two wavelengths in the UV part of spectrum with a synthetic chart of this ratio computed for a variety of ozone values. One of the wavelengths should be appreciably absorbed by ozone compared with the other. Typically choices are 305 and 340 nm. The method is reliable under cloudfree conditions, but increasingly overestimates the ozone column for optically thicker clouds, Mayer et al. (1998).

Within libRadtran the method is implemented with two pieces of software

Gen_o3_tab

Generates a look-up-table of ozone values as a function of the irradiance ratio for a given pair of wavelengths and solar zenith angle.

read_o3_tab

Takes as input the measured ratio and solar zenith angle, reads the look-up-table, and returns an ozone value.

The use of these tools is described in [Section 2.10 \[Geno3tab\]](#), page 47.

1.3 Cloud optical thickness from global irradiance measurements

Stamnes et al. (1991) described a method to derive a representative cloud optical depth from global irradiance measurements. The method compares the measured irradiance at a wavelength where ozone absorption is minimal to irradiances generated by a radiative transfer model as a function of cloud optical thickness. The method is very sensitive to the ground albedo and independent measurement of the albedo are needed when there is snow on the surface.

Within libRadtran the method is implemented with two pieces of software

Gen_wc_tab

Generates a look-up-table of water cloud optical depths for various solar zenith angles for a given wavelength.

read_o3_tab

Takes as input the measured irradiance and solar zenith angle, reads the look-up-table, and returns a cloud optical depth.

The use of these tools is described in [Section 2.11 \[Genwctab\]](#), page 48.

1.4 ... and much more

Apart from the above described programs, the ‘tools’ directory contains related utilities, like e.g. a mie program (**mie**), a tool to calculate wavelength-dependent cloud properties (**cldprp**), a simple shell script to add levels to existing profiles (**addlevel**), some utilities for the calculation of the position of the sun (**zenith**, **noon**, **sza2time**), a few tools for interpolation, convolution, and integration (**spline**, **conv**, **integrate**), and some other small utilities. Generally, these programs will give some information about their purpose and usage when called without arguments.

2 Some useful tools

2.1 uvspec

`uvspec` calculates the radiation field in the Earth's atmosphere for a given set of input parameters. It reads input from standard input, and outputs to standard output. It is normally invoked in the following way:

```
uvspec < input_file > output_file
```

The format of the input and output files are described below. Several realistic examples of input files are subsequently given.

Warning: Please note the error checking on input variables is not complete at the moment. Hence, if you provide erroneous input, the outcome is unpredictable.

2.1.1 The uvspec input file

`uvspec` is controlled in a user-friendly way. The control options are named in a (hopefully) intuitive way. In the following, several examples are given, how to create an input file, how to define a cloudless sky atmosphere, how to add aerosols and clouds, etc. All following examples are taken from the libRadtran examples directory and are part of the `uvspec` self-check. For a complete listing and explanation of all input options, have a look at [section 2.1.3 \[Complete description of input parameters\]](#), page 18.

The `uvspec` input file consists of single line entries, each making up a complete input to the `uvspec` program. First on the line comes the parameter name, followed by one or more parameter values. The parameter name and the parameter values are separated by white space. Filenames are entered without any surrounding single or double quotes. Comments are introduced by a `#`. Blank lines are ignored.

2.1.1.1 Cloudless, aerosol-free atmosphere

The simplest possible input file contains only a few lines:

```
data_files_path ../data/ # Location of internal uvspec data files

                                # Location of atmospheric profile file.
atmosphere_file ../data/atmmod/afglus.dat

                                # Location of the extraterrestrial spectrum
solar_file ../data/solar_flux/atlas_plus_modtran

wvn 310.0 310.0                # Wavelength range [nm]
```

The first three statements define the location of some data files: the uvspec data path (`data_files_path`) where all internal files are expected, the atmospheric profile (`atmosphere_file`), and the extraterrestrial spectrum (`solar_file`). The last line defines the desired wavelength range which is a monochromatic data point in this example. All other data which are not explicitly mentioned assume a default value which is "0" in most cases. Here, the solar zenith angle is 0, the surface albedo is 0, and the atmosphere does not contain clouds nor aerosols. Pressure, temperature, ozone concentration, etc. are read from `atmosphere_file`.

An example of a more complete input file for a clear sky atmosphere is

```
data_files_path ../data/ # Location of internal uvspec data files
                        # Location of atmospheric profile file.
atmosphere_file ../data/atmmod/afglus.dat
                        # Location of the extraterrestrial spectrum
solar_file ../data/solar_flux/atlas_plus_modtran
ozone_column 300.      # Scale ozone column to 300.0 DU
day_of_year 170        # Correct for Earth-Sun distance
albedo 0.2             # Surface albedo
sza 32.0               # Solar zenith angle
rte_solver disort      # Radiative transfer equation solver
deltam on              # delta-M scaling on
nstr 6                 # Number of streams
wvn 299.0 341.0        # Wavelength range [nm]
slit_function_file ../examples/TRI_SLIT.DAT
                        # Location of slit function
spline 300 340 1       # Interpolate from first to last in step

quiet
```

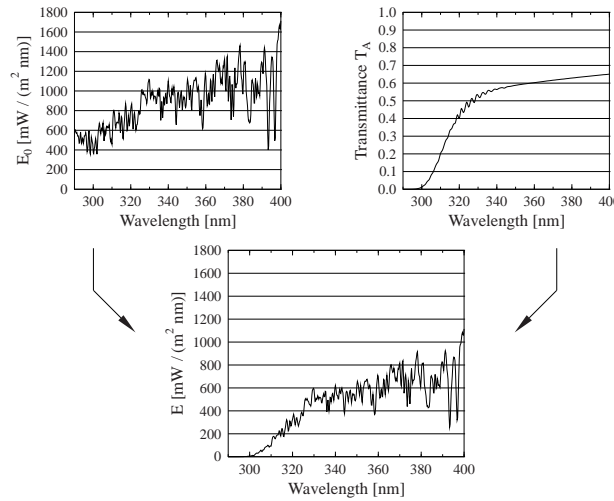
The atmosphere model, i.e. pressure, temperature, and ozone concentration profiles are read from `../data/atmmod/afglus.dat`. The extraterrestrial solar flux is read from the file `../data/solar_flux/atlas_plus_modtran`.

A wavelength dependent **surface albedo** may be specified using `albedo_file` instead of `albedo`. Non-Lambertian surface reflectance (BRDF) for vegetation and water may also be defined (please note that these require the use of `rte_solver disort2`). The BRDF of Vegetation is specified using `rpv_rho0`, `rpv_k`, and `rpv_theta`, following the definition of Rahman et al. (1993). Wavelength-dependent BRDF for vegetation can be defined with `rpv_file`. The BRDF of water surfaces is parameterized following Cox and Munk (1954a, 1954b) and Nakajima (1983). The respective parameters are the wind speed `cox_and_munk_u10`, the pigment concentration `cox_and_munk_pchl`, and the salinity `cox_and_munk_sal`. A complete description of these parameters is given in section [Section 2.1.3 \[Complete description of input parameters\]](#), page 18.

It is helpful to know some details about the **input/output wavelength resolution** in uvspec and how it can be influenced by the user. Basically there are three independent wavelength

grids, the **input grid**, the **internal grid**, and the **output grid**. The essential thing to know is that the internal grid is chosen by `uvspec` itself in a reasonable way, if not explicitly defined in the input file with `transmittance_wl_file`. The output grid is completely independent of the internal grid and is entirely defined by the `solar_file`. The wavelength grid of all other input data (e.g. albedo, optical properties of aerosols and clouds, etc) is also completely independent. These data are automatically interpolated to the resolution of the internal wavelength grid. Hence, only two constraints are set to the gridding of the input data: (1), the wavelength range has to cover all internal grid points; and (2), it should be chosen in a reasonable manner to allow reasonable interpolation (which essentially means, dense enough).

In the ultraviolet/visible, `uvspec` uses an internal grid with a step with of 0.5nm below 350nm and 1nm above 350nm. This is a conservative choice which fully resolves the broad ozone absorption bands and the slowly varying Rayleigh, aerosol, and cloud extinctions. The idea is outlined in the following figure which is taken from Mayer et al. (1997):



The transmittance (or reflectance) is calculated on a moderate resolution grid which reduces the number of calls to the `rte_solver` and hence the computational time. Then, the transmittance is interpolated to the wavelengths in the `solar_file` (which is usually defined with higher spectral resolution), multiplied with the extraterrestrial irradiance, and possibly post-processed. Hence, the wavelength in the output spectrum are those contained in the `solar_file` which has two important implications: (1) Only those wavelengths are output that are contained in the `solar_file`. If e.g. a monochromatic calculation is defined by setting `'wvn 327.14 327.14'`, there will only be output if the wavelength 327.14 is explicitly listed in `solar_file`; (2) this is also true at thermal wavelengths where the extraterrestrial irradiance is zero; hence, even for a calculation in the thermal range a `solar_file` must be specified that defines the output grid in the first column and arbitrary values in the second column. Keeping these points in mind, `solar_file` is a convenient way to define an arbitrary output grid.

If required, a **user-defined internal grid** can be specified with `transmittance_wl_file`. Note that this is a way to speed up the calculation considerably. E.g., for some applica-

tions the internal grid in the UV-A and visible can be set to 10nm which would reduce computational times by up to a factor of 10.

Things are completely different if one of the `correlated_k` parameterizations is selected (see below). In this case all flexibility is taken away from the user which is an inherent feature of the k distribution method. Internal grid as well as the extraterrestrial file are in this case defined by the choice of the parameterization itself.

2.1.1.2 Spectral resolution

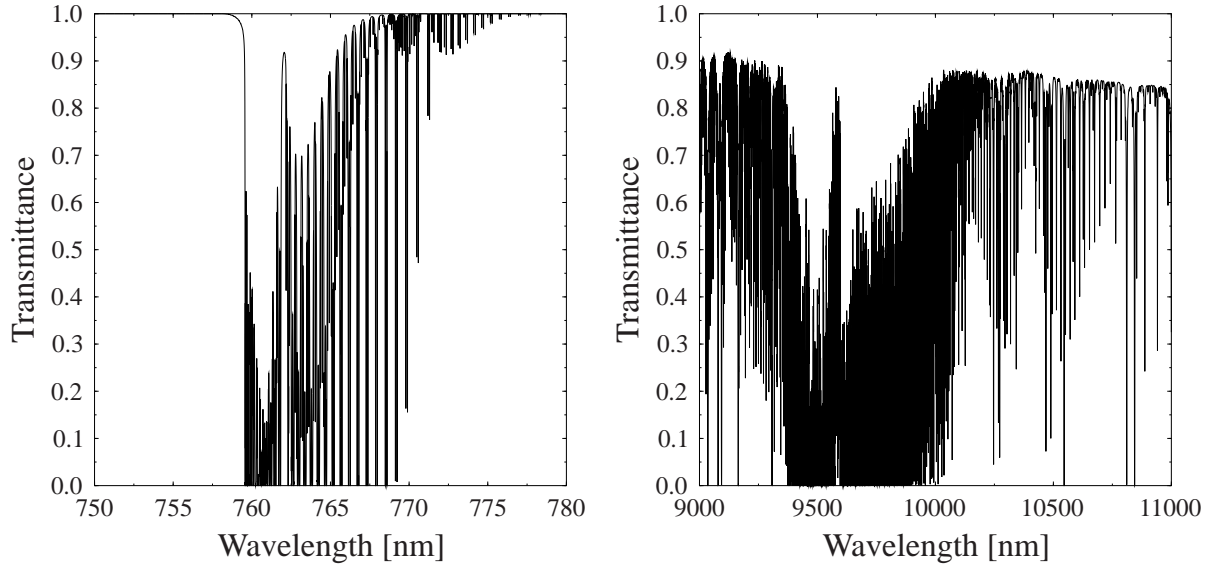
`uvspec` offers four different ways of spectral calculations:

1. **Spectrally resolved calculation** in the UV and visible spectral ranges;
2. **Line-by-line calculation** with user-defined molecular absorption data;
3. **The correlated-k method**.
4. **Pseudo-spectral calculation** with exponential-sum-fit, adopted from SBDART (Ricchi-azzi et al., 1998);

The choice of the method is determined by the problem and the decision is therefore entirely up to the user. The spectrally resolved calculation and the line-by-line calculation are more or less exact methods while the correlated-k distribution and the pseudo-spectral calculation are approximations that provide a compromise between speed and accuracy. In the following it is briefly described which method fits which purpose:

A **spectrally resolved calculation** is the most straightforward way, and will be the choice for all users interested in the ultraviolet and visible spectral ranges. In the UV/vis gas absorption generally occurs in broad bands with only slow spectral variation, the most important of these being the Hartley, Huggins, and Chappuis bands of ozone. Hence, a radiative transfer calculation every 1nm usually is sufficient to fully resolve any spectral variation using the method described in the last section. Absorption cross sections for various species are included, among them the most important O3 and NO2.

In the infrared, however, molecular absorption spectra are characterized by thousands of narrow absorption lines. There are two ways to treat these, either by highly resolved spectral calculations, so-called **line-by-line** calculations, or by a band parameterization. Concerning line-by-line, `uvspec` offers the possibility to define a spectrally resolved absorption cross section profile using `molecular_tau_file`. There is no option in libRadtran to generate such a `molecular_tau_file`, because (1) the HITRAN database which forms the basis for such calculations amounts to about 100 MByte which are updated continuously; and (2), there are sophisticated line-by-line programs available, like e.g. `genln2` (Edwards, 1992). Using `genln2` it is straightforward to create the input for `uvspec` line-by-line calculations. It is also planned to make line-by-line cross sections available for the six standard profiles that come with libRadtran. The following figure shows an example of a line-by-line calculation of the atmospheric transmittance in two selected solar and thermal spectral ranges, the O2A-absorption band around 760nm and a region within the infrared window around 10 micron.



All spectral lines in the left figure are due to absorption by oxygen, while the ones in the left figure are due to ozone, water vapour, and CO₂. Line-by-line is obviously the exact way for radiation calculations. For most applications, however, line-by-line is far too slow. Here one needs a band parameterization, and the most accurate of these is the so-called **correlated-k approximation**. `uvspec` contains several correlated-k parameterizations which are invoked with `correlated_k`, in particular Kato et al. (1999), Fu and Liou (1992/93), Kratz (1995), as well as the possibility to specify a user-defined one. For more information on these parameterizations please refer to the mentioned publications. Correlated-k is a powerful way to calculate spectrally integrated quantities, however, it takes away some flexibility. In particular, this means that the wavelength grid is no longer chosen by the user but by the parameterization, that is, by `uvspec`. The `uvspec` output is then no longer spectral quantities, e.g. $W / (m^2 nm)$, but integrated over the spectral bands, e.g. W / m^2 .

A simple but complete example for a correlated-k approximation of the solar spectrum:

```

# Conditions for the calculation of Figure 3 in
# Kato et al., JQSRT 62, 109-121, 1999.
# To compare the data, the direct irradiance calculated
# by uvspec has to be divided by cos(30 deg) because
# Kato et al. plot direct normal irradiance.

data_files_path ../data/ # Location of internal uvspec data files
                        # Location of atmospheric profile file.
atmosphere_file ../examples/AFGLMS50.DAT
                        # Location of the extraterrestrial spectrum
solar_file ../data/solar_flux/kato

albedo 0.2             # Surface albedo
sza 30.0               # Solar zenith angle
rte_solver twostr      # Radiative transfer equation solver

wvn 200.0 4300.0       # Wavelengths considered

correlated_k KATO      # Correlated-k by Kato et al. [1999]

quiet

```

Here, the solar spectrum is split up into 32 bands according to Kato et al. (1999). In order to calculate integrated shortwave irradiance, simply sum the outputs, or even simpler, add `output sum` to the input file.

For **pseudo-spectral calculations** in the whole spectral range, we have adopted the molecular absorption parameterization from SBDART by Ricchiazzi et al. (1998). The respective section of this paper says:

SBDART relies on low-resolution band models developed for the LOWTRAN 7 atmospheric trans-mission code (Pierluissi and Peng, 1985). These models provide clear-sky atmospheric transmission from 0 to 50000 cm⁻¹ and include the effects of all radiatively active molecular species found in the earth's atmosphere. The models are derived from detailed line-by-line calculations that are degraded to 20 cm⁻¹ resolution for use in LOWTRAN. This translates to a wavelength resolution of about 5 nm in the visible and about 200 nm in the thermal infrared. These band models represent rather large wavelength bands, and the transmission functions do not necessarily follow Beers Law. This means that the fractional transmission through a slab of material depends not only on the slab thickness, but also on the amount of material penetrated before entering the slab. Since the radiative transfer equation solved by SBDART assumes Beers Law behavior, it is necessary to express the transmission as the sum of several exponential functions (Wiscombe and Evans, 1977). SBDART uses a three-term exponential fit, which was also obtained from LOWTRAN 7. Each term in the exponential fit implies a separate solution of the radiation transfer equation. Hence, the RT equation solver only needs to be invoked three times for each spectral increment. This is a great computational economy compared to a higher order fitting polynomial, but it may also be a source of significant error.

The SBDART gas parameterization is invoked with `correlated_k` SBDART. The spectral resolution may be arbitrarily chosen by the user. If not explicitly defined with `transmittance_wl_file`, an internal grid with a step width of 0.5nm below 350nm and 1nm above 350nm is chosen which is practically overkill for most applications in the infrared. An extraterrestrial spectrum covering the complete solar range is provided at two different resolutions, `data/solar_flux/kurudz_1.0nm.dat` and `data/solar_flux/kurudz_0.1nm.dat`. An example for the solar range is shown in `examples/UVSPEC_SBDART_SOLAR.INP`:

```
# uvspec data files
data_files_path ../data/
atmosphere_file ../data/atmmod/afglus.dat
solar_file ../data/solar_flux/kurudz_1.0nm.dat

albedo 0.2          # Surface albedo
sza 30.0           # Solar zenith angle

rte_solver twostr   # Radiative transfer equation solver
wn 250.0 5000.0     # Wavelength range

correlated_k SBDART # select SBDART molecular absorption

quiet
```

while `examples/UVSPEC_SBDART_THERMAL.INP` shows how to do a thermal calculation:

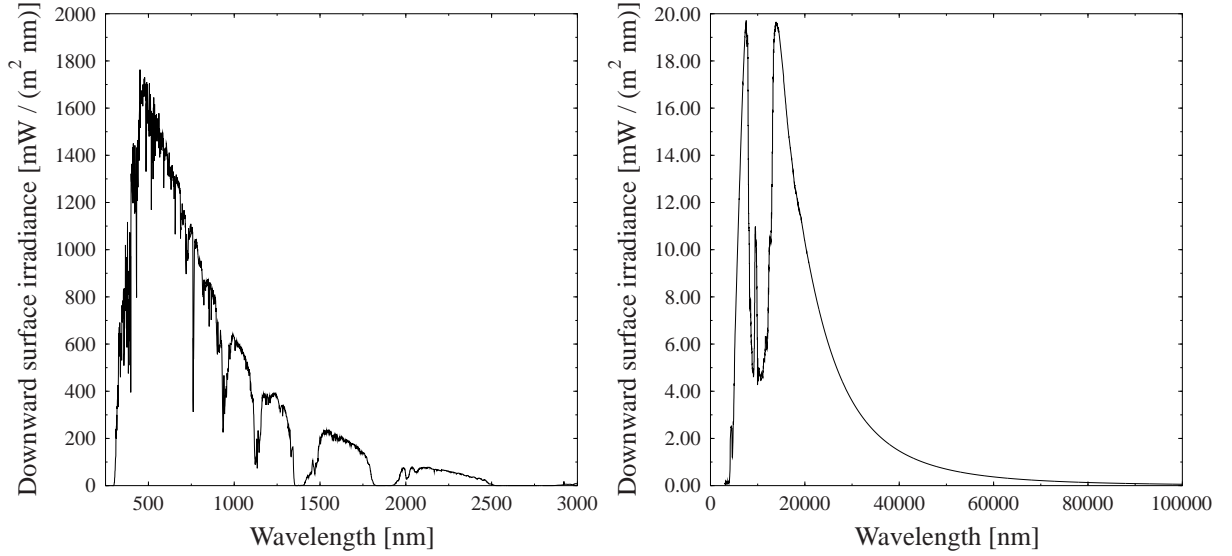
```
# uvspec data files
data_files_path ../data/
atmosphere_file ../examples/AFGLUS.70KM
solar_file ../examples/UVSPEC_SBDART_THERMAL.TRANS

source thermal

rte_solver twostr   # Radiative transfer equation solver
transmittance_wl_file ../examples/UVSPEC_SBDART_THERMAL.TRANS

correlated_k SBDART # select SBDART molecular absorption

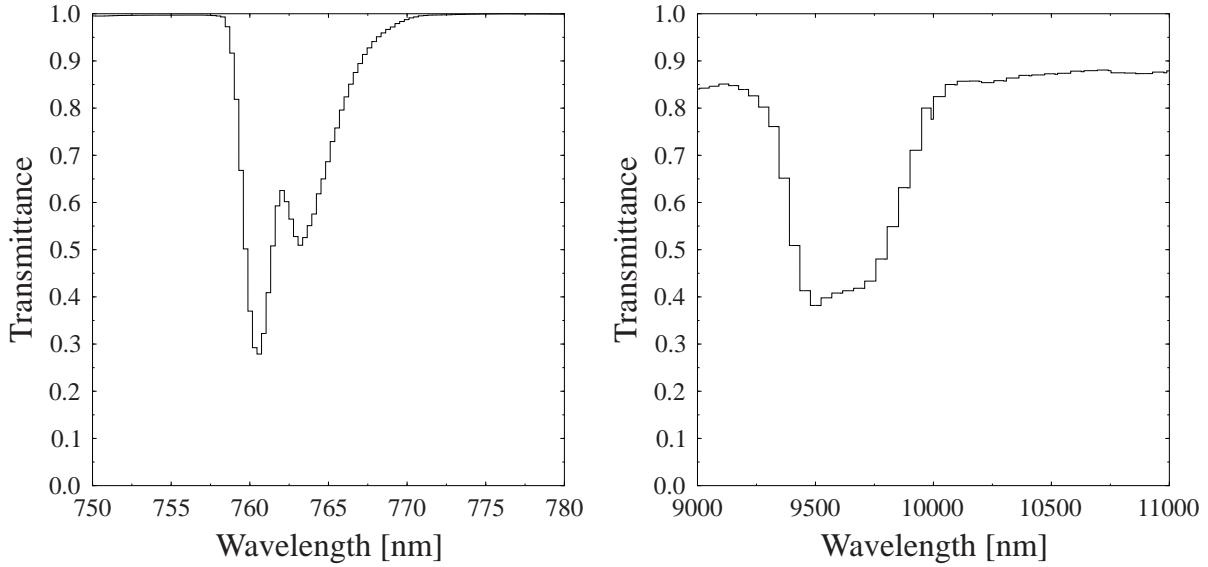
quiet
```



The figure shows the results of the solar and thermal calculations. The water vapour absorption bands in the solar range are clearly visible, as is the absorption window around 10 micron in the thermal. Please note the following points:

- Thermal radiation is per default output in W/(m² cm⁻¹), see documentation of `thermal_bandwith` and `thermal_bands_file`. To convert e.g. to W/(m² nm) multiply with k/λ where k is the wavenumber [cm⁻¹] and λ is the wavelength [nm]. To calculate band-integrated thermal quantities please consider `thermal_bands_file`.
- Even though no extraterrestrial irradiance is required, a `solar_file` has to be specified for the thermal case. The reason is that, as explained initially, the `solar_file` defines the output grid. The second column in `solar_file` can be chosen arbitrarily in this case because it is ignored. Please also note that the `solar_file` and the `transmittance_wl_file` are identical in this example which implies that exactly these wavelengths are output that are processed internally, rather than interpolated numbers.
- For the choice of the wavelength grid for the calculation (`transmittance_wl_grid`) please consider that the resolution of the absorption parameterization is 5cm⁻¹ which converts to 0.3nm at 750nm and to 50nm at 10 micron. Choosing higher resolutions for the internal wavelength grid (`transmittance_wl_file`) is usually a waste of computational time.
- Please also make sure to choose a fine enough spectral resolution in order to capture all absorption features.

The following figure shows two selected wavelength intervals of the solar and thermal range, to demonstrate the spectral resolution of the SBDART absorption parameterization.



The resolution is about 5cm-1 which translates to about 0.3nm in the left figure (oxygen A-band) and 50nm in the right figure (ozone absorption band in the atmospheric window). Compare this figure to the above line-by-line example to get an impression about the differences between both methods.

2.1.1.3 Aerosol

All options to set up and modify aerosol properties start with `aerosol_`. Aerosols may be specified in a hierarchical way. The most simple way to define an aerosol is by the command `aerosol_default` which will set up the aerosol model by E.P. Shettle, "Models of aerosols, clouds and precipitation for atmospheric propagation studies", in "Atmospheric propagation in the uv, visible, ir and mm-region and related system aspects", AGARD Conference Proceedings (454), 1989. The default properties are a rural type aerosol in the boundary layer, background aerosol above 2km, spring-summer conditions and a visibility of 50km. These settings may be modified with `aerosol_haze`, `aerosol_vulcan`, `aerosol_season`, and `aerosol_visibility`. More information can be introduced step by step, overwriting the default parameters. `aerosol_tau_file`, `aerosol_ssa_file`, and `aerosol_gg_file`, can be used to define the profiles of optical thickness, single scattering albedo, and asymmetry parameter. The integrated optical thickness can be set to a constant value using `aerosol_set_tau` or scaled with `aerosol_scale_tau`. The single scattering albedo may be scaled by `aerosol_scale_ssa` or set to a constant value by `aerosol_set_ssa`. The aerosol asymmetry factor may be set by `aerosol_set_gg`. The wavelength dependence of the aerosol optical depth is specified using the `aerosol_angstrom` parameter. `aerosol_moments_file` allows specification of the scattering phase function. If microphysical properties are available these may be introduced by defining the complex index of refraction `aerosol_refrac_index` or `aerosol_refrac_file` and the size distribution `aerosol_sizedist_file`. Finally, one may define the aerosol optical properties of each layer explicitly using `aerosol_files`.

The following list is an overview of the aerosol description parameters. The entries are arranged in a way that a parameter 'overwrites' all values higher up in the list.

aerosol_default
Generate default aerosol according to Shettle (1989)

aerosol_vulcan, aerosol_haze, aerosol_season, aerosol_visibility
Set Shettle (1989) aerosol properties (aerosol type, visibility)

aerosol_files
Specify optical properties of each layer explicitly, that is, extinction coefficient, single scattering albedo, and the moments of the phase function (everything as a function of wavelength).

aerosol_tau_file, aerosol_ssa_file, aerosol_gg_file
Overwrite profiles of optical thickness, single scattering albedo, and asymmetry parameter

aerosol_moments_file
Specify a phase function to be used instead of the Henyey-Greenstein phase function

aerosol_refrac_index, aerosol_refrac file, aerosol_sizedist_file
Calculate optical properties from size distribution and index of refraction using Mie theory. Here is an exception from the rule that ALL values defined above are overwritten because the optical thickness profile is re-scaled so that the optical thickness at the first internal wavelength is unchanged. It is done that way to give the user an easy means of scaling the specifying the optical thickness at a given wavelength.

aerosol_set_gg, aerosol_set_ssa, aerosol_scale_ssa, aerosol_set_tau, aerosol_scale_tau
Overwrite profiles of asymmetry parameter and single scattering albedo

aerosol_angstrom
Overwrite the profile of the optical thickness.

An example for a uvspec aerosol description is

```
aerosol_vulcan 1      # Aerosol type above 2km
aerosol_haze 6        # Aerosol type below 2km
aerosol_season 1      # Summer season
aerosol_visibility 20.0 # Visibility
aerosol_angstrom 1.1 0.2 # Scale aerosol optical depth
                        # using Angstrom alpha and beta
                        # coefficients
aerosol_scale_ssa 0.85 # Scale the single scattering albedo
                        # for all wavelengths
aerosol_set_gg 0.70   # Set the asymmetry factor
aerosol_tau_file ../examples/AERO_TAU.DAT
                        # File with aerosol optical depth profile
```

By combining this with the clear sky example given above a complete uvspec input file including aerosol is constructed.

2.1.1.4 Water clouds

All options to set up and modify water cloud properties start with `wc_`.

The easiest way to define a water cloud is to specify a `wc_file` which defines the liquid water content and effective droplet radius at each model level. By combining the following lines with the clear sky example given above a complete uvspec input file including water clouds is constructed.

```
wc_file ../examples/WC.DAT # Location of water cloud file
wc_set_tau 15.             # Set total water cloud optical depth
```

The file `WC.DAT` is found in the examples directory. Note that, other than for the aerosol, cloud properties are defined at each model level, rather than for a model layer. To define cloud properties per layer, one needs to specify `wc_layer`.

There are different ways to convert the microphysical properties to optical properties. Either a parameterization is used, like the one by Hu and Stamnes (1993) (which is the default), or by Mie calculations. The latter are very time-consuming, hence we decided not to include these online into uvspec but rather have an option to read in pre-calculated Mie tables. The option `wc_properties` controls the method: `hu` selects the Hu and Stamnes (1993) parameterization, `mie` selects pre-calculated Mie tables which are available at <http://www.libradtran.org>. Finally, there is the option to define an arbitrary file which is in the format as generated by Frank Evans' `cloudprp` which comes with SHDOM, see <http://nit.colorado.edu/~evans/shdom.html>.

As for the aerosol, there are several options to modify the optical properties of the clouds. And of course there is also the option of defining all cloud properties explicitly using `wc_files`.

2.1.1.5 Ice clouds

Ice clouds are generated in a similar way to water clouds. All options to set up and modify ice cloud properties start with `ic_`. The main difference between water and ice clouds is that the latter usually consist of non-spherical particles. Hence, the conversion from microphysical to optical properties is much less defined, and several parameterizations are available.

2.1.1.6 Calculation of radiances

To calculate radiances the following lines will do the job when combined with the clear sky example above

```
phi0 40.0                # Solar azimuth angle
umu -1.0 -0.5 -0.2 -0.1 # Output cosine of polar angle
phi 0.0 45. 90. 135. 180.0 225. 270.0 # Output azimuth angles
```

More examples with output are found in the ‘examples’ directory.

2.1.2 The uvspec output

The uvspec output differs for the different solvers and is currently fixed for each solver. We hope to make it user controllable in the future.

2.1.2.1 DISORT, SDISORT and SPSDISORT

For the disort, sdisort and spsdisort solvers uvspec outputs one block of data to standard output (stdout) for each wavelength. The format of the block is

```
lambda rflmdir rfldn flup uavgso uavgdn uavgup
```

if umu is not specified. If umu is specified the format of the block is

```
lambda rflmdir rfldn flup uavgso uavgdn uavgup
umu(0) u0u(umu(0))
umu(1) u0u(umu(1))
.      .
.      .
```

If both umu and phi is specified the output format of each block is

```
lambda rflmdir rfldn flup uavgso uavgdn uavgup
                                phi(0)    ...    phi(m)
umu(0) u0u(umu(0))    uu(umu(0),phi(0)) ... uu(umu(0),phi(m))
umu(1) u0u(umu(1))    uu(umu(1),phi(0)) ... uu(umu(1),phi(m))
.      .              .              .
.      .              .              .
umu(n) u0u(umu(n))    uu(umu(n),phi(0)) ... uu(umu(n),phi(m))
```

and so on for each wavelength.

2.1.2.2 TWOSTR

The format of the output line for the twostr solver is

```
lambda rflmdir rfldn flup uavg
```

for each wavelength.

2.1.2.3 POLRADTRAN

The output from the polradtran solver depends on the number of Stokes parameters, polradtran_nstokes. For each wavelength the output block is

```
lambda down_flux(1) up_flux(1) ... down_flux(is) up_flux(is)
```

if phi is not specified. Here is is the number of Stokes parameters specified by polradtran_nstokes. If phi and umu are specified the block is

```
lambda down_flux(1) up_flux(1) ... down_flux(is) up_flux(is)
                                phi(0)    ...    phi(m)
Stokes vector I
```

```

umu(0) u0u(umu(0))    uu(umu(0),phi(0)) ... uu(umu(0),phi(m))
umu(1) u0u(umu(1))    uu(umu(1),phi(0)) ... uu(umu(1),phi(m))
.      .              .              .
.      .              .              .
umu(n) u0u(umu(n))    uu(umu(n),phi(0)) ... uu(umu(n),phi(m))
Stokes vector Q
.      .              .
.      .              .

```

Note that `polradtran` outputs the total (=direct+diffuse) downward flux. Also note that `u0u` is always zero for `polradtran`.

2.1.2.4 Description of symbols

In the above output blocks the symbols used have the following meaning.

<code>cmu</code>	Computational polar angles from <code>polradtran</code> .
<code>down_flux, up_flux</code>	The total (direct+diffuse) downward (<code>down_flux</code>) and upward (<code>up_flux</code>) irradiances. Same units as extraterrestrial irradiance.
<code>lambda</code>	Wavelength (nm)
<code>rfldir</code>	Direct beam irradiance (same units as extraterrestrial irradiance, e.g. mW/(m ² nm)) if using the ‘ <code>atlas3</code> ’ spectrum in the ‘ <code>data/solar_flux</code> ’ directory.)
<code>rfldn</code>	Diffuse down irradiance, i.e. total minus direct beam (same units as <code>rfldir</code>).
<code>flup</code>	Diffuse up irradiance (same units as <code>rfldir</code>).
<code>uavg</code>	The mean intensity. Proportional to the actinic flux. (same units as <code>rfldir</code>).
<code>uavgso</code>	Direct beam contribution to the mean intensity. (same units as <code>rfldir</code>).
<code>uavgdn</code>	Diffuse downward radiation contribution to the mean intensity. (same units as <code>rfldir</code>).
<code>uavgup</code>	Diffuse upward radiation contribution to the mean intensity. (same units as <code>rfldir</code>).
<code>u0u</code>	The azimuthally averaged intensity at <code>numu</code> user specified angles <code>umu</code> . (units of e.g. mW/(m ² nm sr) if using the ‘ <code>atlas3</code> ’ spectrum in the ‘ <code>data/solar_flux</code> ’ directory.)
<code>uu</code>	The radiance (intensity) at <code>umu</code> and <code>phi</code> user specified angles. (units of e.g. mW/(m ² nm sr) if using the ‘ <code>atlas3</code> ’ spectrum in the ‘ <code>data/solar_flux</code> ’ directory.)
<code>uu_down, uu_up</code>	The downwelling and upwelling radiances (intensity) at <code>cmu</code> and <code>phi</code> angles. (units of e.g. mW/(m ² nm sr) if using the ‘ <code>atlas3</code> ’ spectrum in the ‘ <code>data/solar_flux</code> ’ directory.)

The total downward irradiance is given by

$$\text{irr_down} = \text{rfl_dir} + \text{rfl_dn}$$

The total mean intensity is given by

$$\text{uavg} = \text{uavgso} + \text{uavgdn} + \text{uavgup}$$

If `deltam` is on it does not make sense to look at the individual contributions to `uavg` since they are delta-M scaled.

2.1.3 Complete description of input parameters

The `uvspec` input file consists of single line entries, each making up a complete input to the `uvspec` program. First on the line comes the parameter name, followed by one or more parameter values. The parameter name and the parameter values are separated by white space. Filenames are entered without any surrounding single or double quotes. Comments are introduced by a `#`. Blank lines are ignored. The order of the lines is not important, with one exception: if the same input option is used more than once, the second one will usually over-write the first one.

The various input parameters are described in detail below.

`aerosol_angstrom`

Scale the aerosol optical depth using the Ångström formula. Specify the Ångström alpha and beta coefficients. The optical thickness defined here is the integral from the user-defined `altitude` to TOA (top of atmosphere).

`aerosol_default`

Set up a default aerosol according to Shettle (1989). The default properties are a rural type aerosol in the boundary layer, background aerosol above 2km, spring-summer conditions and a visibility of 50km. These settings may be modified with `aerosol_haze`, `aerosol_vulcan`, `aerosol_season`, and `aerosol_visibility`.

`aerosol_files`

A way to specify aerosol optical depth, single scattering albedo, and phase function moments for each layer. The file specified by `aerosol_files` has two columns where column 1 is the altitude in km. The altitudes must be the same as those specified in the `atmosphere_file`. The second column is a the name of a file which defines the optical properties of the layer starting at the given altitude. The files specified in the second column must have the following format:

Column 1: The wavelength in nm. These wavelengths may be different from those in `solar_file`. Optical properties are interpolated to the requested wavelengths.

Column 2: The extinction coefficient of the layer in units km⁻¹.

Column 3: The aerosol single scattering albedo of the layer.

Column 4-(nmom+4):

The moments of the aerosol phase function.

For some simple examples see the files ‘`examples/AERO_*.LAYER`’. Note that if using the `rte_solver disort2` it makes good sense to make the number of moments larger than `nstr`. For `rte_solver disort` and `rte_solver polradtran` the number of moments included in the calculations will be `nstr+1`. Higher order moments will be ignored for these solvers.

`aerosol_gg_file`

Location of aerosol asymmetry parameter file. The file must have two columns. Column 1 is the altitude in km. The altitude grid must be exactly equal to the altitude grid specified in the file `atmosphere_file`. Column 2 is the asymmetry parameter of each layer. At present, the asymmetry parameter defined here is constant with wavelength but it is planned to introduce the wavelength dependence in the near future. Comments start with `#`. Empty lines are ignored.

`aerosol_haze`

Aerosol type in the lower 2 km of the atmosphere. Integer. See E.P. Shettle, "Models of aerosols, clouds and precipitation for atmospheric propagation studies", in "Atmospheric propagation in the uv, visible, ir and mm-region and related system aspects", AGARD Conference Proceedings (454), 1989.

- 1 Rural type aerosols.
- 4 Maritime type aerosols.
- 5 Urban type aerosols.
- 6 Tropospheric type aerosols.

`aerosol_moments_file`

Location of aerosol moments file, a one-column file containing an arbitrary number of Legendre terms of the phase function. The phase function $p(\mu)$ is

$$p(\mu) = \sum_{m=0}^{\infty} (2m+1) \cdot k_m \cdot P_m(\mu)$$

where k_m is the m 'th moment and $P_m(\mu)$ is the m 'th Legendre polynomial. If not specified, a Henyey-Greenstein phase function is assumed where the asymmetry parameter g is either a default value depending on the aerosol type or may be specified using `aerosol_set_gg`.

`aerosol_refrac_file`

File containing the wavelength-dependent refractive index of the aerosol. Three columns are expected: wavelength [nm] and the real and imaginary parts of the refractive index. Together with `aerosol_sizedist_file` this forms the input to Mie calculations of the aerosol optical properties. Attention: If the aerosol properties are defined using the refractive index and the size distribution, the wavelength dependence of the optical properties is determined by Mie theory. At present there are three ways to define the absolute value of the optical thickness: (1) `visibility` defines the profile at the first *internal* wavelength; for a monochromatic calculation and in correlated-k mode, the first *internal* wavelength equals the first wavelength output by `uvspec`; for spectral calculations,

the first wavelength might be a little bit smaller than the first wavelength output by `uvspec`; (2) `aerosol_tau_file` defines the optical thickness profile at the first *internal* wavelength; or (3) absolute optical thickness and wavelength-dependence are defined by `aerosol_angstrom`. In future it is planned to also allow the specification of absolute particle densities, etc.

`aerosol_refrac_index`

Wavelength-independent refractive index of the aerosol; if wavelength-dependence is required, use `aerosol_refrac_file` instead. Together with `aerosol_sizedist_file` this forms the input to Mie calculations of the aerosol optical properties. See comment at `aerosol_refrac_file`.

`aerosol_scale_ssa`

Scale the aerosol single scattering albedo for all wavelengths and altitudes with a positive number. If the resulting scaled single scattering albedo is larger than 1 it is set to 1.

`aerosol_scale_tau`

Scale the aerosol extinction for all wavelengths and altitudes with a positive number.

`aerosol_set_gg`

Set the aerosol asymmetry parameter for all wavelengths and altitudes to a constant value between -1.0 and 1.0.

`aerosol_set_ssa`

Set the aerosol single scattering albedo for all wavelengths and altitudes to a constant value between 0.0 and 1.0.

`aerosol_set_tau`

Set the aerosol optical thickness for all wavelengths and altitudes to a constant value. The optical thickness defined here is the integral from the user-defined `altitude` to TOA (top of atmosphere).

`aerosol_set_tau550`

Set the aerosol optical thickness at 550nm. Other wavelengths are scaled accordingly. Note that this option requires for technical reasons that the wavelength interval defined by `wvn` does contain 550nm. The optical thickness defined here is the integral from the user-defined `altitude` to TOA (top of atmosphere).

`aerosol_season`

Specify season to get appropriate aerosol profile.

- 1 Spring-summer profile.
- 2 Fall-winter profile.

`aerosol_sizedist_file`

Aerosol size distribution. Two columns are expected: The radius [micron] and the particle number. Together with `aerosol_refrac_index` or `aerosol_refrac_file` this forms the input to Mie calculations of the aerosol optical properties. See comment at `aerosol_refrac_file`.

aerosol_ssa_file

Location of aerosol single scattering albedo file. The file must have two columns. Column 1 is the altitude in km. The altitude grid must be exactly equal to the altitude grid specified in the file **atmosphere_file**. Column 2 is the single scattering albedo of each layer. At present, the single scattering albedo defined here is constant with wavelength but it is planned to introduce the wavelength dependence in the near future. Comments start with **#**. Empty lines are ignored.

aerosol_tau_file

Location of aerosol optical depth file. The file must have two columns. Column 1 is the altitude in km. The altitude grid must be exactly equal to the altitude grid specified in the file **atmosphere_file**. Column 2 is the aerosol optical depth of each layer. To get wavelength dependence use the parameter **aerosol_angstrom**. Comments start with **#**. Empty lines are ignored.

aerosol_visibility

Visibility in km.

aerosol_vulcan

Aerosol situation above 2 km. Integer.

- 1 Background aerosols.
- 2 Moderate volcanic aerosols.
- 3 High volcanic aerosols.
- 4 Extreme volcanic aerosols.

albedo The Lambertian surface albedo, a number between 0.0 and 1.0, constant for all wavelengths. For wavelength dependent surface albedo use **albedo_file**.

albedo_file

Location of surface albedo file for wavelength dependent surface albedo. The file must have two columns. Column 1 is the wavelength, in nm, and column 2 the corresponding Lambertian surface albedo. The wavelength grid may be freely set. The albedo will be interpolated to the wavelength grid used for the radiation calculation. Comments start with **#**. Empty lines are ignored.

angstrom Still supported but obsolete. Replaced by **aerosol_angstrom**.

altitude Set the bottom level in the model atmosphere provided in **atmosphere_file** to be at altitude (km).

```
altitude 0.73    # Altitude of IFU, Garmisch-Partenkirchen
                 # Be aware, for this to work the atmosphere
                 # file must start at 0 km.
```

The profiles of pressure, temperature, molecular absorbers, ice and water clouds are cut at the specified altitude. The aerosol profile is not affected by **altitude** but starts right from the model surface. This is a convenient way for the user to calculate the radiation at other altitudes than sealevel. Note that **altitude** is very different from **zout** where the radiation is calculated at an altitude of **zout**

over the surface. E.g. to calculate the radiation field 1 km above the surface at 0.73 above sealevel, one would specify 'altitude 0.73' and 'zout 1.0'.

A second optional argument may be given to **altitude** as e.g.

```
altitude 0.73 0.5
```

Here the bottom level will be at 0.73 km and the vertical resolution of the model atmosphere will be redistributed to have a spacing between levels specified by the second number, here 0.5 km. Beaware that specifying a fine vertical spacing will produce many layers thus increasing the computing time. Also the radiative transfer equation solvers implemented in Fortran 77 might need to have some array sizes increased (see 'src_f/DISORT.MXD').

atmosphere_file

Location of the atmospheric data file. Must have at least three columns containing the altitude pressure and temperature. Other traces gases are set by **dens_file**. Alternatively the first five columns must be specified and optionally columns 6 - 9. may be omitted. Missing profiles are filled with zero's. The up to nine columns of the file are interpreted as follows (first three columns must be specified):

1. **z(km)** Altitude above sea level in km
2. **p(mb)** Pressure in mbar
3. **T(K)** Temperature in K
4. **air_density(cm-3)**
 Air density in cm-3
5. **o3_density(cm-3)**
 Ozone density in cm-3
6. **o2(cm-3)**
 Oxygen density in cm-3
7. **h2o(cm-3)**
 Water vapour density in cm-3
8. **co2(cm-3)**
 CO2 density in cm-3
9. **no2(cm-3)**
 NO2 density in cm-3

The atmosphere is specified top-down, that is, the top level is the first line in the file, the bottom (surface) level the last line. All properties refer to model *level* *z*, not to model *layer*. It is important that the correct units are used, otherwise unpredictable results are guaranteed. Comments start with #. Empty lines are ignored. libRadtran provides the six standard atmospheres by Anderson et al. (1986):

```
afglt       Tropical
afglms      Midlatitude Summer
```


<code>afglmw</code>	Midlatitude Winter
<code>afglss</code>	Subarctic Summer
<code>afglsw</code>	Subarctic Winter
<code>afglus</code>	U.S. Standard

brightness

Convert radiances / irradiances to brightness temperatures.

ch4_mixing_ratio

The mixing ratio of CH₄ in ppm (default: 1.6 ppm).

co2_mixing_ratio

The mixing ratio of CO₂ in ppm; scale the profile so that the mixing ratio at the user-define `altitude` assumes the specified value.

correlated_k

To calculate integrated shortwave or longwave irradiance, or to simulate satellite instrument channels, choose between the following correlated-k schemes:

<code>Kato</code>	Kato et al. (1999), shortwave
<code>Kato2</code>	Kato et al. (1999), shortwave; optimized version (Seiji Kato, personal communication, 2002)
<code>Fu</code>	Fu and Liou (1992/93), shortwave and longwave
<code>AVHRR_KRATZ</code>	Kratz (1995), AVHRR instrument channels
<code>SBDART</code>	Gas absorption parameterization from SBDART (Ricchiazzi et al., 1998); please see the section on "Spectral resolution".

If correlated_k KATO/KATO2/FU/AVHRR_KRATZ is specified, the extraterrestrial flux is taken from internally defined files specific for each parameterization, not from `solar_file`. The output is the integrated irradiance for each band. To get e.g. integrated shortwave irradiance, simply add all bands of the Kato et al. (1999) or the Fu and Liou (1992/93) parameterization. The five AVHRR channels are weighted sums of the libRadtran output. Examples how to integrate the output in the AVHRR_KRATZ case are included in the `uvspec` self-test which is initiated with `make check`.

cox_and_munk_pcl

Pigment concentration for Cox and Munk ocean surface BRDF (in mg/m³); at present only available with `rte_solver` DISORT2.

cox_and_munk_sal

Salinity for Cox and Munk ocean surface BRDF (in ppt); at present only available with `rte_solver` DISORT2.

cox_and_munk_u10

Wind speed for Cox and Munk ocean surface BRDF (in m/s); at present only available with `rte_solver` DISORT2.

crs_file May be used to specify cross sections of ozone (O3), nitrogendioxide (NO2), bromine oxide (BRO), OCLO, or HCHO to be used instead of one of those coming with libRadtran. No temperature dependence may be specified. Use as follows:

```
crs_file NO2 ../examples/no2_crs.dat
```

The NO2 or O3, BRO or OCLO or HCHO must be specified to identify the specie for which the cross section applies. The cross section file has two columns:

1. wavelength (nm)
2. cross section (cm²)

data_files_path

The path to the directory where all uvspec internal data files live, e.g. the files that are in the subdirectories of the ‘data’ directory that comes with the uvspec distribution.

day_of_year

Integer, to correct the calculated radiation quantities for the Sun-Earth distance for the specified Julian day (1-365). If not given, the Earth-Sun distance is 1 AU (i.e. equinox distances), that is, no correction is applied to the extraterrestrial irradiance **solar_file**.

deltam

Turn delta-M scaling on/off. Set to either **on** or **off**. Note that for the **rte_solver disort2** delta-M scaling is hardcoded to be always on.

dens_column

Set the total column of a density profile. The column is integrated between the user-defined altitude and TOA (top of atmosphere). The syntax is

```
dens_column specie column [unit]
```

where specie is one of O3, NO2, H2O, BRO, OCLO, HCHO and CO2, see also **dens_file**. Column is the total column value of the trace gas and the column is in unit which is optional. The default units are O3 (DU), NO2 (CM_2), H2O (CM_2), BRO (CM_2), OCLO (CM_2), HCHO (CM_2) and CO (CM_2). Here DU is Dobson units and CM_2 = cm⁻². In addition DU may be used for NO2 as well, e.g.

```
dens_column NO2 1.2 DU
```

dens_file

Specify density profiles (or matrix, see below) of various trace gases to be included in the radiative transfer calculation. At the moment ozone (O3), nitrogendioxide (NO2), water vapor (H2O), bromine oxide (BRO), OCLO, HCHO, and carbondioxide (CO2) are included. The various density profiles are identified by their abbreviations given in the parenthesis above.

```
dens_file O3 ../examples/afglus_o3.dat
```

The density file has two columns:

1. **z(km)** Altitude above sea level in km, must be the same as the altitudes in **atmosphere_file**.

2. density of trace gas (cm⁻³)

The density of the trace gas

To scale the profile to a total column value use `dens_column`.

For airmass factor calculations it is for some species necessary to account for the variation of the profile with `sza`. This may be accomplished by using `dens_file` and feeding it a file containing the following:

1. row 1, column 1: a zero, that is 0.0
2. column 1 below row 1: Altitude above sea level in km, must be the same as
the altitudes in `atmosphere_file`.
3. row one after column 1: The solar zenith angle in degrees
4. The rest of the matrix: the density of trace gases as a function of solar
zenith angle and altitude.

The matrix may only be specified for one specie. It may however be combined with profiles of other species. For examples see the `examples` directory. A density matrix may only be used in connection with the `rte_solver sdisort`.

`f11_mixing_ratio`

The mixing ratio of F11 in ppm (default: 0.000268 ppm).

`f12_mixing_ratio`

The mixing ratio of F12 in ppm (default: 0.000503 ppm).

`f22_mixing_ratio`

The mixing ratio of F22 in ppm (default: 0.000105 ppm).

`filter_function_file`

If specified, the calculated spectrum is multiplied with a filter function defined in '`filter_function_file`'. The file must contain two columns. Column 1 is the wavelength, in nm. Column 2 is the corresponding filter function value. Comments start with `#`. Empty lines are ignored. In combination with `output sum` or `output integrate` this option is useful e.g. to calculate weighted irradiances or actinic fluxes or to simulate broadband or satellite observations.

`fisot` Specifies that isotropic illumination is used at top-boundary instead of beam source. Useful for those who want to calculate the reflectance for a homogeneous or inhomogeneous atmosphere. The intensity is still set by `solar_file`.

`flexstor` Output is in flexstor format. May not be combined with `header`. Also, does not currently work when `umu` and/or `phi` is specified.

`h2o_mixing_ratio`

The mixing ratio of H₂O in ppm. Scale the profile so that the mixing ratio at the user-defined `altitude` assumes the specified value.

- h2o_precip** Precipitable water in kg / m² (which is approximately 1mm). The water vapor profile is scaled accordingly. The precipitable water is integrated from the user-defined **altitude** to TOA (top of atmosphere).
- header** Output information on some of the input parameters. May not be combined with **flexstor**.
- n2o_mixing_ratio** The mixing ratio of N₂O in ppm (default: 0.28 ppm).
- ic_file** Location of file defining ice cloud properties. The file must contain three columns. Column 1 is the altitude in km, column 2 the ice water content in grams per cubic meter, and column 3 the effective particle radius in micron. The ice water content and effective particle radius must be specified at the same altitude grid as in **atmosphere_file**. Note that the definition of cloud altitudes in **ic_file** refers to sea level, not to altitude above ground. E.g., when altitude is set to 1.63km, and the first cloud level is defined at 3km, the cloud would start at 1.37km above ground. The optical depth of a layer is calculated using information from the upper and lower levels defining the layer. Comments start with #. Empty lines are ignored. An example of an ice cloud is given in 'examples/IC.DAT'. Per default, the cloud properties are interpreted as properties at a given altitude *level*. If **ic_layer** is defined, they are interpreted as *layer* properties.
- ic_files** A way to specify ice cloud optical depth, single scattering albedo, and phase function moments for each layer. The file specified by **ic_files** has two columns where column 1 is the altitude in km. The altitudes must be the same as those specified in the file **atmosphere_file**. The second column is the name of a file which defines the optical properties of the level starting at the given altitude. The files specified in the second column must have the following format:
- Column 1:** The wavelength in nm. These wavelengths may be different from those in **solar_file**. Optical properties are interpolated to the requested wavelengths.
- Column 2:** The extinction coefficient of the layer in units km⁻¹.
- Column 3:** The single scattering albedo of the layer.
- Column 4-(nmom+4):**
The moments of the scattering phase function.
- Note that if using the **rte_solver disort2** it makes good sense to make the number of moments larger than **nstr**. For **rte_solver disort** and **rte_solver polradtran** the number of moments included in the calculations will be **nstr+1**. Higher order moments will be ignored for these solvers.
- ic_habit** Ice crystal habit for the Yang et al. (2000), Key et al. (2002) parameterization, see also **ic_properties yang**. May be one of solid-column, hollow-column, rough-aggregate, rosette-4, rosette-6, or plate.

ic_ipa_files

A two-column file, defining ice cloud property files (see **ic_file**) in the first column and the corresponding weights in the second column. The radiative transfer calculation is performed independently for each cloud column and the result is the weighted average of all independent columns. If **ic_ipa_files** and **wc_ipa_files** are both defined, both must have the same columns in the same order, otherwise **uvspec** will complain.

ic_layer Specify ice cloud properties for model *layers* instead of *levels* (see also **ic_file**).

ic_properties

Defines how ice water content and effective particle radius are translated to optical properties. Possible choices are

- fu** Parameterization by Fu (1996), Fu et al. (1998), see **ic_file**; this is the default setting. Note that this is a parameterization which has been created to calculate fluxes but not radiances. Note also that the optical properties in the solar range provided by Fu (1996) are delta-scaled properties (that is, the forward peak of the phase function is truncated and optical thickness, asymmetry parameter, and single scattering albedo are reduced accordingly). For wavelengths up to 4 micron Fu (1996) is used while for wavelengths larger than 4 micron Fu et al. (1998) is chosen. Please note that Fu (1996) is based on ray-tracing calculations while Fu et al. (1998) is a mixture of ray-tracing and Mie calculations (which is required for the infrared wavelengths where the geometrical assumption does not hold). Hence, both parameterizations are not fully consistent. Rather, differences of some % are to be expected in the wavelength region where both parameterizations overlap.
- yang** Parameterization by Yang et al. (2000), Key et al. (2002). This parameterization can also be used to calculate radiances because it uses a double-Henyey-Greenstein phase function which better represents both forward and backward peaks. This parameterization covers the region from 0.2 to 4.8 micron. It has been complemented by the parameterization by Yang et al. (2001) which covers the wavelength region 8 to 13 micron.
- mie** Use pre-calculated Mie tables; makes only sense with **correlated_k**; the tables are expected in '**data_files_path**'/correlated_k/./ You may get these tables from <http://www.libradtran.org>. Note that a Mie calculation assumes spherical ice particles, the scattering function of which differs systematically from non-spherical particles.
- filename** Read optical properties from specified filename; file format is as produced by Frank Evans' **cloudprp**; for each of the internal (computational) wavelengths, a separate file is expected (this option is subject to change).

Please note also that, in contrast to spherical particles, there is no unique definition of effective size for non-spherical particles. In particular, the above parameterizations use different definitions which, however, differ only by a constant factor. Yang et al. (2000), Key et al. (2002) use the general definition

$$r_{\text{eff}} = \frac{3 \int V(h)n(h)dh}{4 \int A(h)n(h)dh}$$

where h is the maximum dimension of an ice crystal, $n(h)$ is the number of particles with maximum dimension h in the size distribution, and V and A are the volume and mean projected area of the particles, respectively. The volume and area are based on the spherical diameter with equivalent volume and the spherical diameter with equivalent projected area as defined by Yang et al. (2000). On the other hand, Fu et al. [1996,1998] use hexagonal columns and use the following definition

$$r_{\text{eff}} = \frac{\int D^2 L n(L) dL}{2 \int (DL + \frac{\sqrt{3}}{4} D^2) n(L) dL}$$

where D is the width of the ice crystal (that is, the maximum diameter of the hexagonal area) and L is the length. The integrand in the numerator is proportional to the volume while that in the denominator is proportional to the projected area. Evaluating these formulas one finds that, for the same hexagonal particle, the effective radius would be $3\sqrt{3}/4 = 1.299$ times larger following the Yang et al. (2000), Key et al. (2002) definition than the Fu (1996,1998) definition. As an example, an effective radius of $20\mu m$ with "ic_properties fu" and $1.299 \cdot 20\mu m = 26\mu m$ with "ic_properties yang" would give comparable results for hexagonal columns.

ic_scale_gg

Scale the ice cloud asymmetry factor for all wavelengths and altitudes with a float between 0.0 and 1.0.

ic_scale_ssa

Scale the ice cloud single scattering albedo for all wavelengths and altitudes with a float between 0.0 and 1.0.

ic_set_gg

Set the ice cloud asymmetry factor for all wavelengths and altitudes to a float between -1.0 and 1.0.

ic_set_ssa

Set the ice cloud single scattering albedo for all wavelengths and altitudes to a value between 0.0 and 1.0.

ic_set_tau

Set the total ice cloud optical depth to a constant value for all wavelengths. The optical thickness defined here is the integral from the surface at the user-defined **altitude** to TOA (top of atmosphere). This option is useful only for monochromatic calculations or in wavelength regions where the optical properties of ice clouds can be considered constant, e.g. the ultraviolet region.

ic_set_tau550

Set the ice cloud optical thickness at 550nm. Other wavelengths are scaled accordingly. The optical thickness defined here is the integral from the surface at the user-defined **altitude** to TOA (top of atmosphere). Note that this option requires for technical reasons that the wavelength interval defined by **wvn** does contain 550nm.

include Include a file into the uvspec input. Works exactly like the C **#include** or the Fortran **INCLUDE** statements.

mc_absorption

Calculate MYSTIC absorption of each grid box; may need a considerable amount of memory, depending on the 3D cloud grid. Only meaningful with **rte_solver montecarlo**.

mc_actinic

Calculate MYSTIC actinic flux for each grid box by dividing the absorbed energy by the absorption coefficient; this method is much better than the traditional photon counting which usually comprises spikes (because in the latter method each photon is weighted with $1/\cos(\theta)$ which may be a very large number); may need a considerable amount of memory, depending on the 3D cloud grid. Only meaningful with **rte_solver montecarlo**.

mc_basename

Filename for MYSTIC 3D output (default: **mc**).

mc_albedo_file

Define a MYSTIC 2D albedo input file (default: **albedo2D.dat**).

mc_cloud_grid

3D cloud grid size (N_x, N_y, N_z) for MYSTIC. Weird, but required to keep MYSTIC a function that can be called from a Fortran program. Only meaningful with **rte_solver montecarlo**.

mc_elevation_file

Define a MYSTIC 2D elevation input file (default: **elevation2D.dat**).

mc_emission

Calculate MYSTIC emission of each grid box; may need a considerable amount of memory, depending on the 3D cloud grid. Note that emission is calculated directly without tracing any photons. Increasing **mc_photons** will therefore not change the result. Only meaningful with **rte_solver montecarlo** and **source thermal**.

mc_escape

Calculate MYSTIC radiances via escape probabilities; slows down the tracing but usually requires much less photons. Only meaningful with **rte_solver montecarlo**.

<code>mc_heating</code>	Calculate MYSTIC heating rates for each grid box; may need a considerable amount of memory, depending on the 3D cloud grid. Note that emission is calculated directly without tracing any photons. Increasing <code>mc_photons</code> will therefore not change the result. Only meaningful with <code>rte_solver montecarlo</code> and <code>source thermal</code> .
<code>mc_ipa</code>	Run MYSTIC in independent pixel mode. Only meaningful with <code>rte_solver montecarlo</code> .
<code>mc_momcld_file</code>	Cloud phase function moments file for MYSTIC; can be any filename or 'water' or 'ice'. In the latter case, pre-calculated tables for water and ice are used. For each of the internal (computational) wavelengths, a separate file is expected. Only meaningful with <code>rte_solver montecarlo</code> .
<code>mc_photons</code>	Total number of photons to be traced by the Monte Carlo solver, MYSTIC. Only meaningful with <code>rte_solver montecarlo</code> .
<code>mc_pcld_file</code>	Cloud phase function file for MYSTIC. Only meaningful with <code>rte_solver montecarlo</code> .
<code>mc_photons</code>	Total number of photons to be traced by the Monte Carlo solver, MYSTIC. Only meaningful with <code>rte_solver montecarlo</code> .
<code>mc_photons_file</code>	Distribution of photons over wavelength bands; to be used with <code>correlated_k</code> ; for an example see 'data/correlated_k/kato/photons.dat'. No error checking yet! Do only use if you are absolutely sure what you are doing. Only meaningful with <code>rte_solver montecarlo</code> .
<code>mc_rpv_file</code>	Define a MYSTIC 2D RPV BRDF input file.
<code>mc_rpv_type</code>	File containing wavelength-dependent RPVs to be used in combination with <code>mc_rpv_file</code> .
<code>mc_sample_grid</code>	Sampling grid size (Nx,Ny) for MYSTIC. Only meaningful with <code>rte_solver montecarlo</code> .
<code>mc_truncate</code>	Truncate phase function at the specified polar angle μ . USE ONLY IF YOU REALLY KNOW WHAT YOU ARE DOING!
<code>mc_wcloud_file</code>	Define a MYSTIC 3D cloud input file (default: wcloud3D.dat).

molecular_tau_file

Location of molecular absorption optical depth file. Usually, molecular absorption is calculated from trace gas concentrations provided in **atmosphere_file** (scaled with **ozone_column**, etc. Use this option only if you really want to specify the optical depth directly (e.g. for a model intercomparison) or for a line-by-line calculation. **absorption_tau_file** can be either of the following two formats:

Monochromatic:

Column 1 is the altitude in km where the altitude grid must be exactly equal to the altitude grid specified in **atmosphere_file**.
Column 2 is the absorption optical depth of each layer.

Spectral: The first line contains the level altitudes in decreasing order; the following lines contain the wavelength [nm] in the first column and then the absorption optical depths of each layer.

Comments start with **#**. Empty lines are ignored.

no_absorption

Switch absorption off.

no_molecular_absorption

Switch all (molecular, aerosol, cloud, and ice cloud) absorption off. Note that this option does not affect the single scattering albedo; e.g. with an aerosol optical depth of 1 and a single scattering albedo of 0.7, the scattering optical depth will still be 0.7, even with **no_absorption**.

no_rayleigh

Switch Rayleigh scattering off.

no_scattering

Switch scattering off.

no2_column_du

Obsolete, use **dens_column** instead. Set the NO2 column to a given value. The column is integrated between the user-defined **altitude** and TOA (top of atmosphere). The value must be in Dobson units. If value is negative or **no2_column_du** is not specified, the NO2 column is not scaled.

no2_column_moleccm-2

Obsolete, use **dens_column** instead. Set the NO2 column to a given value. The column is integrated between the user-defined **altitude** and TOA (top of atmosphere). The value must be in molecules / cm². If value is negative or **no2_column_moleccm-2** is not specified, the NO2 column is not scaled.

nscat

The order of scattering for the **sos** radiative transfer equation solver. Default is 20. May also be used with the **sdisort** solver. If set to 1 **sdisort** will run in single scattering mode while if set in to 2, **sdisort** runs in full multiple scattering mode.

nrefrac	For the <code>rte_solver sdisort</code> refraction may be included by specifying <code>nrefrac</code> . If refraction is included also set parameter <code>refraction_file</code> . 0 No refraction, default. 1 Refraction included using fast, but harsh method. 2 Refraction included using slow, but accurate method.
nstr	Number of streams used to solve the radiative transfer equation. Default is 6 for fluxes and 16 for radiances.
o3_crs	Choose between the following ozone cross sections. Bass_and_Paur Bass and Paur ozone cross section. Molina Molina and Molina (1986) ozone cross section. Daumont Ozone cross section by Daumont et al. (1992), Malicet et al. (1995). Molina and Molina is default.
ozone_column	Obsolete, use <code>dens_column</code> instead. Set the ozone column to a given value. The column is integrated between the user-defined <code>altitude</code> and TOA (top of atmosphere). The value must be in Dobson units. If value is negative or <code>ozone_column</code> is not specified, the ozone column is not scaled.
output	Output processing. Choose between the following options: sum Sum output over wavelength. Useful in combination with the <code>correlated_k</code> option. integrate Integrate output over wavelength. Useful for spectral calculations. none No processing - output spectral information (default).
phi	Azimuth output angles (in degrees) in increasing order. The radiance is output at <code>phi</code> and <code>umu</code> .
phi0	Azimuth angles of the sun (0 to 360 degrees). If <code>phi0</code> varies as a function of wavelength use <code>sza_file</code> .
polradtran_aziorder	Order of Fourier azimuth series: 0 is azimuthially symmetric case. Default 0.
polradtran_max_delta_tau	Initial layer thickness for doubling; governs accuracy, 10E-5 should be adequate. Do not go beyond half the real precision, i.e. 10e-8 for REAL*8. Default 1.e-05.
polradtran_nstokes	Number of Stokes parameters 1 for I (no polarization)

- 2 for I,Q
- 3 for I,Q,U
- 4 for I,Q,U,V

Default is 1.

`polradtran_quad_type`

Type of quadrature used:

- G gaussian
- D double gaussian,
- L Lobatto
- E extra-angle(s), this must be used if `polradtran` is used in combination with `umu`. Will internally use Gaussian scheme (G). See also `radtran` documentation.

Default G.

`polradtran_src_code`

Radiation sources included:

- 0 none
- 1 solar
- 2 thermal
- 3 both

Default 1.

pressure The surface pressure (at the user-defined `altitude` in hPa. The pressure profile and density profiles are scaled accordingly.

prndis Specify one or more integers between 1 and 7. Print various disort input and output in disort's own format. See '`disort.doc`' for more information. **Warning:** Produces a lot of output.

quiet If specified, a number of informative messages about what options are used and progress during computations are turned off. Otherwise these messages are output to stderr. See also `verbose`.

`rayleigh_tau_file`

Location of Rayleigh scattering optical depth file. Usually, the Rayleigh scattering cross section is calculated from the air pressure provided in `atmosphere_file` (scaled with `pressure`). Use this option only if you really want to specify the optical depth directly (e.g. for a model intercomparison). Note also that no wavelength dependence can be specified; hence this option is only useful for monochromatic calculations. The `rayleigh_tau_file` must have two columns. Column 1 is the altitude in km where the altitude grid must be exactly equal to the altitude grid specified in `atmosphere_file`. Column 2 is the Rayleigh scattering optical depth of each layer. Comments start with #. Empty lines are ignored.

reflectivity

Calculate transmission / reflectivity instead of absolute quantities. For irradiances / actinic fluxes the transmission T is defined as

$$T = \frac{E}{E_0 \cos \theta}$$

where E is the irradiance / actinic flux, E_0 is the extraterrestrial flux, and θ is the solar zenith angle. The reflectivity R is defined as

$$R = \frac{\pi \cdot L}{E_0 \cos \theta}$$

where L is the radiance, E_0 is the extraterrestrial flux, and θ is the solar zenith angle. Obviously, reflectivities do not depend on Sun-Earth distance. Please note the difference to **transmittance**.

refractive_index_file

Location of refractive index file. The file must have two columns. Column 1 is the altitude in km. The altitude grid must be exactly equal to the altitude grid specified in the file **atmosphere_file**. Column 2 is the refractive index of each layer. Only has effect if used with the **rte_solver sdisort** and **nrefrac** different from zero. Comments start with **#**. Empty lines are ignored.

reverse

Option for the strong and bold. Reverses the atmospheric input to the radiative transfer solvers. That is, the atmosphere is turned on the head. Yes, that is actually useful for some purposes. If you think you need this contact the author. Otherwise, do not use.

rh_file

File that defines a profile of relative humidity. If specified, the water vapour profile in **atmosphere_file** is over-written. If -1 is specified at a level, the value from **atmosphere_file** is used.

rpv_file

4 column file, containing the Rahman, Pinty, and Verstraete (RPV) BDRF parameterization, Rahman et al. (1993). Bidirectional reflectance distribution functions for a variety of surfaces are given in the paper. This option is only supported with DISORT 2.0 and MYSTIC. The columns of the input file are wavelength [nm], rho0, k, and theta. The parameters are interpolated linearly to the internal wavelength grid. To make sure that the results are reasonable, specify the RPV data on a wavelength grid similar or equal to that used internally for the radiative transfer calculation!

rpv_k

Constant RPV rho0, see **rpv_file**. **rpv_k** overwrites the wavelength-dependent value defined in **rpv_file**.

rpv_rho0

Constant RPV rho0, see **rpv_file**. **rpv_rho0** overwrites the wavelength-dependent value defined in **rpv_file**.

rpv_theta

Constant RPV theta, see **rpv_file**. **rpv_theta** overwrites the wavelength-dependent value defined in **rpv_file**.

rte_solver

Set the radiative transfer equation solver to be used. Options are

disort The standard plane-parallel disort algorithm by Stamnes et al. (1988), version 1.3. For documentation see ‘src_f/DISORT.doc’ as well as the papers and the DISORT report at <ftp://climate.gsfc.nasa.gov/pub/wiscombe/Multiple.Scatt/>. To optimize for computational time and memory, please adjust the parameters in src_f/DISORT.MXD for your application and re-compile.

disort2 Version 2 of disort. For documentation see ‘src_f/DISORT2.doc’ as well as the papers and the DISORT report at <ftp://climate.gsfc.nasa.gov/pub/wiscombe/Multiple.Scatt/>. disort2 has several improvements compared to its ‘ancestor’ disort 1.3. Hence we recommend to use disort2 rather than the older version. To optimize for computational time and memory, please adjust the parameters in src_f/DISORT.MXD for your application and re-compile.

sdisort Pseudospherical disort as described by Dahlback and Stamnes (1991). Double precision version. To optimize for computational time and memory, please adjust the parameters in src_f/DISORT.MXD for your application and re-compile.

spsdisort

Pseudospherical disort as described by Dahlback and Stamnes (1991). Single precision version, not recommended for large solar zenith angles. **Warning:** it is not recommended to use sdisort or spsdisort for cloudy conditions. With clouds included it may be numerically unstable and produce wrong results. To optimize for computational time and memory, please adjust the parameters in src_f/DISORT.MXD for your application and re-compile.

polradtran

The plane-parallel radiative transfer solver of Evans and Stephens (1991). Includes polarization. Note that polarization effects of aerosols and clouds are currently not included.

twostr The two-stream radiative transfer solver described by Kylling et al. (1995).

sos A scalar pseudospherical successive orders of scattering code. Works for solar zenith angles smaller than 90 degrees. Can calculate azimuthally averaged radiances. Set **nscat** to specify the order of scattering.

montecarlo

The MYSTIC code, see <http://www.bmayer.de/mystic.html>. Note that MYSTIC is not part of the libRadtran distribution at present. However, it has been given to some users on a collaborative basis.

slit_function_file

If specified, the calculated spectrum is convolved with the function found in the 'slit_function_file'. The file must contain two columns. Column 1 is the wavelength, in nm, and relative to the center wavelength. Column 2 is the corresponding slit function value. It must be unity at the maximum. The wavelength steps in the slit function file must be equidistant. Comments start with #. Empty lines are ignored.

solar_file

Location of file holding the extraterrestrial spectrum. The file must contain two columns. Column 1 is the wavelength, in nm, and column 2 the corresponding extraterrestrial flux. The user may freely use any units he/she wants on the extraterrestrial flux. The wavelength grid specified, will be the wavelength resolution at which results are returned. However, the wavelength range is determined by **wvn**. Comments start with #. Empty lines are ignored. Note that **solar_file** is ignored if **correlated_k** is specified.

source

Solar or thermal source. Set to either **solar** or **thermal**.

spline

Spline interpolate to wavelengths **lambda_0** to **lambda_1** in steps of **lambda_step**, in nm. Specified as e.g.

```
spline 290. 365. 0.5
```

Here, the calculated spectrum is interpolated to wavelengths 290., 290.5, 291., ... , 364.5, 365. For interpolation to arbitrary wavelengths use **spline_file**. The specified wavelength interval must be within the one specified by **wvn**.

spline_file

Spline interpolate to arbitrary wavelengths, in nm, given as a single column in file 'spline_file'. The specified wavelengths must be within the range specified by **wvn**. Comments start with #. Empty lines are ignored.

surface_temperature

Surface temperature, used for thermal infrared calculations. If not specified, the temperature of the lowest atmospheric level is used as surface temperature.

sza

The solar zenith angle. If the solar zenith angle varies with wavelength, use **sza_file**.

sza_file

Location of solar zenith angle file for wavelength dependent solar zenith angle. The file must have two or three columns. Column 1 is the wavelength, in nm, and column 2 the corresponding solar zenith angle. Optionally the third column may contain the corresponding solar azimuth angle. The solar azimuth angle is only needed when calculating radiances. The wavelength grid may be freely set. The solar zenith and azimuth angle will be interpolated to the wavelength grid used for the radiation calculation. Comments start with #. Empty lines are ignored.

thermal_bands_file

File with the center wavelengths and the wavelength band intervals to be used for calculations in the thermal range. The following three columns are expected:

center (or reference) wavelength, lower wavelength limit, upper wavelength limit [nm]. `thermal_bands_file` defines the wavelength grid for the radiative transfer calculation. The RTE solver is called for each of the wavelengths in the first column. The atmospheric (scattering, absorption, etc) properties are also evaluated at these wavelengths. For thermal radiation calculations, the Planck function is integrated over the wavelength bands defined in the second and third columns. The result will therefore be a band-integrated irradiance which does only make sense when the `solar_file` grid equals the `thermal_bands_file` grid.

`thermal_bandwidth`

Specify a constant bandwidth in cm-1 for thermal calculations. The default is 1 cm-1. This option is ignored if the bands are defined explicitly, like with `thermal_bands_file` or `correlated_k` KATO/FU/AVHRR_KRATZ.

`transmittance`

Calculate transmittance / reflectance instead of absolute quantities. That is, set the extraterrestrial irradiance to 1 and do not correct for Sun-Earth distance:

$$T = \frac{E}{E_0}$$

where E is the irradiance / actinic flux / radiance and E_0 is the extraterrestrial flux. Please note the difference to `reflectivity`.

`transmittance_wl_file`

Location of single column file that sets the wavelength grid used for the internal transmittance calculations. The wavelengths must be in nm. Do not use this option unless you know what you are doing. Comments start with #. Empty lines are ignored.

`umu` Cosine of output polar angles in increasing order, starting with negative (downward) values (if any) and on through positive (upward) values. Must not have any zero values. The azimuthally averaged intensity `u0u` is output at `umu`.

`verbose` If specified abundances of informative messages are output to stderr. See also `quiet`.

`wc_cloudcover`

Set the fraction of the horizontal sky area which is covered by clouds. When a cloud cover is specified, the result will be calculated by the independent pixel approximation (IPA), that is, as weighted average of cloudless sky and overcast sky, where the cloud properties are taken from `wc_file`, etc. When `wc_cloudcover` is specified, `wc_ipa` is switched on automatically (see also `wc_ipa`).

`wc_file` Location of file defining water cloud properties. The file must contain three columns. Column 1 is the altitude in km, column 2 the liquid water content in grams per cubic meter, and column 3 the effective droplet radius in micron. The liquid water content and effective droplet radius must be specified at the

same altitude grid as in `atmosphere_file`. Note that the definition of cloud altitudes in `wc_file` refers to sea level, not to altitude above ground. E.g., when altitude is set to 1.63km, and the first cloud level is defined at 3km, the cloud would start at 1.37km above ground. The optical depth of a layer is calculated using information from the upper and lower levels defining the layer. Comments start with `#`. Empty lines are ignored. An example of a cloud is given in `examples/WC.DAT`. Per default, the cloud properties are interpreted as properties at a given altitude *level*. If `wc_layer` is defined, they are interpreted as *layer* properties.

wc_files A way to specify cloud extinction coefficient, single scattering albedo, and scattering phase function for each layer. The file specified by `wc_files` has two columns where column 1 is the altitude in km. The altitudes must be the same as those specified in the file `atmosphere_file`. The second column is the name of a file which defines the optical properties of the layer starting at the given altitude. The files specified in the second column must have the following format:

Column 1: The wavelength in nm. These wavelengths may be different from those in `solar_file`. Optical properties are interpolated to the requested wavelengths.

Column 2: The extinction coefficient of the layer in units km⁻¹.

Column 3: The single scattering albedo of the layer.

Column 4-(nmom+4):

The moments of the scattering phase function.

Note that if using the `rte_solver disort2` it makes good sense to make the number of moments larger than `nstr`. For `rte_solver disort` and `rte_solver polradtran` the number of moments included in the calculations will be `nstr+1`. Higher order moments will be ignored for these solvers.

wc_ipa Independent pixel approximation (IPA) of a 3D cloud field. Cloud information is read from a file named `wcloud3D.dat`. The three-dimensional cloud field is split into individual columns which then are considered one-dimensional and treated separately with the specified RTE solver. The output quantities are averages over all columns. If `wc_cloudcover` is also specified, the cloud is not read from `wcloud3D.dat` but from `wc_file` (see also `wc_cloudcover`). The format of `wcloud3D.dat` is as follows:

```
80 80 94 1 1
0.025 0.025 0.0 0.5 0.75 1.0 1.4 1.5 ...

# layer 3, 0.75 - 1.0 km
1 80 4 1.25 0.86 0.99999
2 80 4 1.25 0.86 0.99999
3 80 4 1.25 0.86 0.99999
4 80 4 1.25 0.86 0.99999
...
```


1. 1st line: Nx Ny Nz gspec wspec
 Nx number of boxes in x direction
 Ny number of boxes in y direction
 Nz number of boxes in z direction
 gspec if > 0, asymmetry parameter is specified for each cell
 wspec if > 0, single scattering albedo is specified for each cell
 Grid boxes are always counted from 1 .. Nx, 1 .. Ny, 1 .. Nz.
2. 2nd line: dx dy z[1] z[2] z[3] ... z[Nz] z[Nz+1]
 dx size of boxes in x direction [km]
 dy size of boxes in y direction [km]
 z[1] .. z[Nz+1]
 z-grid [km]. Note that this grid must be exactly the same as
 the 1D grid in `atmosphere_file`
3. 3rd line, etc. (cloud definition lines)
 ix iy iz ext g w
 ix index of grid box in x direction (1 .. Nx)
 iy index of grid box in y direction (1 .. Ny)
 iz index of grid box in z direction (1 .. Nz)
 ext extinction coefficient [1/km]
 g asymmetry parameter (only used if gspec > 0)
 w single scattering albedo (only used if wspec > 0)

Rules to set up the atmosphere:

1. An 1D atmosphere is built by `uvspec`, as usual. This atmosphere may contain 1D clouds, defined in `wc_filename`.
2. Each layer for which a cloud definition line is found in `wcloud3D.dat` will be overwritten by 3D clouds. If `gspec` or `wspec` are not set, asymmetry parameter or single scattering albedo will be taken from the 1D definition.
3. If only properties for some boxes of a layer are defined, the extinction coefficients of all other boxes of this layer will be set to 0.

`wc_ipa_files`

A two-column file, defining water cloud property files (see `wc_file`) in the first column and the corresponding weights in the second column. The radiative transfer calculation is performed independently for each cloud column and the result is the weighted average of all independent columns. If `ic_ipa_files` and `wc_ipa_files` are both defined, both must have the same columns in the same order, otherwise `uvspec` will complain.

`wc_layer` Specify cloud properties for layers instead of levels (see also `wc_file`). If `wc_layer` is specified, cloud properties are assumed to be constant over the layer.

wc_properties

Define how liquid water content and effective droplet radius are translated to optical properties. Possible choices are

- hu** Parameterization by Hu and Stamnes (1993); this is the default setting. Note that the parameterization is somewhat different for 'correlated_k FU' than for all other cases because in the latter case the parameterization from the newer (March 2000) Fu and Liou code is used while otherwise the data are taken from the original Hu and Stamnes paper. Note that this parameterization has been developed to calculate irradiances, hence it is less suitable for radiances.
- mie** Use pre-calculated Mie tables; makes only sense with **correlated_k**; the tables are expected in **data_files_path/correlated_k/./**. You may get these tables from <http://www.libradtran.org>. This is the correct option to calculate radiances.
- filename** Read optical properties from specified filename; file format is as produced by Frank Evans' **cloudprp**; for each of the internal (computational) wavelengths, a separate file is expected. Use only if you really know what you are doing (this option is subject to change).

wc_scale_gg

Scale the water cloud asymmetry factor for all wavelengths and altitudes with a float between 0.0 and 1.0.

wc_scale_ssa

Scale the water cloud single scattering albedo for all wavelengths and altitudes with a float between 0.0 and 1.0.

wc_set_gg

Set the water cloud asymmetry factor for all wavelengths and altitudes to a float between -1.0 and 1.0. This option is useful only for monochromatic calculations or in wavelength regions where the optical properties of water clouds can be considered constant, e.g. the ultraviolet range.

wc_set_ssa

Set the water cloud single scattering albedo for all wavelengths and altitudes to a float between 0.0 and 1.0. This option is useful only for monochromatic calculations or in wavelength regions where the optical properties of water clouds can be considered constant, e.g. the ultraviolet range.

wc_set_tau

Set the total water cloud optical thickness to a constant value for all wavelengths. The optical thickness defined here is the integral from the surface at the user-defined **altitude** to TOA (top of atmosphere). This option is useful only for monochromatic calculations or in wavelength regions where the optical properties of water clouds can be considered constant, e.g. the ultraviolet range.

wc_set_tau550

Set the water cloud optical thickness at 550nm. The optical thickness defined here is the integral from the surface at the user-defined **altitude** to TOA (top of atmosphere). Other wavelengths are scaled accordingly. Note that this option requires for technical reasons that the wavelength interval defined by **wvn** does contain 550nm.

wvn

Set the wavelength range by specifying first and last wavelength in nm. The default output wavelength grid is that defined in **solar_file**, unless **spline** is specified. Note that the radiative transfer calculations are done on an internal grid which can be influenced with **transmittance_wl_file**.

zout

Output altitudes in km. One or more altitudes may be specified in increasing magnitude. Output altitudes must be within the range defined in the **atmosphere_file**. Note that **zout** does not restructure the atmosphere model. Hence, if you specify **zout 0.730** and have your atmosphere model in **atmosphere_file** go all the way down to sea level, i.e. 0.0km., output is presented at 0.730km and calculations performed with an atmosphere between 0.0 and 0.730 km (and above of course). If you want calculations done for e.g. an elevated site you have to restructure the atmosphere model and make sure it stops at the appropriate altitude. This you may either do by editing the atmosphere file or by using **altitude**. Note that for **rte_solver polradtran** the atmosphere file must contain the altitudes specified by **zout**.

2.2 mie

mie performs Mie scattering calculations for a specified wavelength interval. It reads input from standard input, and outputs to standard output. It is normally invoked in the following way:

```
mie < input_file > output_file
```

The format of the input and output files are described below. Several realistic examples of input files are subsequently given.

Warning: Please note the error checking on input variables is very scarce at the moment. Hence, if you provide erroneous input, the outcome is unpredictable.

2.2.1 The mie input file

The **mie** input file consists of single line entries, each making up a complete input to the **mie** program. First on the line comes the parameter name, followed by one or more parameter values. The parameter name and the parameter values are separated by white space.

Filenames are entered without any surrounding single or double quotes.

Comments are introduced by a **#**. Blank lines are ignored.

The various input parameters are described in detail below.

mie_program

Specify which Mie program to use:

- BH** The Mie scattering program by Bohren and Hoffmann,
ftp://astro.princeton.edu/draine/scat/bhmie/
- MIEVO** The Mie scattering program by W. Wiscombe. For documentation
see src.f/MIEV.doc and the NCAR Mie report at
ftp://climate.gsfc.nasa.gov/pub/wiscombe/Single_Scatt/.

mimcut (positive) value below which imaginary refractive index is regarded as zero
(computation proceeds faster for zero imaginary index). Only used by **mie_**
program MIEVO.

nmom Number of moments of the phase function to be calculated (default: 0). Only
possible with **mie_program** MIEVO.

r_mean The radius [micron] of the particle to calculate single scattering properties of.
Used together with the wavelength information to calculate the Mie size pa-
rameter.

refrac Specify which refractive index to use. The following options are implemented:

ice The complex refractive index is taken from the REFICE function
of W. Wiscombe.

water The complex refractive index is taken from the REFWAT function
of W. Wiscombe.

user <re> <im>
A user defined refractive index. re and im are the real and imagi-
nary parts (both positive numbers).

file <filename>
Read refractive index from a three-column file containing wave-
length [nm], and the real and imaginary parts of the refractive in-
dex (both positive numbers). The Mie calculation is done for each
wavelength defined here.

size_distribution_file

Specify a two column file, r [micron], n(r), which describes a size distribution of
droplets. The Mie calculation is repeated for each value of r found in the size
distribution file, and the final result is a weighted average of these values. The
user himself therefore has to choose a set of r's suited for his specific purpose.

wvn Sets the wavelength range, in nm. Specify first wavelength and last wavelength.
The wavelength step is specified by **wvn_step**. Ignored if **refrac file** is spec-
ified.

wvn_step The wavelength step, in nm. Ignored if **refrac file** is specified.

2.2.2 The mie output

The mie output is currently fixed. We hope to make it user controllable in the future.

mie outputs one line to standard output (stdout) for each wavelength. The format of the output line is

```
lambda refrac_real refrac_img qext omega gg spike pmom(0:nmom)
```

Here

lambda Wavelength, in nm.

refrac_real
The real part of the refractive index.

refrac_img
The imaginary part of the refractive index.

qext The extinction efficiency factor if **r_mean** was specified or the extinction coefficient [km⁻¹] per unit liquid water content [g/m³] if a **size_distribution_file** was specified.

omega The single scattering albedo.

gg The asymmetry parameter.

spike To quote from Wiscombe's 'MIEV0.doc':
(REAL) magnitude of the smallest denominator of either Mie coefficient (a-sub-n or b-sub-n), taken over all terms in the Mie series past N = size parameter XX. Values of SPIKE below about 0.3 signify a ripple spike, since these spikes are produced by abnormally small denominators in the Mie coefficients (normal denominators are of order unity or higher). Defaults to 1.0 when not on a spike. Does not identify all resonances (we are still working on that).

Meaningless if a **size_distribution_file** was specified.

pmom(0:nmom)
The moments of the phase function. The phase function $p(\mu)$ is

$$p(\mu) = \sum_{m=0}^{\infty} (2m+1) \cdot k_m \cdot P_m(\mu)$$

where k_m is the m'th moment and $P_m(\mu)$ is the m'th Legendre polynomial.

2.2.3 Examples of mie input files

An example of a complete input file is

```
mie_program MIEV0
refrac ice
mimcut 0.0000000001
r_mean 200.
wvn 280. 5000.
wvn_step 5.
```

2.3 integrate

integrate calculates the integral between limits `x_min` and `x_max` by interpolating the data points (`x[i]`, `y[i]`) with natural cubic splines or linear interpolation. `x_min` and `x_max` are the minimum and maximum values of the first column in the `input_file`. The x-values in the first column must be in ascending order.

The different options to **integrate** are displayed when executing:

```
integrate -h
```

2.4 spline

spline interpolates discrete data points using natural cubic splines or linear interpolation. The x-values in the first column must be in ascending order.

The different options to **spline** are displayed when executing:

```
spline -h
```

2.5 conv

conv convolutes a spectrum with a given filter function.

The different options to **conv** are displayed when executing:

```
conv -h
```

2.6 addlevel

addlevel is a simple shell script to add a level to one of the existing standard profiles.

The different options to **addlevel** are displayed when executing:

```
addlevel -h
```

2.7 snowalbedo

snowalbedo calculate the diffuse and direct albedo of snow as formulated by Wiscombe and Warren (1980).

The different options to **snowalbedo** are displayed when executing:

```
snowalbedo -h
```

2.8 cldprp

cldprp calculates wavelength-dependent cloud properties using one of several parameterizations.

The different options to **cldprp** are displayed when executing:

```
cldprp -h
```

2.9 cldgen

`cldgen` generates 1D, 2D, and 3D model clouds to be used as input to libRadtran / MYSTIC.

```
cldgen < input_file > output_file
```

The format of the input and output files are described below. Several realistic examples of input files are subsequently given.

Warning: Please note the error checking on input variables is very scarce at the moment. Hence, if you provide erroneous input, the outcome is unpredictable.

2.9.1 The cldgen input file

The `cldgen` input file consists of single line entries, each making up a complete input to the `cldgen` program. First on the line comes the parameter name, followed by one or more parameter values. The parameter name and the parameter values are separated by white space.

Filenames are entered without any surrounding single or double quotes.

Comments are introduced by a `#`. Blank lines are ignored.

The various input parameters are described in detail below.

`cloud_asy`

Cloud asymmetry parameter.

`cloud_ssa`

Cloud single scattering albedo.

`atmosphere_file_input`

Input 1D (uvspec) cloudless sky atmosphere.

`atmosphere_file_output`

Output 1D (uvspec) cloudy sky atmosphere.

`automaton_cloud_fraction`

Cloud fraction.

`automaton_cloud_fraction_file`

Read cellular automaton cloud fractions from a file (one per line).

`automaton_double_upward`

Doubled upward infection distance.

`automaton_periodic`

Periodic cloud (default: cloud generation stops at the boundary).

`automaton_generate_images`

Generate images forv each step of the cellular automaton.

`cascade_fraction`

Fraction for bounded cascade to be re-distributed in each step.

<code>cascade_scale</code>	Scale factor for bounded cascade by which the re-distributed fraction is decreased in each step.
<code>cascade_step</code>	Number of steps for bounded cascade.
<code>cascade2D_p1</code>	p1 for 2D bounded cascade.
<code>cascade2D_p2</code>	p2 for 2D bounded cascade.
<code>cascade2D_H</code>	H for 2D bounded cascade.
<code>cascade2D_step</code>	Number of steps for 2D bounded cascade.
<code>cloud_bottom</code>	Cloud bottom [m] (default: 1000).
<code>cloud_top</code>	Cloud top [m] (default: 2000).
<code>cloud_file_output</code>	Output 1D (uvspec) cloud file.
<code>cloud_extinction</code>	Extinction coefficient [m-1]. Note that only one of both may be defined, <code>cloud_extinction</code> or <code>cloud_tau</code> .
<code>cloud_tau</code>	Average vertical optical thickness of the cloud.
<code>cloud_type</code>	Type of cloud model. <code>cascade1D</code> 1D bounded cascade. <code>cascade2D</code> 2D bounded cascade. <code>automaton</code> 3D cellular automaton. <code>gamma</code> Gamm distribution cloud.
<code>dx</code>	Pixel size in x direction [m] (default: dz).
<code>dy</code>	Pixel size in y direction [m] (default: dz).

Nx Number of pixels in x direction (default: 100).
Ny Number of pixels in y direction (default: 100).
Nz Number of pixels in z direction (default: 100).

2.9.2 The cldgen output

2.9.3 Examples of cldgen input files

An example of a complete input file is

```
cloud_type CASCADE2D

Nx 200
Ny 200

cascade2D_step 7
cascade2D_p1 0.26
cascade2D_p2 0.34
cascade2D_H 0.333333333

atmosphere_file_input ../data/atmmod/afglus.dat

cloud_tau 1

cloud_bottom 8000
cloud_top 10000
```

2.10 Geno3tab

The Perl script **Gen_o3_tab.pl** is used to generate a matrix of ozone values for solar zenith angle versus a chosen ratio of global irradiance at different wavelengths. The table is read by the C program **read_o3_tab** which, for a solar zenith angle and a measured ratio, returns the overhead ozone column. All available options are displayed when executing

```
perl Gen_o3_tab.pl --help
```

and

```
read_o3_tab -h
```

The following different types of tables may be generated.

2.10.1 Simple wavelength ratios with Gen_o3_tab

The simplest type of table is made of ratios of the global irradiance at two single wavelengths. This is the type of table described by Stamnes et al. (1991). This type of table is typically used to analyse measurements of the global irradiance from spectroradiometers. It is generated by the following command (\ is line continuation character)

```
perl Gen_o3_tab.pl --slitfunction slitfncfile --lower_lambda 305. \
--upper_lambda 340. --file table.dat
```

Here ‘slitfncfile’ is the name of the slit function file. It is a two column file where the first column is the wavelength (nm, in relative units) and the second column holds the slit function. The slit function must be normalized to unity at the center wavelength.

The generated table ‘table.dat’ is read by `read_o3_tab` for a measured ratio, `-r 10.0`, and solar zenith angle, `-s 30.0`, corresponding to the modelled ratio in the table

```
read_o3_tab -r 10.0 -s 30.0 table.dat
```

2.10.2 Bandpassed wavelength ratios with Gen_o3_tab

Instead of using single wavelengths it may be of advantage to use ratios of irradiances covering a certain wavelength range and weighted with a bandpass function. This approach may reduce problems due to changes in cloud cover and experimental uncertainties. This approach is also suitable to calculate ozone columns from multichannel, moderate bandwidth filter instruments, Dahlback (1996). Such tables are generated by

```
perl Gen_o3_tab.pl --slitfunction slitfncfile --lower_lambda 305.0 \
--upper_lambda 320.0 --file table.dat \
--bandpasslower bplow.dat --bandpassupper bpupp.dat
```

Here ‘bplow.dat’ and ‘bpupp.dat’ are the bandpass function of the lower and upper wavelength region respectively. The bandpass files have two columns. The first column is the wavelength in nm and relative units to `-lower_lambda` and `-upper_lambda`. If absolute units are specified as for filter instruments, use the `-absolute` option. The second column is the bandpass function.

The tables are read in the same way as the simple wavelength ratio tables.

2.11 Genwctab

The Perl script `Gen_wc_tab.pl` is used to generate a matrix of cloud optical depth for solar zenith angle versus a chosen global irradiance at a selected wavelength. The wavelength should be chosen such that it is not affected by ozone, e.g. 380 nm. The table is read by the C program `read_o3_tab` which, for a solar zenith angle and a measured irradiance, returns the overhead cloud optical depth. All available options are displayed when executing

```
perl Gen_wc_tab.pl --help
```

and

```
read_o3_tab -h
```

The following different types of tables may be generated.

2.11.1 Simple wavelength ratios with Gen_wc_tab

The simplest type of table is made of the global irradiance at a single wavelength. This is the type of table described by Stamnes et al. (1991). This type of table is typically used

to analyse measurements of the global irradiance from spectroradiometers. It is generated by the following command (\ is line continuation character)

```
perl Gen_wc_tab.pl --slitfunction slitfncfile --lambda 380. \
--file table.dat
```

Here ‘slitfncfile’ is the name of the slit function file. It is a two column file where the first column is the wavelength (nm, in relative units) and the second column holds the slit function. The slit function must be normalized to unity at the center wavelength.

The generated table ‘table.dat’ is read by `read_o3_tab` for a measured global irradiance, `-r 10.0`, and solar zenith angle, `-s 30.0`, corresponding to the modelled ratio in the table. The table must be corrected for the Earth–Sun distance for the day of the measurement. This is achieved by specifying `-d 170`, where 170 is the day number. The table is generated for day 1.

```
read_o3_tab -r 10.0 -s 30.0 -d 170 table.dat
```

2.11.2 Bandpassed wavelength ratios with `Gen_wc_tab`

Instead of using a single wavelength it may be of advantage to use irradiances covering a certain wavelength range and weighted with a bandpass function. This approach may reduce problems due to changes in cloud cover and experimental uncertainties. This approach is also suitable to calculate cloud optical depth from multichannel, moderate bandwidth filter instruments, Dahlback (1996). Such tables are generated by

```
perl Gen_wc_tab.pl --slitfunction slitfncfile --lambda 380.0 \
--file table.dat --bandpass bp.dat
```

Here ‘bp.dat’ is the bandpass function of the wavelength region. The bandpass file have two columns. The first column is the wavelength in nm and relative units to `-lambda`. If absolute units are specified as for filter instruments, use the `-absolute` option. The second column is the bandpass function.

The tables are read in the same way as the simple wavelength irradiance tables.

References

- Anderson, G.P., S.A. Clough, F.X. Kneizys, J.H. Chetwynd and E.P. Shettle 1986, 'AFGL Atmospheric Constituent Profiles (0-120 km)', AFGL-TR-86-0110, AFGL (OPI), Hanscom AFB, MA 01736.
- Cox, C. and W. Munk, 'Measurement of the roughness of the sea surface from photographs of the sun's glitter', 1954, *Journal of the Optical Society of America*, 44, 838-850.
- Cox, C. and W. Munk, 'Statistics of the sea surface derived from sun glitter', 1954, *Journal of Marine Research*, 13, 198-227.
- Dahlback, A. and K. Stamnes, 1991, 'A new spherical model for computing the radiation field available for photolysis and heating at twilight', *Planet. Space Sci.*, 39, 671-683.
- Dahlback, A., 1996, 'Measurement of biologically effective UV doses, total ozone abundances, and cloud effects with multichannel, moderate bandwidth filter instruments', *Applied Optics*, 6514-6521.
- Daumont, D., J. Brion, J. Charbonnier, and J. Malicet, 1992, 'Ozone UV Spectroscopy I: Absorption Cross-Sections at Room Temperature', *J. Atm. Chem.*, 15, 145-155.
- Edwards, D.P., 1992, 'GENLN2: A general line-by-line atmospheric transmittance and radiance model: Version 3.0 description and users guide', National Center for Atmospheric Research (NCAR), NCAR/TN-367+STR, Boulder, Colorado.
- Evans, K.F. and G.L. Stephens, 1991, 'A new polarized atmospheric radiative transfer model', *J. Quant. Spectrosc. Radiat. Transfer*, 46, 413-423.
- Fu, Q. and K.N. Liou, 1992, 'On the correlated k-distribution method for radiative transfer in nonhomogeneous atmospheres', *J. Atm. Sci.*, 49, 2139-2156.
- Fu, Q. and K.N. Liou, 1993, Parameterization of the radiative properties of cirrus clouds, *J. Atm. Sci.*, 50, 2008-2025.
- Fu, Q., 1996, 'An accurate parameterization of the solar radiative properties of cirrus clouds in climate models', *Journal of Climate*, 9, 2058-2082.
- Fu, Q., P. Yang, and W.B. Sun, 1998, 'An accurate parameterization of the infrared radiative properties of cirrus clouds in climate models', *Journal of Climate*, 11, 2223-2237.
- Garcia, R., and C. Siewert 1985, 'Benchmark results in radiative transfer', *Transp. Theory and Stat. Physics*, 14, 437-484.
- Hu, Y.X. and K. Stamnes 1993, 'An accurate parameterization of the radiative properties of water clouds suitable for use in climate models', *J. Climate*, 6, 728-742.
- Kato, S., T.P. Ackerman, J.H. Mather, and E.E. Clothiaux, 1999, 'The k-distribution method and correlated-k approximation for a shortwave radiative transfer model' *J. Quant. Spectrosc. Radiat. Transfer*, 62, 109-121.
- Key, J., P. Yang, B. Baum, and S. Nasiri, 2002, 'Parameterization of shortwave ice cloud optical properties for various particle habits', *J. Geophys. Res.*, 107, D13, 10.1029/2001JD000742.
- Kratz, D.P., 1995, 'The correlated k-distribution technique as applied to the AVHRR channels', *J. Quant. Spectrosc. Radiat. Transfer*, 53, 501-517.

- Kylling, A., K. Stamnes, and S.-C. Tsay, 1995, 'A reliable and efficient two-stream algorithm for radiative transfer; Documentation of accuracy in realistic layered media', accepted for publication in *J. of Atmospheric Chemistry*.
- Malicet, J., D. Daumont, J. Charbonnier, C. Parisse, A. Chakir, and J. Brion, 1995, 'Ozone UV Spectroscopy II: Absorption Cross-Sections and Temperature Dependence', *J. Atm. Chem.*, 21, 263-273.
- Mayer, B., G. Seckmeyer and A. Kylling, 1997, 'Systematic long-term comparison of spectral UV measurements and UVSPEC modeling results', *J. Geophys. Res.*, 102, 8755-8767.
- Mayer, B., A. Kylling, S. Madronich, and G. Seckmeyer, 1998, Enhanced absorption of UV radiation due to multiple scattering in clouds: experimental evidence and theoretical explanation, *J. Geophys. Res.*, 103 (D23), 31241-31254.
- Molina, L. T. and M. J. Molina, 1986, 'Absolute Absorption Cross Sections of Ozone in the 185- to 350-nm Wavelength Range', *J. Geophys. Res.*, 91, 14,501-14,508.
- Nakajima, T. and M. Tanaka, 'Effect of wind-generated waves on the transfer of solar radiation in the atmosphere-ocean system', 1983, *J. Quant. Spectrosc. Radiat. Transfer*, 29, 521-537.
- Nakajima, T., and M. Tanaka, 1988, 'Algorithms for radiative intensity calculations in moderately thick atmospheres using a truncation approximation', *J. Quant. Spectrosc. Radiat. Transfer*, 40, 51-69.
- Nicolet, M., 1984, 'On the molecular scattering in the terrestrial atmosphere: an empirical formula for its calculation in the homosphere', *Planet. Space Sci.*, 32, 1467-1468.
- Pierluissi, J.H., and G.-S. Peng, 1985, 'New molecular transmission band models for LOWTRAN', *Optical Engineering*, 24, 541-547.
- Rahman, H. and B. Pinty, and M.M. Verstraete, 1993, 'Coupled Surface-Atmosphere Reflectance (CSAR) Model. 2. Semiempirical Surface Model Usable With NOAA Advanced Very High Resolution Radiometer Data', *J. Geophys. Res.* 98, 20791-20801.
- Ricchiazzi, P., S. Yang, C. Gautier, and D. Sowle, 1998, 'SBDART: A research and teaching software tool for plane-parallel radiative transfer in the Earth's atmosphere', *Bull. Am. Met. Soc.* 79, 2101-2114.
- Shettle E. P. (1989), 'Models of aerosols, clouds and precipitation for atmospheric propagation studies', in AGARD Conference Proceedings No. 454, Atmospheric propagation in the uv, visible, ir and mm-region and related system aspects.
- Stamnes, K., 1982, 'Reflection and Transmission by a Vertically Inhomogeneous Planetary Atmosphere', *Planet. Space Sci.* 30, 727-732.
- Stamnes, K., S.-C. Tsay, W. Wiscombe and K. Jayaweera, 1988, 'Numerically stable algorithm for discrete-ordinate-method radiative transfer in multiple scattering and emitting layered media', *Applied Optics*, 27, 2502.
- Stamnes, K., J. Slusser, and M. Bowen, 1991, 'Derivation of Total Ozone Abundance and Cloud Effects from Spectral Irradiance Measurements', *Applied Optics*, 30, 4418-4426.
- Wiscombe, W., 1977, 'The Delta-M Method: Rapid Yet Accurate Radiative Flux Calculations', *J. Atmos. Sci.* 34, 1408-1422.

- Wiscombe, W.J. and J.W. Evans, 1977, 'Exponential-sum fitting of radiative transmission functions', *Journal of Computational Physics*, 24, 416-444.
- Wiscombe, W.J. and S.G. Warren, 1980, 'A Model for the Spectral Albedo of Snow. I: Pure Snow', *J. Atm. Sci.*, 37, 2712-2733.
- Yang, P. and K.N. Liou and K. Wyser, and D. Mitchell, 2000, 'Parameterization of the scattering and absorption properties of individual ice crystals', *J. Geophys. Res.*, 105(D4), 4699-4718.
- Yang, P., B.-C. Gao, B.A. Baum, X.H. Yong, W.J. Wiscombe, S.-C. Tsay, D.M. Winker, and S.L. Nasiri, 2001, 'Radiative properties of cirrus clouds in the infrared (8-13 micron) spectral region', *J. Quant. Spectrosc. Radiat. Transfer*, 70, 473-504.

Table of Contents

Preface	1
Acknowledgements	1
1 A Brief Overview of libRadtran	3
1.1 Radiative transfer calculations	3
1.2 Ozone retrieval from global irradiance measurements	3
1.3 Cloud optical thickness from global irradiance measurements	4
1.4 ... and much more	4
2 Some useful tools	5
2.1 uvspec	5
2.1.1 The uvspec input file	5
2.1.1.1 Cloudless, aerosol-free atmosphere	5
2.1.1.2 Spectral resolution	8
2.1.1.3 Aerosol	13
2.1.1.4 Water clouds	15
2.1.1.5 Ice clouds	15
2.1.1.6 Calculation of radiances	15
2.1.2 The uvspec output	16
2.1.2.1 DISORT, SDISORT and SPSPDISORT	16
2.1.2.2 TWOSTR	16
2.1.2.3 POLRADTRAN	16
2.1.2.4 Description of symbols	17
2.1.3 Complete description of input parameters	18
2.2 mie	41
2.2.1 The mie input file	41
2.2.2 The mie output	43
2.2.3 Examples of mie input files	43
2.3 integrate	44
2.4 spline	44
2.5 conv	44
2.6 addlevel	44
2.7 snowalbedo	44
2.8 cldprp	44
2.9 cldgen	45
2.9.1 The cldgen input file	45
2.9.2 The cldgen output	47
2.9.3 Examples of cldgen input files	47
2.10 Geno3tab	47
2.10.1 Simple wavelength ratios with Gen_o3_tab	47
2.10.2 Bandpassed wavelength ratios with Gen_o3_tab	48

2.11	Genwctab	48
2.11.1	Simple wavelength ratios with Gen_wc_tab.....	48
2.11.2	Bandpassed wavelength ratios with Gen_wc_tab	49
References		51