libRadtran

Bernhard Mayer, Arve Kylling, Ulrich Hamann, Claudia Emde

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 Bernhard Mayer (bernhard.mayer@dlr.de), Arve Kylling (arve.kylling@helse-sunnmore.no), Ulrich Hamann (ulrich.hamann@dlr.de), and Claudia Emde (claudia.emde@dlr.de), the following people have contributed to libRadtran or helped out in various other ways (the list is almost certainly incomplete - please let us know if we forgot somebody):

- 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). In addition, Ping Yang and Heli Wei kindly provided a comprehensive database of particle single scattering properties which we used to derive a consistent set of ice cloud optical properties for the spectral range 0.2 - 100 micron following the detailed description in 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.
- Luca Bugliaro (`luca.bugliaro@dlr.de`) wrote the analytical TZS solver (thermal, zero scattering).
- Sina Lohmann (`sina.lohmann@dlr.de`) reduced the "overhead time" for reading the Kato et al. tables dramatically which resulted in a speedup of a factor of 2 in a twostr solar irradiance calculation.
- Detailed ice cloud properties were provided by Bryan Baum (`bryan.baum@ssec.wisc.edu`).
- Yongxiang Hu (`yongxiang.hu-1@nasa.gov`) provided the delta-fit program used to calculate the Legendre coefficients for `ic_properties baum`.
- UCAR/Unidata for providing the *netCDF* library.
- 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 2.0-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, `usvpec` 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 absorbtion, 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 2.11 [Geno3tab], page 79.

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 were 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 2.12 [Genwctab], page 80.

1.4 ... and much more

Apart from the above described programs, the 'bin' 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.

uvspec may produce a wealth of diagnostic messages and warnings, depending on your use of `verbose` or `quiet`. Diagnostics, error messages, and warnings are written to `stderr` while the uvspec output is written to `stdout`. To make use of this extra information, you may want to write the standard uvspec output to one file and the diagnostic messages to another. To do so, try (`./uvspec < uvspec.inp > uvspec.out) >& verbose.txt`. The irradiances and radiances will be written to '`uvspec.out`' while all diagnostic messages go into '`verbose.txt`'. This method can also be used to collect uvspec error messages.

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 Section 2.1.3 [Complete description of input parameters], page 21.

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. Be aware that also option in another `included` input file will overwrite options specified before.

2.1.1.1 Cloudless, aerosol-free atmosphere

The simplest possible input file contains only a few lines:

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

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

wavelength 310.0 310.0   # Wavelength range [nm]

quiet
```

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

```
                              # 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
wavelength 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_pcl`, 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 21.

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` or `molecular_tau_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):
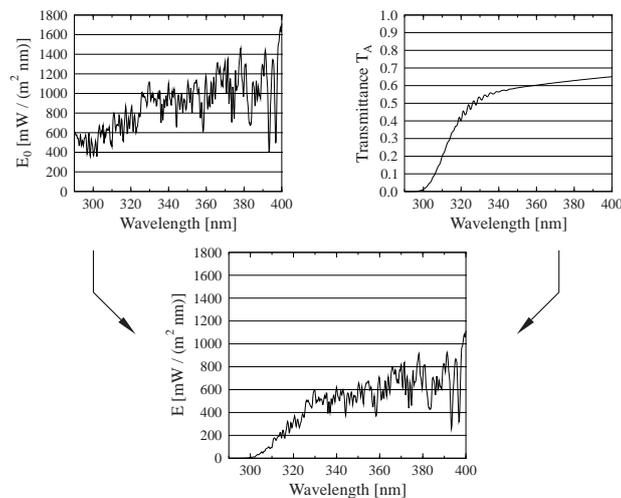
The transmittance (or reflectance) is calculated on a moderate resolution grid which reduces the number of calls to the `rte_solver` and hence the computational time. Then, the transmittance is interpolated to the wavelengths in the `solar_file` (which is usually defined with higher spectral resolution), 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 'wavelength 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` can be specified which 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. `solar_file` may be omitted for thermal radiation calculations (`source thermal`) as well as for `transmittance` and `reflectivity` calculations. If omitted, the output grid equals the internal wavelength grid.

If required, a **user-defined internal grid** can be specified with `transmittance_wl_file` or `molecular_tau_file`. Note that this is a way to speed up the calculation considerably. E.g., for some applications 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, from LOWTRAN; code adopted from SBDART (Ricchiazzi 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 micrometer.
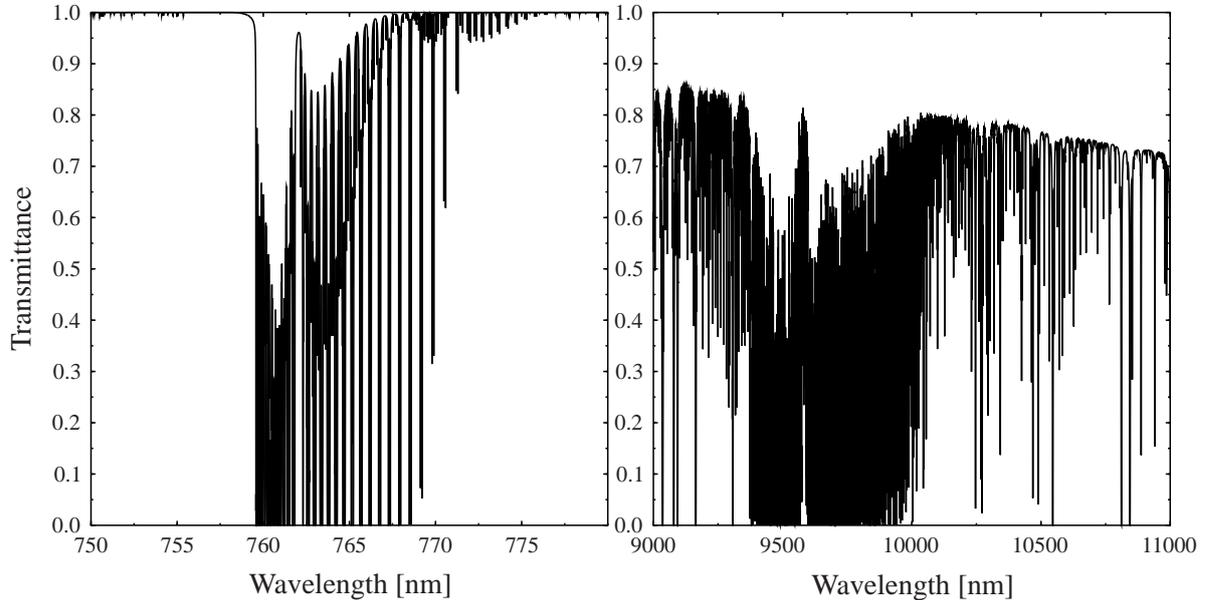
All spectral lines in the left figure are due to absorption by oxygen, while the ones in the right figure are due to ozone, water vapour, and CO2. Line-by-line is obviously the exact way for radiation calculations. For most applications, however, line-by-line is far too slow. Here one needs a band parameterization, 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. Kato et al. (1999) is a accurate parameterization for the solar spectral range. uvspec contains three different versions:

Kato   The original tables provide by Seiji Kato which should correspond to the full version described in Kato et al. (1999); 575 subbands total, that is, 575 calls to the `rte_solver`

Kato2   A new, optimized version of the tables, provided by Seiji Kato, 2003, with only 148 subbands (that is, calls to the `rte_solver`); the uncertainty is only slightly higher than Kato; the absorption coefficients are based on HITRAN 2000.

Kato2.96   Similar to Kato2 but based on HITRAN96.

The following Figure shows a comparison between the three parameterization which are part of libRadtran and the data from Figure 3 by Kato et al. [1999]. It is immediately obvious that the uncertainty is high for all bands above 2.5 micrometer which is probably due to the treatment of band overlap. For this reasons, the results for the indiviudal bands should not be trusted while the integrated shortwave radiation (the sum of all 32 bands) is

calculated with high accuracy because (1) the bands above 2.5 micrometer contribute only little to the integrated irradiance; and (2) errors are random and cancel each other to some degree.
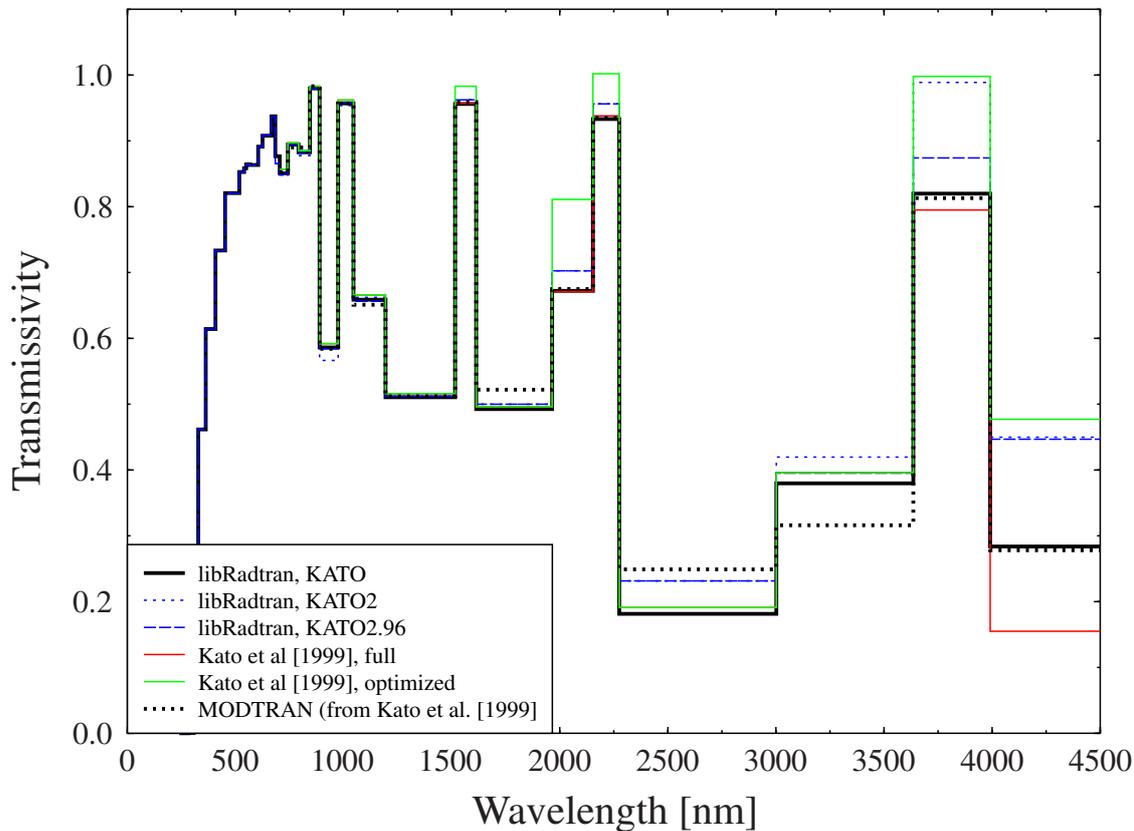
For more information on these parameterizations please refer to the mentioned publications. Correlated-k is a powerful way to calculate spectrally integrated quantities, however, it takes away some flexibility. In particular, this means that the wavelength grid is no longer chosen by the user but by the parameterization, that is, by uvspec. The uvspec output is then no longer spectral quantities, e.g. W / (m2 nm), but integrated over the spectral bands, e.g. W / m2.

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.


                              # Location of atmospheric profile file.
atmosphere_file ../examples/AFGLMS50.DAT
                              # Location of the extraterrestrial spectrum


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

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


output sum                    # Calculate integrated solar irradiance

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 LOWTRAN/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 LOWTRAN/SBDART gas parameterization is invoked with `correlated_k LOWTRAN`. 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`:

```
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
wavelength 250.0 2500.0    # Wavelength range

correlated_k SBDART        # select SBDART molecular absorption

aerosol_default
aerosol_visibility 20

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

output per_nm

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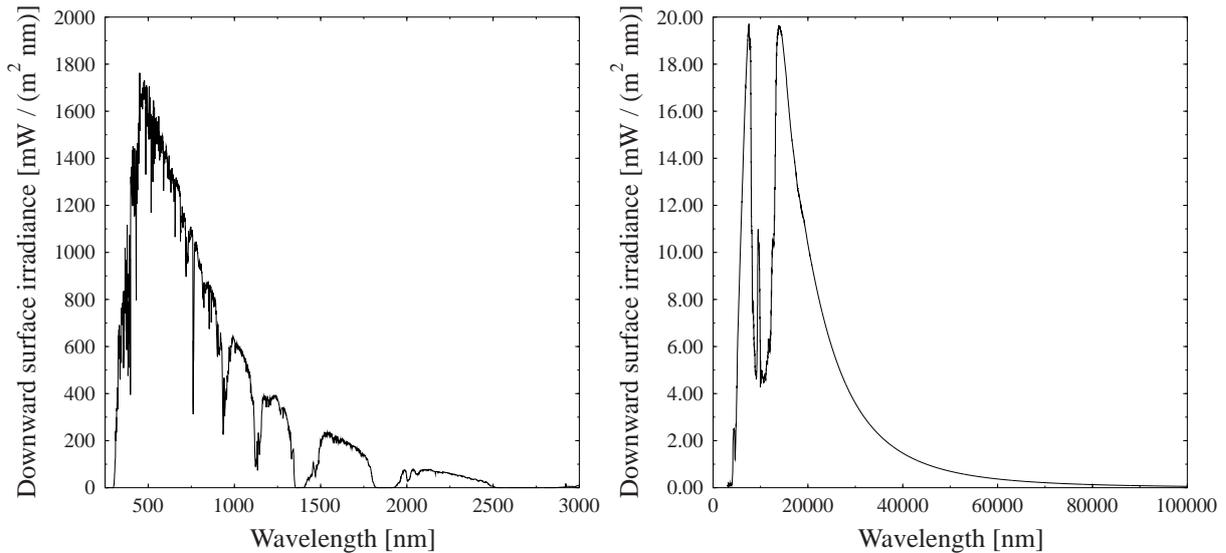
The figure shows the results of the solar and thermal calculations. The water vapour absorption bands in the solar range are clearly visible, as is the absorption window around 10 micrometer in the thermal. Please note the following points:

- Thermal radiation is per default output in W/(m2 cm-1), if the bandwidth is equal to 1 cm-1 (default for `correlated_k LOWTRAN` calculations). Otherwise the output is the integrated flux over the wavenumber interval specified by `thermal_bandwith`, `thermal_bands_file`, or by the `correlated_k` option (Kato, Kato2, Kato2.96, Fu, AVHRR_KRATZ, or Generic). To convert e.g. to W/(m2 nm) use `output per_nm` or multiply with k/lambda where k is the wavenumber [cm-1] and lambda 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` may 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.

- 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 translates to 0.3nm at 750nm and to 50nm at 10 micrometer. 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 LOWTRAN/SBDART absorption parameterization.
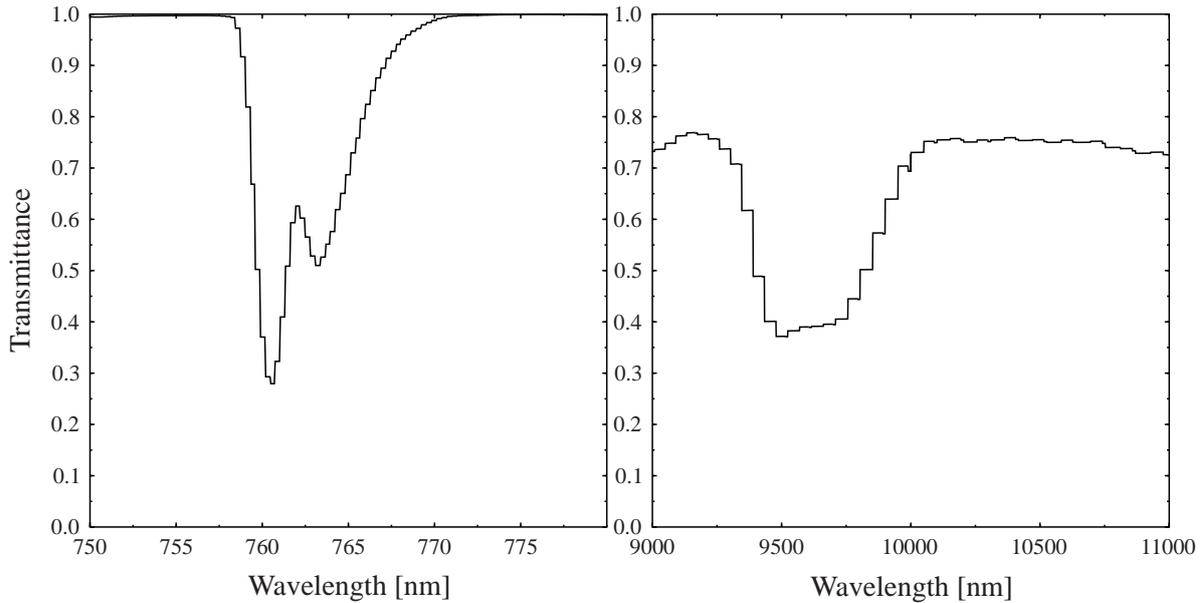
The resolution is about 5cm-1 which translates to about 0.3nm in the left figure (oxygen A-band) and 50nm in the right figure (ozone absorption band in the atmospheric window). Compare this figure to the above line-by-line example to get an impression about the differences between both methods.

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 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 integrated optical thickness (profiles are not changed).

An example for a uvspec aerosol description is

```
include ../examples/UVSPEC_CLEAR.INP

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 layer or level. By combining the following lines with the clear sky example given above a complete uvspec input file including water clouds is constructed.

```
include ../examples/UVSPEC_CLEAR.INP

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

A typical example for a wc_file looks like:

```
#      z      LWC     R_eff
#    (km)   (g/m^3)  (um)
     5.000   0        0
     4.000   0.2     12.0
     3.000   0.1     10.0
     2.000   0.1      8.0
```

The three columns are the level altitude [km], the liquid water content [g/m3], and the effective droplet radius [micrometer]. Per default (since version 1.4), these quantities are interpreted as layer quantities, and in the above example, the cloud would extend from 2 to 5km, with e.g. a LWC of 0.2 g/m3 for the layer between 4 and 5km. Before version 1.4 the `wc_file` was interpreted as level quantities (unless `wc_layer` was specified). That is, the value 0.2 g/m3 referred to altitude 4.0km, as e.g. in a radiosonde profile. The properties of each layer were calculated as average over the adjacent levels. E.g. the single scattering properties for the model layer between 3 and 4km were obtained by averaging over the two levels 3km and 4km. To allow definition of sharp cloud boundaries, clouds were only formed if both liquid water contents above and below the respective layer were larger than 0. Hence, in the above example, the layers between 2 and 3 as well as between 3 and 4km were cloudy while those between 1 and 2km and between 4 and 5km were not. To switch to the old behaviour, use `wc_level`.

To make sure that the clouds really look as you want them to look, it is recommended to use the `verbose` option. This option shows not only where the cloud is actually placed, it rather tells the user exactly how LWC and effective radius are translated into optical properties, depending on the choice of parameterisation. Please also note that the definition of the empty top level at 5km is important to tell uvspec where the cloud ends. If omitted, the cloud would extend all the way to the top of the atmosphere.

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. If `wc_properties mie` is selected, the model expects one Mie cloud property file for each internal wavelength which is useful for the fixed wavelength grids used by the correlated-k parameterisations `correlated_k kato`, `correlated_k fu`, etc. For a spectral calculation with free wavelength grid, there is also the possibility to use a pre-defined set of Mie tables (available at the web site) and to defined `wc_properties_interpolate` to automatically interpolate the Mie properties to the internal wavelength grid. Although this is an extremely useful option, please use it careful because it might consume enourmous amounts of memory. 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 explicitely 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. Please note in addition that there are different definitions of the effective radius and. E.g. the parameterizations by Key et al. (2002) and Baum et al. (2005a,b, 2007) use the same definition whereas Fu (1996) actually uses another definition (see explanation of `ic_properties`). Finally, the sharp forward peak which is typical for ice particles can also be treated differently: E.g., Fu (1996) provides delta-scaled optical properties while Key et al. (2002) uses unscaled parameters (see explanation of `ic_fu_tau`). The following figure illustrates the implications. Plotted are extinction coefficient, asymmetry parameter, and single scattering albedo for ice clouds with an effective radius of 25 micrometers as a function of wavelength. If treated consistently, all parameterizations Key et al. (2002), Fu (1996) and Baum (2005a,b) provide nearly identical results (solid lines, default settings in `uvspec`). If the definition of effective radius by Fu (1996) and delta-scaling is applied the optical properties look different. The effect of delta scaling on a radiative transfer calculation is that the direct irradiance in increased and the diffuse irradiance is decreased, whereas the global irradiance remains unchanged. The definition of the effective radius has a smaller effect but it modifies also the global irradiance. Note that the parameterization by Baum et al. (2005a,b, 2007) is plotted only up to 2200 nm. The reason is that it does not cover the full spectral region, it is available for two spectral regions (from 0.4–2.2 micrometers and from about 3–100 micrometers).
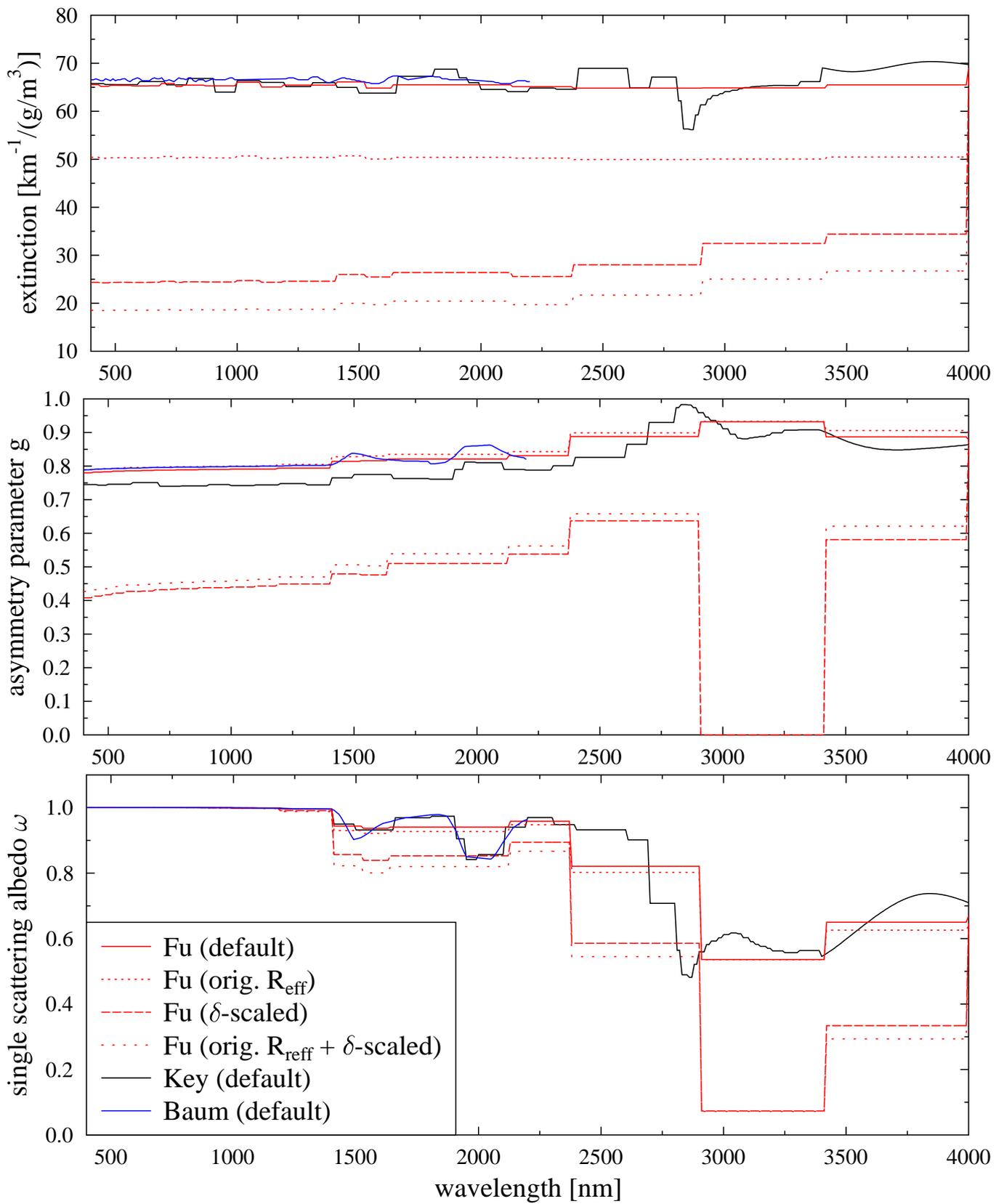
2.1.1.6 Calculation of radiances

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

```
include ../examples/UVSPEC_AEROSOL.INP # Include's may be nested.

rte_solver disort2 # This override what is specified in above file
                   # and files included in that file etc.

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. It may be controlled to some degree using output_user.

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 edir edn eup uavgdir uavgdn uavgup
```

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

```
lambda edir edn eup uavgdir 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 edir edn eup uavgdir 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 edir edn eup 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.

cmu   Computational polar angles from `polradtran`.

down_flux, up_flux
           The total (direct+diffuse) downward (down_flux) and upward (up_flux) irradiances. Same units as extraterrestrial irradiance.

lambda   Wavelength (nm)

edir   Direct beam irradiance (same unit as extraterrestrial irradiance, e.g mW/(m2 nm) if using the 'atlas3' spectrum in the 'data/solar_flux' directory.)

edn   Diffuse down irradiance, i.e. total minus direct beam (same unit as `edir`).

eup   Diffuse up irradiance (same unit as `edir`).

uavg   The mean intensity. Proportional to the actinic flux: To obtain the actinic flux, multiply the mean intensity by 4 pi (same unit as `edir`).

uavgdir   Direct beam contribution to the mean intensity. (same unit as `edir`).

uavgdn   Diffuse downward radiation contribution to the mean intensity. (same unit as `edir`).

uavgup   Diffuse upward radiation contribution to the mean intensity. (same unit as `edir`).

u0u   The azimuthally averaged intensity at `numu` user specified angles `umu`. (units of e.g. mW/(m2 nm sr) if using the 'atlas3' spectrum in the 'data/solar_flux' directory.)

uu   The radiance (intensity) at `umu` and `phi` user specified angles (unit e.g. mW/(m2 nm sr) if using the 'atlas3' spectrum in the 'data/solar_flux' directory.)

uu_down, uu_up

           The downwelling and upwelling radiances (intensity) at `cmu` and `phi` angles (unit e.g. mW/(m2 nm sr) if using the 'atlas3' spectrum in the 'data/solar_flux' directory.)

The total downward irradiance is given by

       irr_down = edir + edn

The total mean intensity is given by

       uavg = uavgdir + uavgdn + uavgup

If `deltam` is on it does not make sense to look at the direct and diffuse contributions to `uavg` separately since they are delta-M scaled (that is, the direct would be larger than expected and the diffuse would be smaller).

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. Be aware that also option in another `included` input file will overwrite options specified before.

The various input parameters are described in detail below.

absorption

           Switch off absorption by individual minor trace gases which are currently only included when `correlated_k lowtran` is chosen. The syntax is

                 absorption species on/off

           where species may be one of O4, N2, CO, SO2, NH3, NO, HNO3. By default all are switched on.

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-definded `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 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. Please note that the uppermost line of the `aerosol_files` denotes simply the top altitude of the uppermost layer. The optical properties of this line are consequently ignored. There are two options for this line: either an optical property file with zero optical thickness is specified or "NULL" instead.

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. The asymmetry parameter defined with this option is constant with wavelength. If you require spectral dependence please use `aerosol_files`. 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_no_scattering

Switch off scattering by aerosols.

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. Please note that only the single-scattering albedo, the scattering phase function, and the wavelength-dependence of the extinction coefficient are affected by the Mie calculation while the absolute value of the extinction coefficient is taken from other sources; generally, the extinction coefficient at the first *internal* wavelength is taken from whatever is available (either default Shettle (1989) or user-defined); the extinction at all other wavelengths is scaled according to the Mie calculation. For this reason, the absolute numbers are not relevant - only the shape of the size distribution matters. In detail: 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 at least 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`. *It is recommended to avoid this option and rather to calculate the aerosol optical properties externally e.g. with mie and to pass them to uvspec with aerosol_files.*

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. Please see the description of `aerosol_refrac_file` to learn how the optical properties are set up. *It is recommended to avoid this option and rather to calculate the aerosol optical properties externally e.g. with mie and to pass them to uvspec with aerosol_files.*

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-definded `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 `wavelength` does contain 550nm. The optical thickness defined here is the integral from the user-definded `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 [micrometer] 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. Please note that only the single-scattering albedo, the scattering phase function, and the wavelength-dependence of the extinction coefficient

are affected by the Mie calculation while the absolute value of the extinction coefficient is taken from other sources; generally, the extinction coefficient at the first *internal* wavelength is taken from whatever is available (either default Shettle (1989) or user-defined); the extinction at all other wavelengths is scaled according to the Mie calculation. For this reason, the absolute numbers are not relevant - only the shape of the size distribution matters. For details see also the description of `aerosol_refrac_file`. *It is recommended to avoid this option and rather to calculate the aerosol optical properties externally e.g. with mie and to pass them to uvspec with aerosol_files.*

`aerosol_species_file`

Specify mass density profiles of a mixture of aerosol types.

`aerosol_species_file profile [aero_1 aero_2 ... aero_n]`

where `aero_1` to `aero_n` are the aerosol species to be included. For each of these species, the optical properties are read from the `aerosol_species_library`, e.g. the OPAC data set provided with libRadtran. The profile file needs to include vertical profiles for each of these species. This file can be either in *netCDF*-format (automatically recognized filename extension `.nc` or `.cdf`) or in ASCII format. The format of the ASCII file is:

```
z1    dens(aero_1, z1) dens(aero_2, z1)  ... dens(aero_n, z1)
z2    dens(aero_1, z2) dens(aero_2, z2)  ...
 .         .             .
 .         .             .
```

where `z` is the height in km, and `dens` are the aerosol mass densities in g/m3. Please make sure to include one column for each of the species `aero_1` to `aero_n` listed after `aerosol_species_file`. For netCDF input it is also possible to specify the unit 'kg kg**-1'; the data are then automatically converted to g/m3.

Some default aerosol mixtures are provided, corresponding to the definitions in Hess et al. (1998). They can simply be invoked by

`aerosol_species_file mixture_name`

where `mixture_name` can be one of the following:

```
continental_clean
continental_average
continental_polluted
urban
maritime_clean
maritime_polluted
maritime_tropical
desert
antarctic
```

`aerosol_species_library`

With this option the *directory* is specified where the optical property files for all aerosols species used in the `aerosol_species_file` are expected: For each species defined in `aerosol_species_file`, *netCDF*-file *species_name*`.nc`, (e.g. `INSO.nc`), which contains the optical properties of the aerosol species, has to be provided. The netcdf format is the one produced by the libRadtran mie tool.

At the libRadtran webpage we provide the OPAC data set (Hess et al. 1998) which can be directly used with uvspec:

```
aerosol_species_library OPAC
```

OPAC contains following aerosol species:

```
INSO  # insoluble
WASO  # water_soluble
SOOT  # soot
SSAM  # sea_salt_accumulation_mode
SSCM  # sea_salt_coarse_mode
MINM  # mineral_nucleation_mode
MIAM  # mineral_accumulation_mode
MICM  # mineral_coarse_mode
MITR  # mineral_transported
SUSO  # sulfate_droplets
```

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. The single scattering albedo defined with this option is constant with wavelength. If you require spectral dependence please use `aerosol_files`. 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 allow wavelength-dependent aerosol optical thickness please use either `aerosol_angstrom` or `aerosol_files`. Comments start with `#`. Empty lines are ignored.

aerosol_visibility

Horizontal visibility in km. Affects the profile according to Shettle (1989) and the optical thickness.

aerosol_vulcan

Aerosol situation above 2 km. Integer.

1   Background aerosols.

2   Moderate vulcanic aerosols.

3   High vulcanic aerosols.

4   Extreme vulcanic 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`. The default albedo is 0.0.

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. An arbitrary wavelength grid may be chosen as the albedo will be interpolated linearely to the wavelength grid used for the radiation calculation. Comments start with #. Empty lines are ignored. A large collection of spectral albedos is available e.g. at http://speclib.jpl.nasa.gov/.

albedo_library

> Albedo libraries are a collection of spectral albedos of different surface types. This option must be used either with `surface_type` or `surface_type_map`, in order to select the specific surface type. There are two possibilities for libraries: the built-in IGBP library or a user defined albedo library.
>
> The built-in library of the International Geosphere Biosphere Programme is selected with `albedo_library IGBP`. The IGBP library contains 20 surface types:
>
> ```
>  1 evergreen_needle_forest
>  2 evergreen_broad_forest
>  3 deciduous_needle_forest
>  4 deciduous_broad_forest
>  5 mixed_forest
>  6 closed_shrubs
>  7 open_shrubs
>  8 woody_savanna
>  9 savanna
> 10 grassland
> 11 wetland
> 12 cropland
> 13 urban
> 14 crop_mosaic
> 15 antarctic_snow
> 16 desert
> 17 ocean_water
> 18 tundra
> 19 fresh_snow
> 20 sea_ice
> ```
>
> Surface types 1 - 17 are defined by the International Geosphere Biosphere Programme (IGBP); additionally there are tundra, fresh_snow, and sea_ice surface types. The spectral albedo of the ground is determined as a function of solar zenith angle, precitable water, and clouds. The spectral resolution equals the grid of the correlated-k Fu/Liou parameterisation. This library originates from the NASA CERES/SARB Surface Properties Project, see Belward and Loveland (1996).
>
> For creating your own albedo library use `albedo_library path`, where `path` is the path of the directory where the albedo data is stored. The files are expected to have the names `albedo_01.dat, albedo_02.dat, ...` If `surface_type 1` is

specified the albedo from `albedo_01.dat` will be used, and so on. Each file is required to 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 linearly to the wavelength grid used for the radiation calculation. Comments start with `#`. Empty lines are ignored. This option is similar to `albedo_file`, except that it offers an easy way to use the option `surface_type_map` in combinition with albedo files.)

albedo_map
> *This option is preliminary and still subject to change (no wavelength dependency yet)!* A possibility to specify a wavelength independent albedo with the help of a *netCDF* file, which is used in combinition with the options `latitude`, `longitude`, and `time`.
>
> > `albedo_map file [variable_name]`
>
> where `file` is the location of the *netCDF* file. With the optional argument the name of the albedo variable in the *netCDF* file can be specified (the default name is `AL`). The albedo must be provided as function of `latitude` and `longitude AL(lat, lon)`, and may also depend on time `AL(time, lat, lon)`. The latitude, longitude, and time grids must be provided as doubles `double lat(lat)`, `double lon(lon)`, and `double time(time)`. libRadtran reads the value at the nearest pixel to the given `latitude` and `longitude`. No spatial interpolation or averaging of the values is done. If a time-dependent albedo is provided, the albedo data nearest to the specified time will be selected (or linear interpolated if `time_interpolate` is switched on).

altitude   Set the bottom level in the model atmosphere provided in `atmosphere_file` to be at the given altitude above sea level (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 above the surface. E.g. to calculate the radiation field 1 km above the surface at a location at 0.73 km above sealevel, one would specify 'altitude 0.73' and 'zout 1.0'. If an altitude is specified which is below the lowest level in the `atmosphere_file`, the atmospheric profiles are extrapolated assuming a constant gradient for temperature and mixing ratios. 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. (Levels 0.73, 1.23, 1.73 ... will be added to the original atmosphere grid and optical properties are devided into the new layers. In order to use interpolated properties use `zout_interpolate`. See

verbose output for details.) Be aware 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').

altitude_map

Specifies an altitude map which is used in combinition with `latitude`, `longitude` in order to select the altitude for the simulation. No interpolation is done between the pixels of the map. The format of the call is:

> altitude_map file [variable_name]

where `file` is the location of the altitude map file. The map is expected to be in *netCDF* format. The file must contain `double lat(lat)`, `double lon(lon)`, and the altitude variable, where `variable_name` is the name of the surface elevation variable in the *netCDF* file. The default name is Z. The altitude variable must be `altitude(lat, lon)`. For format discribtion see also the example map included in libRadtran, data/altitude/ELEVATION_GTOPO_10min.cdf. To use this map in `uvspec`, you may also use `altitude_map GTOPO`. This map has a resolution of 10 arc minutes and the unit of the altitude is meter. Please note that this resolution might not ne adequate for your application. If an altitude in the map is below the lowest level of the `atmosphere_file`, the atmospheric profiles are extrapolated assuming a constant gradient for temperature and mixing ratios.

atmosphere_file

Location of the atmospheric data file. Must have at least three columns containing the altitude, pressure, and temperature. Missing profiles are filled with 0 (e.g., if you did not specify the ozone profile, there will be no ozone absorption!), with exception of the air density which is calculated from pressure and temperature. Other traces gases may be set by `dens_file`. The columns are interpreted as follows:

1. `z(km)`   Altitude above sea level in km

2. `p(mb)`   Pressure in hPa

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. Please note that there is some redundancy: For air as an ideal gas the density can be caluated from pressure and temperature, dens = p / kT. `uvspec` will check if this relation is fulfilled and will stop if it is not. libRadtran provides the six standard atmospheres by Anderson et al. (1986):

```
afglt   Tropical
afglms  Midlatitude Summer
afglmw  Midlatitude Winter
afglss  Subarctic Summer
afglsw  Subarctic Winter
afglus  U.S. Standard
```

If no `atmosphere_file` is defined, libRadtran will automatically select one. If the information `time`, `latitude` and `longitude` are provided in the input file libRadtran will choose from the first 5 files, otherwise it takes the U.S. Standard atmosphere.

`bpdf_tsang_u10`

> Wind speed for ocean BPDF (in m/s) at present only available with `rte_solver MONTECARLO`. The BPDF model has been developed by Tsang et al. (1985). The wind speed is the most important parameter affecting the ocean reflectance matrix. The BPDF model also takes into account shadowing by surface waves. The model has been implemented in a FORTRAN routine by Mishchenko which has been included into libRadtran.

`brdf_ambrals`

> AMBRALS BRDF; specify iso, vol, and geo. Used by MYSTIC and disort2.

`brightness`

> Convert radiances / irradiances to equivalent 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-defined `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:

> `Kato`   Kato et al. (1999), shortwave; based on HITRAN 96. Please note that the bands above 2.5 micrometer are not very reliable which, however, has only little impact on integrated shortwave radiation.

Kato2   Kato et al. (1999), shortwave; optimized version (Seiji Kato, personal communication, 2003); please note that `Kato2` only has 148 subbands (that is, calls to the `rte_solver`) compared to 575 for `Kato` which translates to an increase in computational speed by up to a factor of 4 with only little increase in uncertainty. The absorption data are based on HITRAN 2000. Please note that the bands above 2.5 micrometer are not very reliable which, however, has only little impact on integrated shortwave radiation.

Kato2.96   Kato et al. (1999), shortwave; optimized version (Seiji Kato, personal communication, 2003); similar to `Kato2` but based on HITRAN96. Please note that the bands above 2.5 micrometer are not very reliable which, however, has only little impact on integrated shortwave radiation.

Fu   Fu and Liou (1992/93), shortwave and longwave; fast parameterization, developed for climate models.

AVHRR_KRATZ

     Kratz (1995), AVHRR instrument channels

LOWTRAN   Gas absorption parameterization from LOWTRAN; code adopted from SBDART (Ricchiazzi et al., 1998); please see the section on "Spectral resolution".

SBDART   Identical to LOWTRAN.

If `correlated_k` KATO/KATO2/KATO2.96/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 check which is initiated with **make check**.

cox_and_munk_pcl

     Pigment concentration for Cox and Munk ocean BRDF (in mg/m-3); at present only available with `rte_solver DISORT2`. The number of streams (`nstr`) is automatically increased to 16 if cox_and_munk BRDF is switched on, to avoid numerical problems. The default value is 0.01 mg/m-3. To switch on Cox and Munk BRDF, specify any of the `cox_and_munk` options and define at least `cox_and_munk_u10`.

cox_and_munk_pcl_map

     A possibility to specify pigment concentration (in mg/m3) for the Cox and Munk ocean BRDF with the help of an *netCDF* file, which is used in combinition with options `latitude`, `longitude`, and `time`.

        cox_and_munk_pcl_map file [variable_name]

where `file` is the location of the *netCDF* file. `libRadtran` reads the value at the nearest pixel to the given `latitude` and `longitude`. No spatial interpolation or averaging of the values is done.

The default name of the pigment concentration variable is `chlorophyll`, but can be changed with the optional argument `variable_name`. The pigment concentration must be provided as function of `latitude` and `longitude`, `chlorophyll(lat, lon)`, or additionally may also depend on time `chlorophyll(time, lat, lon)`. If a time-dependent pigment concentration is specified, the pigment concentration will be interpolated according to the option `time_interpolate`. All grids must be provided in the file as `double lat(lat)`, `double lon(lon)`, and `double time(time)`.

`cox_and_munk_sal`

Salinity for Cox and Munk ocean BRDF (in "per mille", 0.1%; this unit is equivalent to the other common units for salinity, ppt - parts per thousand, psu - practical salinity unit); at present only available with `rte_solver DISORT2`. The number of streams (`nstr`) is automatically increased to 16 if cox_and_munk BRDF is switched on, to avoid numerical problems. The default value is 34.3. To switch on Cox and Munk BRDF, specify any of the `cox_and_munk` options and define at least `cox_and_munk_u10`.

`cox_and_munk_sal_map`

Specify ocean salinity (in ppt) for the Cox and Munk ocean BRDF with the help of an *netCDF* file, which is used in combinition with the options `latitude`, `longitude`, and `time`.

>   `cox_and_munk_pcl_map file [variable_name]`

where `file` is the location of the *netCDF* file. `libRadtran` reads the value at the nearest pixel to the given `latitude` and `longitude`. No spatial interpolation or averaging of the values is done.

The expected name of the pigment concentration variable is per default `salinity`, but can be changed with the optional argument `variable_name`. The pigment concentration must be provided as function of `latitude` and `longitude`, `salinity(lat, lon)`, or additionally may also depent on time `salinity(time, lat, lon)`. If a time-dependent salinity is specified, the salinity will be interpolated according to the option `time_interpolate`. All grids must be provided as `double lat(lat)`, `double lon(lon)`, and `double time(time)`.

`cox_and_munk_solar_wind`

Use old definition of wind direction for Monte Carlo simulations. If this switch is set, the wind azimuth is identical to the incoming photon azimuth. Else, the wind azimuth is set by `cox_and_munk_uphi` or is 0. by default.

`cox_and_munk_u10`

Wind speed for Cox and Munk ocean BRDF (in m/s); at present only available with `rte_solver DISORT2`. The wind speed is the most important parameter affecting ocean BRDF. The minimum allowed wind speed is 1 m/s because otherwise the strong specular reflection causes numerical problems. If a lower value is specified, the wind speed is automatically set to 1m/s. Also, the number of streams (`nstr`) is automatically increased to 16 if cox_and_munk BRDF is

switched on, to avoid numerical problems. To switch on Cox and Munk BRDF, specify any of the `cox_and_munk` options and define at least `cox_and_munk_u10`.

`cox_and_munk_u10_map`

Specify wind speed (in m/s) for the Cox and Munk ocean BRDF with the help of an *netCDF* file, which is used in combinition with the options `latitude`, `longitude`, and `time`.

     `cox_and_munk_u10_map file`

where `file` is the location of the *netCDF* file. `libRadtran` reads the value at the nearest pixel to the given `latitude` and `longitude`. No spatial interpolation or averaging of the values is done.

The file must contain the elements of the wind vector `U10` and `V10`. These must be specified as functions of `latitude` and `longitude` `U10(lat, lon)`, `V10(lat, lon)`, or additionally may also depent on time `U10(time, lat, lon)`, `V10(time, lat, lon)`. If the variable time is present in the file, the wind speed will be interpolated according to the option `time_interpolate`. All grids must be provided as `double lat(lat)`, `double lon(lon)`, and `double time(time)`.

`cox_and_munk_uphi`

Wind direction for Cox and Munk ocean BRDF. Default value is 0 degrees, which is wind from the South. 90 degrees corresponds to wind from the West, etc. (Honestly, this was never truly validated. It could possibly be that 0 is wind from the North, 90 is wind from the East, etc.)

`crs_file` May be used to specify cross sections of O3, NO2, BRO, OCLO, or HCHO to be used instead 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 species 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. The default is `../data/`.

`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 distance), that is, no correction is applied to the extraterrestrial irradiance `solar_file`. Alternatively `time` may be used for that purpose.

`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 species column [unit]`
>
> where species is one of O3, O2, H2O, CO2, NO2, BRO, OCLO, or HCHO, see also `dens_file`. The second argument is the total column value, and the optional third argument is the unit, in which the column is given. The unit can be DU (Dobson units) or CM_2 (molecules/cm2). The default units are DU for O3, and CM_2 for all other gases. It is possible to have several `dens_column` commands in the input file (maximum one per species).
>
> > `dens_column NO2 1.2 DU`

dens_file

> Specify density profiles (or matrix, see below) of various traces gases to be included in the radiative transfer calculation. The entry of the input file looks like this:
>
> > `dens_file gas_species [unit] filename`
>
> At the moment following gas_species are included: ozone (O3), nitrogen dioxide (NO2), water vapor (H2O), bromine oxide (BRO), chlorine dioxide (OCLO), formaldehyde (HCHO), and carbon dioxide (CO2). The gas species is identified by their abbrevations given in the parenthesis above. Unit is an optional argument to define the unit of the density. The profiles can be given in particles per cm3 (cm-3), in particles per m3 (m-3), as volume mixing ratio (vmr), as mass mixing ratio (mmr), or as relative humidity (only for water). The default for unit is cm-3. The model expects a density file with two columns:
>
> 1. `z(km)`   Altitude above sea level in km.
>
> 2. `density of trace gas [in the specified unit]`
>   > The density of the trace gase
>
> The altitude grid may be different from the `atmosphere file`. All densities inside the range of the dens_file are replaced. For all other altitudes the values from the atmosphere_file are used. If the density is specified as -1 at a level, the value from atmosphere_file is used.
>
> 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 specifying a `dens_file` in the following format:
>
> ```
>     0.0       SZA1      SZA2 ...
>     z(1)    dens(1,1)    ...
>     z(2)       .          .
>      .         .          .
> ```
>
> where z(i) are the altitude levels above sea level in km, SZA is the solar zenith angle in degrees, and dens is the density [in the specified unit] of the trace gases as 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 can only be used in connection with `rte_solver sdisort`!

earth_radius

Specify the earth radius in km. This is needed by all solvers in spherical geometry, e.g. mystic in combination with option `mc_spherical`. The default value is 6370 km.

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   Provide output in flexstor format. Must 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-define `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   Include information on some of the input parameters in the output. May not be combined with `flexstor`. Please note that the information provided is rather incomplete because this option was introduced quite early and was never updated. For a more complete information please use the `verbose` option.

heating_rate

Calculation of heating rates. Output is only provided at altitudes specified by `zout`. To get heating rate profiles a number of altitudes must thus be specified. Heating rates is the change of temperature with time in units of K/day. For spectral calculations the default output is a matrix:

```
0.0        zout1          zout2 ...
lambda1    heating_rates  ...
lambda2        .
   .           .
   .           .
```

For integrated calculations (`output sum` or `output integrate`) the default output is in two columns with column 1 being the altitude and column 2 the heating rates. The output of `heating_rate` can also be specified with the `output_user` option. Note that heating rates are only well-behaved up to altitudes for which the respective correlated-k options are valid. E.g. about 60 km for `Fu` and about 80 km for `Kato`, `Kato2`, `Kato2.96`, and `LOWTRAN`. Attention: For spectral calculations, the extraterrestrial spectrum is assumed to be in mW/(m2 nm).

Two different methods are implemented to calculate the heating rate, which can be selected with an optional keyword:

> `heating_rate [method]`

where `method` may be either `layer_cd` (heating rates are derived from centered differences of the flux, this is the default method) or `local` (heating rates are derived from the actinic flux). Attention: `heating_rate local` introduces new levels into the profile which slightly affects the model output.

ic_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 `ic_file`, etc. Please note that, if both `wc_cloudcover` and `ic_cloudcover` are set, both must be equal.

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 micrometer. The ice water content and effectice 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. 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 layer properties. Before version 1.4 the default was level properties: The optical depth of a layer was calculated using information from the upper and lower levels defining the layer, see `ic_layer` and `ic_level`. To switch to the old behaviour, use `ic_level`. See the section about ice clouds for a realistic example how the contents of the `ic_file` are converted to optical 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 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 for `rte_solver disort2` it makes good sense to make the number of moments larger than `nstr` because all moments are used in the calculation. 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. Please note that the uppermost line of the `ic_files` denotes simply the top altitude of the uppermost layer. The optical properties of this line are consequently ignored. There are two options for this line: either an optical property file with zero optical thickness is specified or "NULL" instead.

ic_fu_tau

Specify if the Fu (1996) optical properties are delta-scaled or not. With `ic_fu_tau scaled` delta-scaling is switched on, with `ic_fu_tau unscaled` it is switched off. The default is without delta-scaling. Please note that this was changed on July 22, 2008: Before, delta-scaling was switched on by default which might have caused some confusion, because irradiance calculations were not consistent with the other ice cloud parameterizations implemented in uvspec. Using the Fu (1996) parameterization in combination with one of `ic_set_tau/tau550/gg/ssa` or `ic_scale_gg/ssa` you now get consistent results with all other ice cloud parameterizations.

ic_fu_reff

If `ic_fu_reff fu` is specified, the parameterization uses the original definition of the effective radius as specified in Fu (1996/98). By default it uses the same definition of the effective radius as the Key et al. (2002), Yang et al. (2000) and Baum et al. (2005) parameterizations; see discussion of `ic_properties`.

ic_habit   Ice crystal habit for the Yang et al. (2000) and Key et al. (2002) parameterizations, see also `ic_properties key/yang`. May be one of solid-column, hollow-column, rough-aggregate, rosette-4, rosette-6, plate, droxtal, and spheroid. Please note that this parameterization is only valid for a restricted size range, depending on the habit (see table 1 in Key et al. (2002)). Also, some of the habits are only available for wavelengths below 5 micrometer (rosette-4) while others are only available for wavelengths larger than 3 micrometer (droxtal, spheroid).

ic_ipa_files

A two-column file, defining ice cloud property files (see `ic_file`) in the first column and the correspoding 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   Interpret ice cloud properties as layer properties (this is the default behaviour since version 1.4; see also ic_file). Cloud properties are assumed to be constant within each layer. The layer reaches from the level where the properties are defined in the `ic_file` to the level above. For example, the following lines

```
#      z     IWC    R_eff
#     (km)  (g/m^3) (um)
      4.000   0.0    0.0
      3.000   1.0   10.0
```

define a cloud in the layer between 3 and 4 km with sharp boundaries.

ic_level   Interpret cloud properties as level properties (this was the default behaviour before version 1.4; see also ic_file). If `ic_level` is defined, a `ic_file` would be interpreted as follows:

```
#      z     IWC    R_eff
#     (km)  (g/m^3) (um)
      5.000   0      0
      4.000   0.2   12.0
      3.000   0.1   10.0
      2.000   0.1    8.0
```

The value 0.2 g/m3 refers to altitude 4.0km, as e.g. in a radiosonde profile. The properties of each layer are calculated as average over the adjacent levels. E.g. the single scattering properties for the model layer between 3 and 4km are obtained by averaging over the two levels 3km and 4km. To allow easy definition of sharp cloud boundaries, clouds are only formed if both liquid water contents above and below the respective layer are larger than 0. Hence, in the above example, the layers between 2 and 3 as well as between 3 and 4km are cloudy while those between 1 and 2km and between 4 and 5km are not.

ic_no_scattering
          Switch off scattering by ice clouds.

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), whereas `uvspec` uses non delta-scaled properties unless the option `ic_fu_ tau scaled` is specified. By default the parameterization by Fu

(1996) is treated consistently with all other ice cloud parameterizations. For wavelengths up to 4 micrometer Fu (1996) is used while for wavelengths larger than 4 micrometer 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. Also, the wavelength dependence in the solar and infrared parts is treated differently: In the solar part (Fu, 1996) the optical properties are defined for wavelength bands - hence they are assumed constant within each band. In the infrared (Fu et al. 1998) they are defined at certain wavelengths and linearely interpolated in between. If you use this option, please see also the discussion of `ic_fu_tau` and `ic_fu_reff`. The allowed range for the effective radius is from 9.315 - 65.120 micrometer.

echam4   Use the simple two-band parameterization of the ECHAM4 climate model, described in Roeckner et al. (1996); this is probably only meaningful if you want to compare your results with ECHAM4, the two bands are 0.2 - 0.68 micrometer and 0.68 - 4.0 micrometer. Within the two ECHAM4 bands, the optical properties are assumed constant.

key   Parameterization by 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 wavelength region from 0.2 to 5.0 micrometer and is available for the following habits: solid-column, hollow-column, aggregate, rosette-4, rosette-6, and plate.

yang   Parameterization similar to Key et al. (2002) but based on more recent single scattering calculations. Below 3.4 micrometer it actually equals the Key et al. (2002) parameterization while from 3.4 - 100 micrometer new coefficients have been calculated with much higher wavelength resolution and better accuracy. Hence, yang should give a reasonably consistent approximation from 0.2 - 100 micrometer, suitable for spectrally resolved calculations of radiance and irradiance. The covered range for the effective radius depends on the `ic_habit`. (In micrometer: solid columns [5.96, 84.22], hollow columns [4.97, 70.24], rough aggregate [3.55, 108.10],rosettes-4 [2.77, 45.30], rosettes-6 [2.85, 46.01], plates [4.87, 48.18], dendrites [0.45, 1.88], droxtals [9.48, 293.32], spheroids [6.58, 203.39]).

baum   Use ice cloud parameterization from Baum et al. (2005), http://www.ssec.wisc.edu/~baum/Cirrus/IceCloudModels.html.

In combination with the radtiative transfer solvers `disort2` and `mystic`, accurate phase functions are used.

baum_hufit

Similar to the option `baum` but here the phase function is parameterized by 128 Legendre coefficients, calculated with the delta-fit method from Hu et al. (2000). This parameterization covers the region from 0.4 to 2.2 micrometer. If high accuracy is needed e.g. in the vicinity of the halo, the forward peak, or the backscatter peak, `ic_properties baum` is recommended.

mie   Use pre-calculated Mie tables; useful for `correlated_k`; the tables are expected in `data_files_path`/correlated_k/.../
             For spectral or pseudo-spectral (`correlated_k sbdart`) calculations, a set of pre-calculated tables is also available; the wavelength grid points of these data has been carefully selected such that the extinction cross section, single scattering albedo, and the asymmetry parameter are accurate to 1% (compared to the fully-resolved Mie calculation) for all wavelengths between 250nm and 100 micrometer. For spectral or pseudo-spectral calculations `ic_properties_interpolate` has to be defined explicitly to initiate the interpolation of the optical properties to the internal wavelength grid. Please note that this option may be extremely memory-consuming because for each internal wavelength a full set of Legendre moments of the phase function is stored (up to several thousands). The Mie tables are not part of the standard distribution (because of their large size) but they are freely available 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. Hence, `ic_properties mie` is usually not representative of natural ice clouds.

filename   Read optical properties from specified filename; file format is as produced by the mie-tool of the `libRadtran-tool` box (see `output_user cloudprp`) or by Frank Evans' `cloudprp`; for each of the internal (computational) wavelengths, a separate file is expected (this option is subject to change).

The default property is `fu`.

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), and Baum et al. (2005) use the general definition

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

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 `ic_fu_reff fu` and $1.299 \cdot 20\mu m = 26\mu m$ with `ic_properties yang` would give comparable results for hexagonal columns. To use the original definition of the effective radius by Fu et al. (1996,1998) use `ic_fu_reff fu`!

`ic_properties_interpolate`

Interpolate ice cloud optical properties over wavelength; useful for precalculated optical property files defined with `ic_properties`. Please note that this option may be extremely memory-consuming because for each internal wavelength a full set of Legendre moments of the phase function is stored (up to several thousands).

`ic_saturate`

With this option, the relative humidity inside ice clouds can easily be adjusted to a user-defined value, e.g. saturated with respect to water. This option has one mandatory and one optional argument:

`ic_saturate switch [relative_humidity]`

where `switch` is `on`, `off`, or `ipa`. The second optional argument determines the relative humidity in % (with respect to water!) inside the cloud. The default value is 100. ipa is only relevant for independent column calculations.

Details for independent column simulations: Using `switch on`, the air in *all* columns will be saturated, if there is a cloud in at least one of the columns (this option should be used for stratiform clouds). Using `switch ipa`, only cloudy columns are affected (this option should be used for convective cloud fields.)

`ic_scale_gg`

Scale the ice cloud asymmetry factor for all wavelengths and altitudes with a float between 0.0 and 1.0. If you use this option in combination with the ice cloud properties by Fu (1996), please make sure that you understand the explanation of `ic_fu_tau`.

`ic_scale_ssa`

Scale the ice cloud single scattering albedo for all wavelengths and altitudes with a float between 0.0 and 1.0. If you use this option in combination with

the ice cloud properties by Fu (1996), please make sure that you understand the explanation of `ic_fu_tau`.

ic_set_gg

Set the ice cloud asymmetry factor for all wavelengths and altitudes to a float between -1.0 and 1.0. If you use this option in combination with the ice cloud properties by Fu (1996), please make sure that you understand the explanation of `ic_fu_tau`.

ic_set_ssa

Set the ice cloud single scattering albedo for all wavelengths and altitudes to a value between 0.0 and 1.0. If you use this option in combination with the ice cloud properties by Fu (1996), please make sure that you understand the explanation of `ic_fu_tau`.

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. If you use this option in combination with the ice cloud properties by Fu (1996), please make sure that you understand the explanation of `ic_fu_tau`.

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 `wavelength` does contain 550nm. If you use this option in combination with the ice cloud properties by Fu (1996), please make sure that you understand the explanation of `ic_fu_tau`.

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 3D grid box and write it to file `mc_basename.abs.spc` in the following format:

        lambda ix iy iz absorption

Only meaningful with `rte_solver montecarlo`. An optional argument `W_per_m2_and_dz` (default), `W_per_m3`, or `K_per_day` may be specified which converts the result e.g. to heating rates. This option is only available for forward calculations. For backward please use `mc_backward_output abs`.

mc_actinic

Calculate MYSTIC actinic flux for each 3D grid box by dividing the absorbed energy by the absorption coeffcient; 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`. This option is only available for forward calculations. For backward please use `mc_backward_output f`.

`mc_albedo_file`

Define a MYSTIC 2D albedo input file (default: albedo2D.dat). The expected format of the elevation file is:

```
Nx  Ny  dx  dy
ix  iy  albedo
```

where Nx and Ny are the number of grid boxes in x- and y-direction, dx and dy are the size of the pixel in km. In the second and the following lines the indices in x- and y-direction and the albedo of the pixel are specified.

`mc_ambrals_file`

Define a MYSTIC 2D AMBRALS BRDF input file. AMBRALS is the Algorithm for Modeling[MODIS] Bidirectional Reflectance Anisotropies of the Land Surface by Wanner et al. (1995, 1997). See also http://www-modis.bu.edu/brdf/models.html. The file consists of a header followed by the three parameters iso, vol, and geo for each pixel.

`mc_azimuth_old`

Use old MYSTIC azimuth convention (0 degree = looking from the direction of the sun; 180 degree = looking into the direction of the sun; that is, exactly opposite to the disort convention). The MYSTIC azimuth was changed March 1, 2004 - hence this option was introduced for compatibility reasons.

`mc_backward`

Backward tracing of photons. `mc_backward` takes either zero, two or four coordinates:

```
mc_backward [ix_start iy_start] [ix_end iy_end]
```

where `ix_start`, `iy_start` is the index of the sample pixel to be calculated or the pixel area from `ix_start` to `ix_end` and `iy_start` to `iy_end`. All x-indices must be in the range of 0 ... (`Nx-1`) and y-indices the range of 0 ... (`Ny-1`). If no coordinates are specified, all sample pixels will be calculated. `mc_backward` computes radiances and downward diffuse irradiances. If a different quantity is required, please use `mc_backward_output`.

`mc_backward_output`

Specify quantity to be calculated using backward Monte Carlo. So far the following options are implemented:

edir   direct horizontal irradiance

edn   diffuse downward irradiance (default)

eup   diffuse upward irradiance

f   actinic flux

> abs   absorption
>
> emis   emission
>
> heat   heating rates, that is absorption + emission
>
> For `abs`, `emis`, `heat` an optional argument `W_per_m2_and_dz` (default), `W_per_m3`, or `K_per_day` may be specified which converts the result e.g. to heating rates.

mc_backward_writeallpixels
> If set, write all pixels to the output files; otherwise, only those are written which are actually calculated.

mc_backward_writeback
> If set, the distribution of photons contributing to the result is written to a file with extension .bac

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

mc_elevation_file
> Define a MYSTIC 2D elevation input file (default: elevation2D.dat). The expected format of the elevation file is:
>
> ```
> Nx  Ny  dx  dy
> ix  iy  elevation
> ```
>
> where Nx and Ny are the number of grid boxes in x- and y-direction, dx and dy are the size of the grid boxes in km. In the second and the following lines the indices in x- and y-direction and the elevation in km at this point are specified.
>
> Attention: While the other files refer to grid boxes, the elevation is defined at grid points. It has to be this way because each "elevation pixel" contains a surface which is defined by the four cornes of the pixel. If the grid covers an area of 200x200 km^2 and the pixel sizes are dx = 1km and dy = 1km, the elevation has to be defined at 201x201 points (ix = 1 .. Nx, iy = 1 .. Ny).

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 which makes this option very fast. Increasing `mc_photons` will therefore not affect the result. Only meaningful with `rte_solver montecarlo` and `source thermal`. An optional argument `W_per_m2_and_dz` (default), `W_per_m3`, or `K_per_day` may be specified which converts the result e.g. to heating rates. This option is only available for forward calculations. For backward please use `mc_backward_output emis`.

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

mc_heating

Calculate MYSTIC heating rates for each grid box; may need a considerable amount of memory, depending on the 3D cloud grid. Only meaningful with `rte_solver montecarlo`. `mc_heating` is identical to `mc_absorption` for `source solar`. For `source thermal`, however, the emission of a photon in a grid box is counted as cooling. An optional argument `W_per_m2_and_dz` (default), `W_per_m3`, or `K_per_day` may be specified which converts the result e.g. to heating rates. This option is only available for forward calculations. For backward please use `mc_backward_output heat`.

mc_hiddenline

Switch on hidden line removal for the MYSTIC online visualization. Good for topography, not so good for clouds because the latter look much more realistic when the hidden layers are plotted in transparent mode.

mc_icloud_file

Define a MYSTIC 3D ice cloud input file. The format of the 3D ice cloud file is identical to the 3d water cloud file (see `mc_wcloud_file`).

mc_ipa   Run MYSTIC in independent pixel mode. Only meaningful with `rte_solver montecarlo`.

mc_momcld_file

Does not exist anymore as 3D cloud properties are now handled with wc_properties.

mc_panorama Simulate a panorama. Only works with

mc_sensorposition and mc_backward. With mc_backward, the user decides what area of the full 4pi globe is simulated. The first dimension gives theta and goes linearly from 0 (zenith) to 180 (nadir), while the second dimension gives phi and goes from 0+epsilon to 360-epsilon. 0 means north/south/east/west here.

mc_pcld_file

Cloud phase function file for MYSTIC. Only meaningful with `rte_solver montecarlo`. *This is an experimental option.*

mc_photons

Total number of photons to be traced by the Monte Carlo solver, MYSTIC. Only meaningful with `rte_solver montecarlo`.

mc_photons_file

Distribution of photons over wavelength bands; to be used with `correlated_k`; 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 `rte_solver montecarlo`.

mc_polarisation

Switch on polarization for `rte_solver montecarlo`.

`mc_radial_pathlength`

        I3RC case 7, laser beam experiment. Radiances are sampled in radial and pathlength elements. Specify the number of radial (Nr) and time (Nt) intervals. The radius increment is calculated from Nr and the domain size. The time interval may be specified with `mc_radial_pathlength_dt`.

`mc_radial_pathlength_dt`

        Specify time increment for `mc_radial_pathlength`. Time is converted to pathlength assuming a speed of light of 3E8 m/s

`mc_refraction`

        Enable refraction for `rte_solver montecarlo`. Works only with in 1D spherical geometry (with option `mc_spherical`).

`mc_rpv_file`

        Define a MYSTIC 2D RPV BRDF input file.

`mc_rpv_type`

        File containing wavelength-dependent RPVs to be used in combination with `mc_rpv_file`.

`mc_sample_grid`

        Sampling grid size (Nx Ny [dx dy]) for MYSTIC. Only meaningful with `rte_solver montecarlo`.

`mc_randomseed`

        Provide your own random seed (positive integer) for the random number generator. Usually a random seed is determined from current time plus process id. This option is useful to re-run a simulation for debugging.

`mc_readrandomseed`

        Read from file the random seed (positive integer) for the random number generator. Usually a random seed is determined from current time plus process id. This option is useful to re-run a simulation for debugging.

`mc_readrandomstatus`

        Read from file the random status for the random number generator. This option is useful to re-run a simulation for debugging, especially if the buggy photon appears only late in a long simulation. This option automatically toggles on `mc_readrandomseed`.

`mc_sensordirection`

        Define viewing direction of an irradiance sensor in Monte Carlo backward mode. Has been introduced for irradiance calculations in topography and might not properly work with all options. For radiance use the usual `umu` and `phi`.

`mc_sensorposition`

        Define the position of a sensor. Has been introduced for irradiance calculations in topography and might not properly work with all options.

mc_spherical

Spherical geometry in MYSTIC. Works only in "1D" - `wcloud_file` and `icloud_file` are not yet considered. If `mc_spherical` is selected `mc_backward` is switched on automatically.

mc_std   Calculate standard deviation of the average.

mc_surface_reflectalways

Usually, a photon is either absorbed or reflected at the surface, with a probability defined by the surface albedo. If `mc_surface_reflectalways` is specified, each photon is reflected and the albedo is considered by reducing the photon weight. In case of BRDF, `mc_surface_reflectalways` is switched on automatically because the other method is no longer implemented for non-Lambertian BRDFs, due to implementation and numerical problems. For small albedos, the computational time is increased if `mc_surface_reflectalways` is used; however, the accuracy of the upward radiance (reflected by the surface) is increased considerably. In case of clouds, however, computational time might be increased considerably without gaining accuracy.

mc_surfaceparallel

Calculate irradiance parallel to the surface instead of horizonal irradiance. This option is obviously only interesting for topograpy and only for calculations at the surface. For other levels the option is ignored.

mc_temperature_file

Define a MYSTIC 2D temperature input file. The expected format of the temperature file is:

```
Nx  Ny  dx  dy
ix  iy  temperature
```

where Nx and Ny are the number of grid boxes in x- and y-direction, dx and dy are the size of the grid boxes in km. In the second and the following lines the indices in x- and y-direction and the temperature of the pixel are specified.

mc_truncate

Truncate phase function at the specified polar angle mu. USE ONLY IF YOU REALLY KNOW WHAT YOU ARE DOING!

mc_visualize

Switch on OpenGL visualization for MYSTIC.

mc_vroom   Variance Reduction Optimal Options Method.

mc_wcloud_file

Define a MYSTIC 3D water cloud input file. The expected format of the cloud file is:

```
Nx  Ny  Nz  gspec flag
dx  dy  z(1) z(2) ... z(n)
ix  iy  iz  ext  g     ssa  (if flag == 1)
ix  iy  iz  ext  reff       (if flag == 2)
```

```
ix iy iz LWC reff        (if flag == 3)
```

where Nx, Ny and Nz are the number of grid boxes in x, y, and z-direction. gspec and wspec are parameters, which determin the format of the 3rd and following lines. In the second line dx and dy are the sizes of the boxes in x- and y-direction in km. In the third and following lines the indices ix, iy, and iz specify cloudy pixels. The optical properties of the cloud, are given by the other parameters in the line, where ext is the extinction coefficient [1/km], g the assymetry parameter, reff the effective radius [micrometer], and ssa the single scattering albedo. The conversion from microphysical to optical properties is defined by `wc_properties` (identical to the 1D case). For more information please consult the MYSTIC documentation, `doc/README.MC`.

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 want to specify the optical depth directly (e.g. for a model intercomparison) or for a line-by-line calculation. If a spectral `molecular_tau_file` is specified, the wavelength grid defined there is used as internal wavelength grid for the radiative transfer calculation, if not defined otherwise with `transmittance_wl_file`. `molecular_tau_file` can be either of the following three 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, ASCII:

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.

Spectral, netcdf:

netcdf is a common platform independent format; the description, a library to read and write netcdf including some tools to generate netcdf is available at http://www.unidata.ucar.edu/packages/netcdf/. A `molecular_tau_file` must obey certain rules; an example is available at the libRadtran homepage, '`UVSPEC.O2A.afglms.cdf`', a line-by-line spectrum of the oxygen A-Band around 760nm, calculated for the mid-latitude summer atmosphere by Anderson et al. (1986). The advantage of netcdf compared to ASCII is that it is much faster to read, and that the file is a self-contained, including data and a description of the variables and arrays. It is therefore particularly useful for line-by-line calculations where usually many spectral data points are involved.

Comments start with #. Empty lines are ignored.

n2o_mixing_ratio

>   The mixing ratio of N2O in ppm (default: 0.28 ppm).

no_absorption

>   Switch all (molecular, aerosol, cloud, and ice cloud) absorption off. Please
>   note that this option simply sets the absorption optical thickness to 0. If used
>   together with xxx_set_tau this might be a bit confusing but probably the most
>   logical way. E.g. when using `aerosol_default` and `aerosol_set_tau 1`, the
>   aerosol optical thickness is set to 1, with 0.940539 scattering and 0.059461
>   absorption. If `no_absorption` is added, the absorption optical thickness is
>   set to 0 while the scattering optical thickness is preserved at 0.940539 (even
>   though 1 was specified by the user). We find this the most logical solution
>   of the problem because by switching `no_absorption` off and on one tests the
>   effect of the absorber in an isolated way, rather than mixing absorption and
>   scattering. The same is true for water and ice clouds. Note, that thermal
>   emission of molecules is also switched off.

no_molecular_absorption

>   Switch off molecular 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 `rte_solver sdisort` refraction may be included by specifying `nrefrac`.
If refraction is included also set parameter `refraction_file`.

>   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. (For `rte_solver disort` and `disort2` only
            even nstr are possible.)

o2_mixing_ratio
            The mixing ratio of O2 in ppm; scale the profile so that the mixing ratio at the
            user-defined `altitude` assumes the specified value.

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.

optimize_fortran
            When this option is activated, the minimum dimensions of the FORTRAN
            arrays for the specific input conditions are determined and written to `stderr`.

output   Output processing. Choose between the following options:

            sum   Sum output over wavelength.   Useful in combination with
                        the   `correlated_k`   option   (Kato,   Kato2,   Kato2.96,   Fu,
                        AVHRR_KRATZ).

            integrate
                        Integrate output over wavelength for solar and over wavenumber
                        for thermal simulations.   Useful for spectral calculations and
                        `correlated_k` LOWTRAN.

            per_nm   Output is given in W/(m2 nm) or mW/(m2 nm) (W or mW is
                        determined by the extraterrestrial spectrum.)

            per_cm-1   Output is given in W/(m2 cm-1) or mW/(m2 cm-1).

            per_ck_band
                        Output is given in W/m2 or mW/m2 per correlated-k band. (This
                        option can not be used for spectral calculations and `correlated_k`
                        LOWTRAN in the solar range.)

            none   No processing - output spectral information (default).

output_user
            User defined output. This option is case sensitive. Here the user may specify
            the columns desired for output. Default output is the wavelength lambda, edir,
            edn, eup, uavgdir, uavgdn, uavgup for disort, sdisort, and spsdisort, whereas
            the default for twostr is lambda, edir, edn, eup, uavg. The lines containing
            radiances and the output of `rte_solver polradtran` are not affected.

            lambda   Wavelength in nm.

wavenumber
: Wave number in cm-1.

sza
: solar zenith angle

zout
: Output altitude in km.

edir, eglo, edn, eup, enet, esum
: Direct, global, diffuse downward, and diffuse upward irradiance. Net is global - upward, sum is global + upward.

uu
: Radiances: uu(umu(0),phi(0)) ... uu(umu(0),phi(m)) ... uu(umu(n),phi(0)) ... uu(umu(n),phi(m))

fdir, fglo, fdn, fup, f
: Direct, global, diffuse downward, diffuse upward, and total actinic flux.

uavgdir, uavgglo, uavgdn, uavgup, uavg
: Direct, global, diffuse downward, diffuse upward, and total diffuse mean intensity (= actinic flux / 4 pi).

albedo
: Albedo.

heat
: Heating rate in K/day.

It is also possible to gain some information about the atmosphere and the clouds:

p
: pressure [hPa], ,

T, T_d
: temperature [K], dewpoint temperature [K]

T_sur
: surface temperature [K]

theta
: potential temperature [K]

theta_e
: equivalent potential temperature [K]

n_xxx
: number density of the gas xxx [cm-3]

rho_xxx
: mass density of the gas xxx [kg/m3]

mmr_xxx
: mass mixing ratio of the gas xxx [kg/kg]

vmr_xxx
: volume mixing ratio of the gas xxx [m3/m3]

rh
: relative humidity over water [percent]

rh_ice
: relative humidity over ice [percent]

c_p
: specific heat capacity of the air (humidity and temperature dependent)

CLWC
: cloud liquid water content [kg/kg]

CLWD
: cloud liquid water density [g/m3]

CIWC
: cloud ice water content [kg/kg]

CIWD
: cloud ice water density [g/m3]

TCC   total cloud cover [0-1]

where xxx is one of AIR, O3, O2, H2O, CO2, NO2, BRO, OCLO, or HCHO.

ozone_column

Obsolete, use `dens_column` instead. Set the ozone 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 `ozone_column` is not specified, the ozone column is not scaled.

phi   Azimuth output angles (in degrees) in increasing order. The radiance is output at `phi` and `umu`.

```
Sensor in the North (looking South)   0 deg
Sensor in the East  (looking West)   90 deg
Sensor in the South (looking North) 180 deg
Sensor in the West  (looking East)  270 deg
```

For all one-dimensional solvers the absolute azimuth does not matter, but only the relative azimuth `phi-phi0`.

phi0   Azimuth angle of the sun (0 to 360 degrees).

```
Sun in the South    0 degrees
Sun in the West    90 degrees
Sun in the North  180 degrees
Sun in the East   270 degrees
```

For all one-dimensional solvers the absolute azimuth does not matter, but only the relative azimuth `phi-phi0`.

polradtran_aziorder

Order of Fourier azimuth series: 0 (default for irradiance) is azimuthally symmetric case. For radiance computation a higher order is required, thus the default for radiances is 4. This option is only relevant for `rte_solver polradtran`.

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. This option is only relevant for `rte_solver polradtran`.

polradtran_nstokes

Number of Stokes parameters

1   for I (no polarization, default)

2   for I,Q,U (Since V is very small in the atmosphere, it makes sense to compute only I,Q,U. This saves computation time and memory).

3   for I,Q,U,V

Default is 1.

polradtran_quad_type

Type of quadrature used:

G   gaussian

D   double gaussian,

L   Lobatto

E (default)
            extra-angle(s), this must be used of `polradtran` is used in combi-
            nation with `umu`. Will internally use Gaussian scheme (`G`). See also
            radtran documentation.

Default E. This option is only relevant for `rte_solver polradtran`.

polradtran_src_code
            Radiation sources included:

0   none

1   solar

2   thermal

3   both

Default 1. This option is only relevant for `rte_solver polradtran`.

pressure   The surface pressure (at the user-defined `altitude`) in hPa. The pressure profile
            as well as air, O2 and CO2 density profiles are scaled accordingly.

pressure_out
            Specify the output levels in pressure coordinates. The syntax is

                  `pressure_out p1 p2 ...`

            where 'p1 p2 ...' are the output levels in hPa. The pressure output levels must
            be sorted in decreasing order. Output pressure levels must be within the range
            defined in the `atmosphere_file`. You can also use `toa` for `top of atmosphere`
            and `sur` for `surface altitude` and `cpt` for `cold point tropopause`.

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

quiet   If specified, informative messages are turned off. See also `verbose`.

radiosonde
            This option allows to change the temperature and pressure profile, and option-
            ally to specify one or more density profiles. The entry in the input file looks
            like this:

                  `radiosonde filename [gas_species] [unit] ...`

            Currently the following gas_species are included: ozone (O3), nitrogen dioxide
            (NO2), water vapor (H2O), bromine oxide (BRO), chlorine dioxide (OCLO),
            formaldehyde (HCHO), and carbon dioxide (CO2). Each gas species is identified
            by its abbrevations given in parentheses above. Unit is an optional argument
            to defines the unit of the density. The profiles can be given in particles per cm3

(CM-3), in particles per m3 (M-3), as volume mixing ratio (VMR), as mass mixing ratio in kg/kg (MMR), or as relative humidity (RH) (only for water). The default unit is RH for water vapour, MMR for ozone, and CM-3 for all other gases. The radiosonde file must have (2 + number of gases) columns:

1   pressure in hPa

2   temperature in Kelvin

3, 4 ...   density of trace gas in the specified unit

A new z-grid will be calculated, starting at `altitude` and assuming a linear temperature variation between levels. The air density will be recalculated according to the ideal gas law, and the density of the well mixed gases O2 and CO2 will be scaled accordingly. The atmospheric data above the radiosonde data is taken from the `atmosphere_file` level by level, starting at the first pressure level above the radiosonde data. The z-grid of the `atmosphere file` in this height region is shifted accordingly. Also if the density in the radiosonde file is specified as -1 at a level, the value from the `atmosphere_file` is used. Possible calls are

> `radiosonde ../examples/radiosonde.dat`

just in order to change the temperature and pressure profile, or

> `radiosonde ../examples/radiosonde2.dat H2O RH O3 MMR NO2`

where water vapour density will be given as relative humidity, ozone as mass mixing ratio, and NO2 in cm-3 (default).

radiosonde_levels_only

> The atmosphere considered in the simulation has the same height range as the data in the `radiosonde`-file. No further levels are added above those. This option has only an effect in combination with `radiosonde`.

rayleigh_crs

> Choose between the following Rayleigh scattering cross sections.
>
> `Bodhaine`   Bodhaine et al (1999) Rayleigh scattering cross section.
>
> `Nicolet`   Nicolet (1984) Rayleigh scattering cross section.
>
> `Penndorf`   Penndorf (1957) Rayleigh scattering cross section.
>
> Bodhaine et al. is default.

rayleigh_depol

> Rayleigh depolarization factor; the Rayleigh scattering phase function is p(mu) = a + b * mu**2 where a = 1.5*(1+depol)/(2+depol) and b = 1.5*(1-depol)/(2+depol). By default the depolarization is calculated using the expressions from Bodhaine et al. (1999).

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). The optical thickness profile may be either monochromatic or spectral. The format is exactly the same as for `molecular_tau_file`.

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`.

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 in 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 linearely 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! Optionally, a fifth column with a constant scaling factor may be defined which, however, is only used by `rte_solver disort2`.

rpv_k   Constant RPV k, 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`.

rpv_sigma
Constant RPV sigma, to be used for snow (Deguenther and Meerkoetter, 2000). A wavelength dependent sigma is not yet available.

rpv_t1 Constant RPV t1, to be used for snow (Deguenther and Meerkoetter, 2000). A wavelength dependent sigma is not yet available.

rpv_t2 Constant RPV t2, to be used for snow (Deguenther and Meerkoetter, 2000). A wavelength dependent sigma is not yet available.

rpv_scale
Apply a constant scaling factor for the RPV BRDF. Required e.g. if the the albedo should be set to a certain value. This factor is only used by `rte_solver` `disort2`.

rpv_library
The rpv libraries are collections of spectral BRDFs of different surface types, This option must be used either with `surface_type` or `surface_type_map`, in order to select the specific surface type.

For using a `rpv_library` write

        rpv_library library_path

where `library_path` is the path of the directory, where the BRDF data is stored. The files are expected to have the names `IGBP.01.rpv`, `IGBP.02.rpv`, ... If `surface_type 1` is specified the BRDF from `IGBP.01.rpv` will be used, and so on. Each file must have the structure like an `rpv_file`. (This option is quite the same as `rpv_file`, except that it offers you an easy way to use the option `surface_type_map` in combinition with your `rpv_files`.)

rte_solver
Set the radiative transfer equation solver to be used. It not specified the default `rte_solver` is `disort2`. 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. For your application please use `rte_solver disort2` which is the advanced version, unless you e.g. want to explore how a specific feature of disort2 (e.g. the Nakajima intensity correction) improves the disort result.

    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. **Warning:** it is not recommended to use spsdisort for really large solar zenith angles nor for cloudy conditions. For large optical thickness it is numerically unstable and may 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. The full implementation of the pol-Radtran solver is quite new (version 1.4). If you find unusual behaviour, please contact the libRadtran authors.

twostr   The two–stream radiative transfer solver described by Kylling et al. (1995), in pseudo-spherical geometry.

twostrpp   The two–stream radiative transfer solver described by Kylling et al. (1995), in plane-parallel geometry.

sos   A scalar pseudospherical succesive 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. For a more detailed description of MYSTIC read libRadtran/doc/README.MC.

tzs   TZS stands for "thermal, zero scattering" and is a very fast analytical solution for the special case of thermal emission in a non-scattering atmosphere. Please note that TZS does only radiance calculations at top of the atmosphere.

sss   SSS stands for "solar, single scattering" and is an analytical single scattering approximation which might be reasonable for an optically thin atmosphere. Please note that SSS does only radiance calculations at top of the atmosphere. This is an experimental solver - be careful!

null   The NULL solver does not solve the radiative transfer equation. However, it sets up the optical properties, and does the post-processing; useful if you are either interested in the overhead time required by a particular model input or if you are simply interested in the optical properties, as output by `verbose`.

Default: disort2

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. Please note that prior to convolution the spectrum is interpolated to the wavelength steps of the slit function. For this reason, make sure that the resolution of the slit function is high enough even if the slit function is e.g. a simple triangle which could in principle be described with 3 grid points.

solar_file

Location of file holding the extraterrestrial spectrum.

        solar_file filename [unit]

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 defines the wavelength resolution at which results are returned. However, the wavelength range is determined by `wavelength`. `solar_file` may be omitted for thermal radiation calculations (`source thermal`) as well as `transmittance` and `reflectivity` calculations. If omitted, the output resolution equals the internal wavelength grid which the model chooses for the radiative transfer calculation. Comments start with `#`. Empty lines are ignored.

For some purpuses it is useful to tell libRadtran the units of the spectrum. This can be done with the optional second argument. If `unit` is set to `per_nm` libRadtran assumes that the unit of the spectrum is W/(m2 nm), if set to `per_cm-1` it assumes W/(m2 cm-1). Note that `solar_file` is ignored if `correlated_k` is specified.

source   Solar or thermal source. Set to either `solar` or `thermal`. Solar radiation is per default output in W/(m2 nm) for spectral and `correlated_k` LOWTRAN calculations. For all other `correlated_k` options the output is integrated over the wavelength band. Thermal radiation is per default output in W/(m2 cm-1), if the bandwidth is equal to 1 cm-1 (default for `correlated_k` LOWTRAN calculations). Otherwise the output is the integrated flux over the wavenumber interval specified by `thermal_bandwith`, `thermal_bands_file`, or by the `correlated_k` option (Kato, Kato2, Kato2.96, Fu, AVHRR_KRATZ, or Generic).

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 290.0, 290.5, 291.0, ..., 364.5, 365.0. For interpolation to arbitrary wavelengths use `spline_file`. The specified wavelength interval must be within the one specified by `wavelength`.

`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 `wavelength`. 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.

`surface_temperature_map`

Specify a surface-temperature map with a *netCDF* file which is used in combinition with the options `latitude`, `longitude`, and `time`.

```
surface_temperature_map file [variable_name]
```

where `file` is the location of the *netCDF* file. `libRadtran` reads the value at the nearest pixel to the given `latitude` and `longitude`. No spatial interpolation or averaging of the values is done.

`surface_type`

With this option the `surface_type` is selected. This option can be used with `albedo_library` in order to select an spectral albedo or with `rpv_library` in order to select a BRDF function.

```
surface_type surface_type_number
```

where `surface_type_number` is an integer starting from 0, where 0 refers to a black surface and the following numbers to the entries in the specified `library`.

`surface_type_map`

Specify a surface type map, which is used in combinition with `albedo_library`, `latitude`, and `longitude` in order to select the surface type relevant for the simulation. No pixel interpolation is done. The format of the call is:

```
surface_type_map file [variable_name]
```

where `file` is the location of the surface type map file. The map is expected to be in *netCDF* format. The file must contain the variables `double lat(nlat)`, `double lon(nlon)`, and `byte surface_type (nlat, nlon)`. If the name of the surface type variable is different, the optional argument can be used in order to specify the variable name. For format specification see also data/albedo/IGBP/SURFACE_TYPE_IGBP_10min.cdf.

For using the IGBP map, the call is `surface_type_map IGBP`. This map has a resolution of 10 minutes and contains the surface types 1 to 18 defined in the `albedo_library` IGBP. Fresh snow and sea ice are not included, as their extent is too variable. Attention: That implies e.g. that the Arctic is considered ocean_water and not sea_ice!

Locations on the pixel boundaries are interpreted as the pixel northward and eastward respectively. E.g. location 0 N, 0 E is interpreted like the pixel ranging from 0 to 10min North and from 0 to 10min East.

sza   The solar zenith angle (degrees). The default solar zenith angle is 0.

sza_file   Location of solar zenith angle file for wavelength-dependent solar zenith angle. This is actually useful if you want to simulate an instrument which scans so slowly that the solar zenith angle may change significantly during the wavelength scan. 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 used together with `correlated_k` `KATO/KATO2/KATO2.96/FU/AVHRR_KRATZ`.

time   Specifies the time to simulate.

                 time YYYY MM DD hh mm ss

where YYYY is the year, MM the month, DD the day, hh the hour, mm the minute, ss the second in UTC. The time information will be used for a couple of things:

`time` in combination with `latitude`, `longitude`, and any `map`-option is used to select the location where to read the input data.

`time` is used to correct extraterrestrial irradiance for the Sun-Earth distance with the day of year. If not given, the Earth-Sun distance is 1 AU (i.e. equinox distance).

`time` in combination with `latitude` and `longitude` is used to calculate the solar zenith angle if no `sza` is specified.

`time` in combination with `latitude` and `longitude` is used to choose a suitable default atmosphere file, if no `atmosphere_file` is specified.

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 (downwelling radiance, looking upward) values (if any) and on through positive (upwelling radiance, looking downward) values. Must not be zero.

verbose   If specified abundances of informative messages are output to stderr. To make use of this information, you may want to write the standard uvspec output to one file and the diagnostic messages to another. To do so, try (`./uvspec < uvspec.inp > uvspec.out) >& verbose.txt` (depending on your shell you might need a slightly different syntax). The irradiances and radiances will be written to 'uvspec.out' while all diagnostic messages go into 'verbose.txt'. See also `quiet`.

wavelength

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` or `molecular_tau_file`

wavelength_index

Set the wavelengths to be selected. To be used together with predefined wavelength grids, such as `transmittance_wl_file molecular_tau_file` and particularly useful in combination with the `correlated_k` option where often only a specified number of wavelength bands is required. E.g., in combination with `correlated_k AVHRR_KRATZ`, `wavelength_index 15 15` will select wavelength index 15 which corresponds to channel 4, or `wavelength_index 10 14` will select those bands required for channel 3. Indices start from 1.

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. Please note that, if both `wc_cloudcover` and `ic_cloudcover` are set, both must be equal.

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
            (LWC) in grams per cubic meter, and column 3 the effective droplet radius in
            micrometer. Empty lines are ignored. Comments start with #. 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. An example of a cloud
            is given in 'examples/WC.DAT'.

            Per default the cloud properties are interpreted as layer properties. Before
            version 1.4 the default was level properties: The optical depth of a layer was
            calculated using information from the upper and lower levels defining the layer,
            see `wc_layer` and `wc_level`. To switch to the old behaviour, use `wc_level`.
            See the section about water clouds for a realistic example how the contents of
            the `wc_file` are converted to optical properties.

wc_files   A way to specify cloud extinction coefficient, single scattering albedo, and scat-
            tering phase function for each layer. The file specified by `wc_files` has two
            columns where column 1 is the altitude in km. 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 for-
            mat:

            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. Please note that the
            uppermost line of `wc_files` denotes simply the top altitude of the uppermost
            layer. The optical properties of this line are consequently ignored. There are two
            options for this line: either an optical property file with zero optical thickness
            is specified or "NULL" instead.

wc_ipa   Independent column calculation for a 3D cloud field. As argument a name of
            a 3D cloud file must be given. This file has to be in the format as needed by
            MYSTIC.

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`   Interpret cloud properties as layer properties (this is the default behaviour since version 1.4; see also wc‑file). Cloud properties are assumed to be constant over the layer. The layer reaches from the level, where the properties are defined in the `wc_file` to the level above that one. For example, the following lines

```
#      z     LWC    R_eff
#    (km)  (g/m^3) (um)
     4.000   0.0    0.0
     3.000   1.0   10.0
```

define a cloud in the layer between 3 and 4 km with sharp boundaries.

`wc_level`   Interpret cloud properties as level properties (this was the default behaviour before version 1.4; see also wc‑file). If `wc_level` is defined, a `wc_file` would be interpreted as follows:

```
#      z     LWC    R_eff
#    (km)  (g/m^3) (um)
     5.000   0      0
     4.000   0.2    12.0
     3.000   0.1    10.0
     2.000   0.1     8.0
```

The value 0.2 g/m3 refers to altitude 4.0km, as e.g. in a radiosonde profile. The properties of each layer are calculated as average over the adjacent levels. E.g. the single scattering properties for the model layer between 3 and 4km are obtained by averaging over the two levels 3km and 4km. To allow easy definition of sharp cloud boundaries, clouds are only formed if both liquid water contents above and below the respective layer are larger than 0. Hence, in the above example, the layers between 2 and 3 as well as between 3 and 4km are cloudy while those between 1 and 2km and between 4 and 5km are not.

`wc_no_scattering`
          Switch off scattering by water clouds.

`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. This is due to the use of the Henyey-Greenstein phase function as an approximation of the real Mie phase function.

echam4   Use the very simple two-band parameterization of the ECHAM4 climate model, described in Roeckner et al. (1996); this is probably only meaningful if you want to compare your results with ECHAM4, the two bands are 0.2 - 0.68 micrometer and 0.68 - 4.0 micrometer; within these bands, the optical properties are assumed constant.

mie   Use pre-calculated Mie tables; useful for `correlated_k`; the tables are expected in `data_files_path`/correlated_k/../
For spectral or pseudo-spectral (`correlated_k sbdart`) calculations, a set of pre-calculated tables is also available; the wavelength grid points of these data has been carefully selected such that the extinction cross section, single scattering albedo, and the asymmetry parameter are accurate to 1% (compared to the fully-resolved Mie calculation) for all wavelengths between 250nm and 100 micrometer. For spectral or pseudo-spectral calculations `wc_properties_interpolate` has to be defined explicitly to initiate the interpolation of the optical properties to the internal wavelength grid. Please note that this option may be extremely memory-consuming because for each internal wavelength a full set of Legendre moments of the phase function is stored (up to several thousands). The Mie tables are not part of the standard distribution (because of their large size) but they are freely available from http://www.libradtran.org. This is the correct option to calculate radiances, to be preferred over the Henyey-Greenstein approach of Hu and Stamnes (1993).

filename Read optical properties from specified filename; file format is as produced by the mie-tool of the `libRadtran-tool` box (see `output_user cloudprp`) or 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_properties_interpolate
Interpolate water cloud optical properties over wavelength; useful for precalculated optical property files defined with `wc_properties`. Please note that this option may be extremely memory-consuming because for each internal wavelength a full set of Legendre moments of the phase function is stored (up to several thousands).

wc_saturate
With this option, the relative humidity inside water clouds can easily be adjusted to a user-defined value, e.g. saturated with respect to water. This option has one nessesary and one optional argument:

        ic_saturate switch [relative_humidity]

where `switch` is `on`, `off`, or `ipa`. The second optional argument determines the relative humidity in % (with respect to water!) inside the cloud. The default value is 100. `ipa` is only relevant for independent column calculations. Details

for independent column simulations: Using `switch on`, the air in *all* columns will be saturated, if there is a cloud in at least one of the columns (this option should be used for stratiform clouds). Using `switch ipa`, only cloudy columns are affected (this option should be used for convective cloud fields.)

`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 `wavelength` does contain 550nm.

`zout`   This option is used to specify the output altitudes in km *above surface altitude*. One or more altitudes may be specified in increasing magnitude.

              zout 0 1 2 3 4 5 ...

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 due 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`. You can also use `toa` for `top of atmosphere` and `sur` for `surface altitude` and `cpt` for `cold point tropopause`.

zout_sea   like zout, but *above sea surface*

z_interpolate

The profile in the `atmosphere_file` provides the constituents of the atmosphere at the given levels. Where additional levels are introduced and in order to calculate layer properties, an assumption about the variation of the property within the layer is required. These interpolation methods can be changed by the `z_interpolate` option. Two arguments are required, the property, and the interpolation method:

    z_interpolate property interpolation_method

Properties which may be specified are:

O3, O2, H2O, CO2, NO2, BRO, OCLO, HCHO

T   temperature (here linmix is not suitable)

Possible interpolation methods are:

linear   The specified property (number density of the gas or temperature) varies linearly with height.

log   The specified property (number density of the gas or temperature) vaires logarithmically with hight. This is a reasonable option for all well mixed trace gases.

linmix   This option is only possible for gas profiles. The mixing ratio of the gas (assuming a logarithmically varying air density) varies linearly with hight.

For all gas densities the default interpolation method is `linmix`, for temperature it is `linear`.

zout_interpolate

The z-grid of optical properties is determined by the `atmosphere_file`, and, if specified, by other profile files like `dens_file`, `rh_file`, or `refractive_index_file`. Additional levels might be introduced by the `zout` option and the second argument of the `altitude` option. By default (if `zout_interpolate` is not specified) levels introduced by the `zout` option will not affect the optical property profiles, that is, the optical properties are constant within the layers specified by the `atmosphere_file` and profile files. If `zout_interpolate` is specified, the atmospheric profiles (tracegases, temperature ...) are interpolated to the levels introduced by `zout`, and optical properties are determined from the interpolated atmospheric properties. If `heating_rate` or `rte_solver polradtran` is specified, `zout_interpolate` will also be automatically activated. `zout_interpolate` generally causes smoother variation of the optical properties.

2.2 mie

`mie` performes 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 seperated 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.

aerosol_type

With this option Mie calculations are performed for the specified aerosol type. The aerosol properties (refractive index, size distribution, density, humidity) are taken from the OPAC database (see Hess et.al., 1998). Possible types are

`inso`	Water insoluble aerosol consists mostly of soil particles with a certain amount of organic material.
`waso`	Water soluble aerosol originates from gas to particle conversion and consists of various types of sulfates, nitrates, and other, also organic water-soluble substances.
`soot`	Soot is absorbing black carbon, which is not soluble in water. In reality soot particles have a chain-like character, which of course is not accounted for in Mie calculations of optical properties. The optical properties are calculated assuming many very small spherical particles.
`ssam`	Sea salt particles consist of the various kinds of salt contained in seawater. The different modes are given to allow for a different wind-speed-dependant increase of particle number for particles of different size. This aerosol type represents the accumulation mode.
`sscm`	Sea salt particles (coarse mode).

minm   Mineral aerosol or desert dust is produced in arid regions. It con-
              sists of a mixture of quartz and clay minerals and is modeled with
              three modes to allow to consider increasing relative amount of large
              particles for increased turbidity. This aerosol type represents the
              nucleation mode.

miam   Mineral aerosol (accumulation mode).

micm   Mineral aerosol (coarse mode).

mitr   Mineral transported is used to describe desert dust that is trans-
              ported over long distances with a reduced amount of large particles.

suso   The sulfate component is used to describe the amount of sulfate
              found in the Antarctic aerosol. This component is not suited to de-
              scribe antropogenic sulfate aerosols that are included in the water-
              soluble component.

distribution

If specified the effecive radius is converted into a size distribution of droplets.

          distribution distribution_type distribution_parameter

where distribution type is one of the two following:

GAMMA   The Gamma distribuition of cloud droplet sizes is

                    n(r) = a r^alpha exp(-b r),

              where alpha is the distribution paramter given as second argument.
              (a and b are determined automatically.) The effective radius of the
              distribution is r_eff = (alpha+3)/b. A typical value for water clouds
              is alpha=7. For ice clouds a typical value is alpha=1. A large value
              of alpha gives close to a monodisperse distribution.

LOGNORMAL

              The lognormal distribution of cloud droplet sizes is

                    n(r) = a/r exp (-[ln(r/r0)]^2 / [2 sigma^2]),

              where r0 is the logarithmic mode of the distribution (calculated
              automatically) and sigma is the standard deviation, which is given
              by second argument. The effective radius of the distribution is
              r_eff = r0 exp(2.5 sigma^2). A common value for water clouds is
              sigma=0.35.

mass_density

Specifies the mass density of the medium. Useful in combinition with `refrac user` and `output_user cloudprop`, as the format of `cloudprop` specifies the extinction coefficient per mass and not per volumn as usual in this mie program.

mie_program

Specify which Mie program to use:

BH   The Mie scattering program by Bohren and Hoffmann,
              ftp://astro.princeton.edu/draine/scat/bhmie/

MIEV0   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 MIEV0`.

nmom   Number of moments of the phase function to be calculated (default: 0). Only
            possible with `mie_program MIEV0`.

nstokes   Number of Stokes parameters (default: 1). For `nstokes=1` the Legendre poly-
            nomials of the phase function will be calculated. To calculate all phase matrix
            elements required for polarized radiative transfer, set `nstokes=4`.

output_user
            The mie output is one line of output quantities to standard output (stdout)
            for each wavelength and each particle radius. With this option the user may
            specify the columns desired for output:

                 output_user output_1 output_2 ... output_n

            where output_i is one of following arguments:

            lambda   Wavelength in nm.

            wavenumber
                        Wave number in cm-1.

            r_eff   particle radius in micro meter.

            refrac_real
                        The real part of the refractive index.

            refrac_imag
                        The imaginary part of the refractive index.

            qext   The extinction efficiency factor, if `r_eff` was specified, or the ex-
                        tinction coefficient [km-1] per unit concentration [cm3/m3], if a
                        `size_distribution_file` was specified. If the medium is liquid
                        water, 1 cm3/m3 equals a liquid water content of 1g/m3 because
                        the density of water is close to 1 g/cm3. For ice and other sub-
                        stances, the density has to be considered (0.917 g/cm3 for ice at
                        273K).

            qsca   The scattering efficiency factor, if `r_eff` was specified, or the ex-
                        tinction coefficient [km-1] per unit concentration [cm3/m3], if a
                        `size_distribution_file` was specified.

            omega   The single scattering albedo.

            gg   The asymmetry parameter.

            sforw   (Complex) forward-scattering amplitude S1 at 0 degrees.

sback   (Complex) back-scattering amplitude S1 at 180 degrees.

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   The nmom+1 moments (from 0 to nmom, see option `nmom`) 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.

cloudprp   This is a special option which, if specified, must be the only option of `output_user`, as `cloudprp` specifies a whole format of the output. In particular this option is useful when a correlated-k wavelength grid is specified with `wavelength`. If specified, the output will be written in a format, which can be directly used by `libRadtran`. See also `output netcdf`. See `libRadtran`-manual `ic_properties` and `wc_properties` and there the items `mie` and `filename`.

aerosolprp

> This option is similar to the `cloudprp` option. The only difference is that the effective radius dimension is replaced by humidity values of the aerosol.

netcdf   This option writes the same output as `cloudprp` and `aerosolprp` directly in netCDF format. The outputfiles can be used by `uvspec` using the options `ic_properties` and `wc_properties`.

The default output is:

```
lambda refrac_real refrac_imag qext omega gg spike pmom
```

r_eff   The radius [micron] of the particle to calculate single scattering properties of. Used together with the wavelength information to calculate the Mie size parameter.

```
r_eff radius
```

The user can optionally specify a 2nd and 3rd argument to make a loop over several radii:

```
r_eff radius_min radius_max radius_step
```

First calculations is done with radius_min, which will be increased by radius_step until radius_max is reached.

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 imaginary parts (both positive numbers).

   file <filename>
            Read refractive index from a three-column file containing wavelength [nm], and the real and imaginary parts of the refractive index (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.

temperature
         Ambient temperature, used to calculate the refractive indices of water and ice. Temperature dependence is only considered above 10 micron (water) and 167 micron (ice), respectively. Default: 300K.

wavelength
         Sets the wavelength range, in nm.

```
wavelength lambda_min lambda_max
```

The wavelength step is specified by wavelength_step.

For unregular wavelength grid it is also possible to specify a file, where the wavelengthgrid is stored.

```
wavelength wvl_filename
```

where wvl_filename is the path and name of the file, which contains the wavelength grid. It is expected that the wavelength values in nm are stored in the second column. For the correlated_k schemes implemented in uvspec you can use following abbrevations instead of a filename (in this case data_files_path must be specified also): kato, kato2, kato2.96, fu, and avhrr_kratz.

This option is ignored if refrac file is specified.

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

```
wavelength_index
```
Set the wavelengths to be selected. This might be the normal wavelength grid defined by `wavelength` and `wavelength_step` or a `correlated_k` wavelength grid. E.g., in combination with `wavelength AVHRR_KRATZ`, `wavelength_index 15 15` will select wavelength index 15 which corresponds to channel 4, or `wavelength_index 10 14` will select those bands required for channel 3. Indices start from 1.

2.2.2 The mie output

standard output (stdout) of the mie program is one line for each wavelength and each effective radius. The format of the output line is

```
lambda refrac_real refrac_imag qext omega gg spike pmom
```

The keywords here are the same as in input option `output_user`.

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.
wavelength    280. 5000.
wavelength_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 seperated 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.

`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.

`cascade_scale`
> Scale factor for bounded cascade by which the re-distributed fraction is decreased in each step.

`cascade_step`
> Number of steps for bounded cascade.

`cascade2D_p1`
> p1 for 2D bounded cascade.

`cascade2D_p2`
> p2 for 2D bounded cascade.

`cascade2D_H`
> H for 2D bounded cascade.

`cascade2D_step`
> Number of steps for 2D bounded cascade.

`cloud_bottom`
> Cloud bottom [m] (default: 1000).

`cloud_top`
> Cloud top [m] (default: 2000).

`cloud_file_output`
> Output 1D (uvspec) cloud file.

cloud_asy
>   Cloud asymmetry parameter.

cloud_extinction
>   Extinction coeffient [m-1]. Note that only one of both may be defined,
>   cloud_extinction or cloud_tau.

cloud_ssa
>   Cloud single scattering albedo.

cloud_tau
>   Average vertical optical thickness of the cloud.

cloud_type
>   Type of cloud model.

>   cascade1D
>>   1D bounded cascade.

>   cascade2D
>>   2D bounded cascade.

>   automaton
>>   3D cellular automaton.

>   gamma   Gamma distribution of optical thickness.

gamma_gamma
>   Gamma value for gamma distribution of optical thickness.

dx   Pixel size in x direction [m] (default: dz).

dy   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 worldloop

`worldloop` is a tool for making a large amount of `uvspec` calculations either on an latitude/longitude grid of an area (worldloop mode) or an pixel grid of a satellite picture (satellite mode).

The normal call for `worldloop` is:

```
worldloop [optional_arguments] input_template output_file
```

where the `input_template` is very simular to a normal `uvspec` input file, except that `worldloop` will automatically add following input options:

For worldloop mode:

```
latitude  lat
longitude lon
time YYYY MM DD HH mm ss
output_format netCDF output_file
```

For satellite mode:

```
satellite_pixel pixel_x pixel_y
time YYYY MM DD HH mm ss
output_format sat_picture output_file
```

Please have a look at the `uvspec` documentation of theses options.

The `output_file` is a *netCDF* file, which must contain the grid, on which the calculation will be carried out.

For worldloop mode:

```
double lat(lat)    in degrees North
double lon(lon)    in degrees East
double time(time)  in the format YYYYMMDD.FF
```

For satellite mode:

```
double pixel_x(nx) in the system of the satellite_geometry file
double pixel_y(ny) in the system of the satellite_geometry file
double time(time)  in the format YYYYMMDD.FF
```

where YYYYMMDD.FF means year, month, day, and dayfraction like 19780529.25 for 29th May 1978 06UTC. In order to generate such kind of output files, the tools `create_worldloop_grid` and `copy_worldloop_grid` may be useful.

At the beginning `worldloop` will make a test run. This is important in order to check, if all input options given in the `input_template` are proper `uvspec`-input. The user can

- quit the run, if the test output is errornous,
- start the `worldloop` simulation, if the test result is correct, or
- make another test run, if the user can not identify, if the result is correct.

After confirming that the `input_template` is correct, the simulation is started. `worldloop` will make a temporary copy of the `output_file` in the `/tmp` directory. Then `worldloop` will write all results to this copy and move it back to the starting directory, when the simulation is finished. During simulation, the generation process can be marveled with `ncview`.

Following `uvspec` input options, which make use of the latidude/longitude information, are especially useful for worldloop simulations: `albedo_map`, `altitude_map`, `cox_and_munk_pcl_map`, `cox_and_munk_sal_map`, `cox_and_munk_u10_map`, `emissivity_map`, `ECMWF_atmosphere_file`, `ECMWF_wc_file`, `ECMWF_ic_file`, `satellite_geometry` (this is essential for satellite mode simulations!!!), `surface_temperature_map`, `surface_type_map`, and `zout_sea`.

2.10.1 Optional command line arguments

The two most important optional command line arguments are `-w` and `-s` for changing the simulation mode of `worldloop`:

-w   worldloop mode (default) `worldloop` will do calculations on an lat/lon grid.

-s   satellite mode. `worldloop` will do calculations on an pixel_x/pixel_y grid.

There are some optional command line options for worldloop.

-q   worldloop writes only error messages and the most essential messages on the screen

-v   worldloop will give more feedback to the user than normal

-o   Overwrite mode is switched on. In the *netCDF* `output_file` there is a variables `done`, which shows if the simulation at a certain grid point is already carried out. If a run is restarted, `worldloop` will not recalculate the grid points, where done marks finished simulations. But if the overwrite mode is switched on, every grid point is recalculated.

-no_check
            Skipping the test run.

-PN n_processors

> *It is NOT ALLOWED to use this option on the office cluster, but ONLY on the Telekom cluster, as* `worldloop` *is suspected to slow down the internal network!!!* Number of processors (only for parralell run with condor). Instead of performing a normal simulation on one processor, `worldloop` devides the job into `n_processors` working packages and writes a condor script in order to start them. All input and output files must be visible from all computers of the condor cluster. The following steps are submit the condor job with `condor_submit jobfile` and add all parts of the result together with `add_worldloop_result`.

-PE my_processor n_processors

> This is the actual mode, when doing parallel processing. `my_processor` is the number of the processor (1...`n_processors`), and `n_processors` is the total number of processors. The computer `my_processor` computes the pixels `my_processor`, `my_processor` + `n_processors`, `my_processor` + 2 `n_processors` ...

2.10.2 Additional tools for worldloop

This chapter is a summary of small programs, which might be useful in combination with `worldloop`.

`create_worldloop_grid [-s -w -q] output_file lat_min lat_max d_lat lon_min lon_max d_lon time_min [time_max d_time]`

> This is a small program to create the `output_file` containing a lat/lon/time grid, which is nessesary to start `worldloop` simulations. The optional arguments are `-s` for satellite mode, `-w` for worldloop mode (default), and `-q` for quiet. `lat_min` and `lat_max` are the latitude boundaries of the map, and `d_lat` the latitude increment. Same for longitude by `lon_min`, `lon_max`, and `d_lon`, and time with `time_min`, `time_max`, and `d_time`. Latitude must be given in degrees north, longitude in degrees east, and time in the format YYYYMMDD.FF like 19780529.25 for 29th May 1978 06UTC.

`copy_worldloop_grid [-s -w -q] sourcefile resultfile [time_min time_max d_time]`

> This small program copies the calculation grid from a *netCDF* input file: The optional arguments are `-s` for satellite mode, `-w` for worldloop mode (default), and `-q` for quiet. For worldloop mode the source file must contain a latitude/longitude grid like an `ECMWF_atmosphere_file` (see `uvspec` documentation). For satellite mode the source file must contain a pixel_x/pixel_y grid like in an `satellite_geometry` file (see also `uvspec` documentation). The last 3 optional arguments may be given, if the time grid should be different to the source file (or if the source file does not contain a time grid). time should be given in the format YYYYMMDD.FF like 19780529.25 for 29th May 1978 06UTC.

`add_worldloop_results [-o -q -v] output_file1 output_file2`

> Small program to combine two result files from a parallel `worldloop` simulation. `add_worldloop_results` will write all information of `input_file1` to `input_file2`. Both input-files must have an exactly identical *netCDF* structure. The optional arguments means `-q` for quiet and `-q` for verbose feedback. In normal mode information are only copied to `output_file2` if it does not contain any information at these grid points, but if the overwrite mode is switched on with argument `-o` information of `output_file1` will overwrite information of `output_file2`.

`check_worldloop_result [-q -v] limits_file output_file`

> Small program to eliminate results which have an unreasonalbe magnitude (disort instabilities). The `limits_file` is a 3 column ASCII file: 1st column the *netCDF* name of the property, which should be checked, 2nd column the lower limit, 3rd column the upper limit. And `output_file` is the `worldloop` result file, which should be checked. The optional arguments are `-q` for quiet and `-q` for verbose feedback.

2.11 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.

2.11.1 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
```

2.11.2 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.

2.12 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.

2.12.1 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
```

2.12.2 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.

- Belward, A., and T. Loveland, 'The DIS 1km Land Cover Data Set, global change', 1996 The IGBP Newsletter, #27, see also: http://www-surf.larc.nasa.gov/surf/pages/ftp-page.html

- Baum, B. A., A. J. Heymsfield, P. Yang, and S. T. Bedka, 'Bulk scattering models for the remote sensing of ice clouds. Part 1: Microphysical data and models', 2005, J. Appl. Meteor., 44, 1885-1895

- Baum, B. A., P. Yang, A. J. Heymsfield, S. Platnick, M. D. King, Y.-X. Hu, and S. T. Bedka, 'Bulk scattering models for the remote sensing of ice clouds. Part 2: Narrowband models', 2005, J. Appl. Meteor., 44, 1896-1911.

- Baum, B. A., P. Yang, S. L. Nasiri, A. K. Heidinger, A. J. Heymsfield, and J. Li, 'Bulk scattering properties for the remote sensing of ice clouds. Part 3: High resolution spectral models from 100 to 3250 cm-1.', 2007, J. Appl. Meteor. Clim., Vol. 46, 423-434.

- 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.

- Hess, M., P. Koepke, and I. Schult, 'Optical properties of aerosols and clouds: the software package OPAC', Bull. Am. Met. Soc, 79, 831-844, 1998.

- 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.

- Hu, Y.X., B. Wielicki, B. Lin., G. Gibson, S. Tsay and K. Stamnes, 2000,'A fast and accurate treatment of cloud particle scattering phase functions: singular value decomposition least square fitting', J. Quant. Spectrosc. Radiat. Transfer, 65, 681-690.

- 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.

- Wanner, W., X. Li, and A.H. Strahler, 1995, 'On the derivation of kernel-driven models of bi-directional reflectance', Journal of Geophysical Research, 100(D10), 21077-21089.

- Wanner, W., A.H. Strahler,B. Hu, P. Lewis, J.-P. Muller, X. Li, C.L. Barker Schaaf, and M.J. Barnsley, 1997, 'Global retrieval of bidirectional reflectance and albedo over land from EOS MODIS and MISR data: Theory and algorithm', Journal of Geophysical Research, 102(D14), 17143-17161.

- 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