Documentation for RFMDISORT

Nazim Bharmal
n.a.bharmal@rdg.ac.uk

June 30, 2009

There is a list of unresolved issues further in this document. If having trouble, it may be wise to consult it.

1 Précis

This document explains how to use the radiative transfer code RFMDISORT. Within can be found a brief explanation of how the code is structured, a description of how it may be used, and what inputs it expects and what outputs you can expect.

The current capability of the code appears to be: shortwave, longwave, aerosol, fluxes only, heating rates, spectrally resolved (1 cm^{-1} resolution only) and integrated ('broadband') outputs.

2 Overview

The radiation transfer code RFMDISORT is really a combination of two major codes, RFM and DISORT, and a collection of smaller codes written in Fortran and Korn-shell scripts. Collectively, given text input files, the software will calculate radiative fluxes and heating rates at high spectral resolution for typical atmospheric physics. This includes the ability to specify aerosols but does not include the ability to specify clouds.

The user interacts with RFMDISORT via a shell script that calls the various codes in turn and that can be configured partly via command like switches but mostly via a set of text-format input files; each is termed a driver file. Additionally an atmospheric profile must be specified and this contains the temperature, pressure, and gas concentrations for the calculation. There is an overall driver file and then one each for shortwave and longwave calculations. If aerosols are to be used then there additional aerosol definition files are required.

Detailed documentation on RFM can be found at,

http://www.atm.ox.ac.uk/RFM/,

and documentation on DISORT in the `DISORT.doc` file within the `$RFMD_DIR/DISORT` directory. One point of note is that DISORT is called as a function from the RFMDISORT coupler code, `$RFMD_BIN/rfmdisort_coupler.x`.

2.1 Additional information

Nothing at present. I imagine the contact details of the maintainer/person responsible for RFMDISORT ought to be here.

3 Obtaining and configuring the code

Incomplete. I imagine the code will be provided as a TAR file, with this file contained within.

3.1 Building the code

See the appendix for details.

3.2 Quick overview

The precondition for using RFMDISORT is a Fortran 90 compiler (most is written in Fortran 77) and a Korn shell. Currently the compiler commands are specific to Sun Studio v12, so compatible with Solaris computers.

1. Change to the directory `$RFMD_DIR/RFM/source` and type `make`.
2. Change to the directory `$RFMD_DIR/CODE` and type `make`.
3. Change to the directory `$RFMD_DIR/HITRAN` and type `make`.

This should complete the build process.

3.3 For day to day use

Once RFMDISORT is compiled, using it on a daily basis only requires that several environment variables be set in the user's shell. This involves changing to the RFMDISORT directory and 'sourcing' the `sbin/set_rad_env` script. (See section A in the appendix.)

4 Using the code

To use the RFMDISORT code, after it has been installed, first requires that the environment variables be set (see section 3.3). Then input files should be created, consisting of driver files that contain the radiative transfer options and input files that contain the state of the atmosphere and (optionally) the optical properties of any aerosols. Finally, it is a matter of calling the shell `Crun_rfmd` script and then, assuming no errors terminated the process, examining the output files.

4.1 Creating the driver files

The configurable options for RFM and DISORT are found in two (or three) driver files. The first is a general file and exists regardless of the wavelength coverage. The second is specific to shortwave calculations, and the third to longwave calculations. For the latter two driver files, at least one is required but both can be present, depending on the wavelength coverage chosen. These files are in text format and ignorable lines (comments) can be added by prefixing them with a '!' character.

The driver files must exist in the same directory and their filenames should have a common prefix, with the following suffix,

Driver type	Suffix
General	`rfmdisort.drv`
Shortwave	`rfmdisort_sw.drv`
Longwave	`rfmdisort_sw.drv`

The only caveat is that the prefix cannot be blank, meaning that the driver files for a SW-only calculation could be called `test_rfmdisort.drv` & `test_rfmdisort_sw.drv` but could not be called `rfmdisort.drv` & `rfmdisort_sw.drv`. (This is because the `Crun_rfmd` takes the driver files and processes them to create new files whose names are stripped of the prefix.) The default prefix is '`_`', so the general driver is then called `_rfmdisort.drv`.

Following is a brief description of the driver files and the common values, variables, and switches that can be altered within them. A complete guide to their contents is in appendix section D.1.

4.1.1 General driver file

The general driver file has (at least) 10 lines. Within are options controlling whether RFM and DISORT are to be run, the number of atmospheric layers and the filename of the atmospheric input, and the wavebands to do calculations over.

4.1.2 Shortwave driver file

The shortwave driver file has (at least) 18 lines. Within are options controlling the name of the RFM-output optical depths, the wavenumber range, the time-averaging for the calculation, the solar zenith angle *or* date, the gases used to calculate scattering, whether Rayleigh scattering is enabled, whether aerosols are included in the calculation, the number of streams, the number of phase function moments, and the surface albedo.

4.1.3 Longwave driver file

The longwave driver file has (at least) 13 lines. It is similar to the shortwave file. Within are options controlling the name of the RFM-output optical depths, the wavenumber range, the definition of surface (skin) temperature, the gases used to calculate scattering, whether Rayleigh scattering is enabled, whether aerosols are included in the calculation, the number of streams, the number of phase function moments, and the surface albedo.

4.2 Creating the atmospheric input

The state of the atmosphere, baring aerosols, is defined within the atmospheric state file and which can be located in any directory. This is processed exclusively by RFM, and its definition can be found in appendix section D.2. (For detailed documentation, consult the RFM Software User Manual.) In general, the variables are defined at the levels (layer edges) starting at the surface. The first section must be altitude and then the order is arbitary but usually pressure and temperature follow, and then a list of gas concentrations.

A particular customisation exists to account for the lack of HITRAN data above 25232 cm^{-1}. In particular, Ozone is an important gas in the UV region and so an additional gas, X65, is added to the atmospheric state with an identical concentration to O3 (Ozone).

4.3 Creating aerosol input

Whilst the driver files and atmospheric state are necessary to use RFMDISORT, the aerosol inputs are only needed if the aerosol flag in the shortwave or longwave driver file is set to 1. The aerosol data is read in by the DISORT coupler code, meaning if pre-calculated optical depths exist from a RFM calculation then they need not be repeated. This would allow, for example, a RFM and DISORT calculation without aerosols and then a DISORT-only calculation if aerosols are to be included.

There is a hierarchy of files that define the various parameters for each aerosol type, defined in appendix section D.3. First, the file `aerosols.dat` must exist in the same directory as the driver files. This defines the number of aerosols and lists the aerosol control files. Each aerosol control file specifies the density of the aerosol per layer and references two further files; one, aerosol's optical properties (except phase function moments) and two, the phase function moments of the aerosol.

The following structure shows this hierarchy of files,

- `aerosols.dat`, with n aerosols defined, references the file(s)...
  - `aerosol_1.control`, (control file) references the files...
    - `aerosol_1.optprop`, (optical properties),
    - `aerosol_1.pmom`, (phase function moments).

    ...
  - `aerosol_n.control`, (control file) references the files...
    - `aerosol_n.optprop`, (optical properties),
    - `aerosol_n.pmom`, (phase function moments).

There are some (non-obvious) aspects of the aerosol input which are worth noting:

1. The aerosol optical properties and phase function moments are defined using a common set of wavelengths, but this is only referenced in the optical properties file.
2. Each entry in the aerosol optical properties file has two unused values at the end.
3. Each entry in the aerosol phase function moments file has $N + 1$ moments if there are N moments specified in the shortwave or longwave driver file. (The first has the value of 1 so is redundant but must be present.)
4. The aerosol density is specified per *layer*, whereas gas concentrations are given per *level*.

4.4 Examining the outputs

When the `Crun_rfmd` script has finished, there will be several files created in the directory (presuming that the retain option, `-x`, has not been enabled),

- **Spectral fluxes:** A set of files per level will be created with the prefix `spec_flux_sw.dat` and/or `spec_flux_lw.dat`, and the suffix as the layer number.
  The contents are direct and diffuse fluxes for each wavenumber interval.
- **Integrated fluxes:** The fluxes integrated over the wavenumber range and grouped into one file (for all levels), called either `bband_flx_sw.dat` or `bband_flx_lw.dat`.
  The contents are direct and diffuse fluxes, pressure, altitude, net fluxes per level.
- **Heating rates:** Heating rates per layer, called either `bband_heat_sw.dat` or `bband_heat_lw.dat`.
  The contents are layer number and heating rate.
- **Aerosol-free optical depth:** Optical depths are calculated by RFM and used by the RFM-DISORT coupler.
  The name is not fixed and is the first entry in the SW or LW driver file.

Other files are intermediate and can be removed. To usefully examine the spectral or integrated fluxes, there are two IDL codes that convert these text files into NetCDF,

- `rfmdbband2cdf.pro` and,
- `rfmdspec2cdf.pro`.

which can then be examined using the `ncview` utility or similar.

5 Uncategorised Notes

- **IMPORTANT:** There are pathlength limitations which may cause the code to truncate input filenames and then not be able to find them. This can be resolved by moving the offending files into a closer directory.
- **IMPORTANT:** The codes will terminate if there is insufficient memory, but there are insufficient checks made. Usually no helpful output is given explaining the cause of a problem and the process may simply crash.
- **Note:** No ability to calculate radiances.
  - In principle it can be enabled but will require further work.
- **IMPORTANT:** Some checking of shell status in the shell script `Crun_rfmd` but not all errors are correctly accounted for so there are instances where the Fortran codes abort (via a `STOP` command) but the script does not detect this and will attempt to continue the overall algorithm.
- **NOTE:** Test inputs and outputs missing?
  - There are additional (but not comprehensively verified) inputs and outputs available in the `tests` directory.
  - Steven Rumbold has sent the locations of his test inputs. Part of the inputs mentioned below (sub-Arctic winter, 30 degree SZA, no aerosols and high aerosol OT) have been tested and are present in this copy of RFMDISORT, in the `tests/halthore` directory.

```
    ...the comparison with the Halthore et al 2005 study (including
APPRAISE, Imperial, and RAL runs) at:

/home/sws06str/APPRAISE/HALTHORE_TEST/

The results are plotted in: hal11.eps, hal12.eps, hal21.eps and hal22.eps.

black -x-  is Halthore mean and 1 sigma sd
red -o- is ESRAD 2 stream with error bar resulting from band selection.
black o is the Halthore 2 stream code
red star is RFMDISORT
blue star is SBDART from Imperial
Green are the RAL runs (limited to one box due to constraints with their
calculation method)

see also: halthore_researchday.tiff

These plots are generated with the matlab code:  halresult.m.  This is a bit
messy as the data formats for each experiment (RDG, IMP, RAL) were all
different!

The setup for the experiment can be found in:

/home/sws06str/APPRAISE/HAL_SETUP/

This contains:

Rayleigh_xsec.datbis - Rayleigh cross section
halspec.dat - TOA solar flux
testaerosol_low_aod.dat - Low aerosol load case vertical profile
testaerosol.dat - High aerosol load case vertical profile
aerosols.dat - RFMDISORT type aerosols file
halthore_et_al_2005.pdf - Halthore's paper
trotest.atm - tropical atmosphere used
hal_aero_basic - basic aerosol properties
sawtest.atm - sub Arctic winter profile
```

```
hal_aero_pmom - aerosol phase function

This should be enough information to run the idealized cases at 30 and 75
degree solar zenith angles.
```

- **Note:** a 'fort.2' ouput file.
  - This is faulty output, for a stream (2) that has not been opened. The Fortran causing it is thought to be in the file `write_intensity.f`.
- **Undocumented:** Why is the parameter 'MAXWID' in the file 'RFM/source/rfmsiz.inc' increased from 3000 to 50000?
- **NOTE:** The detection of specifying more layers than can be dealt with in DISORT is now enabled. To change the number of supported layers, the values of four variables need changing:
  - In `rfmdisort_coupler.f`: `maxcly`, `maxulv`
  - In `DISORT.f`: `MAXCLY`, `MAXULV`
    (DISORT checks if the number of layers exceeeds the values of these variables and then reports this fact.)
- **NOTE:** For wavenumber larger than 25665, the `X65` gas may have to be removed from the `rfmdisort_sw.drv` file, which is equivalent to removing Ozone. This advice is only relevant if issues occur for this wavenumber.
- **NOTE:** Ensure the RFM (and perhaps DISORT) spectral resolution in the SW or LW (`rfmdisort_Xw.drv`) driver files is written with a decimal point: `1.0` is acceptable but `1` would cause issues.

A Description of Crun_rfmd and set_rad_env

Interaction with RFMDISORT takes place via the driver & aerosol input files, and the shell script called `Crun_rfmd`. This script depends on the environment variables set by `set_rad_env.sh/set_rad_env.csh`.

A.1 Description of set_rad_env

As a pre-requisite to using `Crun_rfmd`, several environment variables need to be set. This can be done by hand or more concisely via one of the two `set_rad_env` scripts, depending on which shell is being used,

```
       for csh:   source sbin/set_rad_env.csh
or bash/ksh:   source sbin/set_rad_env.sh
```

Also, the script will set the IDL path variable so the IDL codes can be used. If successful, the command,

`ls $RFMD_SCRIPT/Crun_rfmd`,

will not produce an error.

A.2 Description of Crun_rfmd

The script `Crun_rfmd` is called from the directory containing the input files using the command,

`$RFMD_SCRIPT/Crun_rfmd`,

with any options (see below) appended. It will process the input driver files and then execute RFM (iteratively) and/or DISORT. It does not process the output files. It can retain the runtime log file produced by RFM, and other debugging information. It will attempt to detect errors in RFM or DISORT and terminate processing, and return a status value of 1 (as standard) to indicate some error. (This doesn't work great with either RFM or the RFM-DISORT coupler but it tries.)

A.2.1 Command line options

No option is mandatory, although `-P` will often be required.

- `-h` number : The maximum wavenumber for which to include HITRAN data. This should not be more than 25232.0, which is the default.
- `-C` string : Comment to be used by RFM. *(It does not appear to be used in the output files, however.)*
- `-n` number : Number of wavebands that RFM can calculate. (Note that a waveband has resolution equal to the spectral resolution.) Default is 1900.
- `-q` : Reduce output from the script.
- `-x` string : Keep intermediate outputs as files prefixed by the string.
- `-P` string : Define the prefix for the driver files. Default is '_'.

B Description of external programs

There are several other programs written to aid the use of RFMDISORT. These are listed and described here. For the IDL codes, it is necessary to add the IDL subdirectory to the IDL path variable. This is taken care of in the `set_rad_env` script.

B.1 IDL program, `es2rfmd.pro`

To convert inputs for Edwards-Slingo (1996), henceforth ES96, into a set of inputs suitable for RFMDISORT, the code `es2rfmd.pro` can be utilised. It should be called using the IDL interpreter and with optional arguments,

```
idl -e ".run $RFMD_DIR/idl/es2rfmd.pro" -args
  -B prefix [-s file] -R low high [-o directory]
  [-S] [-I] [-a] [-r] [-n streams] [-p pressure]
  [-P prefix]
```

B.1.1 Description of options

- `-B` prefix : The ES96 input files prefix, without the full stop but with leading directory information.
  For example, if the humidity is stored in the file `~/exampleinput/mls.q` then a valid prefix is `~/exampleinput/mls`.
- `-s` file (Optional) : ES96 spectralfile, required if aerosols are used for the optical properties and phase function moments but otherwise only used to translate between band number and wavelength range.
- `-R` low high : Depending on if the spectral file is specified,
  - If there is a spectral file, then the lower and upper band numbers, (Translated to wavenumber range internally.)
  - Otherwise, the lower and upper wavenumber range.
- `-o` directory (Optional) : The output directory, which can be '.' to mean the current directory.
  If this option is not given then no output is created, which is useful for testing this program.
- `-S` (Optional) : Create shortwave output.
- `-I` (Optional) : Create longwave output.
  (Note that -S and -I can be used together.)
- `-a` (Optional) : Create aerosol input files. Requires a spectral file to be specified, so does not work with prescribed inputs.
- `-r` (Optional) : Use Rayleigh scattering, default is not to.
- `-n` (Optional) : Specify the number of streams, default is 4.
- `-p` pressure (Optional) : Surface pressure for calculating the isothermal height.
- `-P` prefix (Optional) : The prefix for the driver files, default is `es2rfmd_`.

An example command line is,

```
idl -e ".run $RFMD_DIR/idl/es2rfmd.pro"
 -args -s ~/es_spectra/sp_sw_220_aeronet17r -S -R 90 91
 -B ~/es2rfmd_tst_inputs/mls/sw/mls -a -r  -o ./
```

which will use the spectral file `sp_sw_220_aeronet17r` to convert the input with basename `mls` (in the directory `~/es2rfmd_tst_inputs/mls/sw/`) into a SW-only run over the ES96 bands 90-91 with aerosols & Rayleigh scattering, and output into the current directory.

B.2 IDL program, `rfmdbband2cdf.pro`

To convert the output broadband fluxes and heating rates into NetCDF files, the code `rfmdbband2cdf.pro` should be called with the IDL interpreter,

```
idl -e ".run $RFMD_DIR/idl/rfmdbband2cdf.pro" -args input1 input2 outputprefix
```

where,

- `input1` : The name of the ASCII output file for integrated/broadband fluxes.
  (Either `bband_flx_sw.dat` or `bband_flx_lw.dat`.)
- `input2` : The name of the ASCII output file for integrated/broadband heating rates.
  (Either `bband_heat_sw.dat` or `bband_heat_lw.dat`.)
- `outputprefix` : The prefix for the output NetCDF versions of fluxes and heating rates, called *outputprefix*`flxs.nc` and *outputprefix*`hrts.nc`.

B.3 IDL program, `rfmdspec2cdf.pro`

To convert the output spectrally resolved fluxes rates into a NetCDF file, the code `rfmdspecl2cdf.pro` should be called with the IDL interpreter,

```
idl -e ".run $RFMD_DIR/idl/rfmdspec2cdf.pro" -args inputprefix input2 outputprefix
```

where,

- `inputprefix` : The prefix of the ASCII output file for spectrally resolved fluxes.
  (Stripped of the level suffix, either `spec_flux_sw.dat` or `spec_flux_sw.dat`.)
- `input2` : The name of the ASCII output file for *integrated/broadband* fluxes. (Used only to provide an easy source of information for altitude and pressure.)
- `outputprefix` : The prefix for the output NetCDF version of flux, called *outputprefix*`specflxs.nc`.

C Output file format definitions

Here the format of the output files is given. The definitions below explain each line in the order required by the codes. Each line with a colon will represent (roughly) the Fortran definition of the data on the left and its explanation on the right. The terminology X*$Y(Z)$ means X lines of variables, each consisting of Z values of Y (i.e. a 1-D array).

C.1 Spectral fluxes

The spectral flux files are for either shortwave or longwave, but otherwise identical. They consist of a header (prefixed with an exclamation mark, '!') and then lines for each wavenumber interval, given at the wavenumber bin centre.

C.1.1 File structure

- **`spec_flux_sw.datXXX`, where XXX represents the three digit layer number :**
  - 6*`!`... : Header lines, self-explanatory.
  - N*REAL(4) : Wavenumber bin fluxes, N is the number of wavenumber bins.
    1. Centre of wavenumber bin, cm^{-1},
    2. Direct flux downwards, Wm^{-2},
    3. Diffuse flux downwards, Wm^{-2},
    4. Diffuse (equiv. total) flux upwards, Wm^{-2}.
- **`spec_flux_lw.datXXX`, where XXX represents the three digit layer number :**
  - As above except direct flux is always zero.

C.2 Integrated fluxes

The integrated, or broad-band, fluxes are for either shortwave or longwave, but otherwise identical. They consist of a header (prefixed with an exclamation mark, '!') and then lines for each level.

C.2.1 File structure

- **`bband_flx_sw.dat` :**
  - 9*`!`... : Header lines, self-explanatory.
  - N*REAL(7) : Level fluxes, N is the number of levels.
    1. Pressure, hPa,
    2. Altitude, km,
    3. Diffuse flux upwards, Wm^{-2},
    4. Direct flux downwards, Wm^{-2},
    5. Diffuse flux downwards, Wm^{-2},
    6. Total flux downwards, Wm^{-2},
    7. Net flux, Wm^{-2}.
- **`bband_flx_lw.dat` :**
  - As above except direct flux is always zero.

C.3 Integrated fluxes

The integrated, or broad-band, fluxes are for either shortwave or longwave, but otherwise identical. They consist of a header (prefixed with an exclamation mark, '!') and then lines for each level.

C.3.1 File structure

- `bband_sw_flux.dat` **:**
  - 9*`!`... : Header lines, self-explanatory.
  - N*REAL(7) : Level fluxes, N is the number of levels.
    1. Pressure, hPa,
    2. Altitude, km,
    3. Diffuse flux upwards, Wm^{-2},
    4. Direct flux downwards, Wm^{-2},
    5. Diffuse flux downwards, Wm^{-2},
    6. Total flux downwards, Wm^{-2},
    7. Net flux, Wm^{-2}.
- `bband_lw_flux.dat` **:**
  - As above except direct flux is always zero.

C.4 Integrated heating rates

The integrated, or broad-band, heating rates are for either shortwave or longwave, but otherwise identical. They consist of a one line header and then lines for each level.

C.4.1 File structure

- `bband_hrts_sw.dat` **:**
  - CHARACTER (*) : Header line, self-explanatory.
  - N*REAL(2) : Level fluxes, N is the number of levels.
    1. Layer number (note in reverse order since 1 is the surface),
    2. Heating rate, K day^{-1}.
- `bband_hrts_lw.dat` **:**
  - As above.

D Input file format definitions

Here the format of the input files is given. The definitions below explain each line in the order required by the codes. Each line with a colon will represent (roughly) the Fortran definition of the data on the left and its explanation on the right. The terminology X*$Y(Z)$ means X lines of variables, each consisting of Z values of type Y (i.e. a 1-D array). Sometimes, following the type a variable name is given (in italics) to aid the description.

D.1 Driver files

The driver files contain the configurable options for RFMDISORT. They consist of a general driver, with suffix `rfmdisort.drv`, and one each for the shortwave and longwave, with suffices `rfmdisort_sw.drv` and `rfmdisort_lw.drv`. The `Crun_rfmd` script expects these filenames with a non-blank prefix, the default is '`_`'.

The driver files must have their data in the given order.

D.1.1 File structure

Lines beginning with an exclamantion mark ('!') are ignored so can be used as comments. All lines are processed by the `Crun_rfmd` script so that shell nomenclature (such as environment variables) are expanded. (For example the input '`~/rfmdaero/`' and '`$HOME/rfmdaero/`' would be transformed into identical pathnames.)

- `rfmdisort.drv` **:**
  - CHARACTER (80) : Description.
  - CHARACTER (80) : Prefix for output files.
  - INT : Whether to run RFM (1) or not (0).
  - INT : Whether to run DISORT (1) or not (0).
  - INT n : The number of levels in the atmospheric state.
  - CHARACTER (*) : Filename of atmospheric state input.
  - INT b : If $b = 1$, SW only; $b = 2$, LW only; $b = 3$, both wavebands.
  - REAL p_{trop} : Tropopause pressure, hPa.
  - CHARACTER (*) : Filename of HITRAN binary file.
    Default is: `$RFMD_DIR/HITRAN/HITRAN04.bin`
  - CHARACTER (*) : Flags used by RFM
    Default set is: `OPT OBS ZEN CTM DBL`
- `rfmdisort_sw.drv` **:**
  - CHARACTER (*) : Prefix for optical depth file.
  - REAL w_{low} : Wavenumber lower limit, cm^{-1}.
  - REAL w_{top} : Wavenumber upper limit, cm^{-1}.
  - REAL Δw_{RFM} : Wavenumber resolution for RFM, cm^{-1}.
    NOTE: Values other than 1.0 are not tested.
  - REAL Δw_{DISORT} : Wavenumber resolution for DISORT, cm^{-1}.
    NOTE: Values other than 1.0 are not tested.
  - INTEGER m : Depending on value of m,
    0. non time-averaged (instantaneous) calculation,
    1. diurnal-average calculation between two months,
    2. diurnal-average calculation between two Julian days.
  - REAL *or* INT, INT *or* INT, INT : Depending on m,
    0. REAL : Solar zenith angle, degrees.
    1. INT, INT: Month start and end, inclusive.
    2. INT, INT: Julian day start and end, inclusive.

- INT o : Earth-Sun orbit flag, $o = 0$ is circular orbit, $o = 1$ is calculated orbit.
- CHARACTER (*) : Ozone cross-section filename.
  Default is: `$RFMD_DIR/OZONE/o3.xsc`
- INT r, CHARACTER(*) r_f: Rayleigh scattering, $r = 0$ disables, $r = 1$ enables. The scattering cross-sections are given by r_f.
  Default r_f is: `$RFMD_DIR/RAYLEIGH/ray.dat`
- INT A : Aerosol consideration, if $A =$,
  0. Aerosols not utilised,
  1. Utilised for scattering and absorption,
  2. Utilised for absorption only.
- INT S : Number of streams, S should be an even number greater than 4.
- INT N : Number of phase function moments, where $N \geq S$.
- INT : Type of surface reflectance, currently must be 1 (Lambertian).
- INT $t_a = 1$, REAL *or* INT $t_a = 2$, CHARACTER(*) : Defining surface albedo, depending on t_a,
  1. REAL : grey albedo,
  2. CHARACTER (*) : filename of spectral albedo.
     IMPORTANT NOTE: It is not clear if the ability to have spectral albedos is enabled in the code (`rfmdisort_coupler.f`).

- **`rfmdisort_lw.drv` :**
  - CHARACTER (*) : Prefix for optical depth file.
  - REAL w_{low} : Wavenumber lower limit, cm^{-1}.
  - REAL w_{top} : Wavenumber upper limit, cm^{-1}.
  - REAL Δw_{RFM} : Wavenumber resolution for RFM, cm^{-1}.
    NOTE: Values other than 1.0 are not tested.
  - REAL Δw_{DISORT} : Wavenumber resolution for DISORT, cm^{-1}.
    NOTE: Values other than 1.0 are not tested.
  - INT $t_T = 0$ *or* INT $t_T = 1$, REAL : Skin temperature, depending on t_T,
    1. REAL : skin temperature, K. **IMPORTANT NOTE:** It is not clear if the ability to define the skin temperature using $t_T = 1$ is enabled in the code (`rfmdisort_coupler.f`).
    2. Skin temperature is defined equal to surface level temperature.
  - CHARACTER (*) : List of gases, as in the atmospheric state file. (If `X65` is defined then do not include it here.)
  - INT $r = 0$ *or* INT $r = 1$, CHARACTER(*) r_f: Rayleigh scattering, depending on r,
    0. No Rayleigh scattering.
    1. CHARACTER(*) : Enabled, with the scattering cross-sections in the file r_f.

    Default r_f is: `$RFMD_DIR/RAYLEIGH/ray.dat`
  - INT A : Aerosol consideration, if $A =$,
    0. Aerosols not utilised,
    1. Utilised for scattering and absorption,
    2. Utilised for absorption only.
  - INT S : Number of streams, S should be an even number greater than 4.
  - INT N : Number of phase function moments, where $N \geq S$.
  - INT : Type of surface reflectance, currently must be 1 (Lambertian).
  - INT $t_a = 1$, REAL *or* INT $t_a = 2$, CHARACTER(*) : Defining surface albedo (*not* emissivity), depending on t_a,
    1. REAL : grey albedo,
    2. CHARACTER (*) : filename of spectral albedo.
       IMPORTANT NOTE: It is not clear if the ability to have spectral albedos is enabled in the code (`rfmdisort_coupler.f`).

D.2 Atmospheric state input

The atmosphere is defined as a series of homogenous layers and the relevant variables are described on edge's of these layers, the levels. The structure of the input file is a definition of the level altitudes, the pressure, temperature, and gas concentrations as parts-per-million by volume (ppmv).

D.2.1 File structure

Lines beginning with an exclamantion mark ('!') are ignored so can be used to comment the atmospheric state. The file is divided into sections that contain the values for each variable. Each section begins with a line consisting of a short piece of text, of the form `*LABEL`, that identifies the type of variable and its units. The following lines then contain multiple values until the total number equals the number of levels. This ends a section, and the end of the file is marked by the line `*END`.

- **Atmosphere input filename:**
  - INT n : Number of levels in atmospheric state.
  - CHARACTER (*) : `*HGT`, Label for marking start of level altitude information.
  - **Altitude information: units of** `km`
    - N*REAL : N level values.
    - Above line repeated M times, where $M \times N \leq n$.
    - If $MN \pmod{n} > 0$ i.e. the final levels need values then,
$MN \pmod{n}$*REAL : Final level values.
  - CHARACTER (*) : `*PRE`, Label for marking start of level pressure information.
  - **Pressure information:** `hPa`
    - As for altitude.
  - CHARACTER (*) : `*TEM`, Label for marking start of level temperature information.
  - **Temperature information:** `K`
    - As for altitude.
  - Then the gas specifications follow. Valid labels for each gas are those that RFM recognises. (See a list at `http://www.atm.ox.ac.uk/RFM/#gaslist`.)
    Examples are: `H2O`,`CO2`,`O3`,`O2`.
    NOTE: If Ozone is specified and shortwave calculations are desired (wavenumbers 8920–24665 cm^{-1}) then an additional gas should be specified. This has the label name `X65` and ppmv identical to `O3` i.e a duplicate of Ozone.
  - CHARACTER (*) : `*XXX`, Label for marking start of gas concentration.
  - **Gas concentration information:** `ppmv`
    - As for altitude.
  - CHARACTER (*) : `*END`, Label marking end of file.

D.3 Aerosol input structure

Aerosol data is input into the code by a set of files. The enabling of aerosols is by a flag in the shortwave or longwave driver files. There is a definition file, named `aerosols.dat` which lists the number of aerosols and the names of the control files. Each control file contains the aerosol mass density per layer, and references two further files: one listing the optical properties, *except* the phase function, and a second listing only the phase function expanded as coefficients for the series of spherical harmonics.

D.3.1 File structure

- *Aerosol definition file,* `aerosols.dat`*:*
  - INT n : Number of aerosol types,

- $n\times$CHARACTER(*) : List of aerosol control filenames,
  Aerosol control file:
  - CHARACTER(*) : Aerosol optical properties filename,
    Aerosol optical properties file:
    - CHARACTER(*) : Irrelevant data, could be comment,
    - { 6*REAL :
      Wavelength, (μm),
      Extinction coefficient, (m^{2}g^{-1}),
      Single scattering albedo,
      Asymmetry parameter,
      Unused parameter,
      Unused parameter.
      The above line repeated for the number of wavelengths, w}
    - *Note that the wavelengths can be arbitary and should simply span the wavenumber range being calculated. Intermediate wavenumber values are intepolated from this data.*

    NOTE: The order of the lines is in decreasing wavelength, corresponding to increasing wavenumber.
  - CHARACTER(*) : Aerosol phase function moments filename,
    Aerosol phase function file:
    - { REAL($N+1$) : N is the number of phase function moments. The first term is 1.0, the normalised amplitude), then the other N phase function moments.
      This line is repeated for w wavelengths. }
    - *(Additional lines can be added but will be ignored)*
  - INT l : The number of layers for which aerosol density is given.
  - { INT, REAL : Layer number and then aerosol density.
    This line is repeated for l layers }
    The order of layers is not important.

E Steven Rumbold provided documentation

The remaining pages are from PDF files supplied by Steven Rumbold, the previous author of RFMDISORT. They detail how to modify the versions of DISORT and RFM that are publicly available, and then how to use RFMDISORT. Some of the information for the latter is superceeded by the information in this document.

RFMDISORT INSTRUCTION SHEET (Steve Rumbold, 2009-05-29 v1.0)

1. INSTALLATION
2. RUNNING
3. OUTPUTS

1. INSTALLATION

The code can be found in:

`/home/sws06str/APPRAISE/RFMDAERO`

At present, probably best to copy entire `RFMDAERO` directory to a location on your own home drive.

First, you need to set up HITRAN and RFM components (DISORT components are handled automatically).

HITRAN

Go to: `RFMDAERO/HITRAN/`

Here you will find the program `hitbin.f` provided with RFM.

To compile use: `f77 hitbin.f –o hitbin`

If a `HITRAN04.bin` already exists delete it.

Run: `hitbin`

For all but 2 options, the default is required (press Enter). However for “`Input HITRAN ASCII file:`”, you need: `HITRAN04.par`

For “`New binary file: `”, you need: `HITRAN04.bin`

HITRAN is now set up.

RFM

Go to `RFMDAERO/RFM/source/`

Here you need to compile the code. Use: `f77 *.for –o rfm`

RFM is now set up.

RFMDISORT

To set up the RFMDISORT coupling code go to RFMDAERO/CODE/

Make sure the main script is executable (e.g. `chmod 744 rfmdisort.scr`).

Several pieces of auxiliary Fortran code have to be compiled. They are as follows:

```
f90 –o createrfmlayers.x createrfmlayers.f
f90 –o num_freq.x num_freq.f
f90 –o speclims.x speclims.f
f90 –o a_gt_b.x a_gt_b.f
f90 –o add_band_freq.x add_band_freq.f
f90 –o comprfm1.x comprfm1.f
```

rfmdisort_coupler.f requires special attention to provide an accurate calculation (double precision). This is particularly important for the longwave. Note, the following should be input on one line.

```
f90 –xtypemap=real:64,double:64,integer:64
            –o rfmdisort_coupler.x rfmdisort_coupler.f
```

This also compiles DISORT and its components.

Installation is now complete.

2. RUNNING

The code is run by executing the script rfmdisort.scr. This uses the following main driver files:

rfmdisort.drv, rfmdisort_sw.drv and rfmdisort_lw.drv

These are found in: RFMDAERO/CODE/

The first of these contains general information about the run. It also establishes if it is a shortwave, longwave or combined run. The latter two files contain specific information about the separate shortwave and longwave calculations.

If aerosols are required, then some further files are used (found within RFMDAERO/CODE/).

The main aerosol file is: aerosols.dat

This is a user input file. The first line contains a number. This is the number of aerosols required. The subsequent lines each contain filename. This is the filename of the individual aerosols.

The default `aerosols.dat` contains:

```
1
testaerosol.dat
```

So here we have one aerosol and the detail is contained within `testaerosol.dat`. Potentially there could be any number of aerosols.

`testaerosol.dat` contains two filenames on each of the top two lines. The first is a file containing basic aerosol information e.g. SSA, extinction, asymmetry parameter. The second filename contains the moments of the phase function. These are both standard outputs from the Reading Wiscombe Mie setup (however any source can be used if the numbers are put in the correct form).

The third line contains the number of layers that the aerosol is in. Subsequent lines contain the layers and the amount of aerosol in g m^{-2}. Top layer is layer 1.

The contents of the default `testaerosol.dat` is:

```
b237wis
b237wispmom
15
25   0.00759513974639
26   0.07114406806488
27   0.06956091807256
28   0.06361950512513
29   0.07518202721270
30   0.04274299894643
31   0.02565255917514
32   0.02177552486551
33   0.02512755977798
34   0.02301242684192
35   0.01845869098469
36   0.01829784411607
37   0.02082960860944
38   0.01724444602803
39   0.02107780071876
```

The default settings provide a quick “aerosol on” run for part of both the SW and LW spectrum.

3. OUTPUTS

The main outputs are subdivided into shortwave and longwave components as required.

These are found in: `RFMDAERO/CODE/`

Broad band flux information for each level can be found in:

e.g. `bband_flx_sw.dat`

and heating rate information in:

e.g. `bband_heat_sw.dat`

Spectrally resolved flux data is also provided for each level:

e.g. `spec_flux_sw.dat022` for level 22.

MODIFICATIONS TO RFM AND DISORT
(Steve Rumbold, 2009-06-01 v1.0)

For a fresh installation of RFMDISORT, the following changes have to be made to the raw RFM and DISORT code before compiling.

1. RFM
2. DISORT

1. RFM

The following is true and tested for RFM v4.28.

There are three changes that need to be made to the RFM Fortran code. The first is to comment a line asking for a keyboard input. The second is to comment excessive outputs to the screen that slow down the execution time. The third is to change the spectral limits that RFM is expecting. This is by default hardwired to 25000 cm^{-1} and thus shortwave calculations would be limited.

i. Commenting keyboard input line

The code that needs commenting is in `rfm.for` and at lines 115 to 121. This is shown below (after the already commented lines).

```
C
C Prompt for any run identifier label to be appended to all output file
C
      WRITE ( *, * )
     &  'Optional ID to be appended to filenames (<CR>=none):'
      READ ( *, '(A)' ) RUNID
      IF ( RUNID .NE. ' ' ) THEN
        MESSGE = 'R-RFM:    Filename append string='//RUNID
        WRITE ( *, * ) MESSGE
      END IF
```

Comment all these lines e.g.

```
C
C Prompt for any run identifier label to be appended to all output file
C
C     WRITE ( *, * )
C    &  'Optional ID to be appended to filenames (<CR>=none):'
C     READ ( *, '(A)' ) RUNID
C     IF ( RUNID .NE. ' ' ) THEN
C       MESSGE = 'R-RFM:    Filename append string='//RUNID
C       WRITE ( *, * ) MESSGE
C     END IF
```

ii. Commenting excessive outputs to the screen

The following (line 221) in `rfm.for` needs to be commented.

```
          WRITE (*,*) IWID, NWID
```

The new line therefore is:

```
C         WRITE (*,*) IWID, NWID
```

iii. Changing the spectral limits covered by RFM

The final change is to extend the spectral range of RFM in the shortwave. This is line 54 of `spcchk.for`. This is initially (with line 53 also shown):

```
      DOUBLE PRECISION WNUMAX        ! Maximum Waveno.required for RFM
        PARAMETER (    WNUMAX = 25000.0D0 )
```

This has been tested out to 50000 cm^{-1} (or 200 nm). Thus change value as follows:

```
      DOUBLE PRECISION WNUMAX        ! Maximum Waveno.required for RFM
        PARAMETER (    WNUMAX = 50000.0D0 )
```

RFM is now ready for use with RFMDISORT and should be compiled as in the installation notes.

2. DISORT

There are two changes required for DISORT. The first increases the maximum number layers that can be handled in one go. This is hardwired to 6 computational layers and 5 output levels, which is insufficient for many atmospheric problems. The second change comments out excessive output to the screen.

i. Maximum number of layers

Line 357 in `DISORT.f` needs to be changed. This is displayed along with Lines 355, 356, 358 and 359.

```
      INTEGER   MXCLY, MXULV, MXCMU, MXUMU, MXPHI, MI, MI9M2, NNLYRI,
     &          MXSQT
      PARAMETER ( MXCLY = 6, MXULV = 5, MXCMU = 48, MXUMU = 10,
     &          MXPHI = 3, MI = MXCMU / 2, MI9M2 = 9*MI - 2,
     &          NNLYRI = MXCMU*MXCLY, MXSQT = 1000 )
```

Change MXCLY and MXULV as follows:

```
      INTEGER   MXCLY, MXULV, MXCMU, MXUMU, MXPHI, MI, MI9M2, NNLYRI,
     &          MXSQT
      PARAMETER ( MXCLY = 99, MXULV = 100, MXCMU = 48, MXUMU = 10,
     &          MXPHI = 3, MI = MXCMU / 2, MI9M2 = 9*MI - 2,
     &          NNLYRI = MXCMU*MXCLY, MXSQT = 1000 )
```

This allows for 99 computational layers and 100 output levels.

ii. Commenting excessive outputs to the screen

This involves lines 473 to 475 of the raw `DISORT.f` file.

```
      IF( .NOT.PASS1 .AND. LEN( HEADER ).NE.0 )
     &    WRITE( *,'(//,1X,100(''*''),/,A,/,1X,100(''*''))' )
     &    ' DISORT: '//HEADER
```

This is commented out as follows:

```
c       IF( .NOT.PASS1 .AND. LEN( HEADER ).NE.0 )
c      &    WRITE( *,'(//,1X,100(''*''),/,A,/,1X,100(''*''))' )
c      &    ' DISORT: '//HEADER
```

DISORT is now ready to be compiled when RFMDISORT is compiled.