**CLBLM V1.0 USER GUIDE.**

**Part I: *CLBLM-no-scatt* application**

**Atmospheric and Environmental Research (AER)**

**and**

**National Oceanic and Atmospheric Administration (NOAA)**

**Authors:**

Jean**-**Luc Moncet (AER)

Yingtao Ma (AER)

Igor Polonsky (AER) and

Christopher Brodowsky (AER)

Karen Cady-Pereira (AER)

Betsy Berry (AER)

**Last revised:**

May 1, 2023

Table of Contents

[1. CLBLM overview 2](#_Toc133855115)

[2. CLBLM installation 5](#_Toc133855116)

[2.1. Delivery overview 5](#_Toc133855117)

[2.2. Modifying default pathnames 6](#_Toc133855118)

[2.3. Compiling CLBM source code 7](#_Toc133855119)

[2.4. Input spectroscopic data files 7](#_Toc133855120)

[2.5. Test cases 8](#_Toc133855121)

[3. CLBLM user directives 8](#_Toc133855122)

[3.1. CLBLM-no-scatt output product selection 11](#_Toc133855123)

[3.1.1. CLBLM-no-scatt output RT products 11](#_Toc133855124)

[3.1.2. Using CLBLM to compute monochromatic optical depths only (OD-only mode) 12](#_Toc133855125)

[3.2. Spectral convolution of RT products 13](#_Toc133855126)

[3.3. Spectral grid specification 15](#_Toc133855127)

[3.3.1. RT product spectral grid 16](#_Toc133855128)

[3.3.2. Output optical depth spectral grid in the OD-only mode 16](#_Toc133855129)

[3.4. Activating the NLTE mode 17](#_Toc133855130)

[3.5. Thermal emission and radiation source controls 17](#_Toc133855131)

[3.6. Radiative Flux Controls 18](#_Toc133855132)

[3.7. Path-calculation controls 19](#_Toc133855133)

[3.7.1. Vertical RT grid 19](#_Toc133855134)

[3.7.2. Airmass scaling approximation 19](#_Toc133855135)

[3.7.3. path-calculation-ctrl JSON group 20](#_Toc133855136)

[3.8. Optical depths calculation control flags 21](#_Toc133855137)

[3.9. Scene file and scene selection 22](#_Toc133855138)

[3.10. Overriding scene-file viewing geometry information 23](#_Toc133855139)

[3.11. Convolution-only mode 23](#_Toc133855140)

[4. Appendix A: CLBLM units and conventions 25](#_Toc133855141)

[5. Appendix B: CLBLM input scene file format and content 25](#_Toc133855142)

[6. Appendix C: CLBLM output data file format and content 27](#_Toc133855143)

# CLBLM overview

CLBLM v1.0 is the first official release of a modernized version of the AER LBLRTM line-byline model. CLBLM v1.0 is written in Fortran 2003 and currently contains one executable program *CLBLM-no-scatt*, which includes the capabilities of the original core LBLRTM algorithm (radiance and transmittance calculations in non-scattering atmospheres) as well as Jacobian calculations and radiative fluxes into a single application. A further application, CLBLM-multi-stream, is under development, and will provide the ability to compute radiances in a plane parallel scattering atmosphere.

CLBLM-*no-scatt* is built around the fast LBLRTM Voigt line shape convolution algorithm, modified to avoid the issue of under-sampling of spectrally-isolated narrow absorption lines; it integrates other key elements of the original LBLRTM code in the form of separate libraries: atmospheric path calculations, spectral sampling interval calculations and post-processing functions (spectral convolution with built-in idealized instrument functions or user-supplied instrument functions). It consists of a wrapper performing basic user-interface functions, such as reading in the user-directive file and the multi-scene input scene file, and then calls a radiative transfer driver to perform the requested tasks.

Input atmospheric profiles/geometry

Layer quantities (on RT grid)

OD calculation flags

Post-processing parameters

Input user-directive

file

Input scene file

Scene reader

User-directives reader

Fortran structures definition and arrays declaration

**CLBLM wrapper**

Scanning/

Filtering

SetDV

CLBLM-Path

ODLAY

Loop over scenes

**RT-Driver**

**Scene reader**

**Figure 1**:

CLBLM-Path

SetDV

ODLAY

Filter/scan

Start

**Layer Loop**

Layer merge

Compute layer emission

Spectrally interpolate

Compute layer transmittances

**Figure 2:**

# CLBLM installation

## Delivery overview

The complete CLBLM software is available at https://github.com/AER-RC/CLBLM.

The package contains:

* the CLBLM source code (main programs, modules and subroutines)
* a customizable tool (scene\_writer) to help users create CLBLM scene files from LBLRTM TAPE5s
* a customizable tool (build\_solar) to create CLBLM solar irradiance files
* a makefile for CLBLM, scene writer and build\_solar.
* test input scenes and user directive files, and output test results.

All CLBLM input and output data files, with the exception of the input spectroscopic data (Sec. 2.4), are in NetcDF4 format. Before compiling the code, users must make sure they have the NetCDF-4 Fortran library (version 4.4.1 or higher) available on their system.

Follow the instructions for downloading the software and necessary files on the GitHub page. This process will create a CLBLM directory (hereafter, CLBLM root directory or working directory). The directory structure shown in Figure 3 will appear under the root directory. This is the default directory structure used by the CLBLM applications. Users may want to modify this structure and adopt their own directory structure, in which case they have to ensure that pathnames hardcoded inside the makefiles and the Fortran source code have been correctly set before compiling the code (Sec. 2.2).

The source code for the *CLBLM-no-scatt* application, the NetCDF scene writer, and build\_solar are contained in the src/ subdirectory (see Table 1 for more details). The src/ directory will also hold the object code created by the compiler (see Sec. 2.3). The clblm\_out/ directory is reserved for the output of the CLBLM runs. The static CLBLM input files containing the spectroscopic databases (molecular line absorption parameters and cross-section data) and solar irradiance data are located in clblm\_data/spectroscopy/ and clblm\_data/solar\_irradiance/, respectively. The scratch/ directory is only used by CLBLM to temporarily store partial RT results at run time, when the data volume exceeds available CPU memory space (memory requirements are application dependent). These files are automatically deleted after being closed by the program.

Table 1: ./src/ subdirectories content

|  |  |
| --- | --- |
| **Subdirectory** | **Content/usage** |
| app/ | *CLBLM-no-scatt* and NetCDF scene file writer main wrappers |
| clblm-src/ | CLBLM-specific modules and subroutines |
| json/ | JSON user-directives reader |
| scene-tool/ | scene file writer’s subroutines |
| common/ | subroutines common to CLBLM and scene file writer |
| modules/ | CLBLM module .mod files created by compiler (Sec. 2.2) |



Figure 3: Default CLBLM directory structure. The CLBLM root directory is identified as ‘./’.

## Modifying default pathnames

Directory names for the Fortran source code (see Table 1) are hardcoded inside the delivered makefile. If the source code has been moved into a different directory structure, users have to first ensure that the makefile accesses the correct folders. The same applies to the destination folders for the object and ‘.mod’ files as well as the executables created by the compiler (Sec. 2.3).

The main directory and file names for the CLBLM input databases and temporary output data as well as default[[1]](#footnote-1) directory names for the CLBLM output files (other than temporary scratch files) and user inputs are all hardcoded in the applications source code, inside *clblm\_Config.f90*. Table 2 provides the complete list of hardcoded filenames and directory names that may be modified by the user (and the variables that contains those names).

Table 2: Default pathnames for CLBLM databases (no shading); user files (input scene files, cloud/aerosol and surface property databases and filter files), in green; CLBLM output data folder, in blue.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable name** | **Default value** | **File format** | **Comment** |
| lineFile | 'clblm\_data/spectroscopy/TAPE3' | Binary | CLBLM line parameter file converted from AER line database using program LNFL) |
| xsFilePath | 'clblm\_data/spectroscopy/xs/' | N/A | Cross-section data and FSCDXS index file (all in ASCII format) folder |
| nlteStatPopFile | 'clblm\_data/TAPE4' | ASCII | Needed for NLTE |
| solarFilePath | 'clblm\_data/solar\_irradiance/' | N/A | Contains case-specific user-built solar irradiance file: ‘SOLAR.RAD.nc' (in NetCDF format) – see Sec. 3.5. |
| sceneFile | 'user\_archive/scene\_file/scenes.nc' | NetCDF |  |
| Cloud/aerosol database | 'user\_archive/cloud\_aerosol/' | N/A |  |
| Surface database | 'user\_archive/surface/' | N/A |  |
| filterFunctPath | 'user\_archive/sensor/' | N/A |  |
| clblmOutPath | 'clblm\_out/' | N/A | CLBLM output files directory |

## Compiling CLBM source code

A single *makefile* located in the CLBLM root directory is used to create the object code and build the executables for CLBLM, the CLBLM scene writer and the CLBLM build solar. One can use the ‘*make all*’ command to simultaneously build the CLBLM, scene writer and build solar executables or use ‘*make clblm*’ or ‘*make scene\_writerl or ‘make build\_solar’*’ to build CLBLM, scene writer, or build solar separately. The object files produced during the build process share the same src/ subdirectories as the Fortran source files they have been created from. All executables are placed in the CLBLM root directory. ‘*.mod’* files needed for linking CLBLM subroutines with a main application (CLBLM and external user-application) are also created as part of the build process, and placed in a src/ subdirectory named modules/.

Note that prior to executing the *makefile*, users need to correctly set the variables “NETCDF\_LIBRARY\_LINK” and “NETCDF\_INCLUDES” to point to their local version of the NetCDF library and NetCDF library include modules. In addition, they need to set the “CL” variable to use the desired compiler and compiler options. CLBLM can be compiled with GFortran (version 4.4.6 and beyond) and Intel Fortran Compiler (tested with version 17). Use “make all” command to simultaneously build CLBLM and the scene writer. Use “make clblm” or “make scenetool” to build CLBLM or the scene writer separately.

## Input spectroscopic data files

In this beta version of CLBLM v1.0, the input files containing the line parameters (binary ‘TAPE3’ file) and the ASCII absorption data files for the cross-sections are identical in content and format to the ones used by LBLRTM[[2]](#footnote-2). The first file is created by the program LNFL (downloaded separately from CLBLM from <http://rtweb.aer.com/lnfl_frame.html>) with the complete AER line parameter data base (available from <http://rtweb.aer.com/line_param_frame.html>). To build and run the LNFL program follow the instruction provided on the AER RT website. In order to use the TAPE3 file created by LNFL in CLBLM with the default settings, the file should be moved to clblm\_data/spectroscopy/TAPE3.

The cross-section data files as well as the cross-section index file, FSCDXS are provided via download from GitHub(Note that there is no longer any distinction between ‘line molecules’ and ‘cross-section molecules’ in the input scene file. The way each molecule is being treated in the CLBLM calculations (line parameters or cross-sections) is specified through the FSCDXS file. The FSCDXS file used by CLBLMe) includes an extra column containing an integer flag that indicates for *each cross-section molecule and each spectral interval where a given cross-section molecule is optically active* whether line parameters are also available (0: no line parameters available) and, if line parameters are available, the preferred method for including the molecule in the optical depth calculations for that molecule (1: use line parameters; 2: use cross-section data).

## Test cases

Two test cases area provided for the purpose of verifying that the CLBLM software has been correctly installed on the users’ end. The tests cases each consist of an input user directive file located in the root directory and CLBLM output files in clblm\_out/ and two scenes file in user\_archive/scene\_files/. The user directive file is the same for both cases.

The results were produced with a TAPE3 (little-endian binary file located in clblm\_data/spectroscopy/) containing parameters for all HITRAN molecules covering the spectral range 300-3500 cm-1 created using LNFL v3.2 and the AER aer\_v\_3.8.1line parameter database.

When comparing LBLRTM and CLBLM output it is imperative that all control parameters be set the same way. Some defaults are set differently, so it is safest to check the default values for the parameters not set in the JSON file by examining the clblm\_Config.f90 file.

**2.6 Running CLBLM and the scene\_writer from the command line**

The scene\_writer tool extracts the profile information from one or more LBLRTM TAPE files and creates a scene file. To run the scene\_writer:

*scene\_writer* TAPE5\_ii TAPE5\_ij user\_archive/scene\_files/scenes.nc

Note that the last argument is the path/filename of the scene file CLBLM is hardwired to expect. The user can of course use a different path/filename but will have to copy the file to this location.

To generate a solar irradiance file, run build\_solar by typing:

*build\_solar solar\_config.json*

Note that the last argument is the configuration file (named by the user) used by the build\_solar executable.

To run CLBLM type:

*clblm*

# CLBLM user directives

The language used for entering the CLBLM user-directives is JSON (JavaScript Object Notation), e.g., <https://www.youtube.com/watch?v=wbB3lVyUvAM>. The main elements of the JSON language are key/value pairs. A key is a recognizable keyword (enclosed in *straight* double quotes) which is assigned a value in the user-directive file. Values are separated from the keys by a colon and can be a character string (in *straight* double quotes), a logical (true or false), a single integer or real number, or an array of values of the same or different types enclosed in square brackets, e.g., "key": [true, 5, 1.3e-1, "value", 0.0]. Keys may be organized into groups (or objects). In this case, the keys are enclosed in *curly braces* separated from the group name by a colon, e.g., "group": {"key1": value, "key2": value, key3": value, etc.}.

The set of input CLBLM user-directives can be as simple as shown in the example below:

{

"clblm-out": {"mono rad":"myfolder1/myradfile", "convolved tx**-**profile":"myfolder2/"},

"spectral**-**convolution**-**flags": {"FFT": true, "function ID": 2, "HWHM": 4.0e-02, "boxcar-width-ratio": 0.005},

"output-spectral-grid": {"from": 1000.0, "to": 2500.0, "DV": 2.0e-02}

}

In this example, the user is requesting that CLBLM computes and outputs *monochromatic* radiances and transmittance profiles convolved with a 0.04 cm-1 wide triangular function over the 1000-2500 cm-1 spectral domain using FFT convolution. The convolved transmittance spectra are sampled at uniform 0.02 cm-1 intervals. In the case shown here, calculations are performed in LTE mode (default mode) without solar contribution included, for all scenes contained in the file named ‘. /scenes.nc’ (default scene file name) using the default settings for optical depths calculations. The radiances are written out to a file named ‘myradfile’ in the directory ‘myfolder1/’ and transmittances in files with default filenames in the directory ‘myfolder2/’ (see Appendix A for CLBLM output data file content and default naming conventions). Note that the entire set of JSON directives must be enclosed in curly braces. Any text inserted outside those braces is ignored. All group/key names are *case insensitive* and blank spaces as well as <CR> are ignored except when within quotes. If desired, users can use a JSON validator (<https://jsonlint.com/>) to verify the correctness of their syntax.

Table 3 provides the complete list of JSON keys and groups of keys used by *CLBLM-no-scatt*. Detailed explanations about the meaning of each key and their allowed values are provided in Sec. 3.1 through 3.10. Note that the JSON group **"clblm-out"** appearing in the above example is *mandatory* when starting calculations from scene data. The group **"spectral convolution-flags "**must be included in the user-directive file if at least one output RT product is spectrally convolved. The group **"output-spectral-grid"** containing output product spectral grid specification is used by CLBLM in all situations except when the output product is convolved with a user-defined instrument function. Other CLBLM JSON groups are optional and are used either for overriding the *CLBLM-no-scatt* default settings for RT and optical depth calculations or for performing scene manipulation functions such as scene selection and assigning different viewing geometries to selected scenes without having to edit the input scene file. The last group listed in Table 3 is a special group that activates the CLBLM spectral convolution function only and applies the convolution to existing RT products.

Note that the order in which groups are listed in the user-directive file and the way keys are ordered within groups do not matter. However, in order to make the file easier to read and reduce the chances of mistake we recommend following a logical order with top-level items (e.g., list of output products and post-processing options) first, keys or groups of keys whose presence is required based on the activated functions second, followed by optional settings. When individual optional keys within a group or entire groups of keys are omitted default CLBLM values are used. If a key or group of keys that do not impact the outcome of the CLBLM calculations is present in the directive file, it is simply ignored.

Table 3: Summary of CLBLM JSON groups and keys.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Group** | **Primary keys** | **Comments** | **Secondary keys** | **Comments** |
| **"clblm-out"** | "rad", "total-tx", " tx-profile", "jacobians" | If preceded by **‘mono’**: monochromatic data requested  If preceded by **‘convolved’**: spectral convolution is applied | "jacobian-list" | Only used if "jacobians" key is included |
| "od" (OD-only mode) |  | N/A | |
| **"spectral-convolution-flags"** (only used if one or more output RT product is convolved | "FFT", "function ID", "function-params", "HWHM", "averaging-width" |  | "boxcar-width" | (activates pre-boxcaring) |
| "filter-file" | Must be included if convolving with user-supplied instrument function |  |  |
| **"output-spectral-grid"** | "from", "to", "DV" | Not used if convolving with user-supplied instrument function | "grid-type" | (OD-only mode) |
| **"nlte"** (single key) | N/A | | N/A | |
| **"rt-flags"** | "thermal-source", "linear-in-tau", "solar-source", "solar-cnst", "julday" |  | "linear-in-tau" |  |
| **“flux-flags”** | “flux\_flag”, “dv\_flux”, “nang” |  |  |  |
| **"solar-irradiance"** | "option" |  | "cycle-frac", "facula-var", "spot-var" | Ignored if option = 1 |
| **"path-calculation-ctrl"** | "RT-grid", "airmass-scaling", "reference-path", "v-refrac" |  |  |  |
| **"od-flags"** | "lines-contribution", "continuum-contribution", "line-rejection" |  | "dptmin", dptfac" | (line rejection on) |
| "p-convolution", "collision- partners- broadening" | (lines contribution on) |
| "continuum-scaling" | (continuum contribution on) |
| **"scenes"** | "scene file", "nscenes", "scene-ID" |  | N/A |  |
| **"geometry"** | **"**obs-altitudes", "view angles" |  | N/A |  |
| **"clblm-in"** | "rad", "total-tx", " tx-profile", "jacobians" |  | N/A |  |

## CLBLM-no-scatt output product selection

The selection of CLBLM output products is entered in the JSON group **"clblm-out"**. Unless CLBLM is only used to convolve existing RT products (Sec. 3.10), this group is mandatory and must *always* be included in the user directive file.

The content and format of the CLBLM output files are described in Appendix C.

Users can use *CLBLM-no-scatt* to generate different combinations of monochromatic or spectrally convolved radiative transfer (RT) products (Sec. 3.1.1), or monochromatic optical depths only (this mode is described in Sec. 3.1.2).

### CLBLM-no-scatt output RT products

The RT products generated by the *CLBLM-no-scatt* application, are:

1. Radiances at specified observer level and viewing angle ("rad").
2. Total along-viewing-path transmittances ("total-tx") between observer and surface, if looking down, or between observer and top of the atmosphere (TOA), if looking up.
3. Profile of cumulative transmittances ("tx-profile") between observer and all RT levels (including surface) below observer, if looking down, or between observer and all RT levels (including TOA) above the observer, if looking up.
4. Jacobians ("jacobians"): derivatives of observed radiances with respect to atmospheric temperature or logarithm of concentration for selected molecules.

In the above list, the JSON key for each RT product is indicated in red (in parentheses).

The content of the **"clblm-out"** differs depending on which internal RT driver (*RT-basic* or *RT-Jac*) is called by CLBLM to execute a task. CLBLM automatically selects the driver based on whether or not Jacobians are included in the list of output products. If Jacobians are not requested, the *RT-basic* driver will be called to produce either radiances or transmittances alone or radiances plus transmittances[[3]](#footnote-3). The following provides an example of valid *CLBLM-no-scatt* output product selection directive for this case:

"clblm-out": {"mono rad":"myfolder1/myradfile", "convolved tx**-**profile":"myfolder2/"}.

The keys in the **"clblm-out"** group contain the product names and the character string assigned to each product key (unless empty, in which case default file name and location are used) are user-provided output file directory path or file’s root name (see below). The qualifiers ‘mono’ or ‘convolved’ preceding the product name indicate whether the product should be output in monochromatic form or be convolved with a specified instrument function (Sec. 3.2).

If Jacobians are requested, users have the option to output Jacobians alone or Jacobians and radiances, e.g.,

"clblm-out": {"convolved jacobians":"myfolder3/", "jacobian-list": ["T","O3","CH4"]}, where the character array assigned to the "jacobian-list" key provides the list of desired Jacobians (profile of atmospheric temperature, logarithm of concentration for selected molecules, or…), or Jacobians and radiances,

"clblm-out": {"convolved jacobians":"myfolder3/", "mono rad":" ", "jacobian-list": ["T","O3","CH4"]}.

Jacobians with total transmittance or cumulative transmittance profile are *not allowed* combinations.

If any listed product is convolved, the **"clblm-out "** group *must* be followed by the **"spectral-convolution-flags"** group containing all the information about the function to be applied (see Sec. 3.2 for more details). In this case, the **"output-spectral-grid"** group (Sec. 3.3) provides the bounds of the spectral domain covered by the output product and the spectral sampling interval (DV), e.g.,

"output-spectral-grid": {"from": 1000.0, "to": 2500.0, "DV": 2.0e-02}.

If no output product is convolved, i.e., all outputs of a given CLBLM run are monochromatic then the information contained in **"output-spectral-grid"** applies to the monochromatic spectra.

As mentioned above, the value assigned to each **"clblm-out"** product key overrides the default output data file location or root file name for the selected product (see Appendix C for CLBLM output data file default file naming convention). Users have the option to only enter the location of the folder that will hold the output data (directory and subdirectory names should be followed by “/”, e.g., "mono rad": "myfolder1/mysubfolder2/", where *myfolder1*/ is a subdirectory of the working directory), the *root* file name only (e.g., "mono rad": "myradfile") or both the *root* file name and location of the output directory. As explained in Appendix C, CLBLM will automatically append the qualifier ‘\_mono’ for monochromatic output, geometry information (if applicable), scene number and, for profile data (i.e., Jacobians or transmittance profiles), level or layer number to the user-supplied root file name to construct the complete name of the files containing the output data.

### Using CLBLM to compute monochromatic optical depths only (OD-only mode)

In this mode, CLBLM outputs the optical depths computed along the vertical path for all layers on the RT grid. This mode is triggered by including the "od" key in the **"clblm-out"** group, without any other product appearing included in the group, followed by the spectral grid information in the "**output-spectral-grid"** group (see Sec. 3.3.2).

## Spectral convolution of RT products

The “post-processing” options available in CLBLM are the original LBLRTM “scan” (in wavenumber or FFT space) and “filter” options. The LBLRTM “interpolation” option is no longer explicitly available to the user.

CLBLM can convolve monochromatic spectra either with one of several *built-in* idealized instrument functions (Table 4 and Table 5) or with a custom instrument function read in from a user-supplied external file. Note that the numbering scheme for the functions is different from that used by LBLRTMThe choice of scanning function and the required convolution parameters are entered through the JSON group **"spectral-convolution-flags"**, e.g.,

"spectral**-**convolution**-**flags": {"FFT": true, "function ID": 2, "HWHM": 5.2e-02, "averaging-width": 0.005},

(see Table 6 for the list of keys and allowed key values). This group *must* appear in the user-directive file if one or more output RT product listed in the **"clblm-out"** group (see Sec.3.1) needs to be convolved. Unless the instrument function is read in from file (function ID =0), convolution directives *must* be accompanied by a separate group defining the spectral domain covered by the convolved product and the spectral sampling interval (Sec. 3.3). Note that including the key "averaging-width" as in the above example triggers degrading the resolution of the intermediate monochromatic data product generated by CLBLM by convolving it with a boxcar function with specified width *prior to* performing the convolution with the desired instrument function. This option is highly recommended for convolved Jacobian product as it avoids having to temporarily save monochromatic Jacobians to disk and having to read them back in when the number of RT grid levels or the spectral domain of the calculations are large. By default, the result of the final convolution is automatically corrected for the impact of this “pre-boxcaring” by performing an inverse convolution in Fourier space.

Table 4: CLBLM scanning and apodization function selection. In green: functions available with both LBLRTM SCANFN and FFTSCN functions. Other functions are only available with FFTSCN. Variables used in this table are defined in Table 5. Blue shading: option available for convolution in wavenumber space only; green: both wavenumber and Fourier space; no shading: Fourier space only.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** |  | **Function** | **Wavenumber space** | **Fourier space** | **a/HWHM** |
| 1 |  | Boxcar |  | N/A | 1.0 |
| 2 |  | Triangle |  |  | 2.0 |
| 3 |  | Gauss |  |  | 0.849322 |
| 4 |  | Sinc2 |  | (triangle) | 2.257609 |
| 5 |  | Sinc |  | (rectangle) | 3.314800 |
| 6 |  | Beer |  |  | 2.100669 |
| 7 |  | Hamming |  |  | 2.195676 |
| 8 |  | Hanning |  |  | 2.0 |
| 9 |  | Norton-Beer† | Not implemented | 0 | 2.57027 |
| 10 |  | 2.36771 |
| 11 |  | 2.07176 |
| 12 |  | Brault | Not implemented | where | variable |
| 13 |  | Kaiser-Bessel | Not implemented | where is modified Bessel function of the first kind, order 0 and | variable |
| 14 |  | Kiruna |  | Not defined | variable |

† Values of  for 3 available apodization strengths: weak (9): 0.384093, -0.087577,0.703484, 0., 0., medium (10): 0.152442, -0.136176, 0.983734,0.,0. and strong (11): 0.045355, 0., 0.554883, 0., 0.399782.

Table 5: Description of the variables used in Table 4.

|  |  |
| --- | --- |
| Variable | Definition |
|  | Wavenumber in cm-1 |
| x | Optical path difference in cm |
| u |  |
| *u*’ |  |
| a | = where *L* is the maximum optical path difference of an equivalent interferometer (“maxOPD”) |
| FZ |  |
| CR | Critical value of ratio: (V1-V2)/HWHM |

Table 6: Mandatory keys are in bold. In blue: keys relating to built-in functions *only*. In green: keys relating to user-supplied instrument function *only*. Keys in black are common to both built-in and user-defined functions.

|  |  |  |
| --- | --- | --- |
| **Key** | **Description** | **Default value** |
| "FFT": | true; false | N/A |
| "function-ID" | Integer from 0 to 14, containing ID of selected built-in scanning function (see Table 3 for list of available options). If function ID =0: the filter function is read in from external file. | N/A |
| "function-params" | Real array: [*p*, **off, *c*1, *c*2] † (see Table 3) |  |
| "HWHM" | Function half-width at half-maximum | N/A |
| "averaging-width" | Non-negative real number indicating the width (in cm-1) of the windowto be used for the pre-averaging of monochromatic spectra †† | N/A |
| "filter-file" | Character string containing full path name of user-supplied filter file | Must be provided if function ID = 0 (see xxx for file format). |

† *p*: user input parameter for function-ID=12 (0<p<1) and 13 (2<p<4); **off, *c*1 and *c*2: parameters for function-ID=14.

†† Adjusted internally to be an integer multiple of the DV of the input monochromatic spectrum (if the "averaging-width" key is omitted or if the entered width is smaller than DV, no pre-averaging is performed).

## Spectral grid specification

The lower and upper bounds (V1 and V2) of the spectral domain covered by the CLBLM output *RT product* and the spectral sampling interval (DV) are controlled through the JSON group **"output-spectral-grid"**, e.g.,

"output-spectral-grid": {"from": 1000.0, "to": 2500.0, "DV": 2.0e-02}.

All parameter values are real numbers in wavenumber units (cm-1). The rules that apply when CLBLM is used to generate transmittances, radiances or Jacobians are described in Sec. 3.3.1. In the *OD-only* mode, users have additional control on how DV is computed in each atmospheric layer (Sec. 3.3.2).

### RT product spectral grid

The parameters included in **"output-spectral-grid"** are interpreted differently by CLBLM depending on the types of product listed in **clblm-out"**:

1. *at least one* of the RT products listed in **"clblm-out"** is convolved with
   1. one of the built-in idealized instrument functions (Sec. 3.2), then the spectral grid parameters apply to the convolved product(s),
   2. a *user-supplied* custom instrument function (Sec. 3.2): the spectral grid parameters are part of the instrument function definition and the **"output-spectral-grid"** group, if present in the user-directive file, is *ignored* by CLBLM.
2. all RT products listed in **"clblm-out"** are monochromatic: **"output-spectral -grid"** defines the output spectral grid used for the monochromatic products. In this case, all data (including profile quantities such as cumulative transmittance profiles and Jacobians) is output at the same DV. In the *OD-only* mode, users can also control the dependence of DV on altitude (Sec. 3.3.2).

In cases 1a and 1b, additional monochromatic products listed in the **"clblm-out"** group are output on the spectral grid generated internally by the RT drivers: the V1 and V2 of the monochromatic product include a margin on both ends of the specified spectral domain to avoid truncation of the instrument function at the edges of this interval and the DV in each atmospheric layer depends on the specific RT driver used. In case 2, if the DV key is omitted, then DV is set by the RT driver. Otherwise, the CLBLM product is resampled to the user-supplied DV prior to being output. Like in LBLRTM, the entered DV value should be no larger than 1.2 × the calculated DV for the highest layer along the radiation path[[4]](#footnote-4). Not meeting this criterion may result in the output monochromatic spectra being under-sampled and CLBLM will issue a warning.

### Output optical depth spectral grid in the OD-only mode

Unlike in the case of RT product generation covered in the previous section, when running CLBLM to produce optical depths only, users can control the vertical dependence of DV. In the LBLRTM heritage code, three options are available to the user for setting the layer-dependent DV of the *monochromatic* ODs output by the *ODLAY* module:

* Exact DV computed based on layer pressure, temperature and water vapor amount (for water vapor self-broadening correction), without any adjustment.
* DV adjusted so the ratio of DVs in adjacent layers, DV (upper layer)/DV (lower layer), is one of five different ratios (1/1, 1/2, 2/3, 4/5 or 5/6).
* Vertically uniform DV: optical depths for all layers are interpolated to the finest DV along the radiation path prior to being output.

In CLBLM, the type of DV grid used internally for the monochromatic ODs by *ODLAY* is set by the RT drivers in such a way as to maximize computational efficiency (the *RT\_basic* driver uses adjusted DVs only while *RT\_Jac* operates on a vertically uniform spectral grid). When outputting monochromatic OD’s *only*,users can select one of the vertical grids listed above through the JSON key "grid-type". If "grid-type" is set to “exactDV”, the exact DV is used for each layer. If set to “adjustedDV”, then the adjusted DV is used (default for *RT-basic*) and if set to “uniformDV”, optical depths are output on a vertically uniform spectral grid (default for *RT-Jac*). The same rules provided in Sec. 3.3.1 apply for setting the DV of the highest layer along the radiation path. Note that when not in the *OD-only* mode, the "grid-type" key, if present in the **"output-spectral-grid"** group, is ignored.

When generating monochromatic ODs, CLBLM outputs the OD for each layer on a different file, If the total optical depth is desired the user should set “grid-type” to “uniformDV”, then integrate the ODs over all layers.

## Activating the NLTE mode

By default, Thermodynamical Equilibrium (LTE) is assumed for the *CLBLM-no-scatt* optical depths, radiances and transmittances calculations. Activating the Non-Thermodynamical Equilibrium (NLTE) mode is achieved by simply inserting the single key/value pair,

"nlte": true

in the input JSON directives. Note that the NLTE mode is not available for Jacobian calculations. In this case, the "nlte" flag if present is simply ignored.

## Thermal emission and radiation source controls

The default settings for the radiation sources and treatment of the layer thermal emission in the RT calculation can be overridden using the JSON group **"rt-flags"**.An additional group, **"solar-irradiance"**, is needed when exercising available options for solar variability. The allowed keys and key values in these groups are shown in Table 7. The following example of CLBLM directive,

"rt-flags": {"linear-in-tau": 2, "solar-source": "NRL", "solar-cnst": 1370.25, julday = 182}

modifies the default CLBLM settings (Table 7) to include the solar source (together with thermal) and to replace the formula used by LBLRTM for the linear-in tau approximation [Clough et al., 1992] by the so-called standard formula in the RT calculations. The standard expression is obtained by assuming that the Planck function is of the form  in the vertical integration of the Planck emission across a layer.

Table 7: "rt-flags" keys. Underlined values indicate the CLBLM default.

|  |  |  |
| --- | --- | --- |
| **Key** | **Description** | **Values** |
| "thermal-source" | Turns thermal radiation source ON or OFF | true; false |
| "linear-in-tau" | Selects the approximation for the thermal emission in a layer (0: isothermal layer, 1: LBLRTM linear-in- formulation, 2: standard linear-in- formulation - see text) | 0; 1; 2 |
| "solar-source" | Selects the spectral solar irradiance data to be used in solar contribution calculations: Example: "NRL" (NRLSSI2). If ignored or set to " ", solar-source is turned off. | Solar off |
| "solar-cnst" | Solar constant in W.m-2. If ignored the default value is used (1360.85) |  |
| "option" | Integer (1,2,3). Applies only if "solar-source"= "NRL". If "option" = 3, value entered in for solar constant in "rt-flags" is ignored. | 1, 2, 3 |
| "cycle-frac" | Fraction of the way through the mean 11-year cycle with 0 and 1 defined as the minimum phase of the solar cycle; applies only if "solar-source"= "NRL" and "option"= 2 | ? |
| "facula-var" | Solar variability scaling factors or indices for facular cycle | ? |
| "spot-var" | Solar variability scaling factors or indices sun spot cycle | ? |

If the solar source is turned on, the user needs to create a input solar irradiance filefrom one of the two available solar source functions (build\_comb\_solar\_rad\_avg\_50000plus.nc;build\_comb\_solar\_rad\_multi\_comp\_50000plus.nc; This is accomplished with the build\_solar executable and config file (i.e. solar\_config.json).. The set of input Build\_Solar user-directives is shown in the example below and specified in Table 8.

"inputs": [{"path": "/project/p2326/broot/build\_comb\_solar\_rad\_avg\_50000plus.nc",

"start-wavenumber":22000.0, "end-wavenumber":24000.0 }],

"path": "/project/rc/rc1/clblm\_testbed6/clblm\_dir/clblm\_data/solar\_irradiance/SOLAR.RAD.nc"}

Table 8: Summary of Build\_Solar JSON groups and keys

|  |  |  |
| --- | --- | --- |
| Groups | Keys | Comments |
| **“inputs”** | “Path”, “start-wavenumber”, “end-wavenumber” | Path specified for solar-source function |
| **“path”** |  | Path for generated solar irradiance file |

The solar source functions are derived from the Naval Research Laboratory Solar Spectral Irradiance (NRLSSI2) Model (reference?). It accounts for increases in solar irradiance from bright faculae and decreases in solar irradiance from dark sunspots. When using the multi\_comp (3-component) solar source, an additional group should be added to the CLBLM JSON directives to include variability in sun brightness based on an average 11-year solar cycle[[5]](#footnote-5). This option is only available with multi-component solar irradiance data set:

"solar-irradiance": {"option": 1, "cycle-frac":”0.5” , "facula-var":”0.85” , "spot-var":”1.2” }.

## Radiative Flux Controls

Radiative fluxes can be obtained by setting clblm-out to “rad” and seeting the keys in the flux-flags groups (see Table 9) Radiative fluxes are calculated using first-order quadrature with weights described in Clough et al., 1992. The user can specify 1, 2, or 3 quadrature angles to be used for the fluxes and the the spectral interval (dv\_flux) for the output radiative fluxes The radiances values are then summed over this flux spectral interval.There are two options for specifying the surface term, specular or Lambertian; this is specified in the scene file by setting the sfcPropInputMode variable. ,

Example: "flux-flags": {"flux\_flag": true, "dv\_flux":1.0, "nang":3}.

Table 9: "flux-flags" keys. Underlined values indicate the CLBLM default.

|  |  |  |
| --- | --- | --- |
| **Key** | **Description** | **Values** |
| "flux\_flag" | Turns the flux calculation ON or OFF | true; false |
| "dv\_flux" | The spectral interval for the flux calculations. |  |
| "nang" | Number of quadrature angles to use for the radiative flux calculation. | 1; 2; 3 |

## Path-calculation controls

In CLBLM, the choice of vertical grid used for the numerical integration of the radiative transfer equation (RTE) as well as the use of airmass-scaling approximation for molecular amounts, to accelerate optical depth calculations along thermal and solar radiation paths, are controlled by the JSON group **"path-calculation-ctrl"**.

### Vertical RT grid

The vertical integration of the RTE is performed numerically on a vertical grid (RT-grid) whose levels may be different from the levels used for the input atmosphere specification (so-called ‘input grid’). Three options are available for defining the RT grid:

1. RT calculations for all scenes contained in a given scene file are performed using a same set of fixed pressure levels. The levels are included in the scene file header (see Appendix B).
2. RT levels are determined automatically by CLBLM prior to performing the path calculations (*auto-layering* option)
3. The input grid levels are used for the RTE integration

Specifying which RT grid should be used by CLBLM is done through the JSON key "RT-grid" (see Sec. 3.6.3), where the integer value assigned to this key (1, 2 or 3) indicates the desired option among those listed above.

### Airmass scaling approximation

In the LBLRTM heritage code, air-density-weighted temperature, pressure and molecular amounts in a layer on the RT-grid are performed by numerically integrating temperature, pressure and molecular concentrations along the path followed by the radiation. Atmospheric ray-tracing calculations include effects of refraction and sphericity.

Unlike LBLRTM, CLBLM has the capability to performs full RT calculations with both solar and thermal sources and surface reflection included, in a *single* run. For down-looking sensors, this requires having to perform path calculations along up to three different segments (Figure 4). Because average layer temperatures and pressures produced by the original LBLRTM layer integration scheme are path dependent and amounts for individual molecules do not scale with path length within the layer, “exact” RT calculations require different Voigt line shape convolutions for each path and each layer, which can be time consuming.



Figure 4: Figure illustrating the different radiation path segments used internally in the CLBLM RT calculations for a down-looking sensor: U: path followed by the radiation reflected at the surface and emitted by the surface in the direction of the sensor (extended to TOA); D: downwelling thermal radiation path from TOA to surface; S: path followed by the attenuated direct solar beam. Over specularly reflecting surfaces, segment D, S and U intersect the surface at the same angle. For a sensor looking up, there is no U-segment and unless the sun is directly in the sensor’s line of sight (S=D) solar radiation does not contribute to the observed radiance.

CLBLM provides the ability to perform a single line shape convolution for all the paths by invoking the so-called airmass- scaling approximation. In this approximation, the same average layer temperature and pressure computed along a reference path is used for path segments “U”, “D” and “S” ( Figure 4) within a layer, and layer amounts for any molecule *m*, is obtained by scaling the amounts computed for a reference path ()by a factor *s* that is independent of *m*:

,

where .

It results that total molecular optical depth along any path can be obtained by simply scaling the reference total optical depth:

.

### path-calculation-ctrl JSON group

The following is an example of valid JSON directive for path calculations control:

"path-calculation-ctrl": {"RT-grid": 3, "v-refrac": 5630.0, "airmass-scaling": [true, false],"reference-path":1}.

In this example, the user elects to perform the RT calculations on the same vertical grid as the input profile and to activate the air mass approximation for the downwelling thermal radiation and solar direct beam using the upwelling path as the reference (Table 10). The optional key "v-refrac" used in this example is the equivalent of VBAR in LBLRTM TAPE5 and sets the wavenumber at which the refractive index of the atmosphere is computed (default = ).

Table 10: "path-calculation-ctrl" JSON keys

|  |  |
| --- | --- |
| **Key** | **Allowed values** |
| "RT-grid" | 1: RT grid for all scenes from scene file header |
| 2: Auto-layering option activated |
| 3: RT grid same as input grid |
| "airmass-scaling" | Two-element logical array. First element: thermal radiation path; second element: solar radiation path |
| "reference-path" | 0: airmass-scaling with vertical path as reference |
| 1: airmass-scaling with upwelling path from surface to observer as reference |
| "v-refrac" | Frequency for refractive geometry calculations (default: mid-point of spectral domain of RT calculations) |

## Optical depths calculation control flags

Like LBLRTM, CLBLM provides the user the ability to turn lines or continuum contribution on or off, scale the individual continua and control weak line rejection. By default, all components are included in the optical depth calculations and weak lines rejection is activated. The same criteria as in LBLRTM are used for line rejection, i.e., a line is rejected if the optical depth at the line center, **0, is less than some absolute threshold, *dptmin*, or less than *dptfac*× **cont, where **cont is the molecular continuum optical depth at **0 and *dptmin* and *dptfac* are user-controlled parameters with internal default values: of zero. , respectively.

Default settings can be overridden by including the JSON group **"od-flags"** in the user-directives (Table 11).

Lines and continuum contributions can be independently turned off by inserting the key/value pair "lines-contribution": false or the pair "continuum-contribution": false. In addition, CLBLM provides the ability to scale individual molecular continua and Rayleigh extinction (for solar direct beam only). This is done by entering a seven-element array containing the scaling factors for H2O self and foreign continua, CO2, O3, O2 and N2 continua and Rayleigh extinction, "continuum-scaling": [s1, s2, s3…s7]. Note that as mentioned before, there is no distinction made in CLBLM between line molecules and cross-section molecule other than how their absorption is parameterized inside ODLAY. Therefore, turning off line contribution exclude both from the optical depth calculations. Two other keys, "p-convolution" and "collision-partners-broadening" may be used to turn off the pressure convolution of the cross-section data or force the use of available molecule-pair specific broadening coefficients in the ODLAY line-width calculations (if those coefficients are available in the input spectroscopic database), respectively. The latter key must also be set to handle the O2 lines in the Schumann-Runge range (55000-87000 cm-1), which have predissocation line widths instead of pressure broadened widths.

In the following directive example, the default rejection parameters are modified by the user, the use of molecule-pair specific broadening coefficients is requested and pressure convolution of cross-section data is turned off. All continuum contributions are turned off (scaled by 0) except for water vapor self-broadened continuum and CO2 continuum as well as N2 continuum absorption which is scaled by a factor 2.

{"od-flags": {"line-rejection-params": [dptmin, dptfac], "collision-partners-broadening-on": true, "x-sections-p-convolution": "off", "cntnm-scaling": [1.0, 0.0, 1.0, 0.0, 0.0, 2.0, 0.0]}.

A complete list of **"od-flags"** keys together with allowed and default values is provided inTable 11.

Table 11: List of "od-flags" keys.

|  |  |  |  |
| --- | --- | --- | --- |
| **Keys** | **Description** | **Possible values** | **Default** |
| lines-contribution | Logical character string. Turns off contribution of lines (and x-sections) in OD calculations | true; false | True |
| continuum-contribution | Logical. Turns off contribution of continua in OD calculations | true; false | True |
| collision- partners- broadening | Only used if line contribution is ON. Ignored otherwise. | true; false | False |
| line-rejection | Turns line rejection on or off | true; false |  |
| dptmin | Optical depth below which lines will be rejected | real | 0..0 |
| dptfac | Factor multiplying molecular continuum to determine optical depth below which lines will be rejected | real | 1.0 10-3 |
| p-convolution | Controls pressure convolution of x-section data | true; false | True |
| continuum-scaling | Seven-element real array containing scaling factors for H2O self and foreign continua, CO2, O3, O2 and N2 continua and Rayleigh extinction. Key ignored if continuum contribution is OFF. | [s1, s2..., s7] | all scaling factors = 1.0 |

## Scene file and scene selection

Unless directed otherwise, CLBLM will look for a scene file named *scene.nc* (default name) in the default scene files location (Sec. 2) and will process all the scenes contained in this file. Including the JSON group **"scenes"** in the user-directive file allows the user to provide the path to the desired scene file (if different from the default) and to select among the scenes included in this file the number of scenes to be processed:

"scenes": {"scene file": "mylocation/myfile", "nscenes": *n*, "scene-ID": [ ID (1), ID (2) … ID (*n*)]},

where the value of "nscenes" indicates the number of scenes to be processed and the array "scene-ID" contains the desired scene numbers (which do not have to be contiguous) in increasing order.

Table : Optional scene selection and solar illumination and scene geometry directives

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Group** | **Keys** | **Description** | **Possible values** | **Default** |
| scenes | "nscenes": | number of selected scenes | integer |  |
| "scene-ID": | integer array: [s1, s2, s3…] | 1 to… | N/A |
| Alternate-geometry | "nobs-alts": |  |  |  |
| "nview-angles": |  |  |  |
| "obs-alt": | real array: [z1, z2, ... , z(nobs-alts)] |  |  |
| "view-angle": | real array: [1, 2, ... , (nview-angles)] |  |  |
| "nsun-angles": |  |  |  |
|  | "sun-angles": | real array: [1, 2, ... , (nview-angles)] |  |  |

## Overriding scene-file viewing geometry information

The scenes included in the user-supplied scene file typically contain viewing geometry (observer altitude and angle) as well as solar illumination (solar zenith angle) information. If the selected scenes (Sec. 3.8) include a *complete* description of the atmospheric state and composition (from surface to TOA), it is possible to override the viewing geometry information by using the group **"geometry"** to enter any number of either observer altitudes (in the real array "obs-altitudes") or viewing angles (in the array "view angles") or both:

"geometry": {"obs-altitudes": [ *z*1,*z*2,*z*3…], "view angles": [ **1,**2,**3…]}.

For each selected scene, CLBLM first loops over the different observer altitudes and, for each altitude, loops over the specified viewing angles to compute in each case radiances, transmittances or Jacobians per the directives included in **"clblm-out"**. The default output file name for each product is constructed by appending the observer altitude or view angle to the root file name as described in Appendix A.

## Convolution-only mode

It is possible to use CLBLM to just perform the spectral convolution of *monochromatic RT data* produced in previous runs without activating the optical depth and RT calculations.

In order to use the CLBLM spectral convolution function alone, one must include the JSON group **"clblm-in"** specifying the type(s) of monochromatic products to be convolved (i.e., radiances, transmittances or Jacobians[[6]](#footnote-6)) and associated input file name(s) followed by the spectral convolution and spectral grid directives (Sec. 3.3). The **"clblm-out"** group (Sec. 3.1) can optionally be used to override default output file names, as in the example below [[7]](#footnote-7). More than one product type may be listed. CLBLM can process either a single file per type or batch-process multiple files contained in a *same* directory. In the first case, the user must enter the *full* name of the data file, including the ‘.nc’ extension. If the supplied name does not include the directory location (full path name), CLBLM will assume that the file is contained in the default directory location for its output data (Sec. 2). The same directory rule applies when performing batch-processing. Multiple file selection for batch-processing is done by using one or more wild cards (a wild card is indicated by the single character ‘\*’) in the entered filename. For instance, assuming that the user has adopted the default CLBLM file naming convention (Appendix C), the following set of directives,

{

"clblm-in": {"jacobians": "myfolder1/drad\*-o10km\*", "tx**-**profile ": "myfolder1/tx-profile\_mono\*\_s010-\*"},

"spectral**-**convolution**-**flags": {"FFT": true, "Function ID": 2, "HWHM": 5.2e-02, "boxcar-width-ratio": 0.05}

"output-spectral-grid": {"from": 1000.0, "to": 2500.0, "DV": 2.0e-02}

}

instructs CLBLM to process *all* monochromatic Jacobian files associated with an observer located at 10 km altitude (Sec. 3.9), regardless of viewing angle and scene number, found in the default output data directory and transmittance profile files for all atmospheric levels available for scene # 10 contained in the directory *myfolder1*/. Alternatively, specific input Jacobians may be selected by including the key "jacobian-list" in **"clblm-in"**. Modifying the above example as

{

"clblm-in": {"jacobians": "myfolder1/drad\*-o10km\*", "tx**-**profile ":"myfolder1/tx-profile\_mono\*\_s010-\*", "jacobian-list": ["CO2","CH4"]},

"spectral**-**convolution**-**flags": {"FFT": true, "Function ID": 2, "HWHM": 5.2e-02, "boxcar-width-ratio": 0.05}

"output-spectral-grid": {"from": 1000.0, "to": 2500.0, "DV": 2.0e-02}

}

restricts the list of input Jacobians to CO2 and CH4 Jacobians.

By default, the product of the convolution is saved in the same directory that contains the input file(s). If desired, the output data can be directed to different folders by supplying alternate path names through JSON group "clblm-out", e.g.,

"clblm-out": {"jacobians": "myoutputfolder1/", "tx**-**profile ": "myoutputfolder2/"}.

Note that in the present context there is no mechanism for modifying the name of the output files: CLBLM expects the qualifier ‘\_mono’ to appear at the end of the root name of the input monochromatic files and the name of the file containing the product of the convolution is simply constructed by removing this qualifier from the input filename (all the other elements of this filename are kept intact). The same set of rules applies when processing single-files.

Note that users may elect to depart from the CLBLM naming convention when renaming their files. However, because of the way CLBLM names the output files, one restriction is that the qualifier ‘\_mono’ must appear at the end of the root name of monochromatic files to distinguish them from files containing convolved products (no qualifiers).

# Appendix A: CLBLM units and conventions

Table 13:

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Symbol** | **Unit** |
| Altitude | z | km |
| Pressure | P | mb (or (hPa) |
| Temperature | T | Kelvin (K) |
| Molecular densities |  | molecules.cm-3 |
| Molecular amounts | w | molecules.cm-2 |
| Angles |  | degrees |
| Wavenumber |  | cm-1 |
| Radiance | R | W.cm-2. [cm-1]-1. ster.-1 |
| Radiative Fluxes |  | W/m2 |
| Heating Rate |  | Deg/Day |

Observer viewing angle is defined with respect to the vertical at the observer level. Angles in the 0-90° quadrant indicate an observer *looking up* and angles in the range 90-180° indicate a *down-looking* configuration. Solar illumination geometry is defined by the solar zenith angle at the Earth’ surface and the sun azimuth angle relative to the viewing path (only used with CLBLM-multi-stream in scattering atmospheres, for off-nadir or off-zenith viewing).

# Appendix B: CLBLM input scene file format and content

The NetCDF4 format CLBLM Scene file consists of groups of scene data. One scene per group. Each scene/group can have its own number of molecules, number of levels and surface properties. Number of scenes is defined in the root group. Table 14 gives definitions for dimensions, variables and attributes.

Table14: Definition of NetCDF file for CLBLM scene data.

|  |  |  |
| --- | --- | --- |
| **root group (Header):** | |  |
| **//Global attributes** |  |  |
| **:**fileID | Unique ID number for the scene file |  |
| **:**numScenes | Number of scenes contains in the scene file |  |
| **:**num\_RT\_grid\_Lev | Number of RT grid levels | Optional |
| **:**RT\_grid\_lev\_in\_mb (num\_RT\_grid\_Lev ) | RT grid in pressure unit (mb) | Optional |
| **group (Scene data):** | | A single file can contains multiple groups (scenes) |
| **dimensions:** |  |  |
| numMol | Number of optically active molecules (including isotopologues) |  |
| numPrflLev | Number of profile levels |  |
| numEmisGridNode | Number of emissivity frequency nodes |  |
| numReflGridNode | Number of reflectance frequency nodes |  |
| lenMolName | Molecular name length |  |
|  |  |  |
| **//group attributes** |  |  |
| :sceneNumber | Scene ID number, from 1 to numScenes. |  |
|  |  |  |
| **variables:** | **variable attributes:** |  |
| float earthRadius | long\_name = "Earth radius";  unit = "Km";  FillValue = -999.0;  comment = "Default value is 6371.23 Km" | Optional |
| float latitude1G | long\_name = " observer Latitude ";  unit = "Deg";  FillValue = -999.0;  comment = "Default value is 45 Degree"  valid\_range = [-90.0, 90.0] | Optional |
| float altitude (numPrflLev) | long\_name = "Level altitude";  unit = "Km";  valid\_range = [ -1.0, 120.0] | Optional |
| float **pressure(**numPrflLev**)** | long\_name = "Level pressure";  unit = "mb";  valid\_range = [0.0, 1200.0] |  |
| float **temperature(**numPrflLev**)** | long\_name = "Level temperature";  unit = "Kelvin";  valid\_range = [] |  |
| float **molDensities (**numPrflLev**,** numMol**)** | long\_name = " molecular number densities";  unit = "molecules.cm-3";  valid\_range = [0.0, 1.0e26] |  |
| float **sfcPressure** 1 |  |  |
| float **sfcAltitude** 2 |  |  |
| float obsAltitude 3 | long\_name = "Observer altitude";  unit = "Km";  valid\_range = [0., Inf.] |  |
| float **viewAngle** | long\_name = "Viewing angle at observer height";  unit = "Deg";  valid\_range = [0., 180.0] |  |
| float sfcSkinTemp | long\_name = "Surface temperature";  unit = "Kelvin";  valid\_range = [200.0, 400.0] | Not needed for sensor looking up |
| sfcPropInputMode | 1: **thermal** reflectivity = (1.- emissivity) with surface assumed specular  2**: thermal** reflectivity = (1.- emissivity) with surface assumed Lambertian  In modes 1 and 2, reflectivity of surface-incident solar beam in the direction of the observer is specified in separate sfcRefl array.  3: Surface BRDF (used to describe both **therma**l and **solar** reflectivity) is described as a weighted sum of BRDF kernels: input contains spectrally dependent weights for each kernel (not implemented yet)  In mode 3. Both thermal and solar reflectivity are computed from spectrally dependent BRDF coefficients. |  |
| float sfcPropSpectrGrid (numEmisGridNode) | long\_name = "Surface optical property spectral hinge points";  unit = "cm^-1"  valid\_range = [0.0, 50,000.0] |  |
| float sfcEmissivity 4 | long\_name = "Surface emissivity" |  |
|  |  |  |
| float sfcEmis | long\_name = " surface emissivity at hinge points on sfcPropSpectrGrid";  valid\_range = [0.0, 1.0] |  |
|  |  |  |
| float sfcRefl | long\_name = " surface reflectance " |  |
|  |  |  |
| float sfcRefl (numReflGridNode) | long\_name = " surface reflectivity at hinge points on sfcPropSpectrGrid ";  valid\_range = [0.0, 1.0] |  |

1 If using input profile grid for RT calculations, surface is the lowest level on that grid

2 Should always be supplied, even if altitudes for other levels are not included

3 Observer altitude must be provided if observer is in space (obsPressure =0. Mb) beyond highest level on either the input profile grid or

RT grid, if present)

4 Surface emissivity and reflectance are input either as look up tables or as spectrally independent constant values.

# Appendix C: CLBLM output data file format and content

**File content**

Title

Scene file name and UID and scene number

Product name (od, radiance, total transmittance, transmittance profile, Jacobians)

Type of spectral data (monochromatic or convolved)

V1, V2, DV, NSAMP

If convolved: instrument function (or file name for ID=0)

Optionally FFT and function parameters (ID = 11, 12 or 13)

**Default file names**

According to the CLBLM file naming convention, basic data file names consists of a root name indicating the product type (i.e., optical depths, radiances, transmittances or Jacobians), followed by “\_mono”, for a monochromatic output, and followed by a three-digit scene number (NNN) preceded by the letter “s” (

Table 30). Multi-layer/level products except Jacobians (i.e., optical depths and transmittance profiles) are stored in a single file per layer/level. The layer/level number is indicated by a 3-digit number (LLL) at the end of the file name. For optical depths and Jacobians, this number correspond to the actual layer and level number on the RT grid, respectively. For transmittance profile, this number represents the relative level count above the observer, if looking up (1 being the level immediately above the observer level and the largest number being TOA) or below the observer if looking down (1 being the level immediately below the observer level and the largest number being the surface).

Table 15: Default CLBLM output data root file names (in bold) and *automatically* appended auxiliary information (normal font).

|  |  |  |
| --- | --- | --- |
| **CLBLM output product** |  | **Default file names** |
| Monochromatic optical depths1 |  | **od\_mono** \_sNNN\_LLL.nc |
| Spectrally Convolved (or monochromatic) radiances |  | **rad(\_mono)** \_sNNN.nc |
| Spectrally Convolved (or monochromatic) path transmittances |  | **total-tx(\_mono)**\_sNNN.nc |
| Spectrally Convolved (or monochromatic) transmittance profiles | Up-looking case | **tx-up(\_mono**)\_sNNN\_LLL.nc |
| Down-looking case | **tx-down(\_mono**)\_sNNN\_LLL.nc |
| Spectrally Convolved (or monochromatic) Jacobians2 |  | **drad-dX(\_mono)**\_sNNN\_LLL.nc |
| Radiative Fluxes |  | **FLUX\_OUTPUT.nc** |

1 Optical depths are output in monochromatic form only

2 X=’T’, ‘Tskin’, ‘emis’ or molecule ID (e.g., ‘O3’)

In case the default viewing geometry parameters included in the original scene data have been overridden by the user, additional geometry information (observer altitude or viewing angle, both rounded to the nearest integer) is added by CLBLM before the scene number. For example, ‘rad\_mono-o10km-a40deg \_sNNN.nc’ is the CLBLM default file name for monochromatic radiances computed for an observer at ~10 km altitude looking down at a ~40° angle using atmospheric/surface data from scene number NNN.

1. Theses default may be overridden at run time through the input user-directives (Sec. 3) [↑](#footnote-ref-1)
2. The plan for future releases is to convert the LBLRTM TAPE3 file to NetCDF format. [↑](#footnote-ref-2)
3. Total path transmittance being one element of cumulative transmittance profile, if both appear in the product list, only the latter is output. [↑](#footnote-ref-3)
4. Unless the observer is looking at a target (option not available yet), this layer corresponds to the highest layer on the RT grid. [↑](#footnote-ref-4)
5. Average solar activity over cycles 13-24 (from NRLSSI2) [↑](#footnote-ref-5)
6. Optical depths are not an allowed type. [↑](#footnote-ref-6)
7. Any additional groups present in the input user-directive file are ignored by CLBLM [↑](#footnote-ref-7)