

**The Shortwave (SW) Clear-Sky Detection
and Fitting Algorithm: Algorithm
Operational Details and Explanations**

December 2000

C. N. Long

Work supported by the U.S. Department of Energy,
Office of Energy Research, Office of Health and Environmental Research

Contents

1	Introduction.....	1
2	Preparing the Raw Input Files.....	2
3	Running the Clear ID and Fitting Algorithm (swclid1a).....	4
	3.1 Necessary Files	4
	3.2 Configuration File (swclid1a.cnf).....	5
4	Data Qc Tests and Qc Flags.....	12
5	Contents of the *.SWF Output Files	14
	5.1 Coefficient Data.....	14
	5.2 Irradiance Data.....	15
6	Other Output Files	16
7	Running the Fractional Sky Cover Algorithm.....	19
8	References.....	21

1. Introduction

This document describes some specifics of the algorithm for detecting clear skies and fitting clear-sky shortwave (SW) functions described in Long and Ackerman (2000). This algorithm forms the basis of the ARM SW FLUX ANAL 1Long VAP. In the Atmospheric Radiation Measurement (ARM) case, the value added procedures (VAP) can be described as having three parts: a “front end,” a “black box,” and a “back end.” The “front end” handles the file management of the processing, what range of data files to process in the run, which configuration file to use for each site, extracting the data from the ARM NetCDF files into an ASCII format for the code to process, etc. The “back end” produces ARM-format NetCDF files of the output and other file management. The “black box” is the processing code(s), and is what is discussed in this document. Details on the “front” and “back” ends of the ARM VAP are presented elsewhere.

Before proceeding, it is highly recommended that the reader first review the following documents:

Long, C. N. and T. P. Ackerman, 2000: “Identification of clear skies from broadband pyranometer measurements and calculation of downwelling shortwave cloud effects.” JGR, 105, No. D12, 15609-15626.

Long, C. N., T. P. Ackerman, J. J. DeLuisi, and J. Augustine, 1999: Estimation of fractional sky cover from broadband SW radiometer measurements.” Proc. 10th Conf. on Atmos. Rad., June 28-July 2, 1999, Madison, Wisconsin.

Both of these papers are available in HTML format at <http://playground.arm.gov/~clong>

Click on the “[Clear Sky ID and Cloud Effect Paper](#)“ and “[SW Sky Cover Retrieval Paper](#)“ links, respectively, on the above Web page. These papers detail the science and methodology behind the algorithm; subjects not covered in this document and without which some of the following discussion might not be readily understandable. The intent of this document is to provide details sufficient to run the code(s) and understand some parts included in the code(s) (such as the data QC checks) not covered in the above papers, as well as detail the code outputs and some suggested uses for the output data. These codes are written in FORTRAN77, and are available from the author as stand-alone programs. The wording that follows is derived in part from documentation provided along with the PC Win95/98/NT version of the code currently in use by the atmospheric science community, such as members of the Baseline Surface Radiation Network, under the name “swclid1a.zip.” Thus some references are tailored to the personal computer (PC) environment, such as the referrals to batch files. For the UNIX environment, specifics will need to be garnered from the ARM SW FLUX ANAL 1Long “front” and “back” end documentation.

This algorithm takes input files of total, diffuse and direct SW irradiance 1-minute measurements, detects clear-sky periods, and fits a curve to the clear-sky measurements using the cosine of the solar zenith angle as the independent variable. It then computes the effect of clouds on the downwelling SW irradiance by subtracting the fitted clear-sky irradiance from the measured irradiance. The algorithm is designed to take daily files, determine clear fit coefficients for each day if possible, and linearly

interpolate the coefficients for days that do not have enough clear-sky measurements to fit a curve. The algorithm stores the coefficients from the last day in the data run that had enough clear measurements, thus files from time periods immediately following the data run can be processed and coefficients interpolated between this “last day” and the first day of the new run that has sufficient clear-sky measurements for fitting. In a nutshell, the algorithm is designed as an operational data archive-processing tool. However, depending on the site for which it is set to run operationally, it only makes sense to run the package for a time period sufficiently long that there will certainly be a number of days with sufficient clear-sky measurements to fit curves. For example, the ARM Program of the U.S. Department of Energy (DOE) has a site in the central U.S. near Lamont, Oklahoma, which represents a “typical” northern mid-latitude plains climate. It is recommended that this algorithm be run on a monthly basis for the Lamont site, since during the summer there can be periods of up to 14 days or more without sufficient clear-sky measurements for daily curve fitting. On the other hand, an Australian site near Leigh Creek has climatologically about 250 to 300 days per year sufficiently clear for daily curve fitting. This site could be processed on a weekly basis. And there are some places, like the tropical western Pacific warm pool region, where clear skies are few and far between. In this case, there are rarely clear periods of sufficient duration to calculate daily coefficients. For sites like this, this package can be set to determine one set of coefficients for the entire data period being processed.

NOTE: Because Fortran77 does not allow dynamic setting of array size, all array sizes are “hard-wired” in to the code. In the present form, the code can handle up to five years worth of 1-minute data files. Should you need to process a longer time series, you can either run several batches, or change the code to handle larger arrays. To do this, look for the following parameter statements in the code, and change them accordingly:

PARAMETER STATEMENTS TO SET ARRAY SIZES:

PARAMETER (NMAX = 100000) - NMAX is the total number of points that can be fit in the arrays for a single fitting.

Used in subroutines: swclreq1, medfit, rofunc, swclreq2, and swclreq3

PARAMETER (NDAYMAX = 18500) – NDAYMAX is the total number of days that can be handled in one run of the code.

For 5 years max of data, set NDAYMAX = 1850

For 50 years max of data, set NDAYMAX = 18500

Used in subroutines: clrcof1, clrcof2, and swclrfcg

2. Preparing the Raw Input Files

The Clear ID code is intended to be able to handle many different forms of input data formats. It uses Fortran77 unformatted read statements to read the input data files, and creates data output files (*.cos) in a format that the rest of the code uses for processing. I tried to make this portion of the code as generic as possible, but there are some “rules” I incorporated to make the code practical. The most basic rule,

naturally, is that the input file must be compatible with the unformatted read, i.e., values must be comma or space, etc. delimited in each row. Additional rules are as follows:

- 1) The input data must be in ASCII, chosen as the most basic form of file format. For the ARM SW FLUX ANAL 1Long VAP, the “front end” handles production of the ASCII input files from ARM NetCDF data files.
- 2) All data must be in rows, with the date/time listed before the measurement data on each row, then the measurement values. Each row must have the date/time integers (and other integers if you want) first, then the measurement data as reals. After the initial header lines and such (which you set to skip over in the configuration file explained below), all successive lines must contain data until the end of the file. Note that FORTRAN unformatted read declared for REALS does not understand a character “NaN,” which is sometimes produced by UNIX-run codes. Thus, if the ASCII input files contain any “NaNs,” the code may bomb.
- 3) The date and time info must be in integer format. The date can be separate integers (i.e., YY or YYYY MM DD, where YY or YYYY is the two digit or four digit year, MM is the two digit month, DD is the two digit day of the month.) Or it can be all one integer, i.e., YYYYMMDD. Note that the code does not accept Julian day, or day of the year, but only accepts Gregorian calendar date format. If the date is separate integers, then they can be in any order. The time info can be separate integers (hh mm, where hh is the hour of the day, and mm is the minute of the hour, no seconds are accepted), or all one integer (hhmm). The code in this case will also accept “day minutes” as one integer, i.e., the number of minutes since midnight. If the time is separate integers, then they can be in any order. Also, no matter whether all one integer or separate, the date and time can be in any order, i.e., the time info can come first.
- 4) The input date and time can either be in Universal Time Coordinates (UTC), or Local Standard Time (LST). Note that LST is not necessarily local (political) time. LST is defined strictly from the site longitude, changing one hour for each 15 degrees of longitude, with the zero change LST time zone (from UTC) centered on Greenwich, England. Thus, the “zero offset” LST time zone covers the longitude from 7.5 degrees East to 7.5 degrees West longitude. The following is the FORTRAN code to determine LST hour offset (hrofst) from UTC:

```
hrofst = FLOOR((ABS(longit) + 7.500)/15.000)
IF(longit.lt.0.0) hrofst = -hrofst
```

where longit is the site longitude ranging from -180 to +180 with minus being west longitude and positive being east longitude, ABS is the absolute value, and FLOOR truncates to the integer portion of the real value (i.e., rounds down). For example, if the site longitude is 116.8 degrees west longitude:

```
116.8 W = -116.8
ABS(-116.8) = 116.8
(116.8+7.5)/15.0 = (124.3)/15.0 = 8.286666666667
```

$\text{FLOOR}(8.286666666667) = 8$

Since longit is negative, thus it is less than zero,

Hrofst = -hrofst = -(8)

Thus, LST in this example is UTC - 8 hrs.

- 5) All measurement values to be used by the code must be in “real” numbers. There can be integers in the rows after the final value you will want the code to read, but NO INTEGERS can be “mixed in” in between the last integer at the start of the row and the last input value to be used.
- 6) The input files must contain values for all three SW components (total, direct, and diffuse) even if one of them is just a column of “bad and/or missing data” values such as “-999.0”. The code will work if you only have two of the three measurements, but there must be a column for all three VALUES. If you want, you could generate the third (missing) measurement as the sum/difference of the other two. Unfortunately, if you only have two of the three measurements, then the code cannot perform the automated data QC, but it will produce fitted clear-sky estimates and cloud effect.

3. Running the Clear ID and Fitting Algorithm (swclrid1a)

3.1 Necessary Files

This part involves five files: “swclrid1a.bat,” “swclrid1a.cnf,” “swclrid1a.exe,” “swlastdy.cfn,” and “swlastdy.cfw.” The process runs by executing the batch file. Note that the batch file contains lines that may be remarked out (so they don’t execute) by placing “REM” in front of that line. The important operational parts of the batch file are the last two lines, where a list of files to be processed is piped into a directory file named “inputfls.dir,” and the “swclrid1a.exe” executable is called. The DOS command “dir /b /on filelist > outfile” creates a directory listing (dir) of brief file names (/b) in alphabetical order (/on) of filelist (these are your input files) and pipes the list into an output file called “inputfls.dir” (> inputfls.dir). Do not change the name “inputfls.dir”! Note that the line to list the input files includes a directory path. If the input files are in the same directory as the executable, then just put the DOS wildcard-type listing (ex: “19*.ldr”) there. But the code can accept a path character string up to 160 characters in length (listed in the config file below) so that you can run the code in a different directory than where the input files are. The output files, however, are created in the same directory as the executable resides.

Some other lines in the batch (preceding the last two lines) are there primarily for my convenience in developing/debugging the code. Normally you can remark these lines out, or just delete them. The first two lines of the batch file:

```
copy swlastdy.svn swlastdy.cfn
copy swlastdy.svw swlastdy.cfw
```

Copy “saved” last day coefficient files to overwrite the operating coefficient files used in initial interpolation. DON’T leave these lines active for operational processing! The two files “swlastdy.cfn”

and “swlastdy.cfw” are written and used by the code to keep track of the coefficients of the last “clear enough” fitted day in the previous/current run. The numbers in “swlastdy.cfn” represent, from left to right: date in YYMMDD format, diffuse ratio “a” coefficient, diffuse ratio “b” coefficient, total shortwave (TSW) “a” coefficient, and TSW “b” coefficient. These are coefficients for the equation:

$$Y = a*(\text{Cos}Z)^b,$$

where Y is either the clear-sky TSW or clear-sky diffuse ratio (the diffuse ratio is defined as the diffuse shortwave irradiance over the TSW irradiance), CosZ is the cosine of the solar zenith angle, and a and b are the fitted coefficients. Typical mid-latitude values are given: i.e., 0.10, -0.8, 1250.0, and 1.2 respectively. For the “swlastdy.cfw” file, additional values are included, a pair of coefficients for each value being fitted. The first time ever you run the code on data from a site, you don’t really know what the “ball park” coefficients should be for the climate regime and particular set of instruments and their characteristics. So you run the code once, using these “generic” values, which then generates a “real” set of values in these files. Then run the code again on the same data to “really” process the data. After this initial run, the values left in these files (“swlastdy.cfn” and “swlastdy.cfw”) are used by the code the next time you run it on, say, the NEXT month’s worth of input files. As you can see, this algorithm is intended to be run about once a month, operationally. That’s because the code only works up to the last “clear enough” day in the data set. If there are no clear days in the data being processed, the algorithm fails and zeros are written to these two files. That’s why it’s a good idea at the end of a successful run to copy these two files values to the files “swlastdy.svn” and “swlastdy.svw.” That way, after an unsuccessful run you can un-remark these two lines in the batch file and run the code again when you have more data, but still start with the last clear day coefficients.

The other line in the batch file:

```
copy swclid1a.cnf.siros.e13 swclid1a.cnf,
```

is to make sure I was consistently using the same configuration file, instead of, say, one for the Lauder, NZ, when processing ARM SGP site E13 files. You can remark this line out.

3.2 Configuration File (swclid1a.cnf)

The configuration file for the program is “swclid1a.cnf.” Some of the values in it are pretty much set for you. Each line has a preceding explanation line of what values are being set. The format for this file is fixed, so don’t change it. (This is another reason I had a backup config file). Some lines you will need to change, others you won’t. Here is a “line-by-line” description of the contents of the configuration file, and examples, with a brief explanation following each:

- 0) Data format flag (incl.), 0=BSRN (no), 1=SIROS/SIRS (alb), 2=TWP (UVB), 3=RSR (Ang)

This is an integer value that tells the code what all values you want fitted, in this example set to “1” for “SIROS/SIRS (alb).” Meaning as follows:

- “BSRN (no)” sets the code to only produce fits for downwelling total, diffuse, and direct measurements.
 - “SIROS/SIRS (alb)” sets the code to also produce a fit for surface albedo, using the same “power law” fit equation format as used for the clear total and diffuse ratio. Note that the input data files in this case must contain measurements of upwelling SW. Note also that there isn’t any data QC applied to the upwelling SW measurements, though later in the configuration file there are limits set to preclude fitting for snow-covered (i.e., “too bright”) surfaces.
 - “TWP (UVB)” sets the code to fit some type of solar-driven downwelling ancillary measurement, such as UV, photosynthetically active (PAR), or one spectral channel, etc., using the same “power law” fit equation format as used for the clear total and diffuse ratio. Note that here again no data QC is applied to this input value.
 - “RSR (Ang)” sets the code to include some data value in the input files, but NOT produce any clear-sky fit coefficients for the value. In this example (and where the name came from), I wanted the shading arm angle from vertical to be included in the “*.swf” output files, but it’s not a value that it makes any sense to do a clear-sky fit for. The values are just copied from the input files and included in the “*.swf” files.
- 1) The following are integer values for the number of minutes for the additional test running mean, the running difrat mean, minimum number/day for daily coef, and whether to calculate daily coefficients (1 = yes, 0 = no) (4 integers).

21 11 110 1

The first two integer values tell the code how many minutes of data to use for the running mean and standard deviation calculations used to detect clear skies (21 and 11). The maximum numbers allowed here are both 60 as that is the size of the arrays in the executable. However, normally these numbers are set between 11 and 21, as an hour average severely limits the clear detection capability. These should not need to be changed from the current values (for 1-minute data, which this code is for). The third integer sets the minimum number of 1-minute data detected as clear in a single diurnal cycle for that day to be used for daily fitting. Or, if the last integer is set to “0”, the minimum number of detected clear measurements in a single diurnal cycle needed to include that data in the “one fit for all data” fitting. It is recommended that you don’t set this minimum below 90 for daily coefficients (Long and Ackerman 2000). The last integer sets whether to do daily fitting and coefficient interpolation (usual mode), or to only produce one set of fit coefficients for all input data. This last mode is used, for example, in the tropical western Pacific, where the occurrence of extended periods (hours) needed for a daily fit is rare, and thus interpolation between clear periods would involve months. In this situation, it is better to fit to all detected clear-sky periods. When used in this mode, the third integer needs to be reduced from that of “daily” mode, so that brief periods of clear-sky data

are included in the fitting. However, the value must also be large enough so that the occasional mis-identification of clear sky is not overly included in the fitting. I recommend a value of 15 to 30 minutes minimum number of detected clear-sky 1-minute values as the limit to set the third integer for “one fit for all” mode. In the example above, the code is set to do daily fitting and interpolation for days with at least 110 detected clear-sky data, and uses 21 and 11 minutes for the running mean and standard deviation calculations. A more detailed explanation of these values can be found in Long and Ackerman (2000).

- 2) Solar ephemeris info: Lat (N = +), Long (E = +), Time Flag (0/1 for LST/GMT).

-45.04 169.68 0

You need to change these to the appropriate values associated with the input data. These are used to calculate the solar zenith angle values (if not already provided in the input files), and calculate the UTC/LST offset for the output files. The values must be in degrees and fractions of degrees (not in minutes and seconds of lat/long). In this example, the code is configured for 45.04 degrees South latitude, 169.68 East longitude, and the time in the input files is in LST.

- 3) The following is the total solar normalization exponent (“b” of $TSW/\cos Z^{**b}$), the Normalized Diffuse Ratio Standard Deviation limit, and the delS + Lim factor (*CosZ).

1.18 0.0012 2.0

The first real is a “generic” value used in the first pass for detecting clear-sky periods. This should be set at an “average” value of the SW “b” coefficient determined from typical data. After the first iteration, i.e., in all following iterations in the algorithm, the coefficients used to normalize are those determined in the previous iteration. The second real is the limit that the program uses for the standard deviation of the running normalized diffuse ratio in determining if the sky is clear. The limit 0.0012 works well for most systems. However, if you have a particularly “noisy” data logging/instrument system (for example many MFRSR data sets that I’ve seen), you may have to adjust this parameter if you notice that plots of time series of your data that seem to be clear are not being detected as such. A word of warning, though: the program can’t distinguish between system noise and a small increase in standard deviation due to either just a few puffy Cu or slight variations in a thin (perhaps sub-visual) cirrus deck for a very noisy system. I have set this limit to err on the side of caution, i.e., reject system noise as slightly “cloudy” in order to ensure more accurate clear detection. If you increase this limit, the program isn’t as capable of detecting the above cloud scenarios. The third real is the “change with time” envelope limit constant, the value of 2.0 is correct for 1-minute data. These values are pretty much okay as they are, and shouldn’t need to be changed. A more detailed explanation of these values can be found in Long and Ackerman (2000).

- 4) The following is for inclusion of solar zenith angle calculation or Z type: (0 = yes, calculate; other means data includes: 1 = CosZ, 2 = Z degrees, 3 = Z radians).

0

This integer value tells the code whether to calculate the solar zenith angle, or whether some form of solar zenith angle information is already included in the input files. Acceptable forms of input are the cosine of the solar zenith angle (CosZ), the solar zenith angle in degrees, or the solar zenith angle in radians. Note that the solar elevation angle is not acceptable input. In this example, the code is set to calculate the solar zenith angle using the lat/long given in setting 2), and date and time info in the input files.

- 5) The following are the read variables – number of: header lines to skip, integers in each data line, and reals in each data line. Note: only int, then real allowed.

0 6 10

This tells the code some general things about the input file format. The first integer tells the code how many lines to skip at the beginning of each input file (to skip header lines and blank lines). Note that after these lines are skipped, there must be data on each successive line until the end of the file. The second integer tells the code how many integers are in each line of data at the START of EACH LINE, BEFORE the reals containing the measurement data. The third integer tells the code how many reals follow the integers. Note that this last can be less than the actual number of reals on each line, as long as the number listed includes all input data the code will be looking for, as set in other parts of the configuration file (0, 2, 12, 13, 14, 23, 24). In this example, the code is set for input files with no header or blank lines (the data starts immediately on the first line), there are six integers at the start of each line, and the code should be prepared to read 10 reals after the initial integers on each line.

- 6) Is the date all one integer (ex: yymmdd) or separate integers? (0/1)

1

Tells the code the format of the date. In this example, the date information is in separate integers.

- 7) Of the number of >INTEGERS< declared in 5), which is either date or yr, mn, dy?

1 2 3

This setting tells the code which integers at the beginning of each line contain the date info. In this example, we set setting 6) to tell the code that the date was in separate integers, so we must have three integers here. In the example input files, the year is listed first, followed by the month and day. If we set 6) to “0”, we’d only have one integer on this line.

- 8) Is the year included in data a four or two digit year? (0 = 4 digit/1 = 2 digit)?

0

This setting tells the code whether the year listed in the input files is 2-digit, or full 4-digit. In this example, the input year is 4-digit.

- 9) What are the first two digits of the 4-digit year?

19

You'll need to change this to "20" at the turn of the century. This setting is included for the case where the input file lines include only the 2-digit year. The full 4-digit year is necessary for accurate solar ephemeris calculations (if needed), and for the output file naming.

- 10) Is the time all one integer (hhmm) or separate integers or daymin? (0/1/2)

1

This setting tells the code the format of the time information. In this example, it is set to read separate "hh mm" integers. "daymin" is the time in the format of the number of minutes that have elapsed since midnight on the date listed in 7), with a range of 0 to 1440.

- 11) Of the number of >INTEGERS< declared in 5), what position is either time or hr and min?

4 5

This setting works in conjunction with setting 10), telling the code which of the integers at the beginning of each line is the time information. In this example, since 10) is set for separate integers, we need to have two integers here. These tell the code that the hour is the fourth integer at the beginning of each line, and the minutes are the fifth integer.

- 12) Are the direct measurements direct normal (i.e., NIP) measurements? (0=yes/1=no)

0

This setting tells the code what type of direct SW measurements is included in the input files. The allowed type is either flat plate (sun's rays striking a detector that is aligned parallel to the surface, as with the total and diffuse measurements), or normal incidence (sun's rays striking a detector that is aligned perpendicular to the rays, i.e., measured by a tracking instrument). In this example, the input files contain normal incidence measurements.

- 13) If solar zenith info is included in the data, which of the >REALS< declared in 5) is it (0 for none)?

0

This setting tells the code whether to perform solar ephemeris calculations for each data time, and if not then which of the reals in the input files contain the solar zenith angle information. In this

example, the solar zenith angle information is already included in the input files. Note that this setting allows for processing of data from moving platforms such as ships. While the code itself can only calculate solar zenith angles for a stationary site at the lat/long coordinates listed in 2), if the solar zenith angle information is included in the input files, this restraint is removed.

- 14) Of the >REALS< declared in 5), what position are the TSW, Dif, dir/dirN, and SWu/UVB/ang?

8 9 10 7

This setting tells the code the location of the reals the code will be processing. They are, in order, the locations of the total (global), the diffuse, the direct, and the ancillary (if chosen in setting 0). In this example, the total is the eighth real, the diffuse the ninth, and the direct normal (from setting 12) is tenth. The upwelling SW (from setting 0) is the seventh real.

- 15) What is the low limit for Total SW Irradiance? (real)

1.0

This setting is the lower limit on total SW that the algorithm will include in the *.cos files. Another limit built into the code is a CosZ of 0.0. Fitting within the code requires taking the LOG of the various values, and fitting power law equations with CosZ as the independent variable. Thus, only positive values greater than zero of the total SW, diffuse ratio, albedo, etc., are allowed. Therefore, this value must be set greater than zero (NOT equal to zero). In this example, it is set to 1.0, and all data lines with a total SW less than this value will be excluded from the *.cos files.

- 16) What is the max limit for clear-sky diffuse SW Irradiance? (real)

150.0

This setting is for the constant of the diffuse limit test equation. Typically, a value of around 150.0 is reasonable here. During the winter months, for higher latitudes, or for “hazy” climates such as the tropical western Pacific, it may need to be increased to maybe 200.0 (due to low solar noon sun elevation in the first two cases) if the algorithm seems to be eliminating obviously clear data from fitting near solar noon. An explanation of this limit is in Long and Ackerman (2000).

- 17) What is the max ABS difference (tsw/csw - ssw/cssw) for dif & dir error chk? (real)

0.05

This setting is the limit for the more sophisticated data QC comparison (discussed later). Since the clear-sky fits produced by this method also include the instrument response characteristics, comparing the ratios of measured to clear for the total pyranometer and the sum (direct + diffuse) gives a more

sensitive test for inconsistencies than straight irradiance comparisons. In this example, the limit is set for an absolute ratio difference of 5%. More information on this test can be found in the “swfheadr.txt” file.

- 18) Path for input files up to 160 characters, include final “\”.

C:\rad_dat\asciidat\

This is the path for where to get the input files. NOTE that this path MUST BE THE SAME as the path listed in the “swclid1a.bat” batch file that generates the “inputfiles.dir” files list file. If no path is listed in the batch file (i.e., the input files are in the same directory as the executable), leave this line blank. NOTE also that if NOT left blank, you MUST include the closing “\”.

- 19) Following is the number of iterations between initial and final detections (integer).

4

This setting tells the code how many times to iterate the detection - fitting - detection process. This is in addition to the initial and final iterations. The current value of 4 should not have to be changed. An explanation of this iterative process is in Long and Ackerman (2000).

- 20) Following is the print flag for the *.tst output files (0 = no, 1 = yes)

0

This is a flag to print out additional files used in development for debugging the code. It should be left set at “0”, i.e. do not print out the debug files.

- 21) Flag for the measurement physical limits testing (0 = no, 1 = use tests)

1

This flag tells the code whether to perform the data QC tests or not. Should you choose not to use the built-in automated data QC, set this value to “0”.

- 22) The following are the ALBa and ALBb equation max limits for snow problems (both REALs).

0.4 -0.1

This setting puts limits on the acceptable albedo fitting “a” and “b” coefficients. These are used to, for example, eliminate periods of snow cover from albedo fitting. Any fitting technique presupposes that the independent variable is the dominant factor in the changes of the variable being fitted for. The rapid change in ground cover albedo characteristics during snow accumulation and melting preclude this assumption, and can produce highly irregular results. Thus, if fitting for the “typical”

surface characteristics, the first value should be set a bit above the normally expected value of the site surface. The second value sets a limit on the “shape” of the fitted output. Typically, clear-sky albedo increases with increasing solar zenith angle, as the sun's rays hit the surface at increasing glancing incidence. Thus, the clear-sky “b” coefficient here should be negative, similar to the diffuse radiation “b” coefficient. This setting puts an upper (max) limit on acceptable value. You will need to study your own data and surface characteristics if you will be fitting albedo to properly set these limits. The values above are for typical US mid-western plains grass surfaces.

- 23) You may have up to five more >REAL< variables printed out in the *.cos output files that are in the input files. Enter the number of extra variables you wish to add below, or “0” for none.

2

- 24) Of the >REALS< declared in 5), which are the ones to be printed out and up to a five character header to be included in the output file (ex: 6 SWup). Note: One set (integer and header) per line.

4 IRdn

1 Irup

Settings 23 and 24 allow for the user to include up to five additional data (real) values in the *.cos output files, with a five character header value for each. These are included for convenience if the user would like to keep the “*.cos” files, which in effect include only the original daylight data in the input data set.

4. Data Qc Tests and Qc Flags

As alluded to so far, the algorithm includes some automated data quality checks. These tests are intended so that the algorithm does not use “bad” data in the clear-sky fitting. The algorithm has been developed, as it has to be applied to historical data sets, as well as currently collected data. As such, there is a basic assumption used in these tests and flags, the choice of which was determined by experience with older SW component measurement data sets. That assumption is that, if there is disagreement between the total SW as measured by an unshaded pyranometer, and that derived as the sum of the direct and diffuse SW components, it is more likely that the sum is in error. The reason for this assumption is that the sum is derived using two separate detection instruments (a NIP and shaded pyranometer) mounted on a sun tracking system. The unshaded pyranometer consists of only that, a pyranometer, and nothing else. Thus, the “sum” system is by far the more complex, and generally more susceptible to mechanical, electronic (logging), and computational (solar ephemeris) breakdown, and has double the chance of anomalous obstruction of the detector, etc. This was especially true in the past given the more primitive designs of tracking equipment, compared to more modern tracker systems. The major drawback to unshaded pyranometer measurements is the increased cosine response error, compared to the BSRN recommended sum method. However, as shown in Long and Ackerman (2000), if one takes the ratio of measured over clear fit SW to investigate the effect of clouds on the downwelling SW, by far most of the cosine response error is removed.

In order to decrease the amount of missing or “bad” data, the algorithm tries to “fill in.” For the measurements themselves, first the total SW is checked to make sure it falls within the physically possible limits of -20 Wm^{-2} and 1500 Wm^{-2} . The lower limit includes the typical known IR loss (sometimes referred to as night time) offsets of thermopile pyranometers. The upper limit includes sufficient range so as not to eliminate periods of extreme positive cloud forcing sometimes seen on 1-minute time scales.

The diffuse and direct measurements are checked, to make sure they are not less than -20 Wm^{-2} . No upper limit is checked for these individually, since that is included in the TSW - sum comparison. If either the direct or diffuse fail this test, but the other passes, then the missing or “bad” value is estimated by subtracting the remaining “good” value from the total SW value. The sum is then compared to the total SW. If the absolute difference between the sum and total SW is more than 100 Wm^{-2} , then the diffuse value is set to “bad,” i.e., “-999.0.” Naturally, this last test is nullified if either the direct or diffuse has been estimated by differencing the other two components. However, the more sophisticated tests in the final iteration cover this contingency.

In the event that both the direct and diffuse are “bad,” then no clear-sky detection is possible for those time periods. However, the algorithm keeps track of the sum - total SW difference for any detected clear-sky data. These clear-sky sum - total SW difference data are used to fit (and interpolate) coefficients for a “difference function” similar to the other clear-sky data fitting. In the final iteration, any “missing” sum SW data is “filled in” by using the total SW measurement, and the “difference function” determined and interpolated for that day. These “filled in” sum data are then subjected to the same advanced testing as the original “good” data. The daily coefficients for the “difference function” are included in the “swclreq*.asc” and “clrcoef3.asc” files, and in the header of the “*.swf” output files; along with all the other fit coefficients.

In the final iteration, the code has determined clear-sky estimates for both the total and sum SW. These fits include the instrument characteristics (Long and Ackerman 2000), and thus taking the ratio of the measured over clear fit values (the “cloud effect ratio”) greatly removes these characteristics. This affords us another means to more closely check the data for disagreements. We do this by comparing the cloud effect ratios for the unshaded pyranometer (TSW) and the sum (SSW). A limit is set in the configuration file for an acceptable difference limit. If the comparison exceeds this limit, the sum, diffuse, and direct data are set to “bad.” This test detects when one of the two tracker instruments had a problem, and when the “filling in” of the sum SW using the “difference function” has failed to adequately produce an acceptable approximation of the sum. In addition, the TSW cloud effect ratio is compared to the measured diffuse ratio (diffuse/total SW). If the cloud effect ratio is large (indicating only small cloud effect), but the diffuse ratio is also large (indicating little or no direct component) for a period of time, this indicates that the tracker is out of alignment, i.e. the shaded and unshaded pyranometer measurements are about equal under conditions when they shouldn’t be. In this case again the sum, diffuse, and direct data are set to “bad.”

The results of these tests are displayed as flags in the *.swf output files, along with a clear detection flag. The flags values, and their meanings, are as follows:

- sflg “sum flag” - for Sum data and other QC
- 0 = OK
 - 1 = estimated from TSW plus correction function (orig. bad)
 - 1 = bad diffuse or direct, and couldn't estimate
 - 2 = $\text{CosZ} > 0.25$, $|\text{TSW}/\text{CSW} - \text{SSW}/\text{CSSW}| > \text{swrlim}$
 - 3 = $\text{CosZ} > 0.09$, $|\text{TSW}/\text{CSW} - \text{SSW}/\text{CSSW}| > 2 * \text{swrlim}$
 - 4 = $\text{CosZ} > 0.12$, $\text{TSW}/\text{CSW} > 0.9$ AND $\text{Dif}/\text{TSW} > 0.9$

The “-2” and “-3” tests detect when one of either the direct or diffuse is “bad data.” The “-4” tests detect when the whole tracker was off, i.e., the Sum is about equal to TSW because the Diffuse is unshaded.

- tflg “total SW flag” - for TSW
- 0 = OK
 - 1 = $\text{TSW} < -20.0 \text{ Wm}^{-2}$ (bad data)
 - 2 = $\text{TSW} > 1500 \text{ Wm}^{-2}$ (bad data)

- dflg “diffuse flag” - for diffuse
- 0 = OK
 - 1 = $\text{Dif} < -20.0 \text{ Wm}^{-2}$ (bad data)
 - 2 = absolute difference between TSW and Sum $> 100.0 \text{ Wm}^{-2}$ (bad)
 - 9 = diffuse estimated from TSW-Dir

- rflg “direct flag” - for direct irradiance
- 0 = OK (measured)
 - 1 = original data bad, estimated by TSW-Dif
 - 9 = bad or missing, can't be estimated

- clrf clear detection flag
- 0 = not detected as clear-sky
 - 1 = this measurement detected as clear-sky and used for fitting

5. Contents of the *.SWF Output Files

The “*.swf” files are the primary algorithm output files, containing all the data that went in and passed the QC tests, QC flags to tell you what data failed which test and if the “measured” value here was estimated, and all the clear-sky irradiance estimates. What values are in what columns is explained below, and in the file “swfheadr.txt.”

5.1 Coefficient Data

Each “*.swf” output file contains the clear-sky fit coefficients retrieved or interpolated for that day. These coefficients are named as follows in the output files:

ASCII Files	ARM NetCDF Files	Meaning
Date	coef_date	date in YYYYMMDD format, based on LST
DFRa	coef_clearsky_difratio_a	"a" coefficient calculated for diffuse ratio
DFRb	coef_clearsky_difratio_b	"b" coefficient calculated for diffuse ratio
CSWa	coef_clearsky_gsw_a	"a" coefficient calculated for clear-sky global SW
CSWb	coef_clearsky_gsw_b	"b" coefficient calculated for clear-sky global SW
CSSWa	coef_clearsky_ssw_a	"a" coefficient calculated for clear-sky sum SW
CSSWb	coef_clearsky_ssw_b	"b" coefficient calculated for clear-sky sum SW
SCORc	coef_clearsky_dsw_a	offset coefficient calculated for clear-sky global-sum difference
SCORa	coef_clearsky_dsw_b	"a" coefficient calculated for clear-sky global-sum difference
SCORb	coef_clearsky_dsw_c	"b" coefficient calculated for clear-sky global-sum difference
ALBa/UVBa	coef_clearsky_albedo/uvb_a	"a" coefficient calculated for clear-sky albedo/UV
ALBb/UVBb	coef_clearsky_albedo/uvb_b	"b" coefficient calculated for clear-sky albedo/UV
AvgAU	sun_earth_distance_dailyaverage	daily average earth-sun distance in AUs
NOTE: If "Date" is all zeros, then the code was run in "one fit for all data" mode. In this case, to produce the daily clear-sky CSW and CSSW, divide the "a" coefficient by the square of the "AvgAU."		

5.2 Irradiance Data

The "*.swf" output files also include the following values:

ASCII Files	ARM NetCDF Files	Meaning
Zdate	base_time	date in YYYYMMDD format, based on GMT
Ztim	time_offset	time in hhmm format, based on GMT
Ldate	base_time_LST	date in YYYYMMDD format, based on LST
Ltim	time_offset_LST	time in hhmm format, based on LST
CosZ	solar_cos_z	Cosine of the solar zenith angle (used for fitting)
AU	sun_earth_distance	Earth - Sun distance in AUs
tsw	gswfluxdn_measured	measured downwelling SW from global pyranometer (W/m ²)
csw	gswfluxdn_clearskyfit	clear-sky downwelling SW from global pyranometer (W/m ²)
tswfcg	gswfluxdn_cloudeffect	csw minus tsw (W/m ²)
dif	difswfluxdn_measured	measured downwelling diffuse SW (W/m ²)
cdif	difswfluxdn_clearskyfit	clear-sky downwelling diffuse SW (W/m ²)
diffcg	difswfluxdn_cloudeffect	cdif minus dif (W/m ²)
dir	dirfluxdn_measured	measured downwelling direct SW (W/m ²)
cdir	dirfluxdn_clearskyfit	clear-sky downwelling direct SW (W/m ²)
difr	difratiodn_measured	measured diffuse ratio (dif/tsw) (W/m ²)
cdifr	difratiodn_clearskyfit	clear-sky diffuse ratio (cdif/csw) (W/m ²)
ssw	sswfluxdn_measured	measured downwelling sum (dif+dir) SW (W/m ²)
cssw	sswfluxdn_clearskyfit	clear-sky downwelling sum (cdif+cdir) SW (W/m ²)

sswfcg	sswfluxdn_cloudeffect	cssw minus ssw
sflg	qc_sswfluxdn	“sum flag” for Sum data and other QC
Tflg	qc_gswfluxdn	“total SW flag” for TSW QC
dflg	qc_difswfluxdn	“diffuse flag” for diffuse SW QC
rflg	qc_dirswfluxdn	“direct flag” for direct SW QC
clrf	flag_clearsky_detection	clear detection flag
alb/UVB	albedosw/guvbdn_measured	measured albedo (no units) or UVB (units as in input data)
calb/CUVB	albedosw/guvbdn_clearskyfit	clear-sky albedo (no units) or UVB (units as in input data)

6. Other Output Files

After the processing is completed (successfully), there will be many files besides the *.swf files in the directory where the executable was run. Some of these files have use in such areas as long-term data quality assessment, and a ready record of the “clearest” days in the run. At present, there are no equivalent ARM NetCDF versions of these files, however that may change in the future given experience and user feedback. Brief descriptions:

“*.cos” - files of daylight input data that are generated in the format that all but the front of the algorithm uses for processing. These include all the “good” data (not in the bad_data.cos file) used by the algorithm. These are also the files that contain the “extra” variables set in the configuration file, settings 23 and 24.

“*.cl2” - files used in the LAST iteration of coefficient fitting. These include all data detected as clear by the algorithm tests, plus all “effectively clear” data. These “effectively clear” data are those that are very close to the next to last iteration coefficient based clear irradiance values, and exhibiting very little, but slightly more than the set limit of, allowable variability, and pass a few other statistical tests that have been determined to eliminate data that do not well represent clear-sky equivalent values (i.e. “magic”). The idea here is to “fill in and extend” the “clear” data for better fitting. The initial clear ID is set to err on the side of caution, sometimes eliminating data that really is clear but a little noisy. Also, the clear ID process tends to eliminate early morning/late evening data due to normalization. This last “trick” allows us to use some of the early morning/late evening data and that data that, if not exactly COMPLETELY clear, are virtually identical to the clear value. This enhances the accuracy of the fitting for large solar zenith angles and/or local solar noon.

“*.clr” - files containing the “actually detected as clear” data. This same information is now included in the “*.swf” files as a clear flag (1 = identified as clear, 0 = not identified as clear). However, these files might also be useful, since they contain only the clear data for each day.

NOTE: The following files are “generic” file names, and are overwritten each time the code is run. If, for example, you want to keep a record of “bad data,” then you **MUST** rename the bad_data.cos and bad_data.swf files before you run the code again.

“bad_data.cos” - file includes all data that didn’t pass the initial irradiance comparison physical limits tests and were either replaced by differencing if possible, or if not then eliminated entirely from the processing. This file is useful to detect periods when the input data were inconsistent, as described in “swfheadr.txt.”

“bad_data.swf” - file includes all data that didn’t pass the algorithm advanced QC tests and were either replaced by differencing if possible, or if not then eliminated entirely from the processing. This file is useful to detect periods when the data were more subtly inconsistent, as described in “swfheadr.txt.”

“*.dir” – These files are the various directory file lists that were included in the run. These are helpful in debugging when things go awry, basically to see if the “front end” might have had problems, or whether the problem was internal to the “swclid1a” code.

“cnf_read.asc” – Another debugging tool, this file lists what the code read from the configuration file. This is handy to see if perhaps the configuration file had some lines missing, or not in the proper format, etc.

“swclreq*.asc” – These are the actual clear-sky fit coefficient files produced by the code run. They are useful in determining when the “clear enough” days occurred in the run, and to give an idea as to how clear each day was as they contain the number of clear data points detected each day. This accounting is handy if one is looking for particular type (clear or cloudy) of days in the data record. These files also contain the local solar noon zenith angle information for each day, and fitting check flag codes from the coefficient tests if the coefficients failed a test(s). (These flags replace the number of detected clear data values.) The test failure codes are as follows:

Flag Action

- 44 Not sufficient “high sun” clear points this day. Keep Diffuse ratio “a” and “b” coefficients for this day. Don’t keep the TSW “a” and “b” coefficients, instead interpolate from the last “good” TSW coefficients.
- 55 or -99 Not sufficient “low sun” clear points this day, or diffuse ratio “b” coefficient less than -0.95. Keep Diffuse ratio “a” coefficient and the TSW “a” and “b” coefficients for this day. Don’t keep the diffuse ratio “b” coefficient, instead interpolate from the last “good” “b” coefficient.
- 88 Diffuse ratio “b” coefficient greater than -0.4. Keep TSW “a” and “b” coefficients for this day. Don’t keep the diffuse ratio “a” and “b” coefficients, instead interpolate from the last “good” diffuse ratio coefficients.
- 11 Multiple tests failed, do not keep any coefficients.

The file numbers 1, 2, and 3 represent the initial iteration, last “middle” iteration, and final iteration results, respectively. Files numbers 1 and 2 include fitted coefficients for the total SW and diffuse ratio only, file number 3 includes all fit coefficients.

“clrcoef.asc” and “clrcoef3.asc” – These files contain the interpolated clear-sky fit coefficients (from the “swclreq*.asc” input) for the run. “clrcoef.asc” represents the last “middle” iteration values, and includes only the total SW and diffuse ratio coefficients. “clrcoef3.asc” represents the final iteration values, and includes all fit coefficients.

“swclrid*.day” - These files contain information useful for both debugging problems, and as a record of some aspects of the data run. Files 1 and 2 represent the initial iteration and last “middle” iteration, respectively. They contain the number of detected clear-sky data (N), and the number of daylight data points (TotN) for each day. They also include the average normalized diffuse ratio (AvNDR) and average normalized total SW (AvNSW) values for the detected clear-sky data for that day. The “swclrid3.day” file is for the final iteration, and additionally includes the number of data that were considered for detecting clear-sky (N), the number of clear detected via “normal” means (#clr) and the number detected using the additional *.cl2 method described above (#cl2). Additionally, the number of “partly cloudy” (#pcldy), and “overcast” (#ovc) using a VERY primitive classification system (i.e., only relatively accurate) are included. Comparing these values is useful in describing the type of day each day was. Finally, the local solar noon zenith angle (sncz) and average earth-sun distance in AUs (AvgAU), and the min (czmin) and max (czmax) cosine of the solar zenith angle of the detected clear-sky data, are included. In all, these files are a wealth of descriptive information about the individual days, and how the code treated these days, useful for both characterizing the data set and for debugging.

“posfcg.asc” - This file contains data, in the same format as in the “*.swf” files, for “positive cloud forcing” periods, i.e. when the measured downwelling total SW was significantly greater than the corresponding clear-sky amount. This positive cloud forcing is caused by the enhancement of the downwelling diffuse SW by the presence of clouds, while at the same time the direct SW is not blocked by clouds. This file can be used to study the frequency at which these events occur, and at what magnitude, for each site under study.

“swfcgday.asc” - This file is a daylight average of the data included in the “*.swf” files. The values in this file, does not necessarily well represent each days average amount unless there was little or no missing or “bad” data that day. To determine the representativeness, included in these files are the total number of daylight data in the input files (dayN), and the total number of “good” data for each of the total SW (tswN), the total clear-sky SW (cswN), the diffuse SW (difN) and the clear-sky diffuse SW (cdifN) for each day. By looking at the time series of “dayN”, you can see what days had significant missing data. Note that the dayN exhibits seasonal changes related to day length, as it changes in respect to solar declination angle changes. Comparing the time series of tswN, cswN, difN, and cdifN (along with DayN) then gives information relating to possible instrument measurement problems in the data set. This information then relates to how well it was possible for the algorithm to perform accurate clear-sky detection and fitting for the data run. The algorithm performance is naturally dependent on the quality and continuity of the data input into the code, i.e. GIGO...

“swnulclr.bat” - kind of left over from earlier versions of the code. This is a batch file that is generated each time, and can be run to delete files automatically that were for days that were not clear enough (*.clr), or beyond the last “clear enough” day.

7. Running the Fractional Sky Cover Algorithm

The sky cover algorithm processes the “*.swf” files as input, and is comprised of 3 files in the PC version: “armswfcf.bat,” “armswfcf.cnf,” and “armswfcf.exe.” As before, the batch file first produces a directory file, then calls the executable. The configuration file sets certain parameters used by the executable. Again, you can set a path (this time up to 600 characters in length) to get the input files elsewhere, output files are generated in the same directory as the executable. The batch file creates a list of the “*.swf” input files.

For this retrieval code, all the configuration file values are set for you and should not need to be changed. The only line you’ll need to change is the first one, where the input file path is set. An explanation of the ideas behind this algorithm and the values set for the cloud fraction regression formula can be found in (Long et al. 1999). Again, the format of this configuration file is set for reading by the executable, and must not be changed. The value-by-value description of the configuration file is as follows:

```
out\    /* path for input files (incl. trailing “\”) */
```

This is where the input file path is set.

```
17    /* number of data (reals) in input files */
```

This tells the algorithm the number of reals to expect on each line of the input files. For the “*.swf” files, this value is 17, and should not be changed

```
0    /* print flag for *.swc output files (0 = no *.swc files) */
```

```
0    /* print flag for fitting output files (0 = no files) */
```

These are flags to tell the algorithm whether to print out files used for debugging and algorithm development. Leave these set to zero.

```
2.255 /* Cloud Fraction Equation a coefficient (real) */
```

```
0.9381 /* Cloud Fraction Equation b (exponential) coefficient (real) */
```

These are the empirically derived coefficients for the sky cover estimation equation relating the normalized diffuse cloud effect to sky cover (Long et al. 1999). These were originally left configurable early in the development of the method. However, subsequent study has yet to produce an example of these coefficients not being appropriate. Studies are still in progress using this algorithm, so perhaps a climactic condition wherein these coefficients may have to be adjusted will be found. For now, these should be left set as they are.

```
0.37 /* Difcld limit for ovc detection */
0.95 /* Diffuse ratio limit for ovc detection */
0.05 /* DR StDev limit for Ovc detection */
3 /* number of measurements for running DR StDev */
```

As noted in Long et al. (1999), the normalized diffuse cloud effect ratio is not unique as the sky cover approaches unity. This is because thick overcast and heavy cumulonimbus clouds produce depressed, rather than enhanced, diffuse cloud effect. However, the periods of thick overcast and heavy cumulonimbus occurrence also exhibit distinct characteristics of magnitude and temporal variability of the diffuse ratio, total SW cloud effect, etc. These characteristics are detected in the data time series, and the sky cover is set to 1.0 for these times. These configuration settings are part of the overcast detection, and again have always worked as intended for all 1-minute resolution data sets processed so far. They should be left as they are.

```
0.2 /* CosZ limit for cloud fraction retrieval */
```

Because of the increased uncertainty of the SW measurements, the increased percentage uncertainty in the estimated diffuse clear-sky SW, and the sun angle limitations of the original sky imager retrievals used as “truth” in deriving the sky cover equation coefficients, the sky cover estimates by this method increase in uncertainty at low sun angles. This line sets the limit, as the cosine of the solar zenith angle, for which the algorithm will produce sky cover estimates. It is currently set for a CosZ of 0.2, which translates to a solar elevation angle of about 11 degrees (solar zenith angle of 79 degrees).

```
-8.0 /* “bad data” limit for cosz */
-8.0 /* “bad data” limit for au */
-998.0 /* “bad data” limit for tsw */
-998.0 /* “bad data” limit for csw */
Etc...
```

These lines tell the algorithm what the “bad data” values are for each input value. The algorithm looks for input values LESS THAN the values here, so they must be set slightly greater than the actual “bad data” value. For instance, the “*.swf” “bad data” value for total SW is “-999.0.” Thus the limit set here is “-998.0”, so that the actual “bad data” value is 1 Wm⁻² less than this limit.

The executable takes each input file (*.swf) name stem, and generates an output file with this stem and a “*.c15” extension. The output file contains 15-minute averages of all values in the input file (except for the data QC flags), and a 15-minute average fractional sky cover estimate. The value names are the same as those for the “*.swf” files, noted above. Also included in the output files is 4 “flags,” listed here as ASCII version name first, followed by the ARM NetCDF version name in parenthesis:

Ncf (nsamples_cloudfraction) - the number of data points that went into this 15-minute average of cloud fraction. Ideally should be 15.

Ncsw (nsamples_gswfluxdn_clearskyfit) - the number of data points that went into the 15-minute average of CSW (clear-sky fit to TSW). Ideally should be 15. This serves as an indication of the maximum number of data in the original input files that was available during this 15-minutes.

Nssw (nsamples_sswfluxdn_measured) - the number of data points that went into this 15-minute average of SSW (sum of direct + diffuse). Ideally should be 15. This serves as a check of if either diffuse or direct data was missing or bad.

Nclr (nsamples_clearsky_detection) - the number of data points that went into this 15-minute average of cloud fraction that were originally detected as clear by the clear ID algorithm. If Ncf is 15 and this number is 10, then this means 10 of the 15 1-minute periods were detected as clear originally, and the other 5 that went into this average came from the cloud fraction regression formula.

Another output file is generated which is based in part on the date of the first input file name in the directory file, and ending with an “.a15” extension. This file is the same format as the “*.c15” files, but contains ALL the 15-minute data from this run of the code. At present, there is no equivalent ARM NetCDF version of this file.

8. References

Long, C. N., and T. P. Ackerman, 2000: “Identification of clear skies from broadband pyranometer measurements and calculation of downwelling shortwave cloud effects.” *J. Geophys. Res.*, 105, No. D12, 1,5609-1,5626. Available URL: <http://playground.arm.gov/~clong>

Long, C. N., T. P. Ackerman, J. J. DeLuisi, and J. Augustine, 1999: “Estimation of fractional sky cover from broadband SW radiometer measurements.” In *Proceedings AMS 10th Conf. on Atmos. Rad.*, June 28-July 2, 1999, Madison, Wisconsin. Available URL: <http://playground.arm.gov/~clong>