NREL's SOLPOS 2.0: Documentation

SOLPOS.C

Distributed by the National Renewable Energy Laboratory
Center for Renewable Energy Resources
Renewable Resource Data Center
February 2000

NOTICE

This report was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes an legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial produce, process or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation or favoring by the United States government or any agency thereof. The view and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or any agency thereof.

This C function calculates the apparent solar position and intensity (theoretical maximum solar energy) based on the date, time, and location on Earth. The software has been tested on a variety of platforms, but as noted above, is not guaranteed to work on yours. It is provided here as a convenience.

This document provides only a general overview of the software functionality. The accompanying sample program stest00.c provides additional information by example on how the function is set up and called from an application program. That program serves as the only tutorial for the use of S_solpos.

The module contains three functions:

S_solpos	Performs calculations
S_init	Initializes S_solpos
S_decode	Decodes the return value from S_solpos

To obtain references for the algorithms see the REFERENCES section below. Comments in the source code specify references for each function.

=====================================

```
S_solpos    (computes solar position and intensity from time and place)
```

```
        INPUTS:    (via posdata struct defined in solpos00.h)
                  year, daynum, hour, minute, second, latitude, longitude, timezone,
                  interval
        OPTIONAL:  (via posdata struct)
                  month, day, press, temp, tilt, aspect, function
        OUTPUTS:   EVERY variable in the struct posdata (defined in solpos00.h)


S_init     (optional initialization for all input parameters in the posdata struct)

        INPUTS:    struct posdata*
        OUTPUTS:   struct posdata*
             Initializes the required S_solpos INPUTS above to out-of-bounds conditions,
             forcing the user to supply the parameters; initializes the OPTIONAL S_solpos
             inputs above to nominal values. See listing below for default values
             provided by S_init.


S_decode    (optional utility for decoding the S_solpos return code)

        INPUTS:    long int S_solpos return value, struct posdata*
        OUTOUTS:   Text to stderr
```

ALPHABETICAL LIST OF COMMON VARIABLES

The I/O column contains a letter code:

I: INPUT variable
O: OUTPUT variabl
T: TRANSITIONAL variable used in the algorithm, of interest only to the solar radiation modelers and available to you because you may be one of them.

The FUNCTION column indicates which sub-function within solpos must be switched on using the "function" parameter to calculate the target output variable. All function codes are defined in the solpos00.h file. The default S_ALL mask calculates all output variables. Multiple function masks may be ORed to create a composite function switch. For example, (S_TST | S_SBCF) will force the calculation of the shadow band correction factor as well as all variables required for S_TST (true solar time). Specifying only the functions necessary for required output variables might allow solpos to execute more quickly.

The S_DOY mask works as a toggle between the input date represented as a day of year number (daynum) and an input date represented by month and day of month. To set the switch (to use daynum input), the mask is ORed with the function variable; to clear the switch (to use month and day input), the mask is inverted and ANDed.

For example:

```
pdat->function |= S_DOY /* (sets daynum input) */
pdat->function &= ~S_DOY /* (sets month and day input) */
```

Whichever date form is used, S_solpos will calculate and return the variables(s) of the other form. See the sample program stest00.c for other examples.

```
  VARIABLE    I/O  Function         Description
----------- ---- -------- --------------------------
---------    ---  -------- --------------------------
                           ----

/**** INTEGERS
****/
```

int day	I/O:	S_DOY	Day of month (May 27 = 27, etc.) solpos will CALCULATE this by default, or will optionally require it as input depending on the setting of the S_DOY function switch.
int daynum	I/O:	S_DOY	Day number (day of year; Feb 1 = 32) solpos REQUIRES this by default, but will optionally calculate it from year, month, and day depending on the setting of the S_DOY function switch.
int function	I:		Bit-oriented switch to choose function) for desired output..
int hour	I:		Hour of day, 0 — 24. (Time 24:00:00 is treated internally as time 00:00:00 of the following day.)
int interval	I:		Interval of a measurement period in seconds. Forces solpos to use the time and date from the interval midpoint. The INPUT time (hour, minute, and second) is assumed to be the END of the measurement interval.
int minute	I:		Minute of hour, 0 - 59.
int month	I/O:	S_DOY	Month number (Jan = 1, Feb = 2, etc.) solpos will CALCULATE this by default or will optionally require it as input depending on the setting of the S_DOY function switch.
int second	I:		Second of minute, 0 - 59.
int year	I:		4-digit year (2-digit years NOT allowed)
/**** FLOATS ****/			
float amass	O:	S_AMASS	Relative optical airmass
float ampress	O:	S_AMASS	Pressure-corrected airmass
float aspect	I:		Azimuth of panel surface (direction it faces) N=0, E=90, S=180,

			W=270, DEFAULT = 180
float azim	O:	S_SOLAZM	Solar azimuth angle: N=0, E=90, S=180,W=270
float cosinc	O:	S_TILT	Cosine of solar incidence angle on panel
float coszen	O:	S_REFRAC	Cosine of refraction corrected solar zenith angle
float dayang	T:	S_GEOM	Day angle (daynum*360/year-length) degrees
float declin	T:	S_GEOM	Declination--zenith angle of solar noon at equator, degrees NORTH
float eclong	T:	S_GEOM	Ecliptic longitude, degrees
float ecobli	T:	S_GEOM	Obliquity of ecliptic
float ectime	T:	S_GEOM	Time of ecliptic calculations
float elevetr	O:	S_REFRAC	Solar elevation, no atmospheric correction (= ETR)
float elevref	O:	S_REFRAC	Solar elevation angle, degrees from horizon, refracted
float eqntim	T:	S_TST	Equation of time (TST - LMT), minutes
float erv	T:	S_GEOM	Earth radius vector (multiplied to solar constant)
float etr	O:	S_ETR	Extraterrestrial (top-of-atmosphere) W/sq m global horizontal solar irradiance
float etrn	O:	S_ETR	Extraterrestrial (top-of-atmosphere) W/sq m direct normal solar irradiance
float etrtilt	O:	S_TILT	Extraterrestrial (top-of-atmosphere) W/sq m global irradiance on a tilted surface
float gmst	T:	S_GEOM	Greenwich mean sidereal time, hours
float hrang	T:	S_GEOM	Hour angle--hour of sun from solar noon

			degrees WEST
float julday	T:	S_GEOM	Julian Day of 1 JAN 2000 minusn 2,400,000 days (in order to regain single precision)
float latitude	I:		Latitude, degrees north (south negative)
float longitude	I:		Longitude, degrees east (west negative)
float lmst	T:	S_GEOM	Local mean sidereal time, degrees
float mnanom	T:	S_GEOM	Mean anomaly, degrees
float mnlong	T:	S_GEOM	Mean longitude, degrees
float rascen	T:	S_GEOM	Right ascension, degrees
float press	I:		Surface pressure, millibars, used for refraction correction and ampress
float prime	O:	S_PRIME	Factor that normalizes Kt, Kn, etc.
float sbcf	O:	S_SBCF	Shadow-band correction factor
float sbwid	I:		Shadow-band width (cm)
float sbrad	I:		Shadow-band radius (cm)
float sbsky	I:		Shadow-band sky facto
float solcon	I:		Solar constant (NREL uses 1367 W/sq m)
float ssha	T:	S_SRHA	Sunset(/rise) hour angle, degrees
float sretr	O:	S_SRSS	Sunrise time, minutes from midnight, local, WITHOUT refraction
float ssetr	O:	S_SRSS	Sunset time, minutes from midnight, local, WITHOUT refraction
float temp	I:		Ambient dry-bulb temperature, degrees C, used for refraction correction
float tilt	I:		Degrees tilt from horizontal of panel
float timezone	I:		Time zone, east (west negative)., USA: Mountain = -7, Central = -6, etc.

float tst	T:	S_TST	True solar time, minutes from midnight
floattstfix	T:	S_TST	True solar time - local standard time
float unprime	O:	S_PRIME	Factor that denormalizes Kt', Kn', etc.
float utime	T:	S_GEOM	Universal (Greenwich) standard time
float zenetr	T:	S_ZENETR	Solar zenith angle, no atmospheric correction (= ETR)
float zenref	O:	S_REFRAC	Solar zenith angle, deg. from zenith, refracted.

All functions require the input parameters for time, date, latitude, longitude, time zone, and measurement interval. Some functions may require additional input parameters. The table below indicates with an "X" which, if any, additional input parameters are required for each function. After determining the output variables you require from the above list, make note of the required functions, then determine the required inputs from the table:

Function	**-------------------- Required Inputs ------------------**							
	solcon	press	sbwid	sbrad	sbsky	temp	tilt	aspect
S_AMASS	--	X	--	--	--	X	--	--
S_DOY	--	--	--	--	--	--	--	--
S_ETR	X	X	--	--	--	X	--	--
S_GEOM	--	--	--	--	--	--	--	--
S_REFRAC	--	X	--	--	--	X	--	--
S_PRIME	--	X	--	--	--	X	--	--
S_SOLAZM	--	--	--	--	--	--	--	--
S_SRSS	--	--	--	--	--	--	--	--
S_SSHA	--	--	--	--	--	--	--	--
S_SBCF	--	--	X	X	X	--	--	--
S_TILT	X	X	--	--	--	X	X	X
S_TST	--	--	--	--	--	--	--	--
S_ZENETR	--	--	--	--	--	--	--	--

The S_init function provides nominal values for the above inputs. The values are listed below (note that time and location variables are initialized out of bounds to force the user to provide valid inputs):

```
day      =   -99    /* undefined */
daynum   =  -999    /* undefined */
hour     =   -99    /* undefined */
minute   =   -99    /* undefined */
month    =   -99    /* undefined */
second   =   -99    /* undefined */
year     =   -99    /* undefined */
```

```
interval  =      0     /* instantaneous */
aspect    =  180.0     /* south */
latitude  =  -99.0     /* undefined */
longitude = -999.0     /* undefined */
press     = 1013.0     /* standard pressure */
solcon    = 1367.0     /* NREL uses this */
temp      =   15.0     /* Temperature of the standard atmosphere */
tilt      =    0.0     /* horizontal */
timezone  =  -99.0     /* undefined */
sbwid     =    7.6     /* Eppley shadowband */
sbrad     =   31.7     /* Eppley shadowband */
sbsky     =   0.04     /* Eppley shadowband */
function  =  S_ALL     /* calculate ALL output parameters */
```

Certain conditions exist during which some of the output variables are undefined or cannot be calculated. In these cases, the variables are returned with flag values indicating such. In other cases, the variables may return a realistic, though invalid, value. These variables and the flag values or invalid conditions are listed below:

```
amass      -1.0 at zenetr angles greater than 93.0 degrees
ampress    -1.0 at zenetr angles greater than 93.0 degrees<
azim       invalid at zenetr angle 0.0 or latitude +/-90.0 or at night
elevetr    limited to -9 degrees at night
etr        0.0 at night
etrn       0.0 at night
etrtilt    0.0 when cosinc is less than 0
prime      invalid at zenetr angles greater than 93.0 degrees
sretr      +/- 2999.0 during periods of 24 hour sunup or sundown
ssetr      +/- 2999.0 during periods of 24 hour sunup or sundown
ssha       invalid at the North and South Poles
unprime    invalid at zenetr angles greater than 93.0 degrees
zenetr     limited to 99.0 degrees at night
```

S_solpos returns a long integer error code. Each bit position in the long int represents an error in the range of a particular input parameter. The S_decode function in solpos.c examines the return code for errors and can be used as is or as a template for building an application-specific function.

The bit positions for each error are defined in solpos00.h, and are listed below. (Bit positions are from least significant to most significant.)

```
/*        Code            Bit        Parameter           Range
     ===============      ===  ===================     =========    */
enum {S_YEAR_ERROR,     /*  0   year                  1950 - 2050   */
     S_MONTH_ERROR,     /*  1   month                    1 - 12     */
     S_DAY_ERROR,       /*  2   day-of-month             1 - 31     */
     S_DOY_ERROR,       /*  3   day-of-year              1 - 366    */
     S_HOUR_ERROR,      /*  4   hour                     0 - 24     */
     S_MINUTE_ERROR,    /*  5   minute                   0 - 59     */
     S_SECOND_ERROR,    /*  6   second                   0 - 59     */
     S_TZONE_ERROR,     /*  7   time zone              -12 - 12     */
```

```
    S_INTRVL_ERROR,     /*  8  interval (seconds)        0 - 28800  */
    S_LAT_ERROR,        /*  9  latitude                   -90 - 90  */
    S_LON_ERROR,        /* 10  longitude                -180 - 180  */
    S_TEMP_ERROR,       /* 11  temperature (deg. C)     -100 - 100  */
    S_PRESS_ERROR,      /* 12  pressure (millibars)       0 - 2000  */
    S_TILT_ERROR,       /* 13  tilt                       -90 - 90  */
    S_ASPECT_ERROR,     /* 14  aspect                   -360 - 360  */
    S_SBWID_ERROR,      /* 15  shadow band width (cm)      1 - 100  */
    S_SBRAD_ERROR,      /* 16  shadow band radius (cm)     1 - 100  */
    S_SBSKY_ERROR};     /* 17  shadow band sky factor       -1 - 1  */
```

R E F E R E N C E S

ASTRONOMICAL SOLAR POSITION:

Michalsky, J. 1988. The Astronomical Almanac's algorithm for approximate solar
position (1950-2050). Solar Energy 40 (3), 227-235.

Michalsky, J. 1988. ERRATA: The atronomical almanac's algorithm for approximate
solar position (1950-2050). Solar Energy 41 (1), 113.

DISTANCE FROM SUN TO EARTH

Spencer, J. W. 1971. Fourier series representation of the position of the sun. Search 2
(5), 172.

NOTE: This paper gives solar position algorithms as well, but the Michalsky/Almanac algorithm above is more accurate.

ATMOSPHERIC REFRACTION CORRECTION

Zimmerman, John C. 1981. Sun-pointing programs and their accuracy. SAND81-0761,
Experimental Systems Operation Division 4721, Sandia National Laboratories, Albuquerque, NM.

SHADOW BAND CORRECTION FACTOR

Drummond, A. J. 1956. A contribution to absolute pyrheliometry. Q. J. R. Meteorol.2
Soc. 82, 481-493..

RELATIVE OPTICAL AIR MASS

Kasten, F. and Young, A. 1989. Revised optical air mass tables and

approximation
formula. Applied Optics 28 (22), 4735-4738.

RENORMALIZATION OF KT ("PRIME")

Perez, R., P. Ineichen, Seals, R., & Zelenka, A. 1990. Making full use of the clearness
index for parameterizing hourly insolation conditions. Solar Energy 45 (2),
111-114.

SOLAR POSITION RELATIVE TO EARTH

Iqbal, M. 1983. An Introduction to Solar Radiation. Academic Press, NY.

NOTE: The 1983 edition contains typographic errors in coefficients of some equations. Further, many algorithms given in this book are no longer the best. However, this book gives a complete overview of the issues and methods of measuring and modeling solar radiation

---

Return to RReDC home page (http://www.nrel.gov/rredc)
