solar Documentation

Release

Edwin Lee

April 27, 2016

Contents

Contents:

CHAPTER 1

solar package

1.1 Submodules

1.2 solar.solar module

1.3 Module contents

class solar.**AngularValueType**(*radians=None*, *degrees=None*)

This class combines a numeric value with an angular measurement unit.

Proper construction should call constructor with either radians=x or degrees=y; not both. The constructor will calculate the complementary. The value of the angle can then be retrieve from the .degrees or .radians value as needed.

Another class member, called .valued is available to determine if the class members contain meaningful values.

If the constructor is called without either argument, the .valued variable is False, and the numeric variables are None.

If the constructor is called with both arguments, they will be assigned if they agree to within a small tolerance, or a ValueError is thrown.

solar.**altitudeAngle**(*datetimeInstance*, *daylightSavingsOn*, *longitude*, *standardMeridian*, *latitude*)

Calculates the current solar altitude angle for a given set of time and location conditions. The solar altitude angle is the angle between the sun rays and the horizontal plane.

Parameters

- **datetimeInstance** – [Python native datetime.datetime] The current date and time to be used in this calculation of day of year.
- **daylightSavingsOn** – [Boolean] A flag if the current time is a daylight savings number. If True, the hour is decremented.
- **longitude** – [Float] [degrees west] The current longitude in degrees west of the prime meridian. For Golden, CO, the variable should be = 105.2.
- **standardMeridian** – [Float] [degrees west] The local standard meridian for the location, in degrees west of the prime meridian. For Golden, CO, the variable should be = 105.
- **latitude** – [Float] [degrees north] The local latitude for the location, in degrees north of the equator. For Golden, CO, the variable should be = 39.75.

Returns [AngularValueType] The solar altitude angle in an AngularValueType with both radian and degree versions

solar.**dayOfYear**(*datetimeInstance*)

Calculates the day of year (1-366) given a Python datetime.datetime instance. Basically a wrapper around the native tm_yday parameter to ensure it is a full datetime instance in subsequent calculations. If the type is *not* datetime.datetime, this will throw a TypeError

Parameters **datetimeInstance** – [Python native datetime.datetime] The current date and time to be used in this calculation of day of year.

Returns [Integer] [dimensionless] The day of year, from 1 to 365 for non-leap years and 1-366 for leap years.

solar.**declinationAngle**(*datetimeInstance*)

Calculates the Solar Declination Angle for a given date. The solar declination angle is the angle between a line connecting the center of the sun and earth and the project of that line on the equatorial plane. Calculation is based on McQuiston.

Parameters **datetimeInstance** – [Python native datetime.datetime] The current date and time to be used in this calculation of day of year.

Returns [AngularValueType] The solar declination angle in an AngularValueType with both radian and degree versions

solar.**directRadiationOnSurface**(*datetimeInstance*, *daylightSavingsOn*, *longitude*, *standardMeridian*, *latitude*, *surfaceAzimuthDeg*, *horizontalDirectIrradiation*)

Calculates the amount of direct solar radiation incident on a surface for a given set of time and location conditions, a surface orientation, and a total global horizontal direct irradiation. This is merely the global horizontal direct solar irradiation time the angle of incidence on the surface.

Parameters

- **datetimeInstance** – [Python native datetime.datetime] The current date and time to be used in this calculation of day of year.
- **daylightSavingsOn** – [Boolean] A flag if the current time is a daylight savings number. If True, the hour is decremented.
- **longitude** – [Float] [degrees west] The current longitude in degrees west of the prime meridian. For Golden, CO, the variable should be = 105.2.
- **standardMeridian** – [Float] [degrees west] The local standard meridian for the location, in degrees west of the prime meridian. For Golden, CO, the variable should be = 105.
- **latitude** – [Float] [degrees north] The local latitude for the location, in degrees north of the equator. For Golden, CO, the variable should be = 39.75.
- **surfaceAzimuthDeg** – [Float] [degrees CW from North] The angle between north and the outward facing normal vector of the wall, measured as positive clockwise from south (southwest facing surface: 225, northwest facing surface: 315)
- **horizontalDirectIrradiation** – [Float] [any] The global horizontal direct irradiation at the location.

Returns [Dictionary {DR, Float}] The incident direct radiation on the surface. The units of this return value match the units of the parameter :horizontalDirectIrradiation:

solar.**equationOfTime**(*datetimeInstance*)

Calculates the Equation of Time for a given date. I wasn't able to get the McQuiston equation to match the values in the given table. I ended up using a different formulation here: http://holbert.faculty.asu.edu/eee463/SolarCalcs.pdf.

Parameters **datetimeInstance** – [Python native datetime.datetime] The current date and time to be used in this calculation of day of year.

Returns [Float] The equation of time, which is the difference between local civil time and local solar time

solar.**getDirectDiffuseSplit**(*datetimeInstance*, *daylightSavingsOn*, *longitude*, *standardMeridian*, *latitude*, *horizontalTotalIrradiation*)

Calculates the direct and diffuse split for a given amount of global total horizontal irradiation

Parameters

- **datetimeInstance** – [Python native datetime.datetime] The current date and time to be used in this calculation of day of year.
- **daylightSavingsOn** – [Boolean] A flag if the current time is a daylight savings number. If True, the hour is decremented.
- **longitude** – [Float] [degrees west] The current longitude in degrees west of the prime meridian. For Golden, CO, the variable should be = 105.2.
- **standardMeridian** – [Float] [degrees west] The local standard meridian for the location, in degrees west of the prime meridian. For Golden, CO, the variable should be = 105.
- **latitude** – [Float] [degrees north] The local latitude for the location, in degrees north of the equator. For Golden, CO, the variable should be = 39.75.
- **horizontalTotalIrradiation** – [Float] The total horizontal irradiation at a given time

Returns [Dictionary {String, Float}] A dictionary returning the direct and diffuse portions of a given horizontal irradiation value, two keys: "direct" and "diffuse" are in the dictionary

solar.**hourAngle**(*datetimeInstance*, *daylightSavingsOn*, *longitude*, *standardMeridian*)

Calculates the current hour angle for a given set of time and location conditions. The hour angle is the angle between solar noon and the current solar angle, so at local solar noon the value is zero, in the morning it is below zero, and in the afternoon it is positive.

Parameters

- **datetimeInstance** – [Python native datetime.datetime] The current date and time to be used in this calculation of day of year.
- **daylightSavingsOn** – [Boolean] A flag if the current time is a daylight savings number. If True, the hour is decremented.
- **longitude** – [Float] [degrees west] The current longitude in degrees west of the prime meridian. For Golden, CO, the variable should be = 105.2.
- **standardMeridian** – [Float] [degrees west] The local standard meridian for the location, in degrees west of the prime meridian. For Golden, CO, the variable should be = 105.

Returns [AngularValueType] The hour angle in an AngularValueType with both radian and degree versions

solar.**localCivilTime**(*datetimeInstance*, *daylightSavingsOn*, *longitude*, *standardMeridian*)

Calculates the local civil time for a given set of time and location conditions. The local civil time is the local time based on prime meridian and longitude

Parameters

- **datetimeInstance** – [Python native datetime.datetime] The current date and time to be used in this calculation of day of year.

- **daylightSavingsOn** – [Boolean] A flag if the current time is a daylight savings number. If True, the hour is decremented.
- **longitude** – [Float] [degrees west] The current longitude in degrees west of the prime meridian. For Golden, CO, the variable should be = 105.2.
- **standardMeridian** – [Float] [degrees west] The local standard meridian for the location, in degrees west of the prime meridian. For Golden, CO, the variable should be = 105.

Returns [Float] [hours] Returns the local civil time in hours for the given date/time/location

solar.**localSolarTime**(*datetimeInstance*, *daylightSavingsOn*, *longitude*, *standardMeridian*)

Calculates the local solar time for a given set of time and location conditions. The local solar time is the local civil time that has been corrected by the equation of time.

Parameters

- **datetimeInstance** – [Python native datetime.datetime] The current date and time to be used in this calculation of day of year.
- **daylightSavingsOn** – [Boolean] A flag if the current time is a daylight savings number. If True, the hour is decremented.
- **longitude** – [Float] [degrees west] The current longitude in degrees west of the prime meridian. For Golden, CO, the variable should be = 105.2.
- **standardMeridian** – [Float] [degrees west] The local standard meridian for the location, in degrees west of the prime meridian. For Golden, CO, the variable should be = 105.

Returns [Float] [hours] Returns the local solar time in hours for the given date/time/location

solar.**solarAngleOfIncidence**(*datetimeInstance*, *daylightSavingsOn*, *longitude*, *standardMeridian*, *latitude*, *surfaceAzimuthDeg*)

Calculates the solar angle of incidence for a given set of time and location conditions, and a surface orientation. The solar angle of incidence is the angle between the solar ray vector incident on the surface, and the outward facing surface normal vector.

Parameters

- **datetimeInstance** – [Python native datetime.datetime] The current date and time to be used in this calculation of day of year.
- **daylightSavingsOn** – [Boolean] A flag if the current time is a daylight savings number. If True, the hour is decremented.
- **longitude** – [Float] [degrees west] The current longitude in degrees west of the prime meridian. For Golden, CO, the variable should be = 105.2.
- **standardMeridian** – [Float] [degrees west] The local standard meridian for the location, in degrees west of the prime meridian. For Golden, CO, the variable should be = 105.
- **latitude** – [Float] [degrees north] The local latitude for the location, in degrees north of the equator. For Golden, CO, the variable should be = 39.75.
- **surfaceAzimuthDeg** – [Float] [degrees CW from North] The angle between north and the outward facing normal vector of the wall, measured as positive clockwise from south (southwest facing surface: 225, northwest facing surface: 315)

Returns [AngularValueType] The solar angle of incidence in an AngularValueType with both radian and degree versions. NOTE: If the sun is down, or behind the surface, the Float values in the dictionary are None.

solar.**solarAzimuthAngle**(*datetimeInstance*, *daylightSavingsOn*, *longitude*, *standardMeridian*, *latitude*)

Calculates the current solar azimuth angle for a given set of time and location conditions. The solar azimuth angle is the angle in the horizontal plane between due north and the sun. It is measured clockwise, so that east is +90 degrees and west is +270 degrees.

Parameters

- **datetimeInstance** – [Python native datetime.datetime] The current date and time to be used in this calculation of day of year.
- **daylightSavingsOn** – [Boolean] A flag if the current time is a daylight savings number. If True, the hour is decremented.
- **longitude** – [Float] [degrees west] The current longitude in degrees west of the prime meridian. For Golden, CO, the variable should be = 105.2.
- **standardMeridian** – [Float] [degrees west] The local standard meridian for the location, in degrees west of the prime meridian. For Golden, CO, the variable should be = 105.
- **latitude** – [Float] [degrees north] The local latitude for the location, in degrees north of the equator. For Golden, CO, the variable should be = 39.75.

Returns [AngularValueType] The solar azimuth angle in an AngularValueType with both radian and degree versions. NOTE: If the sun is down, the Float values in the dictionary are None.

solar.**wallAzimuthAngle**(*datetimeInstance*, *daylightSavingsOn*, *longitude*, *standardMeridian*, *latitude*, *surfaceAzimuthDeg*)

Calculates the current wall azimuth angle for a given set of time and location conditions, and a surface orientation. The wall azimiuth angle is the angle in the horizontal plane between the solar azimuth and the vertical wall's outward facing normal vector.

Parameters

- **datetimeInstance** – [Python native datetime.datetime] The current date and time to be used in this calculation of day of year.
- **daylightSavingsOn** – [Boolean] A flag if the current time is a daylight savings number. If True, the hour is decremented.
- **longitude** – [Float] [degrees west] The current longitude in degrees west of the prime meridian. For Golden, CO, the variable should be = 105.2.
- **standardMeridian** – [Float] [degrees west] The local standard meridian for the location, in degrees west of the prime meridian. For Golden, CO, the variable should be = 105.
- **latitude** – [Float] [degrees north] The local latitude for the location, in degrees north of the equator. For Golden, CO, the variable should be = 39.75.
- **surfaceAzimuthDeg** – [Float] [degrees CW from North] The angle between north and the outward facing normal vector of the wall, measured as positive clockwise from south (southwest facing surface: 225, northwest facing surface: 315)

Returns [AngularValueType] The wall azimuth angle in an AngularValueType with both radian and degree versions. NOTE: If the sun is behind the surface, the Float values in the dictionary are None.

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

S

A

D

E

G

H

L

S

W