

APPENDIX: INSTRUCTIONS FOR USING ASTROCALC4R FROM R

1. Find or compile the appropriate library (AstroCalc4R.dll in Windows or AstroCalc4R.so in Linux). All source code, compiled code and R code are available for free at <http://fish.nefsc.noaa.gov/AstroCalc4R>. The steps depend on your compiler. However, if R and the Rtools software (free at www.murdoch-sutherland.com/Rtools/) are installed on your Windows or Linux system, then you probably have all of the software required to compile and link the AstroCalc4R.c program code.
 - a. To make a new library, open a command window, and then navigate to the directory containing AstroCalc4R.c and myfuncs1.c . Use the Rtools command: **R CMD SHLIB AstroCalc4R.c** to make the library.
2. The easiest way to use our software from a library file is probably from the R programming language (version 2.12.1 or later) using our R function (also called AstroCalc4R). The dynamic link library or shared object can be called from other software, such as spreadsheets, but the procedures for doing so are not covered here.
3. The first step is to either paste the AstroCalc4R.r script into the R command window or read the script from a file using the R source function.
4. The arguments passed to AstroCalc4Rs in R may be a single numbers or vectors. However all of the arguments must be the same length. The R function AstroCalc4R will check the input data extensively to be sure it is valid and internally consistent, and stop with a warning message if problems are found.
5. After calculations are completed by the C library, the AstroCalc4R function in R will check to see if any results are for locations near the poles during the polar day (day length = 24 hours, times at sunrise and sunset undefined) or polar night (day length = 0 hours, sunrise and sunset undefined). If calculations for polar night or day are found, the R function gives a warning message and makes sure that results are correct. This is done in the R function because the C library does not completely handle these exceptions. The AstroCalc4R function in R returns a dataframe (if the number of observations > 1) or a vector (if the number of observations =1). The dataframe will include the input data and results or the results only (depending on the argument withinput, see below).
6. The input data supplied to the AstroCalc4R function in R are:
 - a. **day, month and year:** Day, month and year in the local time zone (integers).
 - b. **hour:** Local time for each observation (decimal hours, e.g. 11:30 PM is 23.5, real numbers). All of these input data must be supplied.
 - c. **timezone:** The difference in hours between local time and Greenwich Mean Time (integers). For example, use -5 for local time EST. All of these input data must be supplied.
 - d. **lat:** Latitude for each observation in decimal degrees (0° to 90° in the northern hemisphere and -90° to 0° degrees in the southern hemisphere, real numbers). For example, 42° 30' N is 42.5° and 42° 30' S is -42.5°. All of these input data must be supplied.
 - e. **lon:** Longitude for each observation (-0° to 180° in the western hemisphere and 0° to 180° degrees in the eastern hemisphere, real numbers). For example, 110° 15' W is -110.25° and 110° 15' E is 110.25°. All of these input data must be supplied.

- f. **withinput**: T or TRUE if you want the output dataframe to include the input data and F or FALSE otherwise (logical or integer 0=FALSE, 1=TRUE). The default value is FALSE if the argument is missing.
 - g. **path**: The location of the AstroCalc4R.dll or AstroCalc4R.so library file on your computer (quoted string using single or double quotes). For example, if you keep the library in \Codes\Diel in Windows, use path="\\Codes\\Diel" or path="/Codes/Diel" (R in Windows does not accept the single "\" character between directory names). In Linux use path="/Codes/Diel". Note that slashes are not required on the end of the path variable. If you do not specify the path, then it will default to the current R working directory. You may set the working directory prior to calling AstroCalc4R using the R function setwd. Use getwd to determine what the current working directory is.
7. The output from AstroCalc4R (all real numbers) consists of the following variables. All time values are in decimal hours, local time.
 - a. **noon**: Time at local noon.
 - b. **sunrise**: Time at sunrise (in the morning when the solar zenith is 90.83°).
 - c. **sunset**: Time at sunset (in the evening when the solar zenith is 90.83°).
 - d. **azimuth**: Solar azimuth angle (decimal degrees).
 - e. **zenith**: Solar zenith angle (decimal degrees).
 - f. **eqtime**: Equation of time (decimal minutes).
 - g. **declin**: Solar declination angle (decimal degrees). **daylength**: Time in hours between sunrise and sunset.
 8. The .C function in R can be used access the dynamic link library or shared object directly but AstroCalc4R.r is the easiest approach. See AstroCalc4R.r code for .C function examples.
 9. Short examples calculations with the R AstroCalc4R function are given on the next page.

```

#Work from the directory with the AstroCalc4R software to avoid
# specifying the location of the software in the argument called path.

> setwd("C:\\New_Papers\\DielRefDoc-1\\AstroCalc4R_C_code\\Version1_0")

#Example with one input and output record. The result is a vector.
> AstroCalc4R(day=12,month=9,year=2000,hour=12,
+           timezone=-5,lat=49.9,lon=-110,withinput=T)
  tzzone day month year hhour xlat xlon  noon sunrise  sunset azimuth  zenith
1    -5  12    9 2000   12 49.9 -110 14.26824 7.874134 20.66235 136.5367 54.24775
  eqtime  declin daylight    PAR
1 3.905575 3.871410 12.78821 261.2365

#Example with > 1 input and output record. The output is a data frame.
> AstroCalc4R(day=c(1,2,3,12),month=c(1,12,2,9),
+           year=c(1900,1950,2000,2050),hour=c(0,7,12,23),
+           timezone=c(-7,-9,9,12),lat=c(3,60,-49.9,0),
+           lon=c(-105,-120,0,100),withinput=F)
      noon sunrise sunset azimuth zenith  eqtime  declin daylight
1 12.059396  6.084082 18.034709 182.3910 159.94132  -3.563757 -23.040114 11.950627
2 12.822631  9.606956 16.038306 103.5928 107.57462  10.642146 -21.932818  6.431349
3  5.230462 -2.243980 12.704904 250.6278  84.58528 -13.827698 -16.552014 14.948884
4 -1.733546 -7.789217  4.322125 288.7017  11.60033  4.012744  3.696747 12.111342
      PAR
1  0.00000
2  0.00000
3 16.78328
4 470.70795
>

```