Python Evapotranspiration Applications

Introduction

The Desert Research Institute (DRI) and the Technical Services Center (TSC) of the Bureau of Reclamation (Reclamation) have developed Python applications for calculation of reference, crop and area evapotranspiration (ET) and irrigation requirements. In addition, DRI developed supporting applications to facilitate calibration of estimated solar radiation, calibration of crop growth parameters, and preparation of soil's data need to for crop ET. The ET applications are conversions of VB.net applications that were based on Visual Basic code that was developed by Rick Allen at the University of Idaho.

The ET and supporting applications are in following folders.

et-demands→refET → bin

et-demands→cropET → bin

et-demands→areaET → bin

et-demands→lib

et-demands→prep

et-demands→tools

et-demands→tools → solar_radiation_opt

Overview

The ET applications use a shared module of data management interfaces (DMI's), 'mod_dmis', which is located in folder 'lib'.

All of the ET applications and some of the supporting applications use a configuration file, also known an initialization file or '.ini' file. Configuration files are read by Python 'ConfigParser' module. Configuration files are text (ascii) file that have 'sections'. An initialization file has at least four sections – project, meta data, input and output. In application, a section is designated using brackets and capital letters. For example:

[PROJECT]

The project section is named 'PROJECT' for all applications and is intended to be common denominator of all the applications. The other sections have names specific to the application. Within each section are attributes specific to the section. An attribute and its value are separated by " = ". For instant:

my_attribute = my_value

An attribute name can not have spaces but an attribute value can have spaces. Comment lines begin with "#" and blank lines are skipped. Attributes and values should not begin or end with single or double quotes unless. An attribute can exist in multiple sections. For instance, the input and output sections all have a file type attribute. Its actual value is specific to the section and can vary between sections.

Sections specific to an application and example attributes are provided in application specific documentation below.

Most sections include file specifications. Time series file name specifications often have a basin abbreviation such as "URR' for Upper Red River but those are not required. Time series file specifications usually include wildcards which are recognized by a "%" character within the file specification. Following are conventions for wildcard specifications:

"%s" is station id

"%p" is parameter id

A configuration file consists of these basic types of data – model control, data environment, file specifications, parameters and meta data. Parameters can data specific to ET application or specific to basin being modeled. File specifications are data that tell application what format of data is and where to find it. Meta data sections include parameters and file specifications specific to ET application. Other meta data consist of attributes about time series data. As a minimum, field names and units of all required parameters are needed. Example fleld and units specification are:

tmax_field = Maximum Temperature

tmax_units = F

Other field related specifications are specific to data structure and file type and are discussed in next section. All internal units are metric and most default output is metric.

Data Structure and File Types

Meta data used for the applications can be specified in a workbook or in a tab or comma delimited text file. Time series data can be specified using two broad data structures:

station files with all parameters, aka as 'SF P' data parameters files with all stations, aka as ‘PF S.P’

The ‘S.P’ in the ‘PF S.P’ abbreviation indicates that columns are station and parameter. Examples of each data structure are provided in the application documentation sections.

Another distinguisher of time series files is the file types. Four files types are used by some or all of the ET applications which are:

csv – File type of ‘SF P’ data structure

xls – Excel workbook

csf – column slot file (text version of workbook format)

rdb – text relational database file

File specification also include information for parsing headers lines and a delimiter or worksheet specification. Following are example collective specifications.

Example SF P specification:

file_type = csv

data_structure_type = SF P

name_format = %s_KL_Met.csv

header_lines = 1

names_line = 1

delimiter = ,

Example csf PF S.P specification:

data_structure_type = PF S.P

file_type = csf

name_format = KLDailyS0%p.csv

header_lines = 1

delimiter = ,

Example rdb PF S.P specification:

data_structure_type = PF S.P

file_type = rdb

name_format = KLDaily%p.rdb

header_lines = 1

delimiter = ,

Example xls PF S.P specification:

file_type = xls

data_structure_type = PF S.P

name_format = KLDailyData.xlsx

header_lines = 1

names_line = 1

Workbook specifications require a worksheet for each parameter that is specified in an input or output section. For example:

tmax_ws = TMax

tmin_ws = TMin

ppt_ws = Prcp

All other file specifications are specific to ET application and are noted in examples below.

'%p' wildcard is irrelevant to 'SF P' data structure type. All PF S.P data structure types can use '%p' wildcard to specify parameter in file name. Because station and parameter always exist in column header or as a column value (rdb file type), all attributes can exist in one text file, one worksheet of a workbook or in multiple worksheets of a workbook. '%p' wildcard resolution defaults to parameter specification (called field name attributes in configuration files). Field name file specifications can also be set in configuration file using field name attributes. Following is an example of field names being different from parameter file name specification:

Field name specifications:

tmax_field = Maximum Temperature

tmin_field = Minimum Temperature

ppt_field = Precipitation

File name specifications:

tmax_name = TMax

tmin_name = TMin

ppt_name = Prcp

To summarize, the '_field' attribute is name of the parameter in the time series file and the '_name' attribute is part of time series file name that corresponds to '_field' attribute. Available and required '_field' attributes are specific to application being configured and discussed in application specific sections.

Data and Application Procedures

Each application is independent of the other. However, the application may share meta data and the ET applications are designed to use output of the reference ET application as input of the crop ET application and its output is intended to be used as input to the area ET application. Typical steps in a study would include:

Develop historic daily and average monthly data

Calibrate solar radiation Thorton and Running coefficients

Compute reference ET which includes data filling

Calibrate crop growth parameters (an iteration of crop ET computations and calibration applications)

Compute area ET data

These steps are described in additional detail below.

Data Development

A minimum of three time series are required to compute ET and irrigation requirements using these applications – daily maximum temperature, daily minimum temperature and precipitation. If an incomplete temperature record exists, average monthly temperatures can be provided to the reference ET application that are used to fill the record. A complete precipitation record is needed to compute irrigation requirements.

Computation of reference ET also requires dew point temperature, wind travel, and solar radiation. If one or more of these parameters are missing, reference ET application includes ability to:

use average monthly wind travel in lieu of daily wind travel estimate dew point temperature from average monthly dew point depression (Ko) and minimum temperature

estimate solar radiation as a function of air temperature

DRI created some geographic information system (GIS) based applications to develop soil parameters and cropping data. These applications use publicly available GIS coverages. Files 'readme.txt' and 'preps.txt' in folder 'prep' provide additional information.

Meta Data

Meta data consist of non-time series data[1] provided to applications through the configuration files and through tabular data in workbooks or text files. A good deal of meta data in configuration files are mappings from data store's names and units to application's names and units. These are discussed in additional detail in application sections.

Tabular meta data examples are provided with installation in both workbook form and text form and consist of following types of data:

- Met Nodes
- ET Cells
- Soil Parameters
- Crop Daa
    - Crops types by ET Cell
    - Parameter data by crop type
    - Coefficients by crop curve type
    - Mean cuttings by dairy and beef alfalfa
    - Mixtures as areas or percentages as a repeating time series or annual time series

Crop mixtures (aka cropping patterns) table also include ability to provide user limits on crop and ET Cell irrigation seasons[2].

[1] Cropping patterns can be expressed as a repeated series or as an annual time series.

[2] Crop and ET Cell limits on irrigation season are provided to capture cultural practices and to provide flexibility in setting up irrigation requirements as input for subsequent modeling.

Supporting Time Series Data

Some historic daily solar radiation is needed to calibrate the Thorton and Running coefficients used when solar radiation is estimated. One historic station within a study area is usually sufficient. These data should be in same format of other meteorological data. Solar calibration application is discussed in a later section.

Even if all daily maximum and minimum temperature data are available, the reference ET application requires that average monthly tables be provided. These tables need to be in C.

If dew point temperature data are not available for all stations, it can be estimated from average monthly dew point depression (Ko) and minimum temperature as:

Dew point temperature = Minimum Temperature – Ko

Average monthly Ko in C needs to be developed at least one station in study area.

Average monthly wind travel in meters per second (mps) needs to be developed from at least one station in study area to support using estimated wind travel for other stations.

An example average monthly table is shown on Figure 1.

Wind Stations (Long-term monthly averages)													
Met ID	Met Name	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1001	Yuma	1.959677	1.828571	1.935484	2.21	2.576744	2.26	3.267742	2.787097	2.11	2.316129	2.081395	1.824194
1002	Umatilla	1.596712	1.736722	2.269911	2.373377	2.387891	2.479535	2.336995	2.231843	1.969385	1.906257	1.664363	1.51001
1003	Klamath	1.489807	1.704516	1.903578	2.102404	1.970076	1.830902	1.439877	1.386425	1.36115	1.366308	1.388767	1.541098
1004	Milk River	2.317704	2.322407	2.910053	3.092165	3.169292	2.537009	2.137444	2.18011	2.142683	2.338014	2.246334	2.261894
1005	Middle Rio Grande	1.843914	2.102435	2.257563	2.473569	2.238723	2.011427	1.774032	1.646377	1.625382	1.693415	1.786242	1.711255
1006	Central Valley	1.644473	1.89723	2.087097	2.389231	2.582338	2.606923	2.374194	2.156576	1.905656	1.664824	1.477949	1.617347
1007	Newlands	1.2772	1.6724	2.0153	2.2498	1.8415	1.75	1.4019	1.2508	1.2814	1.0971	1.271	1.5192

Figure 1. Example Average Monthly Data.

Aggregations

Temporal aggregations of daily output to monthly and annual files are available from the ET applications. Because the file names are same for daily, monthly and annual data, timestep dependent data are distinguished by folders as shown in following configuration file examples:

Output flags

daily_refet_flag = True

monthly_refet_flag = True

annual_refet_flag = True

Sub folder names

daily_refet_folder = daily_ret

monthly_refet_folder = monthly_ret

annual_refet_folder = annual_ret

Another option for area ET output as flow is to convert the output to a volume type flow such as 'acre-ft/day', 'acre-ft/month' and 'acre-ft/year'. Following configuration attributes are used to obtain this output:

daily_volume_units = acre-feet   # default is None

monthly_volume_units = acre-feet   # default is None

annual_volume_units = acre-feet   # default is None

For acre-feet output, specify 'cfs' for flow field's Units as:

etflow_units = cfs

irflow_units = cfs

Above example would post daily flow in 'cfs' and monthly and annual flow in 'acre-feet'.

Datetime Precision

Internally, meteorological and evapotranspiration data do not include hours and minutes in the applications. However, Reclamation and other water agencies use of period[3] hour and minutes timestamps. In some cases, this is critical to and software used by them, in particular dates in workbooks. Note that this refers to the precision of the date values, not to the displayed values.

To support these agencies and applications, following optional output date offsets can be specified in reference ET and area ET output sections.

daily_hour_offset = 23

daily_minute_offset = 59

monthly_hour_offset = 23

monthly_minute_offset = 59

annual_hour_offset = 23

annual_minute_offset = 59

[3] End of period dates are also used for reservoir parameters such as pool elevation and storage (aka content).

Reference ET and Data Estimations

Configuration file for Reference ET application and data estimations includes following sections:

[PROJECT]

[RET_META]

[INMET]

[OUTRET]

[OUTMET]

'OUTMET' section is optional but is useful for reviewing and reporting estimated data and for supporting other applications. All other sections are required. Appendix B is content of 'ret_template.ini' example configuration file. To produce optional output meteorological data, specify attributes of 'OUTMET' section and set 'PROJECT' attribute output_met_flag toTrue as:

output_met_flag = True

Copy 'ret_template.ini' or other exiting reference ET configuration file to create an 'ini' file for specific geographic application. Edit "ini' files for application specific items. In particular, edit "basin_id", "project_folder", "start_date", "end_date", meta data file specifications and time series file specifications. In addition, if estimating solar radiation, edit Thornton and Running coefficients as:

TR_b0 = 0.024115

TR_b1 = 0.176397

TR_b2 = -0.224488

Format of 'OUTRET' output is standardized to a 'SF P' data structure but user can specify actual parameters that are posted. Following is example 'OUTRET' output with all most available items posted.

Date,TMax,TMin,Precip,Snow,SDep,EstRs,EsWind,EsTDew,Penm48,PreTay,ASCEr,ASCEg,85Harg

Units,C,C,In*100,In*100,In,MJ/m2,m/s,C,mm/day,mm/day,mm/day,mm/day,mm/day

To use as input the crop ET application, it should include all meteorological data not processed through 'OUTDATA' and at least one of the reference ET methods.

'OUTMET' output can be any of the supported data types shown in Appendix A and include input meteorological data, estimated meteorological data and selected reference ET data. See 'ret_template.ini' and Appendix B for examples.

After setting up an initialization file, edit “bin_ws” line in driver script ‘run_ret.py’ to reflect local installation:

```
bin_ws = r'C:\PythonApps\bortsc\retET\bin'
```

To run application, type:

```
Python run_ret.py -i my_ret.ini
```

‘run_ret’ can also be run with various command switches as shown in these examples:

```
# run ret for one met node with debugging where –m is met node switch

Python run_ret.py -i my_ret.ini -d -m CA9053

# run ret for with multiprocessing and all available processors

Python run_ret.py -i my_ret.ini –mp

# run ret for with multiprocessing and specified number of processors

Python run_ret.py -i my_ret.ini –mp 4
```

Multiprocessing is not available in debug mode, single node, or ‘PF S.P’ output formats.

Arguments for run_ret.py script are:

-h, --help : Show the help message then exit

-i PATH, --ini PATH: Initialization file path; If omitted, script will prompt

-d, --debug : Save debug level comments to debug.txt file

-m MetID, --metid MetID : Specify a particular single Met Node ID to run.

-v, --verbose : Display information level comments

-mp num, --multiprocessing num

```
Date,TMax,TMin,Precip,Snow,SDep,EstRs,EsWind,EsTDew,Penm48,PreTay,ASCEr,ASCEg,85Harg

Units,C,C,In*100,In*100,In,MJ/m2,m/s,C,mm/day,mm/day,mm/day,mm/day,mm/day

…
```

Crop ET and Irrigation Requirement

Configuration file for Crop ET application includes following sections[4]:

[PROJECT]

[CET_META]

[CROP_ET]

[REFET]

[WEATHER]

[HISTTEMPS]

'HISTTEMPS' section is optional and is used as alternative of maximum and minimum temperature when using the constant phenology option. All other sections are required. Appendix C is content of 'cet_template.ini' example configuration file.

Copy 'cet_template.ini' or other exiting crop ET configuration file to create an 'ini' file for specific geographic application. Edit "ini" files for application specific items. In particular, edit "basin_id", "project_folder", "start_date", "end_date", meta data file specifications and time series file specifications.

Format of 'REFET' section data is same as format of reference ET 'OUTRET' output ('SF P'). However, because reference ET and crop ET applications are independent, actual content 'REFET' data can be different and be from another reference application. 'WEATHER' section format can be any of the supported time series formats. Because 'WEATHER' data are available in same files as 'REFET' data, 'WEATHER' section's file specifications will typically be same as 'REFET' section's. 'HISTTEMPS' data can be any of the supported time series formats.

'CROP_ET' output can be one of two formats. The 'DRI' format is a 'SF P' data structure with data of all crops in a station file. The 'RDB' format has data of all crops in one station file as a text relational database file. The 'DRI' format is required for crop parameter calibration application but the 'RDB' format is more convenient for other analyzes. Either format can be used for the area et module.

When using 'DRI' format, 'name_format' specification should include a '%c' wildcard item which is resolved to the crop type number in file name. For example:

name_format = '%s_crop_%c.csv

[4] Original implementation of crop et configuration file had PROJECT, CET_META and CROP_ET combined. Sections were broken out to better support associated applications and scenario management. Original configuration file structure is still supported.

Crop growing output is optionally available using the “gs_output_folder” specification of “CROP_ET” section. If “gs_output_folder” is True, the “gs_name_format” specification can be used to assign a file name other than the default which is a function of crop’s “name_format” specification.

After setting up an initialization file, edit “bin_ws” line in driver script ‘run_cet.py’ to reflect local installation:

bin_ws = r'C:\PythonApps\bortsc\cetET\bin'

To run application, type:

Python run_cet.py -i my_cet.ini

‘run_cet’ can also be run with various command switches as shown in these examples:

run cet for one et cell with debugging where –c is et cell id switch

Python run_cet.py -i my_cet.ini -d -c Klamath_5

run cet for with multiprocessing and all available processors

Python run_cet.py -i my_cet.ini –mp

run cet for with multiprocessing and specified number of processors

Python run_cet.py -i my_cet.ini –mp 4

Multiprocessing is not available in debug mode or ‘PF S.P’ output formats. Multiprocessing can be used to compute all crops of a single et cell.

Arguments for run_cet.py script are:

-h, --help : Show the help message then exit

-i PATH, --ini PATH: Initialization file path; If omitted, script will prompt

-vb, --VB : Mimic calculations in VB code

-d, --debug : Save debug level comments to debug.txt file

-c CellID, --metid CellID : Specify a particular single ET Cell Id to run

-v, --verbose : Display information level comments

-mp num, --multiprocessing num

“DRI” format post data to files by ET Cell and Crop. Typical “DRI” data are:

3 - Alfalfa Hay

Date,Year,Month,Day,DOY,PMETo,ETact,ETpot,ETbas,Kc,Kcb,PPT,Irrigation,Runoff,DPerc,NIWR,Season

Hay crops can include a “Cuttings” column. ‘RDB’ format post all data in same format with ‘Crop Num’ and ‘Crop’ as additional columns before the ‘Date’ column to a single file for an ET Cell.

Area ET and Irrigation Requirement

Configuration file for Area ET application includes following sections:

[PROJECT]

[AET_META]

[INCET]

[OUTAET]

[OUTCET]

[OUTCIR]

'PROJECT', 'AET_META', and 'INCET' sections are required. The 'OUTAET' section provides for posting of several area weighted parameters. The 'OUTCET' and 'OUTCIR' sections provide for posting of crop by crop ET and irrigation requirement in 'SF P' or 'PF S.P' formats. Appendix D is content of 'aet_template.ini' example configuration file.

Copy 'aet_template.ini' or other exiting reference ET configuration file to create an 'ini' file for specific geographic application. Edit "ini' files for application specific items. In particular, edit "basin_id", "project_folder", "start_date", "end_date", meta data file specifications and time series file specifications.

Either 'DRI' or 'RDB' crop ET format can be used for the area et 'INCET' data. All output is optional and can be run individually or collectively. All output can use any of the supported time series formats. However, all crop ET (CET) and irrigation requirements (CIR) data will be in one file whereas 'OUTAET' output can be in parameter by parameter or collective files.

After setting up an initialization file, edit "bin_ws" line in driver script 'run_aet.py' to reflect local installation:

```
bin_ws = r'C:\PythonApps\bortsc\aetET\bin'
```

To run application, type:

```
Python run_aet.py -i my_aet.ini
```

'run_aet' can also be run with various command switches as shown in these examples:

run aet for one et cell with debugging where –c is et cell id switch

```
Python run_aet.py -i my_aet.ini -d –c Klamath_5
```

run aet for with multiprocessing and all available processors

Python run_aet.py -i my_aet.ini –mp

run aet for with multiprocessing and specified number of processors

Python run_aet.py -i my_aet.ini –mp 4

Multiprocessing is not available in debug mode, single node, or 'PF S.P' output formats.

Arguments for run_aet.py script are:

-h, --help : Show the help message then exit

-i PATH, --ini PATH: Initialization file path; If omitted, script will prompt

-d, --debug : Save debug level comments to debug.txt file

-c CellID, --metid CellID : Specify a particular single ET Cell to run

-v, --verbose : Display information level comments

-mp num, --multiprocessing num

Solar Calibration

The solar calibration application can specify all necessary data through command line arguments and defaults or use an initialization file. An example of its initialization file is shown in Appendix D. Main program is 'solar_radation_opt.py' which is run with one or more command line arguments.

If file name specified with '—file' command name argument is file that does not end in '.ini', file is used as source of time series data and data headers must be consistent with example workbook 'RecordedMetData.xlsx'. Additional arguments must be provided with this approach. These are:

"--elev " – elevation in meters

"--lat " – latitude

" --lon " – longitude

"-mc" – number of Monte Carlo iterations

"-sd " – worksheet name or text file delimiter

If filename specified with '—file' ends with '.ini', it is assumed to be a configuration file. The '-mc' argument still needs to be provided. Example wrapper script 'run_solar_rad_opt.py' is provided.

Instructions on using application are in 'UserGuideForMonteCarloAnalysis.docx'.

Crop Parameter Calibration

The crop parameter calibration applications can share the crop ET configuration file. Wrapper script 'run_cgs.py' can be used to run 'compute_growing_season.py'. Wrapper script 'run_pdcd.py' can be used to run 'plot_crop_daily_timeseries'. Examples of each are:

crop growing seasons

Python run_cgs.py -i csv_cet_dri.ini

crop et data plots with file saving and without showing of plots

Python run_pdcd.py -i csv_cet_dri.ini

crop et data plots with file saving and with showing of plots

Python run_pdcd.py -i csv_cet_dri.ini --show

crop et data plots without plot file saving and without showing of plots

Python run_pdcd.py -i csv_cet_dri.ini --no_save –show

Do not use last setup for generating daily plots unless output files were previously generated. Data computations for the plots takes significant computation time and disk space. It is recommended that you a subset of data be used when large numbers of crops and ET Cells exist as explained in following paragraphs.

Crop parameter calibration is an iterative process involving multiple runs of the ET model to set crop start and stop dates, and adjust the crop K_c curve to match known examples. Calibration parameters can be adjusted by the user using the entries in the "CropParams" tab of the meta data workbook. Specific rows are used to effect the calibration, discussed below. An understanding of the growing practices of the area is required for proper calibration, specifically typical planting (or greenup) dates and harvest (senescence) dates are needed. Data specific to the area is best, but general start and stop dates are also available from the U.S. Department of Agriculture (USDA). The most recent nation-wide publication is entitled ***Field Crops: Usual Planting and harvesting Dates, October 2010*** and is (as of this writing) available at: (https://www.nass.usda.gov/Publications/Todays_Reports/reports/fcdate10.pdf).

Once start and stop dates are secured for the area of interest, follow this iterative process.

1) Calibrate one crop at a time. A single crop can be specified by setting only one crop in the "ETCellsCrops" tab of the Meta data workbook. In this table, a "1" indicates crop simulation, and a "0" indicates no simulation for that crop. Remember to always simulate the bare soil types (crop types 44, 45 and 46) to properly account for off season soil moisture conditions which can and do effect crop growth.

2) For a single crop, execute the crop ET model for a single ET Cell. Choose a representative single ET Cell that will act as the calibration cell. All other cells in the simulated area will use this crop calibration. If the area of interest is very large, consider dividing the area into smaller, more homogeneous areas.
3) Execute Python run_cgs.py -i csv_cet_dri.ini. This results in the generation of two files; one showing the start and stop dates for each year of execution, the second showing the average start and stop date for the entire run. Open the annual average file, titled "growing_season_mean_annual.csv" and look at the "MEAN_START_DATE" and "MEAN_END_DATE" fields.
4) Adjust the parameters for start and stop in the "CropParams" tab of the Meta data workbook. The parameters to adjust are explained here.
    a) Row 21 specifies the method used to estimate planting/greenup: 1=Growing Degree Days (GDD), 2= 30-day average temperature (T30), 3=User specified date, 4=Always on
    b) Row 22 specifies the number of GDD or the T30 temperature. If the type displayed in row 21 is 3, enter the start month in row 23. If the MEAN_START_DATE (from step 3 above) is too early, increase the value in row 22.
    c) Row 26 specifies the GDD required for termination. Note that this is the number of GDD after planting/greenup. If the MEAN_END_DATE (from step 3 above) is too early, increase the value in row 26.
        1) Additionally, if the user specifies, an absolute maximum growing season length can be entered in row 29.
        2) For crops that grow until a killing frost, row 30 is used. Enter the temperature required to end the crop growing season.

Once the start and stop dates are achieved, it is time to adjust the crop k_c curve to match example curves. This calibrates the planting/greenup to Effective Full Cover (EFC)/maturity portion of the curve. Once again, this is an iterative process that relies on the judgement of the user to accomplish. Example curves are available online or can be secured from the Reclamation report titled ***West-Wide Climate Risk Assessment: Irrigation Demand and Reservoir Evaporation Projections***. The k_c curves are displayed in Appendix 5, available (as of this writing) at:

http://www.usbr.gov/watersmart/wcra/docs/irrigationdemand/Appendices.pdf

To adjust the curve, use the Python run_pdcd.py -i csv_cet_dri.ini –show command. This processes the crop ET model output and displays the various water use curves in your browser. Comparing the resulting k_c curves to the display curves in Appendix 5 allows the user to change maturity times for the crops, thus dialing in the calibration. Steps for doing this are below:

5) Compare the resulting k_c curve to a known curve for the crop of interest. Look specifically at the time from planting/greenup to full maturity (the top of the curve) relative to the overall length of the life cycle.

6) Adjust the values in row 25 (for GDD and T30 based crops or row 28 (for date based crops) of the “CropParams” tab of the Meta data workbook to adjust the curve peak to better match the example crop curves.

Once a curve for the crop looks good (on average, no two years growing conditions will be exactly the same), choose a different crop on the “ETCellsCrops” tab and repeat for all crops to be simulated.

After each adjustment of the “CropParams” data, re-run crop ET model to get updated simulation results. Unless meteorology is changed, the Reference ET model does not need to be re-run between calibration steps. Typically, start and stop dates are calibrated first (steps 2-4 above), re-executing the ET model between each adjustment of the “CropParams” data. Once the start and stop times are calibrated, move on to the crop curve calibration, executing steps 2, 5 and 6, again re-running the ET model between each parameter adjustment.

Appendix A – Example Time Series Data Files

Table 1. Example SF P Meteorological Data File.

Date,TMax,TMin,Prcp

1950-01-01,31.2093540,12.9753548,0.0455199

1950-01-02,23.8473538,0.7353544,0.0113800

1950-01-03,20.6613539,-11.7206457,0.0257946

1950-01-04,29.1753539,3.4533548,0.0000000

1950-01-05,21.1113539,-8.0486475,0.0000000

1950-01-06,32.2353540,3.5613539,0.0000000

1950-01-07,32.3073540,21.2373542,0.0568998

1950-01-08,32.7213539,16.5933535,0.1433876

1950-01-09,31.0833540,13.1733541,0.0037933

Table 2. Example Column Slot File.

Date,OR1571.TMax,OR8007.TMax,OR1574.TMax

1/1/1950,29.0193016,26.9875193,29.1497082

1/2/1950,28.7133015,22.7395191,26.8637082

1/3/1950,27.7593015,21.2275188,26.8457082

1/4/1950,37.4793017,30.9295192,34.8917083

1/5/1950,36.4533016,26.6095193,35.1977082

1/6/1950,35.7513016,33.1435192,36.4577081

1/7/1950,37.6593015,33.5395191,36.6377083

1/8/1950,37.6593015,33.8815191,35.6297082

1/9/1950,30.2073016,30.1015191,32.5337082

Table 3. Example Text RDB File.

Station	Parameter	Date	Value
OR1571	Maximum Temperature	1950-01-01	29.02
OR1571	Maximum Temperature	1950-01-02	28.71
OR1571	Maximum Temperature	1950-01-03	27.76
OR1571	Maximum Temperature	1950-01-04	37.48
OR1571	Maximum Temperature	1950-01-05	36.45
OR1571	Maximum Temperature	1950-01-06	35.75
OR1571	Maximum Temperature	1950-01-07	37.66
OR1571	Maximum Temperature	1950-01-08	37.66
OR1571	Maximum Temperature	1950-01-09	30.21

Table 4 . Example Workbook Worksheet.

Date	OR1571.Maximum Temperature	OR8007.Maximum Temperature	OR1574.Maximum Temperature
1/1/1950	29.0193016	26.9875193	29.1497082
1/2/1950	28.7133015	22.7395191	26.8637082
1/3/1950	27.7593015	21.2275188	26.8457082
1/4/1950	37.4793017	30.9295192	34.8917083
1/5/1950	36.4533016	26.6095193	35.1977082
1/6/1950	35.7513016	33.1435192	36.4577081
1/7/1950	37.6593015	33.5395191	36.6377083
1/8/1950	37.6593015	33.8815191	35.6297082
1/9/1950	30.2073016	30.1015191	32.5337082

Appendix B. Reference ET Configuration File Example

```
[PROJECT]

basin_id = Klamath

project_folder = C:\PythonApps\bortsc\data

# Limit to a date range (ISO Format: YYYY-MM-DD)

# start_date = None

start_date = 1950-01-01

# end_date = None

end_date = 1999-12-31

time_step = day

# ts_quantity = 1    # default

# Plots sub-folder

daily_plots_folder = daily_plots

# Optional output met data flag

output_met_flag = True

# Average monthly output met data flag (default is False)

# avg_monthly_met_flag = True

[RET_META]

# static (meta) data folder

static_folder = static

# Met node meta data file name and setup

mnmd_header_lines = 1

mnmd_names_line = 1

# text setup

# mnmd_delimiter = ,
```

met_nodes_meta_data_name = MetNodesMetaData.csv

mnmd_delimiter = \t

met_nodes_meta_data_name = MetNodesMetaData.txt

workbook setup

met_nodes_meta_data_name = MetAndDepletionNodes.xlsx

met_nodes_meta_data_ws = MetNodesMetaData

Miscellaneous

elev_units = Feet

Met data configuration

[INMET]

Weather data (Tmax, Tmin, PPT, etc.)

input_met_folder = daily_in_met

data structure type, file type (xls, csv, cdf, rdb) and file name format

station file format

file_type = csv

data_structure_type = SF P

name_format = %s_KL_S0_Met.csv

Parameter formats

data_structure_type = PF S.P

individual parameter files example - %p is wildcard for "_name"'s

name_format = KLDaily%pS0Data.xlsx

shared parameter files example

name_format = KLDailyS0Data.xlsx

additional file specs - used if %p in name_format for parameter formats

tmax_name = TMax

tmin_name = TMin

ppt_name = Prcp

csf setup

file_type = csf

name_format = KLDailyS0Data.csv

rdb setup

file_type = rdb

name_format = KLDailyS0Data.rdb

workbook setup

file_type = xls

name_format = KLDailyS0Data.xlsx

worksheet names

tmax_ws = TMax

tmin_ws = TMin

ppt_ws = Prcp

1's based indices

header_lines = 1

names_line = 1

delimiter = \t

delimiter = ,

Field names

date_field = Date

tmax_field = TMax

tmin_field = TMin

ppt_field = Prcp

rs_field = Incident Solar Radiation

wind_field = Wind Travel

tdew_field = Dew Point Temperature

q_field = Q

snow_field = Snow

depth_field = Snow Depth

Units

tmax_units = F

tmin_units = F

ppt_units = inches/day

rs_units = MJ/m2

wind_units = m/s

tdew_units = F

#q_units = kg/kg

snow_units = inches/day

depth_units = inches

Wind gage height in meters

wind_height = 2

Thorton and Running coefficients for solar radiation filling

TR_b0 = 0.0307075712855

TR_b1 = 0.1960418743712

TR_b2 = -0.2454592897026

Data filling support files, etc

All optional if all time series data of parameter exists

File should exist in static data folder

```
# csv text file setup
# avgm_tmax_name = TMaxMon.csv
# avgm_tmax_delimiter = ,
# avgm_tmin_name = TMinMon.csv
# avgm_tmin_delimiter = ,
# avgm_Ko_name = KoMon.csv
# avgm_Ko_delimiter = ,
# avgm_wind_name = WindMon.csv
# avgm_wind_delimiter = ,
# avgm_wind_header_lines = 2
# workbook setup
avgm_tmax_name = MetAndDepletionNodes.xlsx
avgm_tmax_ws = TMaxMon
avgm_tmin_name = MetAndDepletionNodes.xlsx
avgm_tmin_ws = TMinMon
avgm_Ko_name = MetAndDepletionNodes.xlsx
avgm_Ko_ws = KoMon
avgm_Ko_header_lines = 2
avgm_wind_name = MetAndDepletionNodes.xlsx
avgm_wind_ws = WindMon
avgm_wind_header_lines = 2
[OUTRET]
refet_units = mm/day
# Output flags
daily_refet_flag = True
```

```
# monthly_refet_flag = False
monthly_refet_flag = True
# annual_refet_flag = False
annual_refet_flag = True
# Sub folder names
daily_refet_folder = daily_ret
monthly_refet_folder = monthly_ret
annual_refet_folder = annual_ret
# ret_method is only used if ret_field is specified (defaults to ASCEg)
# etr_method is only used if etr_field is specified (defaults to ASCEr)
# eto_method is only used if eto_field is specified (defaults to ASCEg)
ret_method = ASCEg
etr_method = ASCEr
eto_method = ASCEg
# data structure type, file type (xls, tab, csv, rdb) and file name format
file_type = csv
data_structure_type = SF P
name_format = %s_KL_S0_RET.csv
header_lines = 2
# 1's based indices
names_line = 1
# delimiter = \t
delimiter = ,
# daily_date_format = %d-%m-%Y
# daily_float_format = %10.6f
```

```
date_field = Date

# year_field = Year

# month_field = Month

# day_field = Day

# doy_field = DOY

# Met data (TMax, TMin, Prcp, Rs, Wind, TDew, Snow, SDep) that is posted with reference ET

# Met field names

tmax_field = TMax

tmin_field = TMin

ppt_field = Precip

rs_field = EstRs

wind_field = EsWind

tdew_field = EsTDew

snow_field = Snow

depth_field = SDep

# Met Units

# tmax_units = F

# tmin_units = F

# tdew_units = F

tmax_units = C

tmin_units = C

tdew_units = C

ppt_units = In*100

rs_units = MJ/m2

wind_units = m/s
```

snow_units = In*100

depth_units = In

Ref ET output fields (units are common to all)

enter value for ret_field to get column with that name. ret_method is used as posted values

enter value for etr_field to get column with that name. etr_method is used as posted values

enter value for etr_field to get column with that name. etr_method is used as posted values

ret_field = None

ret_field = RefET

etr_field = None

etr_field = Alfalfa RET

eto_field = None

eto_field = Grass RET

penm_field = Penm48

pretay_field = PreTay

ascer_field = ASCEr

asceg_field = ASCEg

harg_field = 85Harg

[OUTMET]

Weather data (Tmax, Tmin, PPT, etc.)

daily_output_met_flag = True

monthly_output_met_flag = False

monthly_output_met_flag = True

annual_output_met_flag = False

annual_output_met_flag = True

Sub folder names

daily_output_met_folder = daily_out_met

monthly_output_met_folder = monthly_out_met

annual_output_met_folder = annual_out_met

data structure type, file type (xls, tab, csv, rdb) and file name format

station file format

file_type = csv

data_structure_type = SF P

name_format = %sFilledMetData.csv

header_lines = 2

Parameter formats

data_structure_type = PF S.P

header_lines = 1

individual parameter files example - %p is wildcard for "_name"'s

name_format = KLDaily%pS0Data.xlsx

shared parameter files example

name_format = KLDailyS0Data.xlsx

additional file specs - used if %p in name_format for parameter formats

tmax_name = TMax

tmin_name = TMin

ppt_name = Prcp

csf setup

file_type = csf

name_format = KLDaily%pS0Data.csv

rdb setup

```
# file_type = rdb

# name_format = KLDaily%pS0Data.rdb

# workbook setup

# file_type = xls

# name_format = KLDaily%pS0Data.xlsx

# worksheet names

# tmax_ws = TMax

# tmin_ws = TMin

# ppt_ws = Prcp

# 1's based indices

names_line = 1

# delimiter = \t

delimiter = ,

# date formats

# daily_date_format = %d-%m-%Y

# daily_float_format = %10.6f

# text ISO data formats

# daily_date_format = %Y-%m-%d

# monthly_date_format = %Y-%m

# annual_date_format = %Y-%m

# Excel ISO data formats

# daily_date_format = yyyy-mm-dd

# monthly_date_format = yyyy-mm

# annual_date_format = yyyy-mm

# Excel RiverWare data formats
```

daily_date_format = m/d/yyyy

monthly_date_format = m/yyyy

annual_date_format = m/yyyy

Field names

date_field = Date

year_field = Year

month_field = Month

day_field = Day

doy_field = DOY

tmax_field = TMax

tmin_field = TMin

tavg_field = TAvg

ppt_field = Precip

rs_field = EstRs

wind_field = EsWind

tdew_field = EsTDew

snow_field = Snow

depth_field = SDep

refet_field = RefET

Units

tmax_units = F

tmin_units = F

tavg_units = F

ppt_units = inches/day

rs_units = MJ/m2

wind_units = m/s

tdew_units = F

snow_units = inches/day

depth_units = inches

refet_units = inches/day

Appendix C. Crop ET Configuration File Example

[PROJECT]

basin_id = Klamath

project_folder = C:\PythonApps\bortsc\data

Limit to a date range (ISO Format: YYYY-MM-DD)

Date are picked up time series file if unspecified

start_date = None

start_date = 1950-01-01

end_date = None

end_date = 1999-12-31

Historic (constant) phenology options

This option is used to support climate change meteorology scenarios

because many crop curves (especially annual crops) are temperture based.

Using historic maximum and minimum temperatures produces more realistic

crop curve evolution. If a non zero option is specified,

[HISTTEMPS] must exist with specifications for historic

maximum and minimum temperature data.

Options are:

0 - none (default)

1 - annual crops only

2 - perennial crops only

3 - both annual and perennial crops

phenology_option = 0

[CET_META]

static (meta) data folder

```
static_folder = static
# ET cells properties file name and setup
cell_properties_header_lines = 1
cell_properties_names_line = 1
# text setup
# cell_properties_delimiter = ,
# cell_properties_name = ETCellsProperties.txt
# cell_properties_delimiter = \t
# cell_properties_name = ETCellsProperties.txt
# cell_properties_delimiter = \t
# workbook setup
# cell_properties_name = MetAndDepletionNodes.xlsx
# cell_properties_ws = ETCellsProperties
cell_properties_name = MetAndDepletionNodes.xlsx
cell_properties_ws = ETCellsProperties
# ET cells cuttings file name and setup
cell_cuttings_header_lines = 2
cell_cuttings_names_line = 2
# text setup
# cell_cuttings_delimiter = ,
# cell_cuttings_name = MeanCuttings.txt
# cell_cuttings_delimiter = \t
# workbook setup
# cell_cuttings_name = MetAndDepletionNodes.xlsx
# cell_cuttings_ws = MeanCuttings
```

```
cell_cuttings_name = MetAndDepletionNodes.xlsx

cell_cuttings_ws = MeanCuttings

# ET cells crops file name and setup

cell_crops_header_lines = 3

cell_crops_names_line = 2

# text setup

# cell_crops_delimiter = ,

# cell_crops_name = ETCellsCrops.txt

# cell_crops_delimiter = \t

# workbook setup

# cell_crops_name = MetAndDepletionNodes.xlsx

# cell_crops_ws = ETCellsCrops

cell_crops_name = MetAndDepletionNodes.xlsx

cell_crops_ws = ETCellsCrops

# crop parameters

crop_params_header_lines = 4

crop_params_names_line = 3

# text setup

# crop_params_name = CropParams.txt

# crop_params_delimiter = \t

# workbook setup

# crop_params_name = MetAndDepletionNodes.xlsx

# crop_params_ws = CropParams

crop_params_name = MetAndDepletionNodes.xlsx

crop_params_ws = CropParams
```

crop coefficients

crop_coefs_header_lines = 6

crop_coefs_names_line = 4

text setup

crop_coefs_name = CropCoefs.txt

crop_coefs_delimiter = \t

workbook setup

crop_coefs_name = MetAndDepletionNodes.xlsx

crop_coefs_ws = CropCoefs

crop_coefs_name = MetAndDepletionNodes.xlsx

crop_coefs_ws = CropCoefs

Miscellaneous

elev_units = Feet

[CROP_ET]

ET Demands folder

crop_et_folder = C:\PythonApps\bortsc\data\daily_cet

#crop_test_list = 3

#crop_skip_list = 55, 56, 57

DRI

data_structure_type = DRI

name_format = %s_crop_%c_S0.csv

RDB

data_structure_type = RDB

name_format = %s_crop_S0.csv

data_structure_type = DRI

```
name_format = %s_crop_%c_S0.csv

header_lines = 1

names_line = 1

delimiter = ,

# date and float formats

# daily_date_format = %d-%m-%Y

# daily_float_format = %10.6f

# monthly_date_format = %m-%Y

# monthly_float_format = %8.4f

# annual_date_format = %Y

# annual_float_format = %9.4f

# output flags

daily_stats_flag = True

# monthly_stats_flag = False

monthly_stats_flag = True

# annual_stats_flag = False

annual_stats_flag = True

# growing_season_stats_flag = False

growing_season_stats_flag = True

# Computation switches

# crop one flag

# False sets crop 1 to alfalfa peak with no cuttings

# True sets crop 1 to nonpristine alfalfa w/cuttings

crop_one_flag = True

# crop one (alfalfa) reduction factor
```

crop_one_reducer = 0.9

Output alfalfa cuttings

cutting_flag = True

Output net-irrigation water requirement (NIWR)

niwr_flag = True

Output crop coefficient (Kc)

kc_flag = True

CO2 correction

co2_flag = False

co2_grass_list = 1-6,9-18, 21-67, 69, 71-73, 75, 79-81, 83-85

co2_trees_list = 19, 20, 70, 74, 82

co2_c4_list = 7, 8, 68, 76-78

cet output folder names

daily_output_folder = daily_cet

monthly_output_folder = monthly_cet

annual_output_folder = annual_cet

gs_output_folder = growing_season_cet

growing season name format

dafault is a function of cet output name

gs_name_format = %s_gs_crop_%c.csv

Plots sub-folder names

daily_plots_folder = daily_plots

[REFET]

RefET folder (ETo or ETr)

refet_folder = daily_ret

refet_type = ETo

station file format

default 'file_type' is 'csv'

default 'data_structure_type' is 'SF P'

name_format = %s_KL_S0_RET.csv

header_lines = 2

name_format = %s_KL_S0_RET.csv

header_lines = 2

Parameter formats

data_structure_type = PF S.P

individual parameter files example - %p is wildcard for "_name"'s

name_format = KLDaily%pS0RET.xlsx

shared parameter files example

name_format = KLDailyS0_RET.xlsx

header_lines = 1

additional file specs - used if %p in name_format for parameter formats

refet_name = RefET

csf setup

file_type = csf

name_format = KLDailyS0_RET.csv

rdb setup

file_type = rdb

name_format = KLDailyS0_RET.rdb

workbook setup

file_type = xls

```
# name_format = KLDailyS0_RET.xlsx

# worksheet names

# tmax_ws = RefET

# 1's based indices

names_line = 1

# delimiter = \t

delimiter = ,

# Field names and units

date_field = Date

etref_field = ASCEg

etref_units = mm/day

[WEATHER]

# Weather data (Tmin, Tmax, PPT, etc.)

weather_folder = daily_ret

# station file format

# default 'file_type' is 'csv'

# default 'data_structure_type' is 'SF P'

# name_format = %s_KL_S0_RET.csv

name_format = %s_KL_S0_RET.csv

header_lines = 2

# Parameter formats

# data_structure_type = PF S.P

# individual parameter files example - %p is wildcard for "_name"'s

# name_format = KLDaily%pS0RET.xlsx

# shared parameter files example
```

name_format = KLDailyS0_RET.xlsx

header_lines = 1

additional file specs - used if %p in name_format for parameter formats

tmax_name = TMax

tmin_name = TMin

ppt_name = Prcp

csf setup

file_type = csf

name_format = KLDailyS0_RET.csv

rdb setup

file_type = rdb

name_format = KLDailyS0_RET.rdb

workbook setup

file_type = xls

name_format = KLDailyS0_RET.xlsx

worksheet names

tmax_ws = TMax

tmin_ws = TMin

ppt_ws = Prcp

1's based indices

names_line = 1

delimiter = \t

delimiter = ,

Parameter field names

date_field = Date

tmin_field = TMin

tmax_field = TMax

ppt_field = Precip

#snow_field = Snow

#depth_field = SDep

wind_field = EsWind

tdew_field = EsTDew

#q_field = Q

Parameter units

tmin_units = C

tmax_units = C

ppt_units = In*100

#snow_units = In*100

#depth_units = In

wind_units = m/s

tdew_units = C

#q_units = kg/kg

Wind height in meters

wind_height = 2

[HISTTEMPS]

historic temperature data (Tmin and, Tmax) for historic phenology computions

hist_temps_folder = daily_ret

hist_temps_folder = daily_out_met

station file format

default 'file_type' is 'csv'

```
# default 'data_structure_type' is 'SF P'

file_type = csv

data_structure_type = SF P

name_format = %sS0MetData.csv

header_lines = 2

# Parameter formats

# data_structure_type = PF S.P

# individual parameter files example - %p is wildcard for "_name"'s

# name_format = KLDaily%pS0Data.xlsx

# shared parameter files example

# name_format = KLDailyS0Data.xlsx

# header_lines = 1

# additional file specs - used if %p in name_format for parameter formats

tmax_name = TMax

tmin_name = TMin

# csf setup

# file_type = csf

# name_format = KLDailyS0Data.csv

# rdb setup

# file_type = rdb

# name_format = KLDailyS0Data.rdb

# workbook setup

# file_type = xls

# name_format = KLDailyS0Data.xlsx

# worksheet names
```

```
# tmax_ws = TMax

# tmin_ws = TMin


# 1's based indices

names_line = 1

# delimiter = \t

delimiter = ,

# Field names

date_field = Date

mint_field = TMin

maxt_field = TMax

mint_units = C

maxt_units = C
```

Appendix D. Area ET Configuration File Example

[PROJECT]

basin_id = Klamath

project_folder = C:\PythonApps\bortsc\data

Output cet flag

output_cet_flag = False

output_cet_flag = True

Output cir flag

output_cir_flag = False

output_cir_flag = True

Limit to a date range (ISO Format: YYYY-MM-DD)

start_date = None

start_date = 1950-01-01

end_date = None

end_date = 1999-12-31

end_date = 1951-12-31

Plots sub-folder name

daily_plots_folder = daily_plots

[AET_META]

static_folder = static

ET Cell crop mix specifications

ccm_ts_type is 0 for variable crop mix, 1 for constant crop mix

ccm_ts_type = 1

ccm_ts_type = 0

ccm_mix_type is 0 for percentages, 1 for areas

```
ccm_mix_type = 1

# area_units_type is 0 hectare, 1 for acre

area_units_type = 1

ccm_delimiter = \t

ccm_header_lines = 1

ccm_names_line = 1

# Text setup

# cell_mix_name = ETCellsCropMix.txt

# cell_mix_delimiter = \t

# Excel setup

# cell_mix_name = MetAndDepletionNodes.xlsx

# cell_mix_ws = ETCellsCropMix

# cell_mix_ws = TSCropMix

# cell_mix_ws = VarCropMix

cell_mix_name = MetAndDepletionNodes.xlsx

cell_mix_ws = ETCellsCropMix

# cell_crops_name = ETCellsCrops.txt

# ET Cells crops (flags)

cell_crops_header_lines = 3

cell_crops_names_line = 2

# cell_crops_delimiter = ,

# Text setup

# cell_crops_name = ETCellsCrops.txt

# cell_crops_delimiter = \t

cell_crops_delimiter = \t
```

```
# Excel setup
# cell_crops_name = MetAndDepletionNodes.xlsx
# cell_crops_ws = ETCellsCrops
cell_crops_name = MetAndDepletionNodes.xlsx
cell_crops_ws = ETCellsCrops
# Crop parameters
crop_params_header_lines = 4
crop_params_names_line = 3
# Text setup
# crop_params_name = CropParams.txt
# crop_params_delimiter = \t
# Excel setup
# crop_params_name = MetAndDepletionNodes.xlsx
# crop_params_ws = CropParams
crop_params_name = MetAndDepletionNodes.xlsx
crop_params_ws = CropParams
# Miscellaneous
time_step = day
# ts_quantity = 1    # default
[INCET]
# input crop et data specifications
# cet time series data folder names
daily_input_cet_folder = daily_cet
# data structure type, file type (xls, tab, csv, rdb) and file name format
# DRI
```

```
# data_structure_type = DRI

# name_format = %s_crop_%c_S0.csv

# RDB

# data_structure_type = RDB

# name_format = %s_crop_S0.csv

data_structure_type = DRI

name_format = %s_crop_%c_S0.csv

file_type = csv

header_lines = 1

names_line = 1

delimiter = ,

# Set data field names

date_field = Date

ret_field = PMETo

ppt_field = PPT

etAct_field = ETact

etPot_field = ETpot

sir_field = Irrigation

sro_field = Runoff

dperc_field = DPerc

season_field = Season

cir_field = NIWR

# Units

ret_units = mm

ppt_units = mm
```

etAct_units = mm

etPot_units = mm

sir_units = mm

sro_units = mm

dperc_units = mm

[OUTAET]

Area weighted cell ET data specifications

daily_output_aet_flag = True

daily_output_aet_flag = False

monthly_output_aet_flag = False

monthly_output_aet_flag = True

annual_output_aet_flag = False

annual_output_aet_flag = True

Sub folder names

daily_output_aet_folder = daily_out_aet

monthly_output_aet_folder = monthly_out_aet

annual_output_aet_folder = annual_out_aet

aet data structure type, file type (xls, tab, csv, rdb) and file name format

station setup

data_structure_type = SF P

file_type = csv

name_format = %s_KL_S0_AET.csv

parameter setup

individual parameter files example - %p is wildcard for "_name"'s

name_format = KLDaily%pS0Data.rdb

```
# shared parameter files example
# name_format = KLDailyS0Data.rdb
# data_structure_type = PF S.P
# file_type = csf
# name_format = KL_S0_AET_%p.csv
# data_structure_type = PF S.P
# file_type = rdb
# name_format = KL_S0_AET_%p.rdb
# data_structure_type = PF S.P
# file_type = xls
# name_format = KL_S0_AET_%p.xlsx
# used format
data_structure_type = SF P
file_type = csv
name_format = %s_KL_S0_AET.csv
header_lines = 1
# 1's based indices
names_line = 1
# delimiter = \t
delimiter = ,
# date formats
# daily_date_format = %d-%m-%Y
# daily_float_format = %10.6f
# text ISO data formats
# daily_date_format = %Y-%m-%d
```

monthly_date_format = %Y-%m

annual_date_format = %Y-%m

Excel ISO data formats

daily_date_format = yyyy-mm-dd

monthly_date_format = yyyy-mm

annual_date_format = yyyy-mm

Excel RiverWare data formats

daily_date_format = m/d/yyyy

monthly_date_format = m/yyyy

annual_date_format = m/yyyy

Set data field names

date_field = Date

year_field = Year

month_field = Month

day_field = Day

et_field = Evapotranspiration

et_field = ET

nir_field = Net Irrigation Requirement

nir_field = NIR

etflow_field = ET Demand

etflow_field = ET_Flow

irflow_field = NIR Demand

nirflow_field = NIR_Flow

ret_field = Reference Evapotranspiration

ret_field = RET

```
# ppt_field = Precipitation

ppt_field = Prcp

# nirfrac_field = "NIR Fractions"

nirfrac_field = NIR_Frac

# Units

et_units = inches

nir_units = inches

etflow_units = cfs

irflow_units = cfs

ret_units = inches

ppt_units = inches

nirfrac_units = fraction

# Flow as volume units

# if None or non existing, use units specified in by Units

# otherwise, convert to volume units as specified

# default is None

# For acre-feet output, specify 'cfs' for Units specification

# daily_volume_units = acre-feet

# daily_volume_units = None

monthly_volume_units = acre-feet

# monthly_volume_units = None

annual_volume_units = acre-feet

# annual_volume_units = None

[OUTCIR]

# Output cell crop type CIR specifications
```

daily_output_cir_flag = True

monthly_output_cir_flag = True

annual_output_cir_flag = True

Sub folder names

daily_output_cir_folder = daily_out_cir

monthly_output_cir_folder = monthly_out_cir

annual_output_cir_folder = annual_out_cir

cir data structure type, file type (xls, tab, csv, rdb) and file name format

station setup

file_type = csv

data_structure_type = SF P

name_format = %s_CellCIRData.csv

parameter setup - no wildcards

data_structure_type = PF S.P

file_type = csf

name_format = KLCellCIRData.csv

file_type = rdb

name_format = KLCellCIRData.rdb

file_type = xls

name_format = KLCellCIRData.xlsx

used format

data_structure_type = SF P

file_type = csv

name_format = %s_CellCIRData.csv

cir_sheet = CIR

```
header_lines = 1

# 1's based indices

names_line = 1

# delimiter = \t

delimiter = ,

# date formats

# daily_date_format = %d-%m-%Y

# daily_float_format = %10.6f

# text ISO data formats

# daily_date_format = %Y-%m-%d

# monthly_date_format = %Y-%m

# annual_date_format = %Y-%m

# Excel ISO data formats

# daily_date_format = yyyy-mm-dd

# monthly_date_format = yyyy-mm

# annual_date_format = yyyy-mm

# Excel RiverWare data formats

# daily_date_format = m/d/yyyy

# monthly_date_format = m/yyyy

# annual_date_format = m/yyyy

# Set data field names

date_field = Date

# year_field = Year

# month_field = Month

# day_field = Day
```

```
# cir_field = Crop Evapotranspiration

cir_field = CIR

# Units

cir_units = inches

[OUTCET]

# Output cell crop type CET specifications

daily_output_cet_flag = True

monthly_output_cet_flag = True

annual_output_cet_flag = True

# Sub folder names

daily_output_cet_folder = daily_out_cet

monthly_output_cet_folder = monthly_out_cet

annual_output_cet_folder = annual_out_cet

# cet data structure type, file type (xls, tab, csv, rdb) and file name format

# station setup

# file_type = csv

# data_structure_type = SF P

# name_format = %s_CellCETData.csv

# parameter setup - no wildcards

# data_structure_type = PF S.P

# file_type = csf

# name_format = KLCellCETData.csv

# file_type = rdb

# name_format = KLCellCETData.rdb

# file_type = xls
```

```
# name_format = KLCellCETData.xlsx

# used format

file_type = csv

data_structure_type = SF P

name_format = %s_CellCETData.csv

header_lines = 1

# 1's based indices

names_line = 1

# delimiter = \t

delimiter = ,

cir_sheet = "CET"

# Set data field names

date_field = Date

# cet_field = Crop Evapotranspiration

cet_field = CET

# Units

cet_units = inches
```

Appendix D. Solar Radiation Calibration Configuration File Example

```
[PROJECT]

project_folder = C:\PythonApps\bortsc\tools

# Limit to a date range (ISO Format: YYYY-MM-DD)

# Specify None or do not specify to use file contents

# start_date = None

# start_date = 1950-01-01

# end_date = None

# end_date = 1999-12-31

[SOLAR_META]

# station data

elev_units = meter

elevation = 422

latitude = 35.14887

longitude = 98.46607

# miscellaneous

missing_data_value = NaN

# missing_data_value = -999

# weather data section (Tmax, Tmin, TDew, Rs)

[INMET]

input_met_folder = solar_radiation_opt

# text setup

# file_type = csv

# file_name = RecordedMetData.csv

# header_lines = 1
```

```
# names_line = 1
# sheet_delim = ,
# workbook setup
# file_type = xls
# file_name = RecordedMetData.xlsx
# header_lines = 1
# names_line = 1
# sheet_delim = sheet1
# used setup
file_type = xls
file_name = RecordedMetData.xlsx
header_lines = 1
names_line = 1
sheet_delim = sheet1
# additional file specs
# field names
# date_field = Date
year_field = Year
month_field = Month
day_field = Day
tmax_field = TmaxC
tmin_field = TminC
tdew_field = TdewC
rs_field = Rs_MJ_m2
```

Units

tmax_units = C

tmin_units = C

rs_units = MJ/m2

tdew_units = C