swb Documentation

Release 5.0.0

Antonis Christofides

Aug 13, 2021

Contents

CHAPTER 1

`calculate_soil_water()` — Calculation of soil water balance

1.1 Usage

```
from swb import calculate_soil_water

results = calculate_soil_water(
    theta_s=0.425,
    theta_fc=0.287,
    theta_wp=0.14,
    zr=0.5,
    zr_factor=1000,
    p=0.5,
    draintime=2.2,
    timeseries=a_pandas_dataframe,
    theta_init=0.19,
    refill_factor=0.5,
)
```

See the reference section below for what all this means.

1.2 Methodology

1.2.1 Introduction

This module provides the *`calculate_soil_water()`* function, which calculates soil water content.

The soil is, so to speak, full of water when it is at field capacity and empty when it is at wilting point.

The **field capacity** is below saturation point. When the soil is saturated, any excess water runs off immediately; when it is between field capacity and saturation, the soil drains downward, usually in 2-3 days; when the soil is at or below field capacity, water is lost only through evapotranspiration. The field capacity is roughly the amount of water that the soil can keep indefinitely without percolation. It is a property of the soil.

The **wilting point** (also called "permanent wilting point") is the amount of water needed for plants to not wilt. It is a property of the soil.

We are interested only in the topmost part of the soil, which is called the **root zone**. It is the part of the soil that contains plant roots. The depth of the root zone is a property of the crop.

The **depletion** (more precisely the "root zone depletion") is the amount of water missing from the soil (more precisely the root zone). When the soil is "full" (at field capacity) the depletion is zero. When it is "half-empty", the depletion is the amount of water that would be needed to "fill it up"; that is, the amount of water that is needed to reach field capacity. The depletion is normally measured in mm.

This is the relation between depletion D_r and water content θ:

$$D_r = (\theta_{fc} - \theta)\ Z_r$$

(*FAO56*, p. 170 eq. 87)

where:

- θ is the water content. It is a proper number (m^3/m^3).
- θ_{fc} is the water content at field capacity.
- Z_r is the root zone depth.

Often Z_r is in meters and depletion in mm, so the equation becomes

$$D_r = (\theta_{fc} - \theta) \times Z_r \times 1000$$

(In the rest of this text, we call 1000 the "root depth factor" or *zr_factor*).

As we said, the soil is "full" at field capacity and "empty" at wilting point. The difference between these two is the **total available water** or TAW. In other words, if the soil was a reservoir, the TAW would be its capacity. The TAW is normally measured in mm.

$$TAW = (\theta_{fc} - \theta_{wp}) \times Z_r \times zr\_factor$$

(*FAO56*, p. 162 eq. 82)

Since saturation is above field capacity, the soil can be "overfull", so to speak, in which case depletion is negative. The **drain time** is the time the soil needs to go from saturation (θ_s) to field capacity (θ_{fc}) solely because of downward movement (percolation) (i.e. if evapotranspiration is zero).

Although plants can theoretically survive whenever the water content is above wilting point, there's a threshold below which they are stressed. This is different from crop to crop. The difference between field capacity and this threshold is the **readily available water**:

$$RAW = p\ TAW$$

(*FAO56*, p. 162 eq. 83)

The factor *p*, called "soil water depletion fraction for no stress", is a property of the crop.

When the water content is above the threshold ($D_r < RAW$), the crop evapotranspiration is $K_c \times ET_0$, where ET_0 the reference evapotranspiration and K_c the crop coefficient. When the water content is below the threshold ($D_r > RAW$), the crop is stressed and decreases the amount of evapotranspiration to $K_s \times K_c \times ET_0$ where K_s is the **water stress coefficient**:

$$K_s = (TAW - D_r) / (TAW - RAW) = (TAW - D_r) / ((1-p)\ TAW)$$

(*FAO56*, p. 169 eq. 84)

When the water content reaches the threshold (i.e. when D_r reaches RAW), we need to irrigate. Normally the amount of water we irrigate with is RAW. But sometimes we prefer to throw in a fraction of that amount. This will result in more frequent irrigations thereafter and is beneficial in some cases. This fraction is called the **refill factor**.

1.2.2 Calculation of depletion

The basis for the calculation is this formula:

$$D_{r,i} = D_{r,i-1} - (P_i - RO_i) - IR_{n,i} - CR_i + ET_{c,i} + DP_i$$

(*FAO56*, p. 170 eq. 85)

where:

- i is the current time period (i.e. the current day).
- $D_{r,i}$ is the root zone depletion at the end of time period i.
- P_i is the effective precipitation (see below).
- RO_i is the runoff (see below).
- $IR_{n,i}$ is the net irrigation depth (see below).
- CR_i is the capillary rise.
- $ET_{c,i}$ is the crop evapotranspiration.
- DP_i is the water loss through deep percolation.

subject to this constraint:

$$D_{r,i} \quad TAW$$

(*FAO56*, p. 170 eq. 86. That equation also has the constraint $D_{r,i}$ 0; however we allow $D_{r,i}$ to have negative values, because water content can actually exceed field capacity and reach saturation.)

CR_i is ignored and considered zero.

The evapotranspiration $ET_{c,i}$ is the reference evapotranspiration multiplied by the crop coefficient K_c.

The **runoff** is the amount of water that exceeds saturation after heavy rainfall:

$$RO_i = P_i + (\theta_{i-1} - \theta_s)\ Z_r \text{ when larger than zero}$$

(*Malamos et al., 2016*, eq. 5)

The **effective precipitation** is the precipitation that actually falls on the soil. It is essentially the total precipitation minus the amount that is held by the leaves. `swb` does not contain any model that converts total precipitation to effective precipitation; you need to make this conversion and call `calculate_soil_water()` with the effective precipitation. (A trivial model that you can use is multiply total precipitation by a factor, p_{eff}, usually 0.8; it's quite crude, but it's better than nothing.)

The **net irrigation depth** is the amount of water that reaches the soil during irrigation. It is the total amount of water consumed for irrigation minus losses. `swb` does not convert between total and net irrigation; it accepts net irrigation as input (and includes net irrigation in its output).

The **deep percolation** is zero if we are at or below field capacity. If we are above field capacity ($\theta_{fc} < \theta < \theta_s$) it is this:

$$DP_i = (\theta - \theta_{fc}) * Z_r / draintime$$

If $\theta > \theta_s$ (which technically can't happen, but θ can have this value as calculated in the previous step, notably if there has been too much irrigation) then θ_s is used instead of θ in the above equation.

1.3 Reference

`calculate_soil_water`(***kwargs*)

Calculates soil water balance. Example:

```
results = calculate_soil_water(
    theta_s=0.425,
    theta_fc=0.287,
    theta_wp=0.14,
    zr=0.5,
    zr_factor=1000,
    p=0.5,
    draintime=2.2,
    timeseries=a_pandas_dataframe,
    theta_init=0.19,
    refill_factor=0.5,
)
```

Parameters

- **theta_s** (*float*) – Water content at saturation.
- **theta_fc** (*float*) – Water content at field capacity.
- **theta_wp** (*float*) – Water content at wilting point.
- **zr** (*float*) – The root depth.
- **zr_factor** (*float*) – If the root depth is in a different unit than the water depth variables (such as evapotranspiration, precipitation, irrigation and depletion) `zr_factor` is used to convert it. If the root depth is in metres and the water depth variables are in mm, specify `zr_factor=1000`.
- **p** (*float*) – The soil water depletion fraction for no stress.
- **draintime** (*float*) – The time, in days, needed for the soil to drain from saturation to field capacity.
- **timeseries** (*dataframe*) – A pandas dataframe indexed by date, containing two or three columns with input time series. The dataframe and its time series must be continuous and have no missing values. The columns are "crop_evapotranspiration", "effective_precipitation" and "actual_net_irrigation". All time series should be in mm; more precisely, in the same unit as the resulting depletion.

  The "crop_evapotranspiration" is the potential crop evapotranspiration (that is, the reference evapotranspiration multiplied by the crop coefficient K_c).

  The "actual_net_irrigation" is the applied net irrigation (that is, the total applied irrigation multiplied by the irrigation efficiency). Each record of the actual net irrigation is either a floating point number expressing the amount of water, or the string "model", or the string "fc".

  If it is the string "model", then it is assumed that the amount of water equals the recommended amount that was calculated by the model.

  If it is the string "fc", it is assumed that the amount of water is what the model calculates that would be required to bring the soil to field capacity. If it's already at field capacity but below saturation, it is what the model calculates that would be required to bring the soil to saturation. If it's already at saturation or higher, zero is assumed. The purpose of this option is to make some assumption about the amount of water when it is known that a field was irrigated (presumably sufficiently) but the amount of water is unknown.
- **theta_init** (*float*) – The initial water content (that is, the water content at the first date of the time series).
- **refill_factor** (*float*) – The refill factor.

Return type dict

Returns

A dictionary with the results. It contains the following items:

raw The readily available water.

taw The total available water.

timeseries The original dataframe with additional columns added, namely:

`dr` The depletion.

`theta` The soil moisture.

`ks` The water stress coefficient.

`recommended_net_irrigation` The calculated recommended net irrigation.

`assumed_net_irrigation` This equals `actual_net_irrigation` where the latter's value is a floating point number. Where `actual_net_irrigation`'s value is "model" or "fc", `assumed_net_irrigation` contains the assumed amount of water.

The original dataframe is changed in place (so the caller doesn't really need it returned), but the original columns and index are untouched.

1.4 References

R. G. Allen, L. S. Pereira, D. Raes, and M. Smith, Crop evapotranspiration - Guidelines for computing crop water requirements, FAO Irrigation and drainage paper no. 56, 1998.

N. Malamos, I. L. Tsirogiannis, and A. Christofides, Modelling irrigation management services: the IRMA_SYS case, International Journal of Sustainable Agricultural Management and Informatics, 2 (1), 1–18, 2016.

CHAPTER 2

calculate_crop_evapotranspiration() — Get ET_c from ET_0

2.1 Usage

```
from swb import KcStage, calculate_crop_evapotranspiration

calculate_crop_evapotranspiration(
    timeseries=a_pandas_dataframe,
    planting_date=dt.date(2019, 3, 21),
    kc_offseason=0.3,
    kc_plantingdate=0.7,
    kc_stages=(
        KcStage(35, 0.7),
        KcStage(45, 1.05),
        KcStage(40, 1.05),
        KcStage(15, 0.95),
    ),
)
```

`timeseries` is a pandas dataframe that contains a `ref_evapotranspiration` column with the reference evapotranspiration. Two time series will be calculated and added to the dataframe: `kc` and `crop_evapotranspiration`. `kc_stages` specifies the K_c stages, that is, a sequence of (number of days in stage, K_c at end of stage) pairs. `KcStage` is a named tuple whose items are `ndays` and `kc_end`. The planting_date corresponds to the beginning (day 1) of the first stage. At each stage, we do linear interpolation between the K_c at the end of the stage and the K_c at the end of the previous stage (or `kc_plantingdate` if there's no previous stage).

This is a generalization of the methodology of *FAO56*, where three values are given for K_c ($K_{c\,\text{ini}}$, $K_{c\,\text{mid}}$, $K_{c\,\text{end}}$), and there are four development stages: initial, development, middle, and late. The example above is equivalent to $K_{c\,\text{ini}}$ = 0.7, $K_{c\,\text{mid}}$ = 1.05, $K_{c\,\text{end}}$ = 0.95, initial = 35 days, development = 45 days, middle = 40 days, late = 15 days. It also assumes a $K_{c\,\text{os}}$ (off-season) of 0.3, i.e. how much water is evaporated by the soil before planting. (Technically this isn't "crop evapotranspiration" and the coefficient isn't K_c; but in order to calculate depletion we need to know how much water has evaporated from the soil and put this in the resulting time series.)

2.2 References

R. G. Allen, L. S. Pereira, D. Raes, and M. Smith, Crop evapotranspiration - Guidelines for computing crop water requirements, FAO Irrigation and drainage paper no. 56, 1998.

CHAPTER 3

get_effective_precipitation() — Get P_{eff} from P

3.1 Usage

```
from swb import get_effective_precipitation

get_effective_precipitation(timeseries)
```

`timeseries` is a pandas dataframe in daily step that contains a `precipitation` column and a `ref_evapotranspiration` (reference evapotranspiration) column. The function adds an `effective_precipitation` column to the data frame. This is calculated with a simple model: The effective precipitation is 0.8 times the precipitation, unless the daily precipitation is less than a fifth of the reference evapotranspiration, in which case the effective precipitation is zero.

CHAPTER 4

License

Index

C