bifacialvf

Release 0.1.6

Sep 28, 2022

Contents:

1 Citing bifacialvf 3

2 Contents 5
- 2.1 What's New . . . 5
  - 2.1.1 v1.8 (18 / 03 / 2022) . . . 5
- 2.2 Installation . . . 6
- 2.3 API . . . 6
  - 2.3.1 BifacialVF Package . . . 6
  - 2.3.2 View Factors . . . 6

3 Indices and tables 13

Python Module Index 15

Index 17

bifacialvf is a NREL supported tool that provides a set of functions and casses for simulating the performance of bifacial PV systems through view factor method.

The intended audience ranges from PV performance researchers, Engineering Procurement Construction (EPC) companies, installers, investors, consumers and analysts of the PV industry interested in predicting and evaluating bifacial photovoltaic systems.

The source code for bifacial_radiance is hosted on github.

Please see the *Installation* page for installation help.

CHAPTER 1

Citing bifacialvf

Many of the contributors to bifacial_radiance work in institutions where citation metrics are used in performance or career evaluations. If you use bifacial_radiance in a published work, please cite:

> Marion, Bill, MacAlpine, S, Deline, Chris, Asgharzadeh, Amir, Toor, Fatima, Riley, D., Stein, Joshua S., Hansen, C. A practical Irradiance Model for Bifacial PV Modules. 2017 IEEE Photovoltaic Specialists Conference, Washington DC, 2017. URL: https://www.nrel.gov/docs/fy17osti/67847.pdf

CHAPTER 2

Contents

2.1 What's New

These are new features and improvements of note in each release.

2.1.1 v1.8 (18 / 03 / 2022)

- `agriPV` as input, so that .csv output has the Ground irradiance at every timestep. The ground irradiance is given in 100 segments between start of the module and start of the module in the next row.
- Capability to run Mismatch Analysis along with the bifacial calculations: can use PVMismatch (set input calculatePVMismatch to True) or Bilinear Interpolation (set input calculateBilInterpol to True)
- Input can be TMY, EPW, or a dataframe with the right column headings. See documentation for the needed headings.
- New Sphinx documentation.
- New logo!
- Sunposition now calculated from PVLib.
- Sunposition can now be calculated at the timestamp, or with a delta of time before or after the timestamp. Sunrise and Susnet positions are also calculated and the sunposition for that hour is corrected (i.e. if it's right labeled data, and sun rises at 7:40, sunposition is calculated at 7:50 not at 7:30)
- Albedo gets read from a column for every timepoint provided.
- Deprecated "D" (spacing from module edge to next row module edge) deprecated
- Important bug fix on Sunset hour position calculation.
- Some Changes throughout to take advantage of arrays, like moving all tracking calculations outside of the main loop.

- Changed how results are printed for tracking scenarios; before when sazm switched from 90 to 270, the Sensor position changed (so S1 became Slast), same for ground. Now the positoins are internally inverted so the output is consistent for spatil location.
- Updated requirements
- Updated TUTORIALS

Contributors

- Silvana Ovaitt (@shirubana)
- Chris Deline (@cdeline)
- Mark Mikofski (@mikofski)
- Nick DiOrio (@nickdiorio)

2.2 Installation

pip install bifacialvf

2.3 API

Modules, methods, classes and attributes are explained here.

2.3.1 BifacialVF Package

2.3.2 View Factors

ViewFactor module - VF calculation helper files for bifacial-viewfactor

@author Bill Marion @translated to python by sayala 06/09/17

Get Back Surface Irradiances

`bifacialvf.vf.`**`getBackSurfaceIrradiances`**(*rowType*, *maxShadow*, *PVbackSurface*, *beta*, *sazm*, *dni*, *dhi*, *C*, *D*, *albedo*, *zen*, *azm*, *cellRows*, *pvBackSH*, *rearGroundGHI*, *frontGroundGHI*, *frontReflected*, *offset=0*)

This method calculates the AOI corrected irradiance on the back of the PV module/panel. 11/19/2015

Added rowType and other changes to distinguish between types of rows. 4/19/2016

Added input of offset of reference cell from PV module back (in PV panel slope lengths) for modeling Sara's reference cell measurements, should be set to zero for PV module cell irradiances.

Added while loop so projected Xs aren't too negative causing array index problems (<0) 12/13/2016:

```
while (projectedX1 < -100.0 || projectedX2 < -100.0):
    # Offset so array indexes are >= -100.0  12/13/2016

    projectedX1 += 100.0;
    projectedX2 += 100.0;
```

Parameters

- **rowType** (*str*) – Type of row: “first”, “interior”, “last”, or “single”
- **maxShadow** – Maximum shadow length projected to the front(-) or rear (+) from the front of the module
- **PVbackSurface** – PV module back surface material type, either “glass” or “ARglass”
- **beta** – Tilt from horizontal of the PV modules/panels (deg) (for front surface)
- **sazm** – Surface azimuth of PV panels (deg) (for front surface)
- **dni** – Direct normal irradiance (W/m2)
- **dhi** – Diffuse horizontal irradiance (W/m2)
- **C** – Ground clearance of PV panel (in PV panel slope lengths)
- **D** – Horizontal distance between rows of PV panels (in PV panel slope lengths)
- **albedo** – Ground albedo
- **zen** – Sun zenith (in radians)
- **azm** – Sun azimuth (in radians)
- **pvBackSH** – Decimal fraction of the back surface of the PV panel that is shaded, 0.0 to 1.0
- **rearGroundGHI** (*array of size [100]*) – Global horizontal irradiance for each of 100 ground segments (W/m2)
- **frontGroundGHI** (*array of size [100]*) – Global horizontal irradiance for each of 100 ground segments (W/m2)
- **frontReflected** (*array of size [cellRows]*) – Irradiance reflected from the front of the PV module/panel (W/m2) in the row behind the one of interest
- **offset** – Offset of reference cell from PV module back (in PV panel slope lengths), set to zero for PV module cell irradiances

Returns

- **backGTI** (*array of size [cellRows]*) – AOI corrected irradiance on back side of PV module/panel, one for each cell row (W/m2)
- **aveGroundGHI** (*numeric*) – Average GHI on ground under PV array

Notes

1-degree hemispherical segment AOI correction factor for glass (index=0) and ARglass (index=1)

Get Front Surface Irradiances

`bifacialvf.vf.getFrontSurfaceIrradiances`(*rowType*, *maxShadow*, *PVfrontSurface*, *beta*, *sazm*, *dni*, *dhi*, *C*, *D*, *albedo*, *zen*, *azm*, *cellRows*, *pvFrontSH*, *frontGroundGHI*)

This method calculates the AOI corrected irradiance on the front of the PV module/panel and the irradiance reflected from the the front of the PV module/panel. 11/12/2015

Added row type and MaxShadow and changed code to accommodate 4/19/2015

Parameters

- **rowType** (*str*) – Type of row: "first", "interior", "last", or "single"
- **maxShadow** – Maximum shadow length projected to the front (-) or rear (+) from the front of the module row (in PV panel slope lengths), only used for *rowTypes* other than "interior"
- **PVfrontSurface** – PV module front surface material type, either "glass" or "ARglass"
- **beta** – Tilt from horizontal of the PV modules/panels (deg)
- **sazm** – Surface azimuth of PV panels (deg)
- **dni** – Direct normal irradiance (W/m2)
- **dhi** – Diffuse horizontal irradiance (W/m2)
- **C** – Ground clearance of PV panel (in PV panel slope lengths)
- **D** – Horizontal distance between rows of PV panels (in PV panel slope lengths)
- **albedo** – Ground albedo
- **zen** – Sun zenith (in radians)
- **azm** – Sun azimuth (in radians)
- **pvFrontSH** – Decimal fraction of the front surface of the PV panel that is shaded, 0.0 to 1.0
- **froutGroundGHI** (*array of size [100]*) – Global horizontal irradiance for each of 100 ground segments in front of the module row

Returns

- **frontGTI** (*array of size [cellRows]*) – AOI corrected irradiance on front side of PV module/panel, one for each cell row (W/m2)
- **frontReflected** (*array of size [cellRows]*) – Irradiance reflected from the front of the PV module/panel (W/m2)
- **aveGroundGHI** (*numeric*) – Average GHI on the ground (includes effects of shading by array) from the array frontGroundGHI[100]

Notes

1-degree hemispherical segment AOI correction factor for glass (index=0) and ARglass (index=1). Creates a list containing 5 lists, each of 8 items, all set to 0

Get Ground Shade Factors

`bifacialvf.vf.getGroundShadeFactors(rowType, beta, C, D, elv, azm, sazm)`

This method determines if the ground is shaded from direct beam radiation for points on the ground from the leading edge of one row of PV panels to the leading edge of the next row of PV panels behind it. This row-to-row dimension is divided into 100 ground segments and a ground shade factor is returned for each ground segment, with values of 1 for shaded segments and values of 0 for non shaded segments. The fractional amounts of shading of the front and back surfaces of the PV panel are also returned. 8/20/2015

4/18/2016 - Modified to account for different row types. Because the ground factors may now be different depending on row, they are calculated for the row-to-row dimension to the rear of the leading module edge and to the front of the leading edge. Also returned is the maximum shadow length projected to the front or rear from the front of the module row

Parameters

- **rowType** (*str*) – "first", "interior", "last", or "single"
- **beta** – Tilt from horizontal of the PV modules/panels (deg)
- **C** – Ground clearance of PV panel (in PV panel slope lengths)
- **D** – Horizontal distance between rows of PV panels (in PV panel slope lengths)
- **elv** – Sun elevation (in radians)
- **azm** – Sun azimuth (in radians)
- **sazm** – Surface azimuth of PV panels (deg)

Returns

- **pvFrontSH** (*numeric*) – Decimal fraction of the front surface of the PV panel that is shaded, 0.0 to 1.0
- **pvBackSH** (*numeric*) – Decimal fraction of the back surface of the PV panel that is shaded, 0.0 to 1.0
- **rearGroundSH** (*array of size [100]*) – Ground shade factors for ground segments to the rear, 0 = not shaded, 1 = shaded
- **frontGroundSH** (*array of size [100]*) – Ground shade factors for ground segments to the front, 0 = not shaded, 1 = shaded
- **maxShadow** (*numeric*) – Maximum shadow length projected to the front(-) or rear (+) from the front of the module row (in PV panel slope lengths), only used later for rowTypes other than "interior"

Get Sky Configuration Factors

`bifacialvf.vf.getSkyConfigurationFactors(rowType, beta, C, D)`

This method determines the sky configuration factors for points on the ground from the leading edge of one row of PV panels to the leading edge of the next row of PV panels behind it. This row-to-row dimension is divided into 100 ground segments and a sky configuration factor is returned for each ground segment. The sky configuration factor represents the fraction of the isotropic diffuse sky radiation (unobstructed) that is present on the ground when partially obstructed by the rows of PV panels. The equations follow that on pages in the notebook dated 8/12/2015. 8/20/2015

4/15/2016 Modifed for calculations other than just the interior rows. Row type is identified with the string *rowType*, with the possilbe values:

- first = first row of the array

- interior = interior row of array
- last = last row of the array
- single = a single row array

Because the sky configuration factors may now be different depending on row, they are calculated for the row-to-row dimension to the rear of the leading module edge and to the front of the leading edge.

Parameters

- **`rowType`** (*`str`*) – "first", "interior", "last", or "single"
- **`beta`** (*`float`*) – Tilt from horizontal of the PV modules/panels (deg)
- **`C`** (*`float`*) – Ground clearance of PV panel (in PV module/panel slope lengths)
- **`D`** (*`float`*) – Horizontal distance between rows of PV panels (in PV module/panel slope lengths)

Returns

- **rearSkyConfigFactors** (*array of size [100]*) – Sky configuration factors to rear of leading PVmodule edge (decimal fraction)
- **frontSkyConfigFactors** (*array of size [100]*) – Sky configuration factors to rear of leading PVmodule edge (decimal fraction)

Notes

The horizontal distance between rows, *D*, is from the back edge of one row to the front edge of the next, and it is not the row-to-row spacing.

Row Spacing

`bifacialvf.vf.`**`rowSpacing`**`(`*beta*, *sazm*, *lat*, *lng*, *tz*, *hour*, *minute*`)`

This method determines the horizontal distance D between rows of PV panels (in PV module/panel slope lengths) for no shading on December 21 (north hemisphere) June 21 (south hemisphere) for a module tilt angle beta and surface azimuth sazm, and a given latitude, longitude, and time zone and for the time passed to the method (typically 9 am).

(Ref: the row-to-row spacing is then `D + cos(beta)`) 8/21/2015

Parameters

- **`beta`** (*`double`*) – Tilt from horizontal of the PV modules/panels (deg)
- **`sazm`** (*`double`*) – Surface azimuth of the PV modules/panels (deg)
- **`lat`** (*`double`*) – Site latitude (deg)
- **`lng`** (*`double`*) – Site longitude (deg)
- **`tz`** (*`double`*) – Time zone (hrs)
- **`hour`** (*`int`*) – hour for no shading criteria
- **`minute`** (*`double`*) – minute for no shading

Returns **D** (*numeric*) – Horizontal distance between rows of PV panels (in PV panel slope lengths)

Tracking Bifacial Values Calculator

`bifacialvf.vf.`**`trackingBFvaluescalculator`**(*beta*, *hub_height*, *r2r*)

1-axis tracking helper file

Parameters

- **`beta`** (*`series of floats, or float`*) – Tilt from horizontal of the PV modules/panels, in radians
- **`hub_height`** (*`float`*) – tracker hub height
- **`r2r`** (*`float`*) – Row-to-row distance (in PV panel slope lengths)

Returns

- **C** (*float*) – ground clearance of PV panel
- **D** (*float*) – row-to-row distance (each in PV panel slope lengths)

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

b

bifacialvf, 6
bifacialvf.vf, 6

Index

B

bifacialvf (*module*), 6
bifacialvf.vf (*module*), 6

G

getBackSurfaceIrradiances() (*in module bifacialvf.vf*), 6
getFrontSurfaceIrradiances() (*in module bifacialvf.vf*), 8
getGroundShadeFactors() (*in module bifacialvf.vf*), 9
getSkyConfigurationFactors() (*in module bifacialvf.vf*), 9

R

rowSpacing() (*in module bifacialvf.vf*), 10

T

trackingBFvaluescalculator() (*in module bifacialvf.vf*), 11