A twelve-year retrospective on pvlib: open-source PV performance modeling libraries

Joshua S. Stein[1], Clifford Hansen[1]

With acknowledgments to Robert Andrews[2], Will Holmgren[3], Mark Mikofski[3], Kevin Anderson[4], Adam Jensen[5], Anton Driesse[6]

[1] Sandia National Laboratories

[2] Heliolytics

[3] DNV

[4] NREL

[5] DTU

[6] PV Performance Labs

SAND2022-14270 PE

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

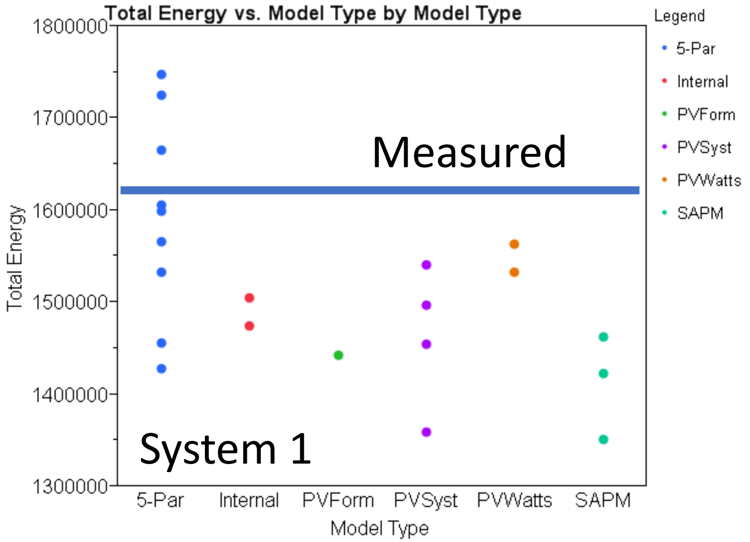
PV_LIB, PV_Lib Toolbox, pvlib-python

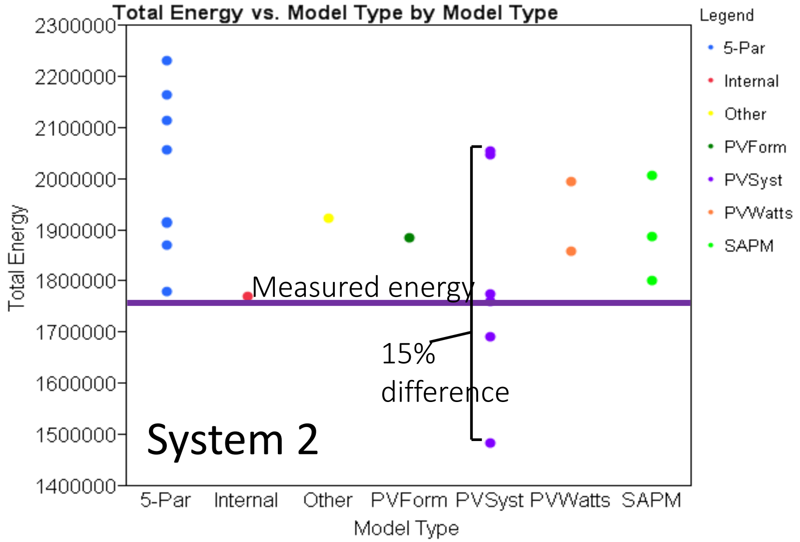
In 2010 Sandia National Laboratories identified problems with PV system modeling:

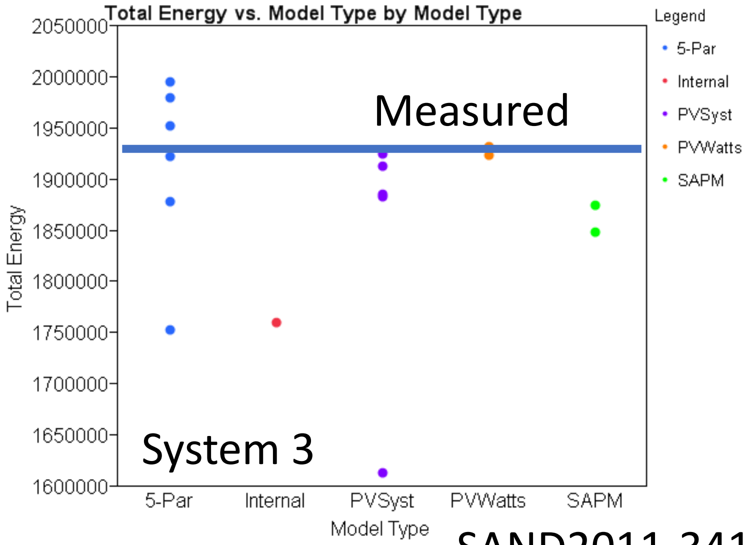
- A system model requires use of many component models.
- Each researcher implemented (coded) these models in the language/tool of their choice for their personal use.
- Cross comparison between different researchers discovered that few of these implantations of the same model yielded the same result.
- To quantify the effect on system models, Sandia ran the 1st PV Performance Modeling Blind Comparison.

2010 1st Blind PV Modeling Comparison

- Weather and system design data provided for 3 systems. Modelers asked to predict power output.
- 25 contributors
- In one case differences of 15% were seen between two modelers using the same model from the same company!







SAND2011-3419

How to make modeling more consistent?

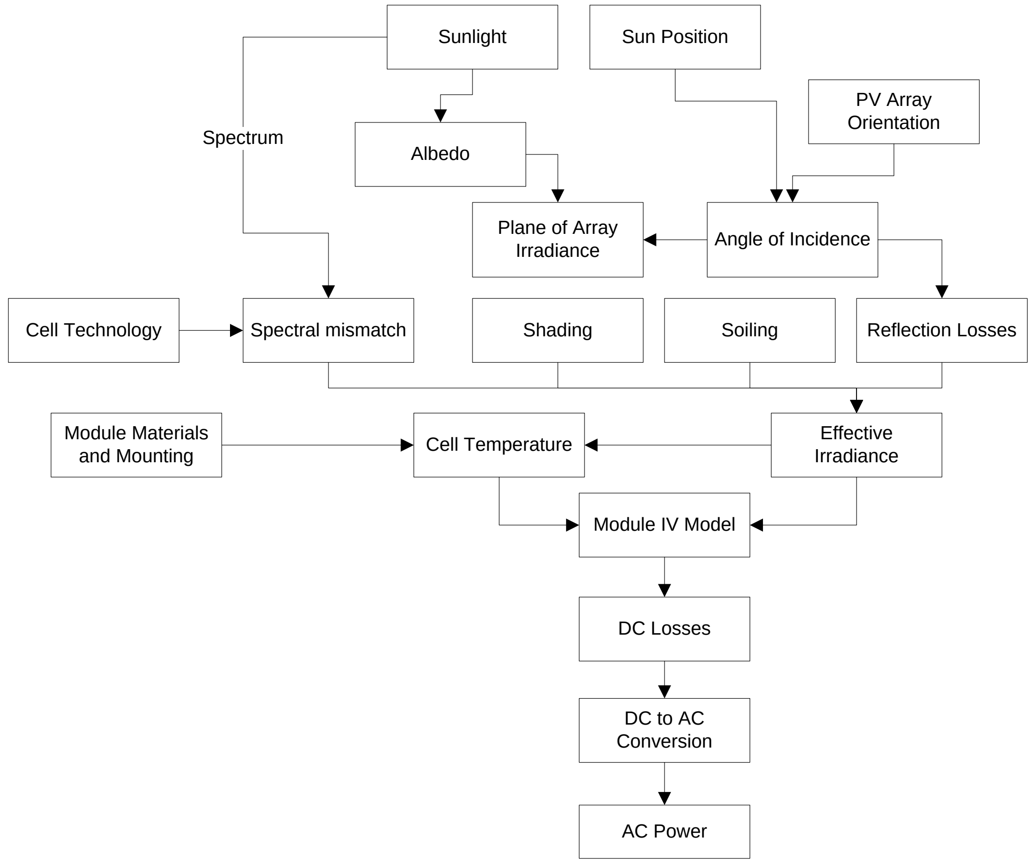
Standardize modeling practice (Modeling Chain Concept)

- Each step of the chain has options for submodels
- Standardize inputs and outputs
  - Locations (lat, long, elev, tz)
  - Timeseries format conventions
- Documentation and examples

Share community software

- PV_LIB or PVLIB Toolbox for Matlab was born!
- Designed and coded by the Sandia PV group
- Offered for download on PVPMC website

- PVLIB Matlab has moved to Github
- Code base has fallen behind pvlib-python and is relatively stagnant.



Stein, J. S. (2017). Energy Prediction and System Modeling. Photovoltaic Solar Energy: From Fundamentals to Applications. A. Reinders, P. Verlinden, W. v. Sark and A. Freundlich. Chichester, West Sussex UK, Wiley: 564-578.

pvlib-python origins

In 2013 Rob Andrews (graduate student at Queen's University) proposed translating PV_LIB Matlab to python.

- Python was free to use
- Jupyter notebooks made documenting analyses easier

PV_LIB Version 1.1 was born (renamed pvlib-python and re-versioned to 0.1).

- 31 Functions were included
- Initial tests were proposed
- Developed on Github
- Documents on readthedocs

TABLE I

FUNCTIONS INCLUDED IN VERSION 1.1 OF PV_LIB . ALL REFERENCES ARE INCLUDED IN THE ON LINE PV_LIB DOCUMENTATION

Irradiance and atmospheric functions	
pvlib.pvl_alt2pres(altitude)	Determine site pressure from altitude
pvlib.pvl_pres2alt(pressure)	Determine altitude from site pressure
pvlib.pvl_getaoi(SurfTilt, SurfAz, SunZen, SunAz)	Determine angle of incidence from surface tilt/azimuth and apparent sun zenith/azimuth
pvlib.pvl_disc(GHI, SunZen, Time[, pressure])	Estimate Direct Normal Irradiance from Global Horizontal Irradiance using the DISC model
pvlib.pvl_ephemeris(Time, Location[, ...])	Calculates the position of the sun given time, location, and optionally pressure and temperature
pvlib.pvl_spa(Time, Location)	Calculate the solar position using the PySolar package
pvlib.pvl_extraradiation(doy)	Determine extraterrestrial radiation from day of year
pvlib.pvl_globalinplane(SurfTilt, SurfAz, ...)	Determine the three components on in-plane irradiance
pvlib.pvl_grounddiffuse(SurfTilt, GHI, Albedo)	Estimate diffuse irradiance from ground reflections given irradiance, albedo, and surface tilt
pvlib.pvl_makelocationstruct(latitude, ...)	Create a structure to define a site location
pvlib.pvl_relativeairmass(z[, model])	Gives the relative (not pressure-corrected) airmass
pvlib.pvl_absoluteairmass(AMrelative, Pressure)	Determine absolute (pressure corrected) airmass from relative airmass and pressure
pvlib.pvl_clearsky_ineichen(Time, Location)	Determine clear sky GHI, DNI, and DHI from Ineichen/Perez model
pvlib.pvl_clearsky_haurwitz(ApparentZenith)	Determine clear sky GHI from Haurwitz model
Irradiance Translation Functions	
pvlib.pvl_perez(SurfTilt, SurfAz, DHI, DNI, ...)	Determine diffuse irradiance from the sky on a tilted surface using one of the Perez models
pvlib.pvl_haydavies1980(SurfTilt, SurfAz, ...)	Determine diffuse irradiance from the sky on a tilted surface using Hay & Davies' 1980 model
pvlib.pvl_isotropicsky(SurfTilt, DHI)	Determine diffuse irradiance from the sky on a tilted surface using isotropic sky model
pvlib.pvl_kingdiffuse(SurfTilt, DHI, GHI, SunZen)	Determine diffuse irradiance from the sky on a tilted surface using the King model
pvlib.pvl_klucher1979(SurfTilt, SurfAz, DHI, ...)	Determine diffuse irradiance from the sky on a tilted surface using Klucher's 1979 model
pvlib.pvl_reindl1990(SurfTilt, SurfAz, DHI, ...)	Determine diffuse irradiance from the sky on a tilted surface using Reindl's 1990 model
Data Handling	
pvlib.pvl_readtmy2(FileName)	Read a TMY2 file in to a DataFrame
pvlib.pvl_readtmy3(FileName)	Read a TMY3 file in to a DataFrame
System Modeling functions	
pvlib.pvl_physicaliam(K, L, n, theta)	Determine the incidence angle modifier using refractive models
pvlib.pvl_ashraeiam(b, theta)	Determine the incidence angle modifier using the ASHRAE transmission model
pvlib.pvl_calcparams_desoto(S, Tcell, ...[, ...])	Applies the temperature and irradiance corrections to inputs for pvl_singlediode
pvlib.pvl_retreiveSAM(name[, FileLoc])	Retrieve latest module and inverter info from SAM website
pvlib.pvl_sapm(Module, Eb, Ediff, Tcell, AM, AOI)	Performs Sandia PV Array Performance Model to get 5 points on IV curve given SAPM module parameters, Ee, andcell temperature
pvlib.pvl_sapmcelltemp(E, Wspd, Tamb[, modelt])	Estimate cell temperature from irradiance, wind speed, ambient temperature, and module parameters (SAPM)
pvlib.pvl_singlediode(Module, IL, I0, Rs, ...)	Solve the single-diode model to obtain a photovoltaic IV curve
pvlib.pvl_snlinverter(Inverter, Vmp, Pmp)	Converts DC power and voltage to AC power using Sandia's Grid-Connected PV Inverter model
pvlib.pvl_systemdef(TMYmeta, SurfTilt, ...)	Generates a dictionary of system parameters used throughout a simulation

Andrews, R. W., et al. (2014). Introduction to the Open Source PV LIB for Python Photovoltaic System Modelling Package. 40th IEEE PV Specialists Conference. Denver, CO.

pvlib-python maturation (2014 – 2022)

The Dark Ages (2014-2018)

Will Holmgren (University of Arizona, now DNV) was researching PV forecasting and started using pvlib-python for his work.

- Principal maintainer from 2014-2018.
- Instituted Python coding standards.
- Added classes (PVSystem, ModelChain) for higher level functionality
- Added continuous integration testing
- Recruited and coached new maintainers

The Renaissance (2019-2021)

- New content added as a byproduct of other work.

Examples:

- SolarForecastArbiter – rework of many function APIs, publication of the primary reference [1]
- SolarPerformanceInsight – multiple Array capabilities

The Modern Era (present)

- Team of 6 maintainers (5 institutions)
- NumFocus affiliation
    - GSoC 2021: build-out of iotools
- Ongoing PVPMC project – 2D bifacial model, parameter translation and estimation capabilities.

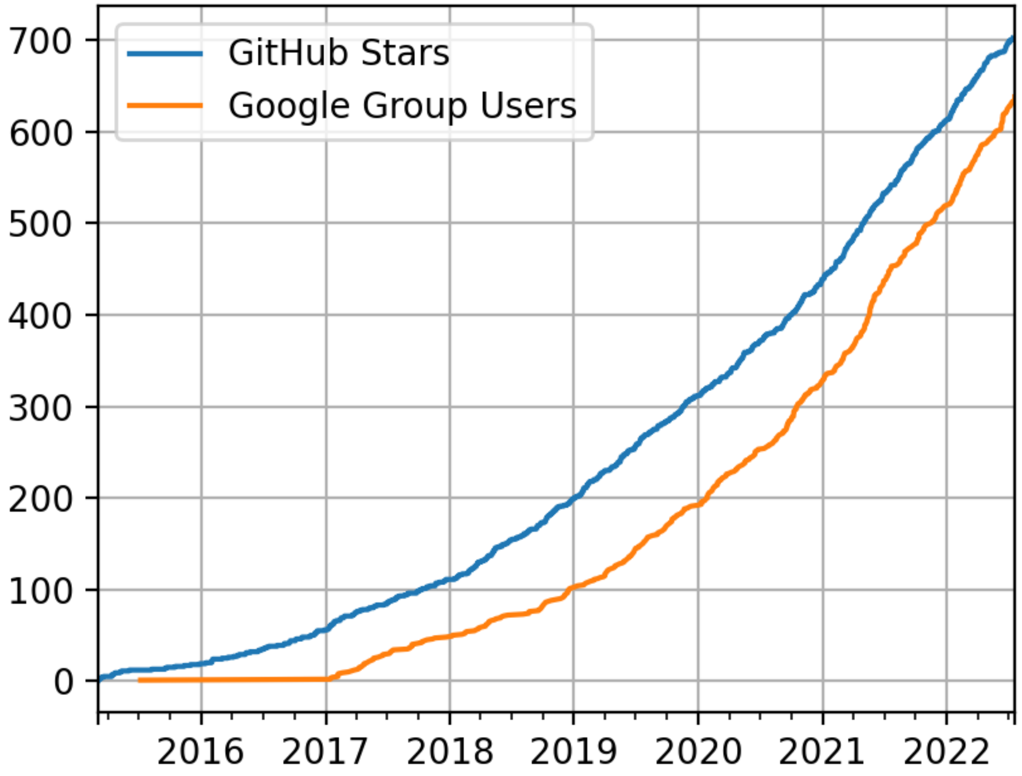
NUMFOCUS [AFFILIATED PROJECT]

1. William F. Holmgren, Clifford W. Hansen, and Mark A. Mikofski. "pvlib python: a python package for modeling solar energy systems." Journal of Open Source Software, 3(29), 884, (2018). https://doi.org/10.21105/joss.00884

Community Growth

Google Group (user discussion, announcements)
- 600+ members
- Roughly quadrupled since 2019 workshop
- https://groups.google.com/g/pvlib-python

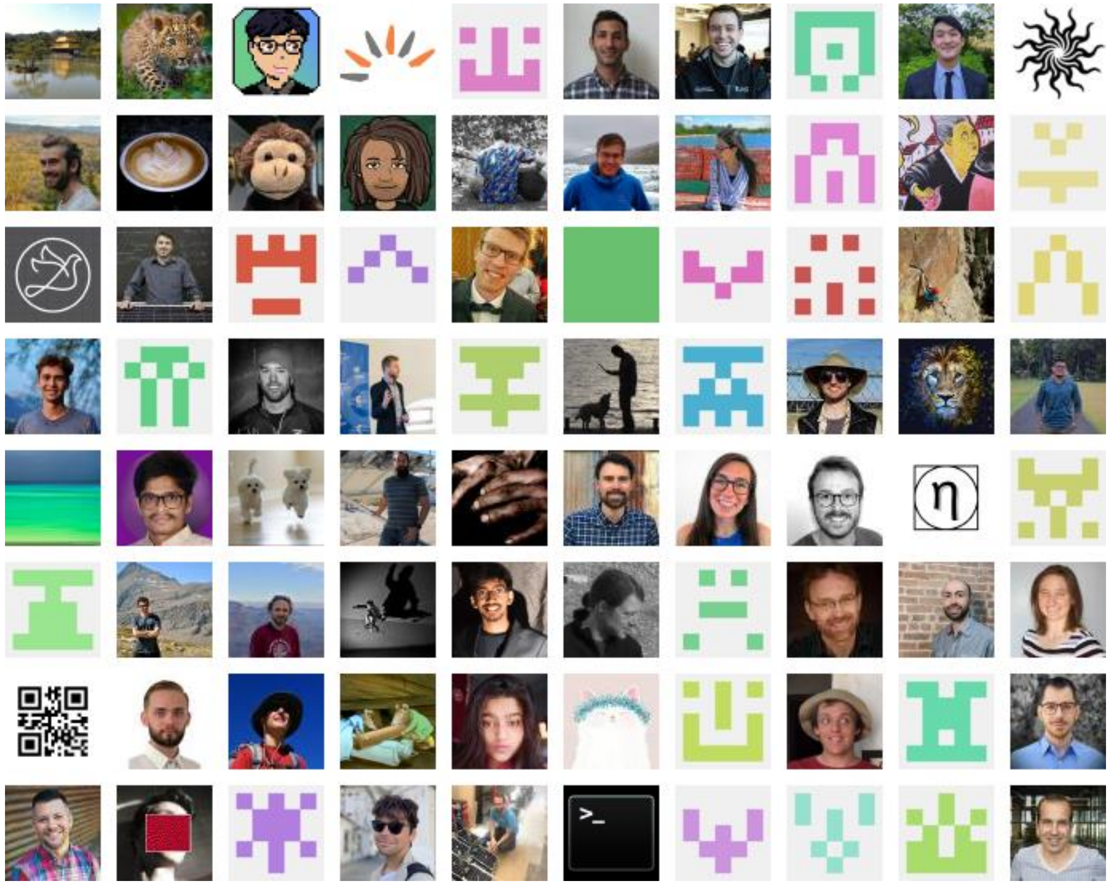
GitHub (code development)
- 700+ pull requests
- Code contributions from 80+ people
- https://github.com/pvlib/pvlib-python

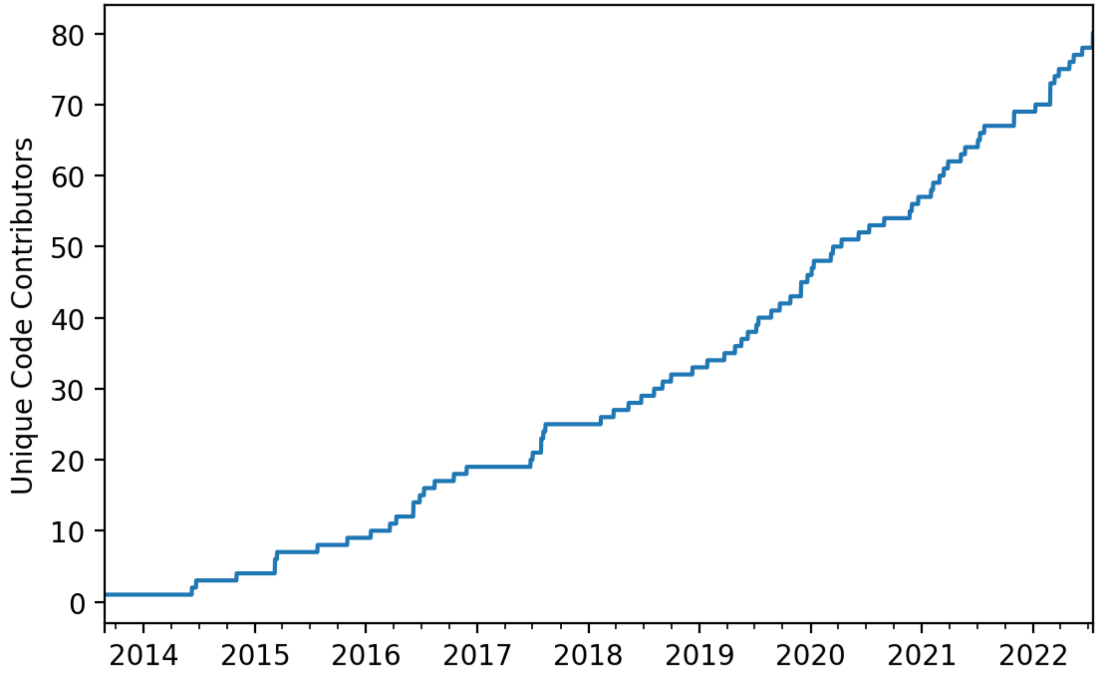


Numbers as of 2022-07-20

From: K. Anderson et al., pvlib python 2022 update, PV Performance Modeling and Monitoring Workshop, Salt Lake City, UT. August 23, 2022.

GitHub Contributors





*Not all contributions are code!

This software is made possible by contributions from people like you. <u>You can help!</u>

https://pvlib-python.readthedocs.io/en/stable/contributing.html

From: K. Anderson et al., pvlib python 2022 update, PV Performance Modeling and Monitoring Workshop, Salt Lake City, UT. August 23, 2022.

Known commercial users of pvlib

pvlib

DNV

BLACK & VEATCH
100% EMPLOYEE-OWNED

POWER FACTORS

EDF renewables

ERTHOS

GREAT SNIPE ANALYTICS

Clean Power Research®

Gantner instruments

Southern Company

pvlib citations increase every year (https://pvpmc.sandia.gov/applications/pv_lib-toolbox/)

- 223 in 2021
- 193 in 2020
- 160 in 2019
- 110 in 2018
- 43 in 2017
- 27 in 2016
- 8 in 2015
- 6 in 2014
- 1 in 2013

Principles for pvlib-python's success

- Community property, not of any institution.
- Re-use is prioritized. "pv-lib" stands for "PV library."
    - Base layer of functions.
    - Higher layers of Class methods automate frequent tasks, e.g., ModelChain.
    - Free from any particular workflow (there is no GUI).
- Transparency and credibility:
    - Algorithms and models closely follow published references.
    - Open review process, full-coverage testing, compliance with python conventions, automated documentation.
- Sustained, frequent activity by maintainers.
    - github.com/pvlib/pvlib-python.git is not a "ghost town."
- A diverse team of committed maintainers.

Open-source is not simply publicly-shared

While the software is free to use, it is not free to:

<u>Sustain</u>

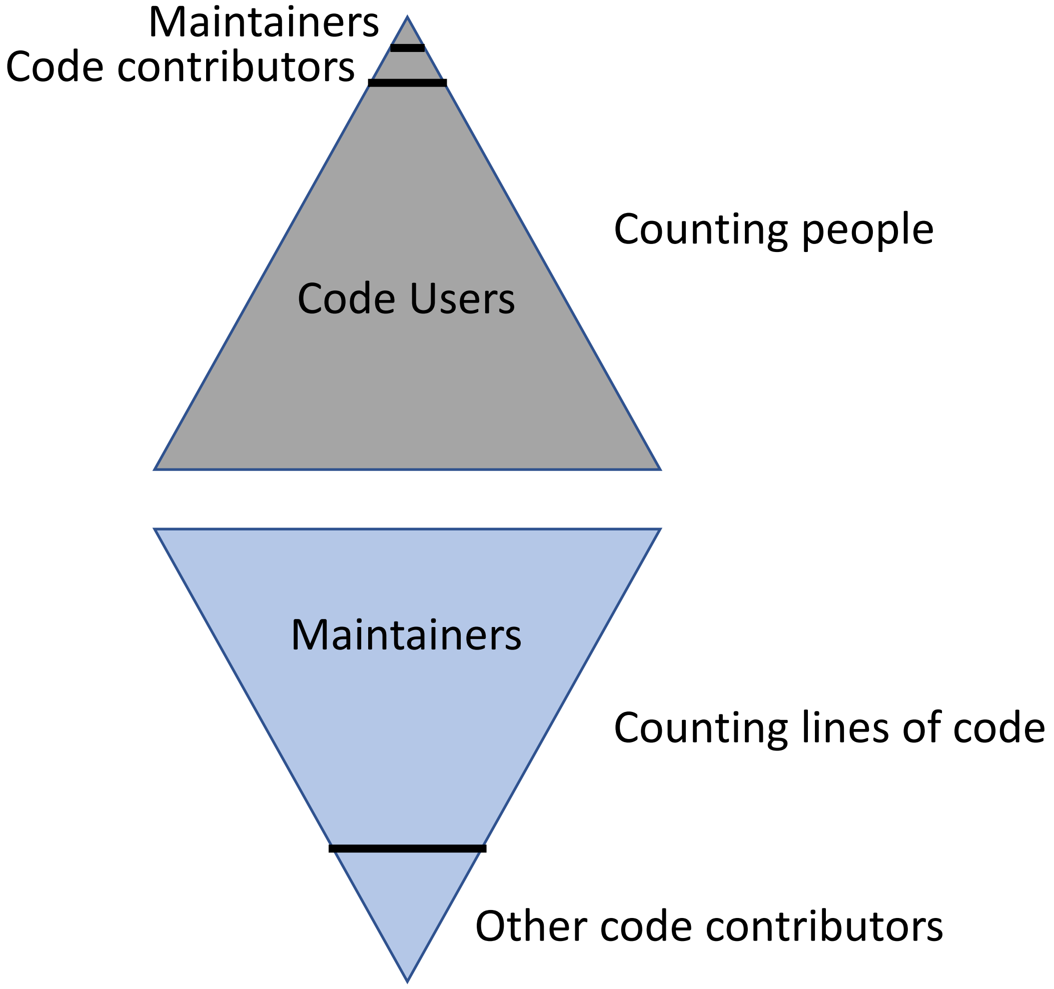
- Keep software running as python and package versions change
- Engage community – online Q&A, user group meetings, tutorials

<u>Maintain</u>

- Manage Issues, pull requests, code reviews, tracking engagement

<u>Develop</u>

- Add capabilities (bifacial modeling, shade/snow losses, interfaces with other tools/data)
- Improvements to speed, adapt to new hardware innovations



- *SETO provides some support for pvlib through the PV Performance Modeling Collaborative project at Sandia. Other maintainers are self supported and volunteer their time to the project.*

Some Benefits of Open Source pvlib-python

pvlib

- Great way to find bugs and get them fixed
- People who you do not know contribute cool things.
- Your community grows
- Provides diverse opportunities for new development and leadership
- Builds international networks

pvlib
PyData

Sascha Birk, PhD Student from University of Applied Science Cologne has a series of at least nine YouTube videos on using pvlib

NUMFOCUS
[AFFILIATED PROJECT]

pvFactors

Google
Summer of Code
@ pvlib

pvilb-python is used all over the world

PVLib Introducción con Python - YouTube

PVLib Introducción con **Python**. 498 views Mar 28, 2022 Taller impartido para el curso EL5854 Sistemas Fotovoltaicos de la...

YouTube · Hugo Andrés Sánchez Ortiz · Mar 28, 2022