pvlib: a snapshot

Rodrigo Amaro e Silva

rasilva@fc.ul.pt

18/11/2025

Intro

- **pvlib** is a Python library, which is free and is regularly updated

- It is not a PV model *per se*, but more of a repository of algorithms that aims to make the job of a PV modeller easier

Intro

It is a fully open-source, designed and developed by volunteers.

Open to feedback & contributions:

https://github.com/pvlib/pvlib-python
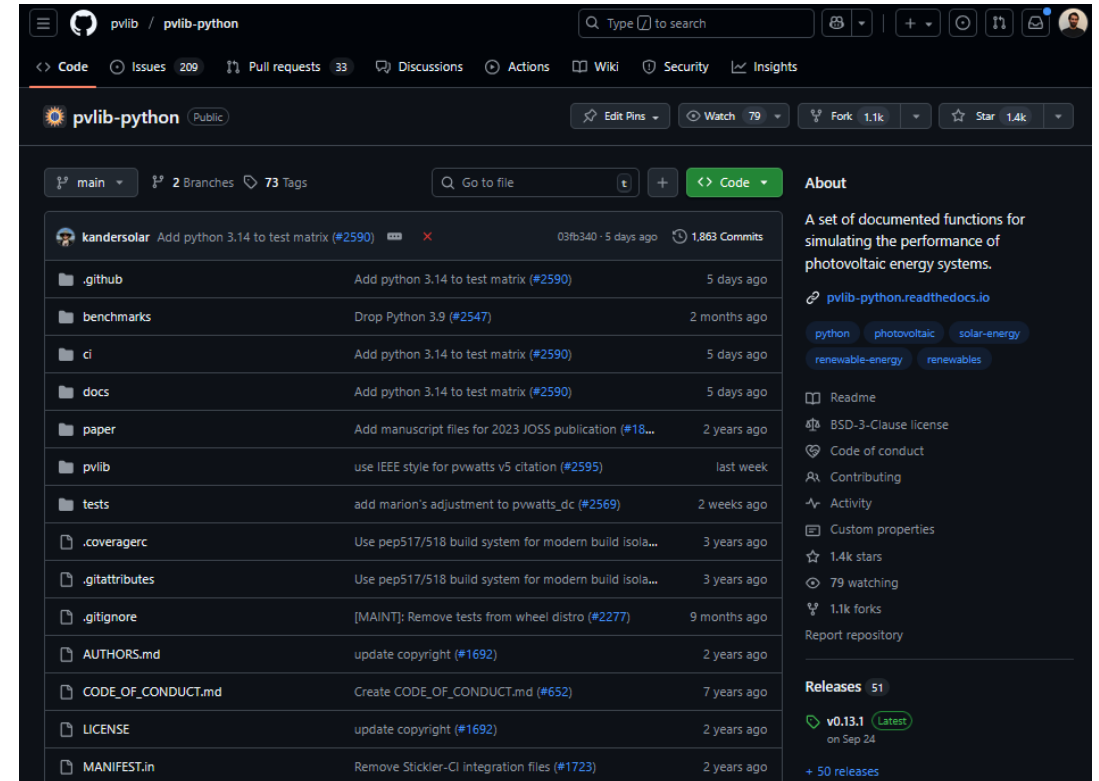
pvlib main functionalities

- **PV modelchains**, to quickly implement full modelling pipelines

- **Individual algorithms**, for a customized modelling of each step of the pipeline

- **APIs**, to easily access weather and PVGIS data

First step: designing a case study

pvlib.location.Location

```
class pvlib.location.Location(latitude, longitude, tz='UTC', altitude=None, name=None)
```

(define where your project is located)

pvlib.pvsystem.PVSystem
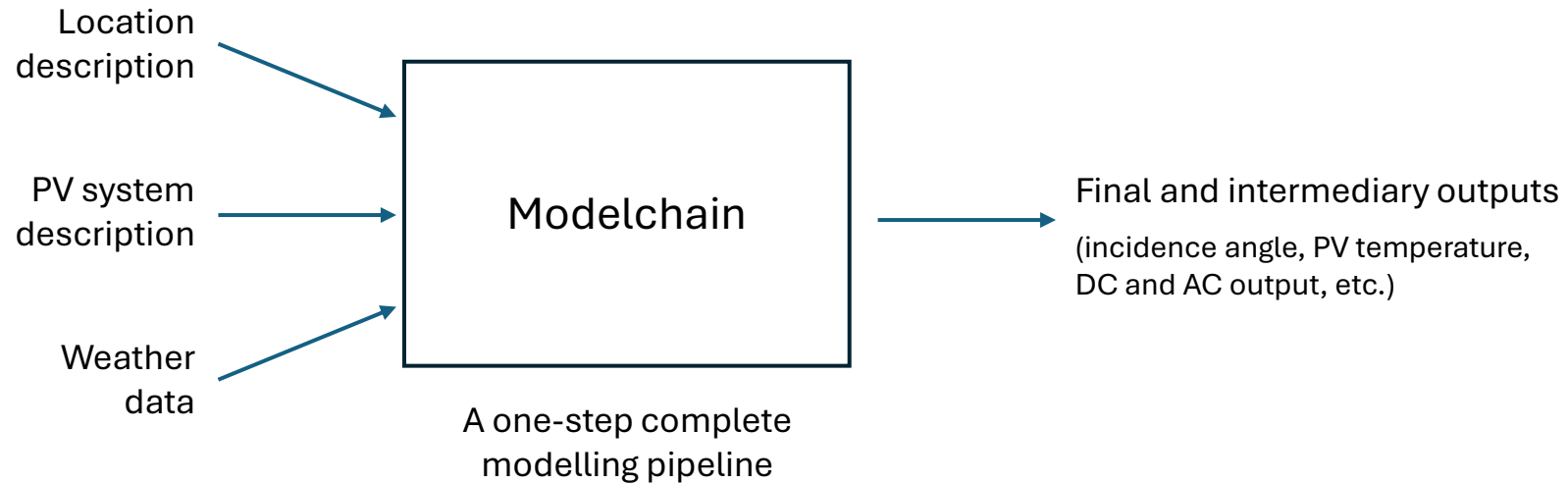
```
class pvlib.pvsystem.PVSystem(arrays=None, surface_tilt=0, surface_azimuth=180, albedo=None,
surface_type=None, module=None, module_type=None, module_parameters=None,
temperature_model_parameters=None, modules_per_string=1, strings_per_inverter=1, inverter=None,
inverter_parameters=None, racking_model=None, losses_parameters=None, name=None)
```

(define your PV project)

Can do it in more detail for diode models,
or with less parameters for PVWATTS

Whenever in a document you see "parameter=something" the something is the default value assumed
by the function when no value is provided by the user (e.g. if you don't define a surface_tilt, it will be 0 by default)

Modelchain: in a nutshell

Location description →

PV system description →

Weather data →

Modelchain

A one-step complete modelling pipeline

→ Final and intermediary outputs

(incidence angle, PV temperature, DC and AC output, etc.)

Modelchain: setting up

pvlib.modelchain.ModelChain

```
class pvlib.modelchain.ModelChain(system, location, clearsky_model='ineichen',
transposition_model='haydavies', solar_position_method='nrel_numpy', airmass_model='kastenyoung1989',
dc_model=None, ac_model=None, aoi_model=None, spectral_model=None, temperature_model=None,
dc_ohmic_model='no_loss', losses_model='no_loss', name=None)                          [source
```

The modelchain defines the algorithm used in each modelling step.
As you can see, there is a default configuration already set up (but unlikely to be the best)

Modelchain: running

pvlib.modelchain.ModelChain.run_model

`ModelChain.run_model(weather)`

(only requires weather data, assumes you have horizontal irradiance data)

pvlib.modelchain.ModelChain.run_model_from_poa

`ModelChain.run_model_from_poa(data)`

(same as above, for you when have plane-of-array irradiance data)

Individual algorithms: examples

Doing step by step, with more control/customization. Examples below:

pvlib.iam.martin_ruiz

(example of optical loss algorithm,
pvlib.iam contains other options)

pvlib.irradiance.perez

(example of radiation transposition algorithm,
pvlib.irradiance contains other options)

pvlib.temperature.sapm_cell

(example of PV temperature algorithm,
pvlib.temperature contains other options)

APIs: pvlib as a data facilitator

Access to several databases, examples below:

pvlib.iotools.get_cams   (solar radiation estimated
                                        by satellite, from CAMS)

pvlib.iotools.get_pvgis_tmy   veather data for a Typical
                                       meteorological year, from PVGIS)

pvlib.iotools.get_pvgis_hourly   (PV and weather data, from PVGIS)

And many others: https://pvlib-python.readthedocs.io/en/latest/reference/iotools.html
                Jensen et al. (2023)

kaggle and notebooks shared

Kaggle allows you to access/run notebooks without having to install anything (simply create an account in https://www.kaggle.com/)

- Notebook part1: getting weather data + implementing modelchain

- Notebook part2: step-by-step modelling (without chain)

You can run your own pvlib project in Kaggle if you want (would just suggest that you do it outside the teaching notebooks – to keep them as they are)

- Teaching notebooks are meant as reference for the logic and coding behind pvlib

kaggle : one configuration step I forgot

From the start, once you have a Kaggle account you are free to access, create, edit, and run Kaggle notebooks.

However, to be able to install new packages in the environment like pvlib, there is one extra configuration step:

> Once you have access to the notebooks, you will have to verify your identity once and ensure the notebook is connected to the internet (more details here).
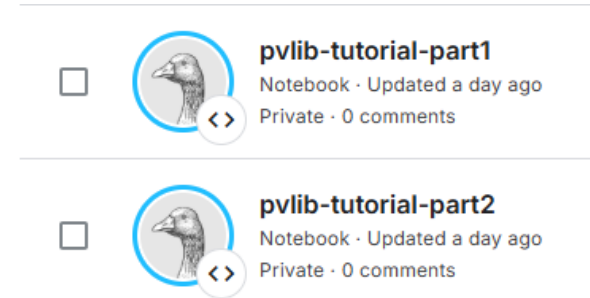
kaggle : how to use

Once you login, go to the "Code" page.
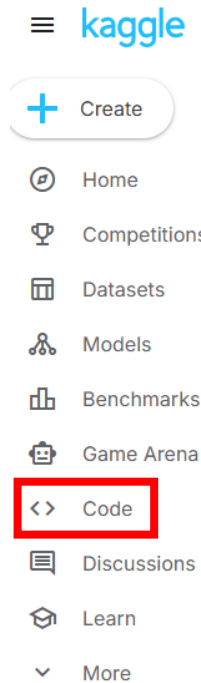
Then, "Your Work".

You should see two notebooks.
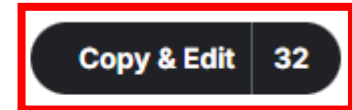
kaggle : how to use

Because I only gave you "viewing permission, you cannot direclty edit it.

You will have to create a local, editable copy of your own.

Author · 5MO AGO · 175 VIEWS · PRIVATE 30 Copy & Edit 32

Notebook title
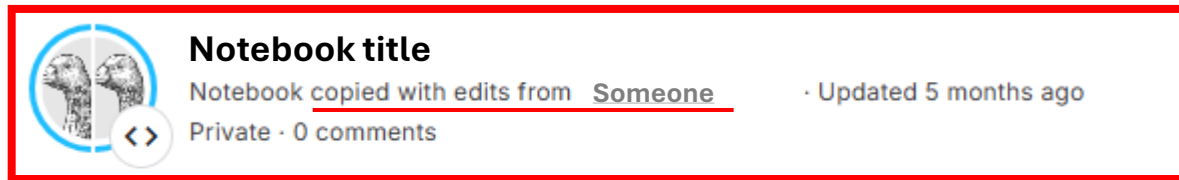
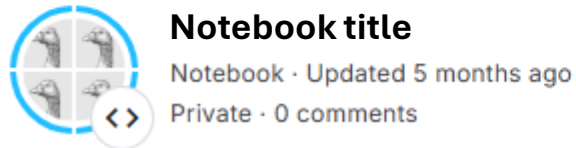Notebook   Input   Output   Logs   Comments (0)

kaggle : how to use

Because I only gave you "viewing permission, you cannot direclty edit it.

You will have to create a local, editable copy of your own.

Your local copy of the original notebook

After having your own local copy, you're free to do whatever you want.

Suggestion: keep these copies for playing with the teaching materials, and create a separate notebook for your optional PV project.