- make sure you have installed all the required libraries (requirements.txt)

- open the file in any IDE (Ex: PyCharm) and run the file

- A Graphical User Interface (GUI) window appears for inputs

- Use the example inputs for reference in the images provided

Contents:

Location:

- Latitude
- Longitude
- Time zone
- Plant name

Weather:

- Weather data file
- GHI column
- DHI column
- DNI column
- Temperature column
- Wind speed column
- Start date

System:

- Plant capacity
- Module's mounting parameters
- Surface tilt
- Surface azimuth
- Albedo
- Losses

pvlib python is a community developed toolbox that provides a set of functions and classes for simulating the performance of photovoltaic energy systems and accomplishing related tasks. (https://pvlib-python.readthedocs.io)

This Graphical User Interface helps to generate an annual hourly feed-in timeseries profile with normalised DC and AC power for a location with weather data using PVlib python. The output can be used to model OEMOF simulation.

User Guide PVlib GUI

Location:

Latitude: Latitude of the location of your plant

Longitude: Longitude of the location of your plant

Time zone: Specify the time zone of your location

Plant name: Name of your plant, can be anything

Weather data:

Weather data file: browse and upload

- hourly weather data in .csv format must contain temperature, Global Horizontal Irradiance, Diffuse Horizontal Irradiance, Direct Normal Irradiance, wind speed data for every hour in the year
- See example.csv file for reference

(make sure that the data must be cleaned and no null values are present)

GHI column: Column name for the data containing values for Global Horizontal Irradiance

DHI column: Column name for the data containing values for Diffuse Horizontal Irradiance

DNI column: Column name for the data containing values for Direct Normal Irradiance

Temperature column: Column name for the data containing values for temperature in C

Wind speed column: Column name for the data containing values for wind speed in m/sec

Start date: Start date of your data, must be specified in the same format

While entering the column names, please make sure you enter the names as in the dataset.

System:

Plant capacity: The plant capacity in KW

Module's mounting parameters: choose the mounting parameters from the available list

Surface Tilt: Specify the tilt angle for your system in degrees

Surface Azimuth: Specify the azimuth angle of your system (Ex: 180 is for south direction)

Albedo: Ground reflectance, also known as the reflection coefficient, More on albedo for different surfaces can be found in the following link.

https://www.researchgate.net/figure/Typical-albedo-values-of-different-kind-of-surfaces-3_tbl1_275956502

Losses: The losses model using which the losses of the system can be calculated.

Only PV watts model is available and it takes the default values which can be found here, https://pvlib-python.readthedocs.io/en/stable/reference/generated/pvlib.pvsystem.pvwatts_losses.html#pvlib.pvsystem.pvwatts_losses

Then, press the load “Upload inputs” so that your inputs are fed to program.

After verifying your inputs in the text area, press the “Simulate” button.

After you close the plot, the normalised output of your simulation is automatically saved into the current working directory, where your current project is running.

Please note:

- Only these inputs can be given by the user and the rest were set to defaults for valid reasons.
- The number of panels are calculated and rounded to lower integer.
- The module and inverter can be changed in the program.
- The number of strings is limited to 1 for ease of simulation.