PVFree!
NREL-SAM Component Database, PVLIB API, and Interactive PV Modeling Tutorial
Mark Mikofski
PVPMC-12: Open Source Modeling Project Collaboration
May 15, 2019
Outline

• Motivation
• Component Database
  • Human browsable web pages
  • API usage, filters
• Pvlib Python API
• Interactive PV Modelling Tutorial
• Hardware, Software Setup
• PR Puppies
Motivation

- Many different sources of modelling parameters, different formats
- Data updates difficult, no data validation, repeats, missing data
- No modelling parameter API, data passed around in spreadsheets, need to keep track of versions
- Wanted to make a pvlib demo site
- Wanted to make interactive tutorial
### NREL-SAM Component Database

- **CEC Inverters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Vac</th>
<th>Paco</th>
<th>Vdco</th>
<th>Pdco</th>
<th>C0</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>Pso</th>
<th>Pnt</th>
<th>Vdcmx</th>
<th>Idcmx</th>
<th>MPPT low</th>
<th>MPPT high</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV Powered: PVP100 kW-206 (208V) 208V [CEC 2018]</td>
<td>208.0</td>
<td>100000.0</td>
<td>341</td>
<td>105191</td>
<td>-2.74E-07</td>
<td>3.50E-05</td>
<td>2.29E-03</td>
<td>-5.66E-04</td>
<td>445.7</td>
<td>42.0</td>
<td>480.0</td>
<td>308.47</td>
<td>295.0</td>
<td>480.0</td>
</tr>
<tr>
<td>PV Powered: PVP100 kW-480 (480V) 480V [CEC 2018]</td>
<td>480.0</td>
<td>100000.0</td>
<td>341</td>
<td>104680</td>
<td>-2.90E-07</td>
<td>5.27E-05</td>
<td>2.49E-03</td>
<td>2.29E-04</td>
<td>476.5</td>
<td>36.0</td>
<td>480.0</td>
<td>306.97</td>
<td>295.0</td>
<td>480.0</td>
</tr>
<tr>
<td>PV Powered: PVP100 kW-206 208V [CEC 2018]</td>
<td>208.0</td>
<td>100000.0</td>
<td>341</td>
<td>105191</td>
<td>-2.74E-07</td>
<td>3.50E-05</td>
<td>2.29E-03</td>
<td>-5.66E-04</td>
<td>445.7</td>
<td>42.0</td>
<td>480.0</td>
<td>0.208333</td>
<td>295.0</td>
<td>480.0</td>
</tr>
<tr>
<td>PV Powered: PVP100 kW-480 480V [CEC 2018]</td>
<td>480.0</td>
<td>100000.0</td>
<td>341</td>
<td>104680</td>
<td>-2.90E-07</td>
<td>5.27E-05</td>
<td>2.49E-03</td>
<td>2.29E-04</td>
<td>476.5</td>
<td>36.0</td>
<td>480.0</td>
<td>0.208333</td>
<td>295.0</td>
<td>480.0</td>
</tr>
<tr>
<td>PV Powered: PVP100 kW-600 600V [CEC 2012]</td>
<td>600.0</td>
<td>100000.0</td>
<td>341.7</td>
<td>104547.2</td>
<td>-2.33E-07</td>
<td>3.85E-05</td>
<td>2.34E-03</td>
<td>2.91E-04</td>
<td>536.0</td>
<td>99.56</td>
<td>600.0</td>
<td>356.0</td>
<td>295.0</td>
<td>595.0</td>
</tr>
<tr>
<td>PV Powered: PVP100 kW-600 600V [CEC 2018]</td>
<td>600.0</td>
<td>100000.0</td>
<td>341</td>
<td>104450</td>
<td>-2.20E-07</td>
<td>4.07E-05</td>
<td>2.56E-03</td>
<td>7.51E-04</td>
<td>517.0</td>
<td>99.56</td>
<td>480.0</td>
<td>306.306</td>
<td>295.0</td>
<td>480.0</td>
</tr>
<tr>
<td>PV Powered: PVP100 kW-600 600V [CEC 2018]</td>
<td>120.0</td>
<td>1100.0</td>
<td>182</td>
<td>1194.1</td>
<td>-2.95E-05</td>
<td>5.71E-05</td>
<td>2.08E-03</td>
<td>6.23E-04</td>
<td>22.1</td>
<td>3.6</td>
<td>380.0</td>
<td>6.5696</td>
<td>100.0</td>
<td>380.0</td>
</tr>
</tbody>
</table>
NREL-SAM Component Database

- CEC Inverters
- CEC Modules
NREL-SAM Component Database

- CEC Inverters
- CEC Modules
- Sandia Modules
# NREL-SAM Component Database

- CEC Inverters
- CEC Modules
- Sandia Modules
- Searchable
- Sortable
- Human Browsable

<table>
<thead>
<tr>
<th>Pso</th>
<th>Pnt</th>
<th>Vdcmax</th>
<th>Idcmax</th>
<th>MPPT low</th>
<th>MPPT high</th>
</tr>
</thead>
<tbody>
<tr>
<td>66E-04</td>
<td>445.7</td>
<td>42.0</td>
<td>480.0</td>
<td>308.477</td>
<td>295.0</td>
</tr>
</tbody>
</table>
NREL-SAM Component Database

- CEC Inverters
- CEC Modules
- Sandia Modules
- Searchable
- Sortable
- Human Browsable
- Details, Plots
API Usage, filters

```python
In [1]: import requests

In [2]: r = requests.get('https://pyfree.herokuapp.com/api/v1/pvinverter/?Name__istartswith=KACO&Paco__gt=2030&Paco__lt=2050')

In [3]: r
Out[3]: <Response [200]>

In [4]: r.json()
Out[4]:
{
    'meta': {
        'limit': 20,
        'next': None,
        'offset': 0,
        'previous': None,
        'total_count': 2},
    'objects': [{
        'C0': 1e-05,
        'C1': 3.62e-05,
        'C2': 7.54e-05,
        'C3': -0.00148,
        'Idcmax': 11.0,
        'Mppp_high': 518.0,
        'Mppp_low': 190.0,
        'Name': 'KACO: blueplanet 2.0 TL1 M1 WM OD U53x (280V) 288V [CEC 2015]',
        'Paco': 2040.0,
        'Pdco': 2169.374224,
        'Pnt': 4.67,
        'Pso': 20.54972872,
        'Vac': 200.0,
        'Vdcmax': 600.0,
        'Vdco': 389.8024,
    }
```
Pvlib Python API

- WIP currently exposes: solarposition, linke-turbidity, and airmass
Interactive PV Tutorial

- Single page app
- Steps user through modelling

PVLIB API
Software for simulating photovoltaic solar energy systems.

PVlib
This is a tutorial and API to the pvlib python library.

PV Modeling Steps
- Solar Position
  It's important to know where the sun is.
- Linke Turbidity
  Aerosols and dust in the atmosphere can affect PV power more than you think.
- Air Mass
  The further light travels through the air, the more its color changes.

Solar Position
Interactive PV Tutorial

- Single page app
- Steps user through modelling
- Solar Position
Interactive PV Tutorial

- Single page app
- Steps user through modelling
- Solar Position
- Generates code and warnings

Example
Try the following: `/api/v1/pvlib/solarposition/?lat=36&lon=-122&start=2018-01-01 7:00&end=2018-01-01 8:00&freq=1Hz&tz=-8`

Latitude: 37.8520851975281
Longitude: -122.26959285156
Start Timestamp: 2018-01-01 7:00
End Timestamp: 2018-01-01 8:00
Frequency: 1Hz
Timezone: -8

Submit

Click on the map and enter the dates and frequency fields above, then click submit to see what your request looks like, a plot of azimuth vs. zenith, and a table of solar positions below:

```
{"start": "This field is required.", "end": "This field is required."}
```

Example
Try the following: `/api/v1/pvlib/solarposition/?lat=36&lon=-122&start=2018-01-01 7:00&end=2018-01-01 8:00&freq=1Hz&tz=-8`

Latitude: 37.890280013712
Longitude: -122.174833209078
Start Timestamp: 2019-01-01
End Timestamp: 2019-01-02
Frequency: 1Hz
Timezone: -8

Submit

Click on the map and enter the dates and frequency fields above, then click submit to see what your request looks like, a plot of azimuth vs. zenith, and a table of solar positions below.
Interactive PV Tutorial

- Single page app
- Steps user through modelling
- Solar Position
- Generates code and warnings
- Outputs data and plots
Hardware, Software Setup

- Python
- Pvlib Python
- Django, Tastypie
- Bokeh, Altair
- Bootstrap
- GitHub
- Heroku
- ElephantSQL
Thank You

https://pvfree.herokuapp.com