

---

# **openpvtools Documentation**

**openpvtools contributors**

**Sep 12, 2023**



---

## Contents

---

<b>1</b>	<b>Open Source PV Modeling Tools</b>	<b>3</b>
<b>2</b>	<b>Funding</b>	<b>7</b>
<b>3</b>	<b>Community</b>	<b>9</b>
<b>4</b>	<b>Additional Challenges and Opportunities</b>	<b>11</b>
<b>5</b>	<b>Conclusions</b>	<b>13</b>
<b>6</b>	<b>References</b>	<b>15</b>
<b>7</b>	<b>Open Source Tools for PV Modeling</b>	<b>17</b>



Open source software is code that is publicly available and free to use according to the terms of a license [1]. Black Duck Software estimates that 65% of companies used open source software in 2016 [2]. The open source PV modeling community continues grow, as well. Therefore, it is important to evaluate the benefits and challenges of open source software in the solar industry. The purpose of this website is to:

1. Promote the use of open source software in the PV modeling community,
2. Summarize the state of open source for PV modeling, and
3. Stimulate discussion of how to support open source PV modeling projects in the future.

Open source software can be beneficial for a number of reasons. It encourages reproducibility and replicability in scientific studies [3]. Open source analyses complement scientific publications by encouraging transparency and collaboration, e.g. [4, 5]. According to GitHub, open source software promotes collaboration, adoption, and transparency [6]. Google states that open source is good engineering economics in its open source guide [7]. In the PV industry, open source software may reduce project financing soft costs because algorithms and implementations are transparent and have been vetted by users, so they are inherently less risky. Open source software also has a number challenges [8]. These challenges include funding, community, version control, programming expertise, quantifying impact, and professional credit. We will discuss these challenges in the context of PV modeling.



---

## Open Source PV Modeling Tools

---

The table below shows a summary of the open source PV modeling tools known to the authors as of this writing. Here, we briefly describe each tool.

PVLib Matlab is a general purpose PV modeling toolkit for the Matlab platform. It is primarily developed at Sandia National Laboratories, but contains contributions from members of the PV Performance Modeling Collaborative (PVPMC; see [www.pvpmc.org](http://www.pvpmc.org)). PVLib Matlab includes extensive documentation and extensive examples.

PVLib Python is a general purpose PV modeling toolkit written in Python. PVLib Python originated as a port of PVLib Matlab and contains many of the Matlab project's features. The PVLib Python and Matlab projects are developed independently, but some developers contribute to both projects. PVLib Python is a dependency of other open source tools. PVLib Python includes extensive function-level documentation, and a collection of examples in its online documentation and Jupyter notebooks.

System Advisory Model is a popular desktop application for PV, CSP, wind, and financial modeling. It is developed at the National Renewable Energy Laboratory. SAM has a MIT license for commercial entities and a GPL 3 license for national labs, universities, and non-profits. The SAM Simulation Core (SSC) is the computational library for SAM. NREL has also developed the SAM SDK for writing scripts to interact with SAM and the SSC.

rdtools is a library for PV degradation analysis. It is developed at NREL and written in Python. It includes a Jupyter notebook with an example analysis. rdtools uses PVLib Python.

PVFree is a web application and API for obtaining PV modeling parameters.

SolarUtils is a package that contains Python wrappers for NREL's solar position and spectral decomposition packages (written in C). It was developed by SunPower.

Pecos is a performance monitoring tool originally developed for PV but with applications to other systems. Pecos includes online documentation and examples. Pecos uses PVLib Python. It is developed at Sandia and written in Python.

SolPy is a general purpose PV modeling tool.

PVMismatch is a tool for IV curves for modules with heterogeneous cells. It is developed at SunPower and written in Python.

photovoltaic is a general purpose PV modeling package. It is developed at Arizona State University for educational use. It is written in Python.

feedinlib is a tool for PV time series modeling. It is part of the Open Energy Modelling Framework.

CASSYS is a simulation tool for grid-tied PV systems. CASSYS is developed by Canadian Solar O&M Inc. The user interface is a Microsoft Excel spreadsheet, and the calculations are run by a program. CASSYS includes a user guide, a model-description document (useful for general PV modeling), and a comparison to PVSyst document.

Table 1: Table of Open Source PV Tools. A \* denotes that the project is under active development.

Name	Purpose	Years developed	Documentation Website	Development Website	Primary Language	License
PVLib Matlab	General purpose PV modeling	2012? - *	<a href="https://pvpmc.sandia.gov">https://pvpmc.sandia.gov</a>	<a href="https://github.com/sandialabs/MATLAB_PV_LIB">https://github.com/sandialabs/MATLAB_PV_LIB</a>	Matlab	BSD 3
PVLib Python	General purpose PV modeling	2013 - *	<a href="https://pvlib-python.readthedocs.io">https://pvlib-python.readthedocs.io</a>	<a href="https://github.com/pvlib/pvlib-python">https://github.com/pvlib/pvlib-python</a>	Python	BSD 3
System Advisor Model (SAM)	Desktop app for PV, wind, CSP modeling, financial	2004 - *	<a href="https://sam.nrel.gov">https://sam.nrel.gov</a>	<a href="https://github.com/NREL/SAM">https://github.com/NREL/SAM</a>	C++	Mixed MIT/GPL 3
ssc	Compute modules for SAM	2004 - *	<a href="https://sam.nrel.gov">https://sam.nrel.gov</a>	<a href="https://github.com/nrel/ssc">https://github.com/nrel/ssc</a>	C, C++	Mixed MIT/GPL 3
RdTools	Technical analysis of PV time-series data	2016 - *	<a href="https://rdtools.readthedocs.io">https://rdtools.readthedocs.io</a>	<a href="https://github.com/NREL/rdtools">https://github.com/NREL/rdtools</a>	Python	MIT
PVFree	API for obtaining PV modeling parameters	2015 - *	<a href="https://pvfree.azurewebsites.net">https://pvfree.azurewebsites.net</a>	<a href="https://github.com/BreakingBytes/pvfree">https://github.com/BreakingBytes/pvfree</a>	Python	BSD 2
SolarUtils	Python wrappers of C solar position and spectral decomposition	2016	<a href="https://sunpower.github.io/SolarUtils/">https://sunpower.github.io/SolarUtils/</a>	<a href="https://github.com/SunPower/SolarUtils">https://github.com/SunPower/SolarUtils</a>	Python	BSD 3
Pecos	Performance monitoring	2016 - *	<a href="https://pecos.readthedocs.io">https://pecos.readthedocs.io</a>	<a href="https://github.com/sandialabs/pecos">https://github.com/sandialabs/pecos</a>	Python	BSD 3
Solpy	General purpose PV modeling	2011-2015	<a href="https://solpy.readthedocs.io">https://solpy.readthedocs.io</a>	<a href="https://github.com/nrcharles/solpy">https://github.com/nrcharles/solpy</a>	Python	LGPL 2.1
PVMismatch	IV curve	2012 - *	<a href="https://sunpower.github.io/">https://sunpower.github.io/</a>	<a href="https://github.com/SunPower/">https://github.com/SunPower/</a>	Python	BSD 3



Funding for open source PV modeling software may be categorized by funding source (public or private) and by funding kind (direct or indirect). Here, we define direct funding as resources that are specifically allocated for developers to support or create an open source tool. In contrast, we define indirect funding as resources that are used to support or create an open source tool in the process of achieving a larger goal. Next, we provide examples of each combination of funding source and kind. Public/Direct: DOE EERE supports development and maintenance of PVLib Matlab and SAM. Public/Indirect: EERE initially funded the PVPMC and its workshop, which features open source tools and thus indirectly supports open source PV modeling tools. Private/Direct: Southern Company Services and the Electric Power Research Institute funded UA developers to contribute a forecast module to PVLib Python. Private/Indirect: Developers at companies such as SunPower and First Solar have used PVLib to conduct research that furthers their businesses. Developer time spent reporting issues or contributing to the software is an indirect funding source.

A challenge for open source software is that direct funding for it is inconsistent. This is especially true for funding that goes towards maintaining existing functionality rather than adding new features. Indirect funding of open source PV software has enabled packages such as PVLib Python to receive minor, regular updates. Whether or not indirect funding is sufficient for the long-term health of a project is an open question.

Funding agencies demand to know the impact of their awards, however, it can be difficult to accurately quantify the impact of an open source software project. This is especially true if the project is used by or incorporated into additional projects. The developers of many of the tools described here have made efforts towards cataloging their impacts but are often able to report only statistics indicating interest, such as numbers of downloads and webpage views. To our knowledge, there is no comprehensive public repository with this information for open source PV tools.

Most public funding for open source PV modeling software has been through the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy. EERE proposals often require data management plans that describe how a project's data, including software code, will be disseminated. EERE's data management plan instructions suggest open source and websites such as GitHub as acceptable methods to disseminate results. Funding agencies could provide further indirect support for open source software by explicitly encouraging open source by-products of funding.

Some users of open source PV tools may be interested in providing relatively small amounts of funding to sustain or improve the tools that they use. However, many of the lead developers of the projects discussed here are employed by government labs, universities, or large companies. Contracting with large institutions can be difficult, and this effectively creates a minimum contract value that is often more than a single user can contribute. We suggest that the open source PV tools developers explore alternatives such as spin-off companies focused on open source development, utilizing intermediaries such as the Numfocus organization or donations through a university foundation, and

micropayment methods such as Patreon.

---

### Community

---

User and developer communities ultimately drive the success or failure of an open source project. Developing these communities is a challenging task. The PVPMC has helped to establish a community that supports many of the projects discussed here. A challenge going forward is to grow the community while maintaining or improving its character.

The open source PV modeling community should recognize that professional conduct is critical to attracting potential contributors, encouraging diversity, and increasing the overall impact of a project. The challenge for open source leaders is to create systems that promote positive contributors and prevent negative contributions. The open source PV modeling community is currently small enough that many contributors know each other, and this may help to maintain the professionalism of the community. As PV grows globally, we hope that our open source communities will also grow. Additional efforts, such as implementing a code of conduct for open source projects and meetings [10], may be useful for maintaining a positive, collaborative environment as the community grows.



---

### Additional Challenges and Opportunities

---

Here we discuss miscellaneous additional challenges and opportunities for open source PV modeling. Some would-be contributors are deterred by unfamiliarity with open source tools such as version control and the GitHub website. Contributors may also be deterred by a feeling that their code is not of sufficient quality, and by a projects lack of support for helping potential contributors improve their code. The PVPMC has previously invited talks on open source tooling, and this may be an area to further develop. Inspiration may be found by looking at related fields. For example, the American Meteorological Society’s annual conference includes a Symposium on Advances in Modeling and Analysis Using Python. A similar software-focused session could be added to the PVSC or other PV-related conferences. The 2018 AMS Python Symposium included a “Show me your stack!” panel session in which the community could learn how four different developers approach problems.



## CHAPTER 5

---

### Conclusions

---

The PV modeling community is developing a set of open source tools that are growing in popularity. Outstanding challenges include identifying sustainable funding sources for project maintenance and expansion, maintaining a positive, collaborative environment as the community grows, and lowering barriers to entry.



---

## References

---

- [1] Open Source Initiative <https://opensource.org>
- [2] Black Duck Software, Inc. “The Tenth Annual Future of Open Source Survey” <https://www.blackducksoftware.com/2016-future-of-open-source>
- [3] D. C. Ince, L. Hatton, and J. Graham-Cumming, “The case for open computer programs,” *Nature* 482, p. 485, 2012.
- [4] M. Mikofski, “pvsc44-clearsky-aod” <https://github.com/mikofski/pvsc44-clearsky-aod>
- [5] M. Mikofski, et. al. “Use of Measured Aerosol Optical Depth and Precipitable Water to Model Clear Sky Irradiance,” 44th IEEE Photovoltaic Specialists Conference (PVSC), Washington, D.C., 2017.
- [6] GitHub, “Open Source Guides” <https://opensource.guide>
- [7] Google, “Google Open Source” <https://opensource.google.com>
- [8] N. Eghbal, “Roads and Bridges: The Unseen Labor Behind Our Digital Infrastructure,” Ford Foundation, 2016.
- [9] W. Holmgren, et. al., “OpenPVTools” <https://github.com/openpvtools/openpvtools>
- [10] Contributor Covenant, “A Code of Conduct for Open Source Projects,” <https://www.contributor-covenant.org>



---

## Open Source Tools for PV Modeling

---

Solar power researchers and engineers are developing a growing number of open source software tools for energy modeling. This website aims to catalog these tools.

Please see the companion paper in the Proceedings of the WCPEC-7 (2018).

Also see the companion presentation at the [10th PVPMC Workshop and 2018 Systems Symposium](#), plus additional talks on open source PV software from DOE, NREL, and Sandia. Additionally, some national labs maintain their own software indexes: [Sandia](#), [NREL](#).

We welcome your contributions to this archive! To contribute or update information on a tool, please submit a pull request to the [openpvtools GitHub](#) repository.

Table 1: Table of Open Source PV Tools. A \* denotes that the project is under active development.

Name	Purpose	Years developed	Documentation Website	Development Website	Primary Language	License
PVLib Matlab	General purpose PV modeling	2012? - *	<a href="https://pvpmc.sandia.gov">https://pvpmc.sandia.gov</a>	<a href="https://github.com/sandialabs/MATLAB_PV_LIB">https://github.com/sandialabs/MATLAB_PV_LIB</a>	Matlab	BSD 3
PVLib Python	General purpose PV modeling	2013 - *	<a href="https://pvlib-python.readthedocs.io">https://pvlib-python.readthedocs.io</a>	<a href="https://github.com/pvlib/pvlib-python">https://github.com/pvlib/pvlib-python</a>	Python	BSD 3
System Advisor Model (SAM)	Desktop app for PV, wind, CSP modeling, financial	2004 - *	<a href="https://sam.nrel.gov">https://sam.nrel.gov</a>	<a href="https://github.com/NREL/SAM">https://github.com/NREL/SAM</a>	C++	Mixed MIT/GPL 3
ssc	Compute modules for SAM	2004 - *	<a href="https://sam.nrel.gov">https://sam.nrel.gov</a>	<a href="https://github.com/nrel/ssc">https://github.com/nrel/ssc</a>	C, C++	Mixed MIT/GPL 3
RdTools	Technical analysis of PV time-series data	2016 - *	<a href="https://rdtools.readthedocs.io">https://rdtools.readthedocs.io</a>	<a href="https://github.com/NREL/rdtools">https://github.com/NREL/rdtools</a>	Python	MIT
PVFree	API for obtaining PV modeling parameters	2015 - *	<a href="https://pvfree.azurewebsites.net">https://pvfree.azurewebsites.net</a>	<a href="https://github.com/BreakingBytes/pvfree">https://github.com/BreakingBytes/pvfree</a>	Python	BSD 2
SolarUtils	Python wrappers of C solar position and spectral decomposition	2016	<a href="https://sunpower.github.io/SolarUtils/">https://sunpower.github.io/SolarUtils/</a>	<a href="https://github.com/SunPower/SolarUtils">https://github.com/SunPower/SolarUtils</a>	Python	BSD 3
Pecos	Performance monitoring	2016 - *	<a href="https://pecos.readthedocs.io">https://pecos.readthedocs.io</a>	<a href="https://github.com/sandialabs/pecos">https://github.com/sandialabs/pecos</a>	Python	BSD 3
Solpy	General purpose PV modeling	2011-2015	<a href="https://solpy.readthedocs.io">https://solpy.readthedocs.io</a>	<a href="https://github.com/nrcharles/solpy">https://github.com/nrcharles/solpy</a>	Python	LGPL 2.1
<b>18</b>			<b>Chapter 7. Open Source Tools for PV Modeling</b>			
PVMismatch	IV curve	2012 - *	<a href="https://sunpower.github.io/">https://sunpower.github.io/</a>	<a href="https://github.com/SunPower/">https://github.com/SunPower/</a>	Python	BSD 3

Please see the linked sections to the left for discussion, or click the Next button below.