

## Assessing Solar: An Interactive Best Practices Guide to Solar Resource Assessment

Adam R. Jensen<sup>1</sup>, Javier Lopez-Lorente<sup>2</sup>, Philippe Blanc<sup>3</sup>, and Yves-Marie Saint-Drenan<sup>3</sup>

<sup>1</sup> Department of Civil Engineering, Technical University of Denmark, Kgs. Lyngby, (Denmark)

<sup>2</sup> PV Technology Laboratory, FOSS Research Centre for Sustainable Energy, University of Cyprus, Nicosia (Cyprus)

<sup>3</sup> Centre Observation, Impacts, Energy, MINES ParisTech, Sophia Antipolis (France)

### Abstract

Solar energy has become a key technology in the renewable energy transition. Particularly, the decrease in the cost of solar technologies has led to increasing investments and significant penetration into energy systems. This has, in turn, led to a greater need for reliable and accurate solar resource data. Fortunately, there has been a great effort in advancing the state of the art of solar resource assessment, including enhanced solar radiation models and more accurate solar resource datasets. While these research advances are published in great detail in scientific journals, a large gap is left in the path to widespread adoption by the scientific community and industry. The high complexity of these new methods and models often means that they are, in practice, only available to a small group of experts. To that end, *Assessing Solar* presents a practical guide to solar resource assessment using the Python programming language. The guide consists of several sections on different topics, which are organized by theme. Each section provides best practices and includes several examples on how to apply the methods to real-life applications, allowing users to understand the code and, with minimal effort, modify it to fit their own applications. The content is written in the Jupyter Notebook format, allowing for an interactive experience and seamless integration of computational content, figures, equations, text, references, and more. The guide is available at [AssessingSolar.org](http://AssessingSolar.org).

*Keywords: solar energy, solar resource, Jupyter Notebook, open-source, Python.*

---

## 1. Introduction

Solar resource data are key in many environmental and energy applications, such as thermal, concentrating, and photovoltaic (PV) solar energy systems. The use of solar resource data ranges from the preliminary design stage, where it is the basis for determining the profitability of economic investments, to the operational stage, where it is used to ensure the correct performance of these systems. Notably, solar PV is at the core of decarbonization policies in the electricity sector and is predicted to become the “king of the world’s electricity markets” (International Energy Agency, 2020). Thus, with the increasing development of solar energy projects worldwide, the need for reliable and accurate solar resource data has become ever more important (Sengupta *et al.*, 2021).

Fortunately, the state of the art in solar resource assessment has been significantly advanced in recent years, including improved solar radiation algorithms, down-scaling models, and solar resource datasets with increased spatial and temporal resolution. These new and enhanced methods are usually published in great detail in international scientific journals and specialized conference proceedings. However, the high complexity of these new innovative methods and models often limits their widespread adoption, leaving them only practically available to a small group of researchers and experts. This represents a critical gap in the dissemination of best practices and state-of-the-art methods to the wider research community and industry professionals.

In this paper, we present *Assessing Solar*, a practical guide to solar resource assessment in Python. The guide aims to disseminate solar resource assessment best practices and reduce the barriers that limit users from employing the most suitable and accurate methods. The guide covers a diverse range of topics, each containing a short overview, examples for applying best practices, and recommendations from a group of experts. At the core of

each section is a practical demonstration of the concepts, which uses the Python programming language to demonstrate how the methods can be applied in practice and provide code for users to adopt for their own applications.

The guide is an outcome of the collaborative effort between participants in Task 16 of the International Energy Agency's (IEA) Photovoltaic Power Systems Programme on *Solar Resource for High Penetration and Large-scale Applications*. The guide is publicly available for free at the website *AssessingSolar.org* (AssessingSolar.org, 2021).

The remainder of this paper is structured as follows: Section 2 presents the methodology used in developing the guide, Section 3 introduces the structure of the guide, and Section 4 provides a shortened example from the website. The conclusions and outlook are presented in Section 5.

## 2. Methodology

*Assessing Solar* is an open-source community-supported project that provides best practices and practical guidance on solar resource assessment using the Python programming language. The guide has a strong focus on using examples based on real-life applications, making it as relevant as possible to the users.

All of the topics addressed in the guide involve manipulating and modeling solar irradiance data, which is relevant to most users – and less on how to make solar irradiance measurements. Therefore, the majority of the examples deal with applying models using computer code. For this, we have chosen to use Python, as it is very versatile and the most popular programming language for data analysis and visualization (Perkel, 2015). Additionally, Python is open-source and used by most open-source solar-related libraries (Holmgren, 2018), making it the ideal language for this project. Specifically, this guide makes extensive use of the pvlib-python library, which provides numerous solar radiation models, is peer-reviewed, and has a well-established contributor base (Holmgren, Hansen and Mikofski, 2018).

A core concept of this guide is to provide practical examples of how to apply models and manipulate data in Python, thus, being able to incorporate computational content was a must. This led to choosing to write the content in the format of Jupyter Notebooks. Jupyter Notebook is a document format that allows interactive computational content integrated together with explanatory text, figures, equations, and references. Jupyter Notebooks have become popular in industry and academic research and are starting to make their way into education. The educational research goals of using Jupyter Notebooks include: (i) the use of a collaborative environment; (ii) self-directed learning; and (iii) an interactive learning environment (Suárez-García *et al.*, 2021). From a developer's perspective, this means that all of the content for a section can be written in a single document, simplifying the writing process and allowing code examples to be at the center. From a user's perspective, it is convenient that each section can be downloaded and executed locally, making the code easily adaptable. Overall, using Jupyter Notebooks allows for an enriched, interactive experience that is not available from traditional word processing software. The individual notebooks of *Assessing Solar* were compiled into a website and a publication-quality book using the Jupyter Book tools (Executable Books Community, 2020).

The majority of users will only interact with the AssessingSolar.org website, where the guide is hosted and the best user experience is achieved. The guide is meant to be continually updated and improved as new methods are developed and more contributors join the community. To facilitate a collaborative experience, the entire project, hereunder the Jupyter Notebooks and website source files, are located in a GitHub repository. This allows anyone to contribute to the guide, raise an issue, or start a discussion. These features were the main reasons for choosing GitHub, which provides free hosting of open-source projects. Using a software development platform offers the additional advantage that every contribution goes through a review process, ensuring that the content is accurate and peer-reviewed. Every time a new contribution has been reviewed and approved, this update will be reflected in a new version of the *Assessing Solar* website within minutes. In this way, unlike the writing of traditional textbooks, the development of *Assessing Solar* is a continuous and dynamic process, where new material can be added quickly.

### 3. Structure of Assessing Solar

The sections within the guide are organized by themes and cover pre-processing, manipulation, modeling, and analysis of solar resource data. At the time of writing, the guide has four major themes or chapters: (I) solar fundamentals, (II) solar radiation modeling, (III) solar resource data, and (IV) applications.

#### 3.1. Solar fundamentals

The first chapter introduces the fundamentals and basic principles of solar resource assessment, including the components of solar radiation (global, direct, and diffuse radiation), definition of solar angles (azimuth and zenith), and how to calculate the sun position in practice. This chapter also includes a section on how to manipulate solar radiation time-series data, which demonstrates many of the essential Python functions used throughout the rest of the guide. Users or students who are inexperienced with solar radiation data and models are recommended to start with this chapter. More experienced users are urged to go straight to the section that deals with their particular topic of interest.

#### 3.2. Solar radiation modeling

The second chapter covers solar radiation modeling, including algorithms and techniques that are frequently used in solar resource assessment, such as:

- Solar decomposition models, which permits estimating the direct and diffuse irradiation from global horizontal irradiance
- Solar transposition models, which permit estimating the global tilted irradiation, also known as plane-of-array (POA) irradiance in photovoltaic applications (*planned*)
- Clear-sky radiation models (*planned*)
- Normalization of solar radiation data, e.g., clear sky index and clearness index (*planned*)

#### 3.3. Solar resource data

Within the theme of solar resource data, the guide offers guidance on how to retrieve time-series of ground measured and satellite-derived solar irradiance from multiple databases. For example, the subsection titled “BSRN” introduces the Baseline Surface Radiation Network (BSRN), a global network of high-quality solar radiation monitoring stations, and demonstrates how to retrieve and parse data. The function for parsing the BSRN station-to-archive files was developed within this project and contributed to the pvlib-python library.

Additionally, procedures for quality control (QC) of solar data are provided, guiding the user through the process of flagging and removing bad data points. QC is an essential but underprioritized step, where currently, individual users tend to develop their own methods. By providing a reference method, this guide aims to provide users with better, streamlined, and standardized QC procedures while developing a deeper understanding of the potential errors.

#### 3.4. Applications

The applications chapter covers various aspects of applying solar resource data. For example, the section “*Solar Power Modeling*” demonstrates how solar resource data can be used to model the power output of solar photovoltaic systems. A second section covers site adaption, which is the process of using ground measured irradiance data to remove systematic biases in satellite-derived irradiance datasets.

### 4. An illustrative example of the use of AssessingSolar.org

This section presents an example of how *Assessing Solar* seeks to increase the use of state-of-the-art methods and disseminate best practices. To this end, we have chosen to present the section on decomposition models because it is a good illustration of how the online guide facilitates the application of models from the literature and the handling of in-situ reference data. Decomposition models separate global horizontal irradiance (GHI) into its two components: diffuse horizontal irradiance (DHI) and direct normal irradiance (DNI). The best practices are derived from an extensive study carried out by Gueymard and Ruiz-Arias (2016), where 140 decomposition models were compared to ground measured irradiance data. The study found that the DIRINDEX (Perez *et al.*, 2002) and Engerer2 (Bright and Engerer, 2019) models were the best overall and noted that many popular and frequently cited models did not perform well. Nevertheless, citations in recent years show that the popular but less

accurate models are still widely used.

We believe that the main reasons for the lack of adoption of the more accurate models are the high complexity required for implementing the models and that they remain unknown to many users. This illustrates the two key obstacles that *Assessing Solar* aims to solve: making state-of-the-art models accessible and disseminating best practices to new and experienced users alike.

#### 4.1. Retrieving BSRN solar data

The section on decomposition models starts with demonstrating how to obtain a reference dataset. Reference data is retrieved for one of the high-quality stations in the Baseline Surface Radiation Network (BSRN). In most cases, using data from the BSRN is the best practice, as it is the only network that provides high-quality ground measurements from many different climate zones and continents. However, the BSRN data is stored in a BSRN-specific station-to-archive file format, making accessing the data time-consuming and complicated, a limiting factor for many users. To reduce this barrier, the *Assessing Solar* contributors have developed a standard function for accessing BSRN data from the FTP server or a dedicated THREDDS server using a CF-compliant NetCDF format as part of the *Assessing Solar* framework. The code for accessing the FTP server has been contributed to the pvlib-python library and is available in version 0.9.0 and higher. By having these functions available, users do not have to understand the station-to-archive file format or FTP protocols, but only need to interact with the function documentation and the pvlib-python library. Using the function is then as simple as specifying the inputs: station name, start/end date, and credentials for the BSRN FTP server.

```
# Retrieving the measurement data
df, meta = pvlib.iotools.get_bsrn(
    station='CAB',
    start=pd.Timestamp('2021-01-01'),
    end=pd.Timestamp('2021-07-31'),
    username=bsrn_username,
    password=bsrn_password)

# Resample the 1-minute data to hourly
df = df.resample('1h').mean()

# Display six rows of the dataset
df['2021-01-01 07:00:00': '2021-01-01 12:00:00']
```

	ghi	ghi_std	ghi_min	ghi_max	dni	...	temp_air	pressure
2021-01-01 07:00:00+00:00	0.600000	0.088333	0.483333	0.700000	1.083333	...	-0.190000	1007.200000
2021-01-01 08:00:00+00:00	21.183333	0.206667	20.850000	21.583333	6.550000	...	0.891667	1008.000000
2021-01-01 09:00:00+00:00	77.783333	1.036667	76.150000	79.533333	22.183333	...	1.846667	1008.000000
2021-01-01 10:00:00+00:00	156.900000	2.688333	152.400000	161.283333	144.450000	...	3.618333	1008.000000
2021-01-01 11:00:00+00:00	157.766667	1.931667	154.700000	160.850000	49.983333	...	5.143333	1008.000000
2021-01-01 12:00:00+00:00	87.283333	0.646667	86.283333	88.416667	3.050000	...	5.228333	1008.000000

Fig. 1: Code example demonstrating how to retrieve BSRN data.

A code example from *Assessing Solar* is presented in Fig. 1, demonstrating how to retrieve data from the Cabauw BSRN station from January to July 2021. For each code example, a copy button is provided in the top right corner to encourage users to use the code and apply it in their own applications.

#### 4.2. Applying decomposition models

The pvlib-python library also provides functions for several different decomposition models. By relying on decomposition models from the pvlib library, users can be confident that the models are implemented correctly, as each code contribution has undergone a rigorous review and testing process. However, the library does not provide guidance on which decomposition model is suitable for a given application, what accuracy can be expected, or how the models differ.

*Assessing Solar* aims to bridge this gap by introducing and comparing the different models and demonstrating how to use them in practice. For example, in Fig. 2, the DIRINDEX model is introduced and applied in a practical example.

```

DIRINDEX model

The DIRINDEX model (Pérez et al., 2002) was found to be one of the two best
decomposition models in an extensive study by Gueymard and Ruiz-Arias (2016).
The model is an enhancement of the DIRINT model by considering clearsky data.

# Retrieve clear-sky irradiance from CAMS McClear
clearsky, meta = pvlib.iotools.get_cams(
    latitude=location.latitude,
    longitude=location.longitude,
    altitude=location.altitude,
    start=df.index[0],
    end=df.index[-1],
    email=sodapro_username)

# Estimation of the DIRINDEX model
dirindex = pvlib.irradiance.dirindex(
    ghi=df['ghi'],
    ghi_clearsky=clearsky['ghi_clear'],
    dni_clearsky=clearsky['dni_clear'],
    zenith=solpos['zenith'],
    times=df.index,
    pressure=df['pressure']*100, # atmospheric pressure in Pascal
    temp_dew=df['temp_dew']) # Dew temperature in Degree Celsius

```

Fig. 2: Screenshot from *AssessingSolar.org* on how to obtain clear-sky data from CAMS and apply the DIRINDEX model.

The section on decomposition models introduces and applies four different decomposition models, including the DIRINDEX. Subsequently, the output of the models, i.e., a time-series of DNI, is graphically compared to the measured values in Fig. 3.

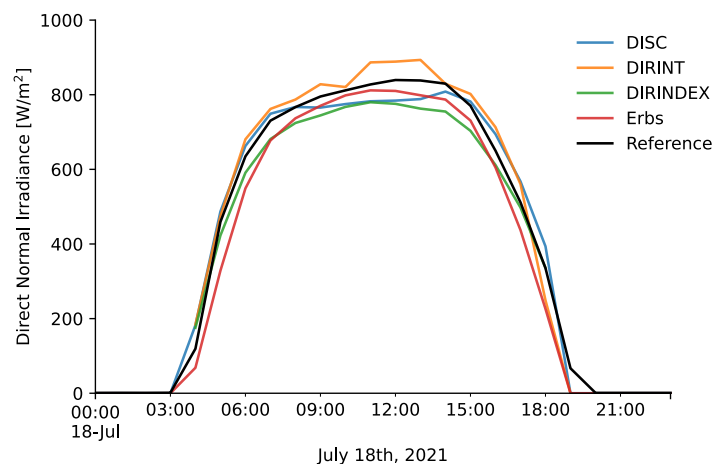


Fig. 3: Comparison of measured and modeled DNI for one sunny day.

To quantify and compare the performance of the different models, several performance metrics were calculated for each model for the analyzed period. The results showed that the DIRINDEX model outperformed the other models, confirming the findings by Gueymard and Ruiz-Arias (2016). Specifically, the mean bias deviation (MBD) of the DIRINDEX model was  $-3.5 \text{ W/m}^2$ , whereas the MBD were  $41.7$ ,  $18.3$ , and  $14.1 \text{ W/m}^2$  for the DISC, DIRINT, and Erbs models, respectively.

## 5. Conclusion

This paper has introduced *Assessing Solar*, a practical open-source guide to solar resource assessment. The guide is structured around several themes of solar resource assessment, including topics such as quality control of solar irradiance data, solar radiation data, and how to apply various solar radiation models. The examples illustrated in *Assessing Solar* make use of real data to represent real-life applications, allowing users to learn and directly apply the code and methods to their own applications.

*Assessing Solar* is a collaborative effort by participants of the IEA PVPS Task 16, and its development is a continuous process as new advances in the field are made. In order to enable the continuous update and expansion of the guide in the future, *Assessing Solar* has been set up in a collaborative manner through an online GitHub repository. The vision for AssessingSolar.org is to become the main online reference for best practices for solar resource assessment.

## 6. Acknowledgments

The authors would like to thank the members of the IEA PVPS Task 16 for their feedback and ideas during the development of AssessingSolar.org.

Adam R. Jensen would like to acknowledge the financial support of the Danish Energy Agency's Energy Technology Development and Demonstration Program (EUDP) under grant number 64020-1082.

## 7. References

AssessingSolar.org (2021) AssessingSolar.org - Solar Resource Assessment in Python: Best practice guide with interactive examples. Available at: <https://assessingsolar.org/> (Accessed: 15 October 2021).

Bright, J. M. and Engerer, N. A. (2019) 'Engerer2: Global re-parameterisation, update, and validation of an irradiance separation model at different temporal resolutions', *Journal of Renewable and Sustainable Energy*, 11, p. 033701. doi: 10.1063/1.5097014.

Executable Books Community (2020) 'Jupyter Book (v0.10)'. doi: 10.5281/zenodo.4539666.

Gueymard, C. A. and Ruiz-Arias, J. A. (2016) 'Extensive worldwide validation and climate sensitivity analysis of direct irradiance predictions from 1-min global irradiance', *Solar Energy*, 128, pp. 1–30. doi: 10.1016/j.solener.2015.10.010.

Holmgren, W. F. (2018) 'Review of Open Source Tools for PV Modeling', in 10th PV Performance Modeling Workshop. doi: 10.1109/PVSC.2018.8548231.

Holmgren, W. F., Hansen, C. W. and Mikofski, M. A. (2018) 'Pvlib Python: a Python Package for Modeling Solar Energy Systems', *Journal of Open Source Software*, 3(29), p. 884. doi: 10.21105/joss.00884.

International Energy Agency (2020) World Energy Outlook 2020. Paris, FR. Available at: [www.iea.org/reports/world-energy-outlook-2020](http://www.iea.org/reports/world-energy-outlook-2020).

Perez, R. et al. (2002) 'A new operational model for satellite-derived irradiances: Description and validation', *Solar Energy*, 73(5), pp. 307–317. doi: 10.1016/S0038-092X(02)00122-6.

Perkel, J. M. (2015) 'Programming: Pick up Python', *Nature*, 518, pp. 125–126. doi: 10.1038/518125a.

Sengupta, M. et al. (2021) Best Practices Handbook for the Collection and Use of Solar Resource Data for Solar Energy Applications: Third Edition. Golden, CO. Available at: <https://www.nrel.gov/docs/fy21osti/77635.pdf>.

Suárez-García, A. et al. (2021) 'Teaching structural analysis theory with Jupyter Notebooks', *Computer Applications in Engineering Education*, pp. 1–10. doi: 10.1002/cae.22383.