SolarSize Solar Model – Research and Implementation

SSE Capstone – University of Regina

SolarSize

Winter 2022

Tristan Brown-Hannibal

Karlee Fidek

Kaden Goski

Table of Contents

Introduction

Beginning our project, we knew next to nothing about solar panels and solar power production. Accordingly, we began by researching and scouring the internet for solar power panel information and production methodologies and calculations. We read various scholarly papers and found some great resources on the matter, however determining which methodology or approach was the best was not something that could be done by testing alone due to our solar irradiance data constraints.

It all came down to which solar irradiance information we had available, in comparison to which solar irradiance data the evaluation models required. Our data source is the NASA POWER API that provides free access to 10 km spatial hourly global irradiance values along with temperature, cloud cover, dew points, and wind speeds. Given this data, we determined that we would need to split up the given global irradiance data into its components, direct and diffuse beams. This could be done by a model called the DISC/DIRINT solar estimation model which we utilized from the Python pvlib library.

Research

To determine how to size solar panel installations and calculate ROI's, we had to figure out how to model and estimate solar power production from solar panels. This involved researching online and looking for scholarly resources to determine the methodology and variables needed to do so.

Sources

From this, we were able to find some great resources in the pvlib Python project and the pveducation website [1][2]. Both of these were created by researchers and professors from various universities and institutes and have been cited in peer reviewed papers.

Solar Irradiance API

Following the guidance of the pveducation website, we discovered that we would need a solar irradiance data source as a reference to estimate how much power could be produced by a panel. This led us down a path of discovering how expensive live access to such datasets can be and eventually finding the free to use NASA POWER API [3]. The NASA API however, had a delay of about 3-6 months of data and therefore would not be useful for any sort of live prediction or estimation. After discussing with GreenWave Innovations about how important a live-data source was, it was determined that we could use the NASA POWER API and forgo the need for any sort of live prediction model due to budgetary constraints.

DIRINT Model

The pveducation website described in great detail all the properties of solar radiation and irradiance and the various variables needed to calculate the output of a solar panel. It specified the need for calculating module azimuth (angle of the sun related to the location of the installation), the declination angle of the Earth on a given date, the hour angle (local time into degrees of sun movement), and the components of solar irradiance, direct normal irradiance (DNI) and diffuse horizontal irradiance (DHI). The calculations for module azimuth, declination angle, and hour angle

involved formulas that were given and implemented in our Python script. They can be found in the appendix.

Unfortunately, we ran into issues when the NASA API dataset had global horizontal irradiance values (GHI) and we needed its components, direct normal irradiance and diffuse horizontal irradiance to calculate how much would hit an angled solar panel. Usually, backtracking from GHI to DNI and DHI is fairly simple, as you would know what original values were used in this calculation. However, these were not provided by the API and we would have to use a model to estimate these values. This led to us researching and using the DIRINT model [4].

The DIRINT model is implemented in the pvlib Python library and as they mention "GHI is measured and a model, such as the DISC or DIRINT models, is used to estimate the DNI" [5]. It works by utilizing variables such as GHI and dew points, which we were able to get from the NASA API, to perform time derivatives and parameterization binning. From there it estimates the DNI which can be used algebraically to get the DHI (see appendix A for the formulas).

Implementation and Remarks

After applying the formulas and calculations from pveducation, accessing data from the NASA POWER API, and utilizing the DIRINT model to get the DNI and DHI from the GHI values, we were able to create a working Python script model. Testing of this model was difficult as we only had one dataset from GreenWave Innovations to go off of, however our model showed to be within ~10% assuming a 20% system loss (see Appendix B).

This was important as for our purposes, to size solar installations for optimal ROI, our model would work best if it slightly underestimated the power produced and therefore the ROI. It would not provide outlandish suggestions and instead would underestimate how much panels could produce. While this was still not ideal, we are not experts in the solar power field and were content with the performance of the model. Any further tuning or modifications would be on GreenWave Innovations and would be easier for them to perform as they get more datasets to use and compare against to reduce biases.

Testing

By analysing the chart in Appendix B, we can see that the model is accurate within 5% of peak production of the reference data from GreenWave Innovations, with a maximum value of 75KwH compared to a real value of 72.5 kwH. This is good as it does not overestimate how much the system will produce at peaks.

Looking at the winter months, we can see how much production is lost due to snow and ice coverage in the real data. While it is often not worth it to clean snow off of panels due to man hours costing more than the power saved, this demonstrates the real effects of snow being on the panels.

Lastly, we can see that the model sometimes underfits with how much power is produced, this can be attributed to the fact that the NASA API data is at a resolution of 10km and their methodology of producing GHI values, along with our model have losses which are apparent here.

Appendix A – Formulas

Declination Angle [6]

$$\delta = 23.45 \ \sin\left[\tfrac{360}{365}(284 + N)\right],$$

where N = year day, with January 1 + 1.

Solar Zenith Angle [1] [7]

SZA = 90° - α (Solar Elevation Angle)

Where:

α (Solar Elevation Angle) = 90 + Latitude – Declination Angle

Local Solar Time [1] [7]

LST (Local Solar Time) = LT + TC/60

LT (Local Time) = Longitude

TC (Time Correction) = 4 * (Longitude – LSTM) + EoT

LSTM (Local Standard Time Meridian) = ΔT_{UTC} * 15° (Difference between local time zone and UTC, with +UTC being west and -UTC being East.)

EoT (Equation of Time) = 9.87sin(2B) – 7.53cos(B) – 1.5sin(B); where B = 360/365 (day of year – 81)

HRA (Hour Angle) = 15° (LST – 12)

Solar Azimuth Angle [8]

$$Azimuth = cos^{-1}\left[\frac{sin\delta cos\varphi - cos\delta sin\varphi cos(HRA)}{cos\alpha}\right]$$

Where: α is the solar elevation angle, Φ is the latitude, and δ is the declination angle

Appendix A – Formulas (cont.)

DNI – Direct Normal Irradiance/Radiation [1]

$$B = DNI\,(\sin(\delta)\sin(\varphi)\cos(\beta) -$$
$$- \sin(\delta)\cos(\varphi)\sin(\beta)\cos(\psi) +$$
$$+ \cos(\delta)\cos(\varphi)\cos(\beta)\cos(HRA) +$$
$$+ \cos(\delta)\sin(\varphi)\sin(\beta)\cos(\psi)\cos(HRA) +$$
$$+ \cos(\delta)\sin(\psi)\sin(HRA)\sin(\beta))$$

δ is the Declination Angle

φ is the Latitude

β is the panel tilt

ψ is the panel azimuth angle (panel direction/orientation angle)

HRA is solar hour angle
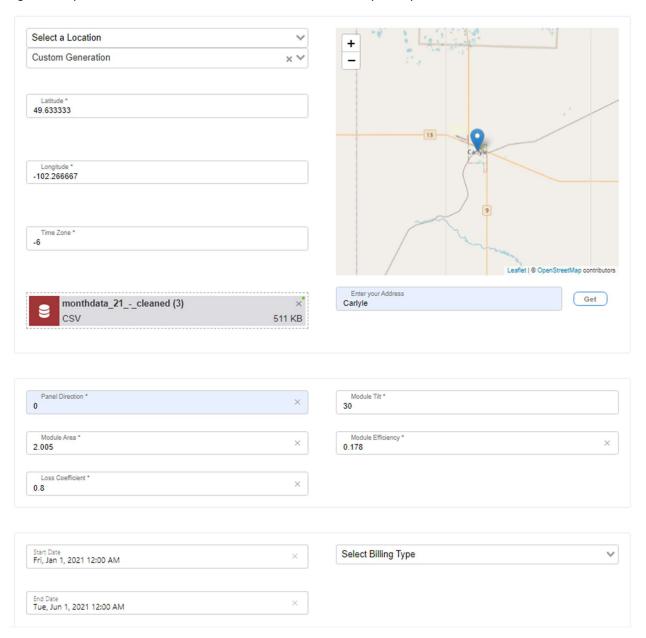
GHI – Global Horizontal Irradiance/Radiation [9]

GHI = DHI + DNI * cos (Z)

where

Z is the solar zenith angle.

DHI – Diffuse Horizontal Irradiance/Radiation
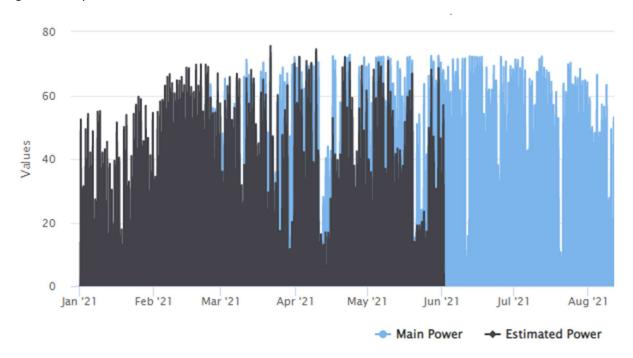
DHI = GHI – DNI * cos (SZA)

Appendix B – Results and Testing

Figure 1. Inputs used on the website to calculate the solar output vs production

Appendix B – Results and Testing (cont.)

Figure 2. Outputs – Black = Estimation Model, Blue = Real Production Data

References

[1] A. L. and Hegedus, A. Luque, Hegedus, R. Perez, Ineichen, Seals, Michalsky, and Stewart, "Making use of TMY Data," *PVEducation*. [Online]. Available: https://www.pveducation.org/pvcdrom/properties-of-sunlight/making-use-of-tmy-data. [Accessed: 03-Apr-2022].

[2] W. F. Holmgren, C. W. Hansen, and M. A. Mikofski, "Pvlib python: A python package for Modeling Solar Energy Systems," *Journal of Open-Source Software*, vol. 3, no. 29, p. 884, 2018.

[3] "NASA Power API," *NASA*. [Online]. Available: https://power.larc.nasa.gov/docs/services/api/temporal/. [Accessed: 03-Apr-2022].

[4] P. Ineichen, R. Perez, R. Seal, E. Maxwell, and A. Zalenka, "[PDF] Dynamic Global-to-direct irradiance conversion models: Semantic scholar," *undefined*, 01-Jan-1992. [Online]. Available: https://www.semanticscholar.org/paper/Dynamic-global-to-direct-irradiance-conversion-Ineichen-Perez/0da5f4e6bdb0f42eb10d45607cce1df13e08961a. [Accessed: 03-Apr-2022].

[5] "Global horizontal irradiance," *PV Performance Modeling Collaborative*. [Online]. Available: https://pvpmc.sandia.gov/modeling-steps/1-weather-design-inputs/irradiance-and-insolation-2/global-horizontal-irradiance/. [Accessed: 03-Apr-2022].

[6] I. Sarbu and C. Sebarchievici, *Solar heating and cooling systems: Fundamentals, experiments and applications*. Amsterdam: Academic Press, 2017.

[7] D. D. Rooij, "Elevation angle," *Manage risks and maximize ROI for your PV and energy storage projects*, 01-Mar-2022. [Online]. Available: https://sinovoltaics.com/learning-center/basics/elevation-angle/. [Accessed: 03-Apr-2022].

[8] A. L. and Hegedus, A. Luque, Hegedus, R. Perez, Ineichen, Seals, Michalsky, and Stewart, "Azimuth Angle," *PVEducation*. [Online]. Available: https://www.pveducation.org/pvcdrom/properties-of-sunlight/azimuth-angle. [Accessed: 03-Apr-2022].

[9] "Solar Resource Glossary," *NREL.gov*. [Online]. Available: https://www.nrel.gov/grid/solar-resource/solar-glossary.html. [Accessed: 03-Apr-2022].