

Estimating installation parameters of photovoltaic systems by mathematical model fitting in northern latitudes

Timo Salola

MSc Thesis

November 2023

Department of Physics and Mathematics
University of Eastern Finland

Preface

As someone whose interests and strengths lie in geometry and programming, the real world problem of panel installation parameter estimation was appealing from the beginning. I would like to thank William Wandji and Juha Karhu for their insights on the effects of clouds, snows and temperature on solar PV installation data. And Luna for being the softest and purriest of cats.

Helsinki, the 15th of August 2023

Timo Salola

CONTENTS

1	Introduction	1
2	Datasets	3
2.1	Visualizing the data	5
2.2	Data pre-processing	7
2.2.1	Clear day detection algorithm	8
2.3	Assumptions and possible issues	9
3	Solar irradiance simulation tools	10
3.1	PVlib POA python function	11
3.1.1	Function input parameters	12
3.2	PVlib POA evaluation	14
3.3	Influence of different parameters on the PVlib poa model	16
3.3.1	Influence of different longitudes	18
3.3.2	Influence of different latitudes	19
4	Estimating geographic location	20
4.1	Estimating geographic longitude	21
4.1.1	Longitude estimation results	23
4.1.2	Possible issues and further development ideas	24
4.2	Estimating geographic latitude	26
4.2.1	Latitude algorithm	26
4.2.2	Latitude algorithm visualization	27

4.2.3	Improving the algorithm	27
4.2.4	Latitude estimation results	29
4.2.5	Possible issues and further development ideas	30
4.3	Combined latitude and longitude estimations	31
5	Estimating panel angles	32
5.1	Prediction error function	34
5.2	Simulation error function	36
5.2.1	Area based error function	37
5.2.2	Alternative simulation error functions	37
5.3	Multiplier matching	38
5.3.1	Area based multiplier matching	38
5.3.2	Segmented multiplier matching	39
5.3.3	Translating multiplier values to PV system rated power values	42
5.4	Angle space discretization	43
5.4.1	Importance of lattice density	43
5.5	Solving panel angles	46
5.5.1	Evaluation of results	48
5.5.2	Possible issues and further development ideas	50
5.6	Final remarks	52
References		53

CHAPTER I

Introduction

This thesis examines a specific applied mathematics problem suggested by the Finnish Meteorological Institute(FMI). The goal is to solve the geographic location and panel installation angles of photovoltaic solar power installations using only the power output data. The chosen method breaks the problem of solving the geographic location and panel angles into separate algorithms. In practice, this means that the algorithms used can be less complex as they do not have to solve every unknown variable simultaneously and visualizations of individual algorithms should also be more straightforward as 2-dimensional data can be graphed with ease. Using multiple algorithms also splits the parameter space into smaller spaces, thus improving the performance of fitting algorithms. And the final benefit is the ability to focus on solving only the unknown parameters. For example, if the geolocation of a system is known and the panel installation angles are not, there is very little reason to use a model that would solve geolocation and panel angles without taking into account that some parameters are already known.

The applications of parameter estimation algorithms would be in improving the quality of metadata in solar PV datasets. This could have implications for solar PV research but the existance of such algrithms poses some privacy and security related questions as well. Whether these research benefits and concerns are realized, depends on the accuracy, and to some extent the ease of use of the algorithms.

Perhaps the most similar study was done by N. Haghdi et al. in 2017 [1]. In Haghdi's paper, the standard deviation and mean absolute deviation for longitude are both less than 1.5° in the five listed case studies, reaching as low as 0.08° in case study 1-2. The standard deviation of latitudes is less than 3.5° for each case study;

the panel tilt deviation is less than 11.5° , and the azimuth is less than 27.5° .

Another article of relevance written by M.K. Williams et. al. in 2012 [2] proposes multiple methods for determining the locations and orientations of solar PV installations. Perhaps the most interesting contribution of the 2012 article is the network approach to determining geographic location. This method relies on a grid of installations with known and accurate geolocation and installation angles. According to the authors, this networked approach works up to a 10-mile accuracy when the grid of known installations is dense around the estimated installation. This could make the network approach preferable for electric companies or institutions with large amounts of data. But as of now, it does not seem usable for FMI.

CHAPTER II

Datasets

The two primary solar power datasets for this thesis were provided by FMI. Datasets from these installations were previously used in Herman Böök's *Photovoltaic output modeling* [3] and thus this data has to some extent been verified. Another benefit of FMI datasets is the high temporal resolution. One measurement per minute should ensure that predictions are not significantly affected by the data resolution.

The table 2.1 shows the general structure of solar PV energy output datasets with timestamps and power output values. The differences between FMI datasets and datasets from other sources are likely to be the timestamp formats, temporal resolution, power measurement units and the amount of different power measurement sources. The algorithms in this thesis will work on any dataset with timestamps and power values, but the temporal resolution is likely to influence the accuracy of algorithms and interpolation has to be done in order to match the time resolution of measurements and the algorithms.

As seen in the table 2.1, the dataset is missing some values. In the case of the first and last daily minutes, this could be the result of the inverter or other instruments turning off when electricity generation is close to zero, but there could be other reasons as well. Due to the possibility of missing measurements and presence of NaN-values, the data has to be preprocessed.

Another point of interest are the multiple different power values included in the datafile. Columns String 1 and String 2 correspond to the output power from two identical sets of solar panels installed in the same installation site. Having two identical installations at the same location has some value for data verification as both panel strings should provide almost identical outputs, but this is not the main



Figure 2.1: FMI Kumpula solar power installation string.

focus of the thesis and thus these values are ignored.

In addition to the two panel set output values, there are two inverter values. Out of these the inverter input is one step closer to the power generation of solar PV systems and thus algorithms that utilize this input value could be more accurate. However as the inverter output represents the amount of usable power generated by the PV system, the output is the most likely power value to be included in datasets and thus it will be used in this thesis.

	Helsinki	Kuopio
Latitude	60.204°	62.892°
Longitude	24.961°	27.634°
Nominal capacity	21 kW	20.28 kW
Panel tilt	15°	15°
Panel angle	135°	217°
Elevation	17m	10m

Table 2.2: Parameters for the FMI's Kumpula(Helsinki) and Kuopio PV installations as listed in Böök 2020 [3].

Timestamp[UTC]	Inverter out	Inverter in	String 1	String 2
2015 – 08 – 26 03 : 34	<i>NaN</i>	<i>NaN</i>	0.5	<i>NaN</i>
2015 – 08 – 26 03 : 36	11.1	7.5	2.6	4.9
2015 – 08 – 26 03 : 37	25.4	26.1	9.8	16.3
2015 – 08 – 26 03 : 38	30.7	<i>NaN</i>	<i>NaN</i>	0.4
2015 – 08 – 26 03 : 39	46.4	44.8	20	24.8
2015 – 08 – 26 03 : 40	3.3	<i>NaN</i>	<i>NaN</i>	0.4
2015 – 08 – 26 03 : 41	29.3	18	9.1	8.9
2015 – 08 – 26 03 : 42	33.1	27.4	10.6	16.9
:	:	:	:	:
2015 – 08 – 26 12 : 42	12374.8	14619.1	7152	7467.1
2015 – 08 – 26 12 : 43	15442.2	15482.1	7708.9	7773.2
2015 – 08 – 26 12 : 44	14085.8	12898.7	6387	6511.8
:	:	:	:	:

Table 2.1: A section from FMI’s Kumpula solar site PV production data, only the timestamp and inverter output values are used by the algorithms in this thesis. All power measurements are in watts.

2.1 Visualizing the data

Before choosing a method for solving the geographic location and panel angles, it would be helpful to determine whether there are any useful patterns in the data. In figure 2.2 we can see that the measurements during certain days can be fairly smooth and they clearly form well defined structures, while measurements from other days may form patterns that appear almost random. While this deviation from the ideal in the power generation figures can be caused by a multitude of factors, for the sake of convenience, the terms *cloudy days* and *cloud free days* or *clear days* will be used to describe whether days contain characteristics typical to the smooth, cloud free days or the noisy, cloudy days regardless of the actual cause.

The clear day [2017-123] shown in the figure 2.2 is representative of the very best data within datasets and so days similar to it should be used for installation parameter estimation. The traits which can be easily extracted from similar days are the first and last non-zero power minutes and the peak power generation minute.

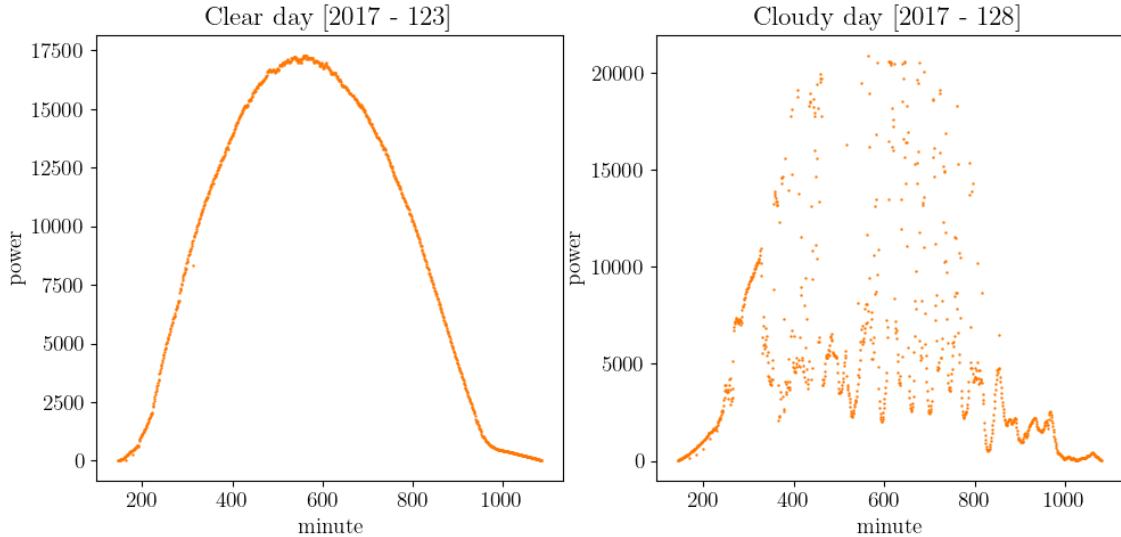


Figure 2.2: Two days from FMI Kumpula dataset with different characteristics.

It is worth pointing out that even with the high amount of noise in the cloudy day [2017-128], visual inspection suggests that the first and last non-zero minutes of the measurements seem to be the same as during the clear day [2017-123]. This would suggest that even days with a high amounts of apparent noise may contain usable data. Therefore whether noisy data can be utilized or not has to be decided on a case by case basis.

The peak power generation minute signifies a moment in time during which the system is performing optimally. Geometrical intuition would suggest that this occurs when the angle of incidence of direct sunlight reaching the panel is at its minimum and thus the peak generation minute could prove to be useful for parameter estimation. Despite the apparent importance, the peak power minute was not utilized by any of the cited research papers. This could be due to the uncertainty in timing caused by variation in power generation or redundancy as system parameters can be estimated through other means.

In addition to the patterns visible in single day sections from the datasets, patterns also exist in multi-day sections. This is visible in the scatterplot 2.3 where the added day axis highlights the existence of patterns along the day axis. The two noteworthy patterns are the 3D dome-shape formed by the individual days and the shapes formed by the first and last non-zero minutes.

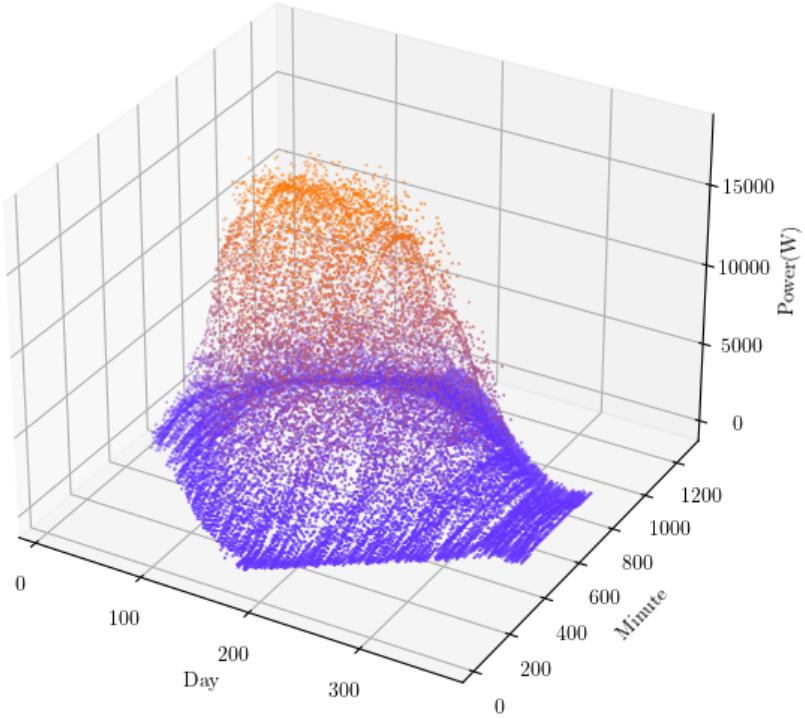


Figure 2.3: One year of data from FMI Kumpula installation as a 3D point cloud.

2.2 Data pre-processing

The not-a-number or *Nan* values mentioned earlier and visible in the table 2.1 could cause issues with some data processing algorithms. In the python programming language, many mathematical functions will always return *Nan* if even a single *Nan* value is present in the input. Another issue with datasets is that they include gaps, meaning that rows corresponding to some minutes are not present in the dataset. If these gaps are significant enough, their existence could cause errors and biases in processing algorithms.

One solution for this problem would be to design every algorithm to accept two inputs, the measured power and the measuring time but this would come at a cost to algorithm complexity. The easier option and the method used for processing the FMI datasets is the removal of all rows with *Nan* power values and the use of linear interpolation in order to approximate the missing rows. In this case linear interpolation is applied on the minute dimension, meaning that if a gap in power measurements is detected on the minute axis, the missing minutes are filled so that

the gap is filled with a linear transition breaching the gap.

2.2.1 Clear day detection algorithm

Another important form of data pre-processing is the selection of days which contain as little noise as possible. These cloud free days are useful as their more defined structure makes them ideal for model fitting. The following listing contains the steps of a clear sky detection algorithm.

1. Separate the dataset into individual days.
2. Take one day and save the power output values from that day into a list *powers*.
Compute the length of a day by calculating euclidean distance over a day of measurements with the sum function: $\sum_{i=1}^n \sqrt{1 + (powers[i] - powers[i - 1])^2}$
3. Perform a low pass filter on the power values. In this thesis filtering was accomplished with a Fourier transformation based method which preserves the longest wavelengths present within the measurements. Filtering can also be done with running average filters or any other means of smoothening out the measurements.
4. Compute Euclidean distance from point to point for the low pass version of measurements by using the same algorithm as in step 2.
5. Compute the average difference between the values from steps 2 and 4.
6. Repeat steps 2-5 for each day in the dataset. If the average difference from step 5 is on average higher than a given threshold value, reject the day.

Note that the algorithm listed above is highly dependent on measurement intervals and further tuning could be needed when operating with datasets that have different temporal resolutions. And as is, the algorithm selects days based on their proportionally low high frequency component, thus in theory this algorithm should classify zero power output days as cloud free days. Despite this fault the algorithm seems to work well for the FMI datasets.

2.3 Assumptions and possible issues

If metadata such as the geographic location or panel installation angles is missing from the datafiles, it is very likely that other critical pieces of information could be left out as well. Were additional modules were installed during operation? Could some panels be installed in different panel angles? What if the panels are installed in tracking mounts and thus the panel angles vary during each day? These questions are left unanswered and thus some assumptions have to be made. In this thesis we will assume that the panels are installed on fixed mounts, no changes were done during data gathering period and all panels are oriented similarly. We will also assume that there are no major obstacles casting shadows on the panels and that the panels are not self-shadowing, meaning that the panels are not casting shadows on one another.

Another source of uncertainty is data collection itself. The device responsible for measuring power output values and logging the values has to have a clock for measuring time, but this clock could have been running too slow or fast, resulting in a drifting error in the timestamps. Similarly if the system clock is running at the right speed but it is off by a minute or two, this could cause a bias in the data which would be hard to detect. There is also the question of how measurement timing is done. If the time resolution of the logging device is 15 minutes, is the power value at 12:45 taken during the 45th minute or is the power value the average of the previous 15 minutes as is often done in meteorology? Or could the power value be the average of measurements taken during the interval 12:38 to 12:52? In meteorology, the last period average would be the standard, but standards may not always be followed.

CHAPTER III

Solar irradiance simulation tools

Having a mathematical model which would simulate the output of a PV system would allow for the parameters of a PV installation to be solved with model fitting. In the best case scenario, we would have a physics based model which would take geographic location, panel installation angles, time of year and panel surface area or power rating as inputs and the output would be similar to the data from FMI Kumpula installation seen in table 2.1. Creating a such model is rather challenging as the model has to take into account atmospheric scattering, Sun angles, Sun-Earth distance variation and a multitude of other factors, the consideration of which are far beyond this mathematics thesis. Luckily the modeling of the energy output of solar PV installations has uses for the cost-benefit analysis of solar PV installations and thus pre-existing modeling algorithms are publicly available.

This thesis uses a plane of array irradiance simulation function from the python library PVlib. The function takes geographic location, timestamp and panel angles as inputs. The outputs contain power values which describe the amount of direct and atmospherically scattered light that would hit a square meter sized imaginary solar panel with the input parameters. The sum of these sources is referred as plane of array (POA) irradiance and this value can be used to estimate the output of solar power installations. A section of simulated data is included in table 3.1.

As the model simulates radiation values during clear sky conditions and not the power output of pv installations, the model should be seen as an approximation which is accurate to a certain degree. The errors in the model could be due to reflectivity of the solar panels, weather conditions, temperature related changes in efficiency, atmospheric composition or a multitude of other factors which the model

does not take into account.

Timestamp[UTC]	Minute	POA(W)
2018 - 05 - 30 00 : 00	0	0.0
2018 - 05 - 30 00 : 01	1	0.0
2018 - 05 - 30 00 : 02	2	0.0
⋮	⋮	⋮
2018 - 05 - 30 07 : 34	454	800.691861
2018 - 05 - 30 07 : 35	455	802.110516
2018 - 05 - 30 07 : 36	456	803.517424
⋮	⋮	⋮
2018 - 05 - 30 23 : 57	1437	0.0
2018 - 05 - 30 23 : 58	1438	0.0
2018 - 05 - 30 23 : 59	1439	0.0

Table 3.1: One day of simulated plane of array irradiance values. Note that the minute column is added to the table for convinience and it is redundant as minutes can be read from the timestamps.

3.1 PVlib POA python function

Taking a look at the code responsible for the plane of array irradiance simulations can give some insights into the behavior of the estimations and problem of parameter estimation. The python function responsible for simulating plane of array irradiance values over a day is defined with the header 3.1. In this header we can see that the function takes 7 inputs each of which is mandatory, meaning that when the functions is used as a tool for parameter estimation, some of the parameters have to be guessed or randomly assigned.

Listing 3.1: PVlib POA simulation function header.

```
def get_irradiance_with_multiplier(year, day, latitude, longitude, tilt,
                                   azimuth, multiplier):
```

3.1.1 Function input parameters

The following listing contains the parameters of the plane of array irradiance function and their domains.

- Year $\in \mathbb{N}$
- Day $[1, 365/366] \in \mathbb{N}$
- Latitude $[-90, 90] \in \mathbb{R}$
- Longitude $[-180, 180] \in \mathbb{R}$
- Tilt $[0, 90] \in \mathbb{R}$
- Azimuth $[0, 360[\in \mathbb{R}$
- Multiplier $]0, \infty[\in \mathbb{R}$

Note: While the function does accept the full latitude and longitude ranges as inputs, it may be beneficial to restrict the range of the coordinate parameters when the approximate location of the installation is known. For example, Finland fits within subrange $[19, 32]$ on the longitude axis and thus it could make sense to restrict the longitude range when examining installations located within Finland.

The combination of these parameters and their ranges can be thought to form a subspace in seven-dimensional Euclidean space. This so-called parameter space and its "volume" are both concepts that can be used for analyzing the difficulty of parameter estimation problems, behavior of parameter estimation functions and their efficiency. In general, the more parameters and thus dimensions there are, the larger the resulting parameter space is and the harder the problem becomes. And the more parameters an algorithm is attempting to solve at once, the slower the algorithm can be expected to be.

With solar PV installation parameter estimation, there are 5 unknown parameters as the year and day parameters are always known. If each of the remaining parameters is discretized to 20 evenly spaced values, solving all the 5 parameters at once by testing out every possible combination would require evaluating 20^5 or 3.2 million unique combinations. However if the parameters could be solved one by

one, isolated from the influence of other parameters, there would only be $20 * 5$ or 100 unique combinations, a reduction of 32000 to 1. This highlights how important it is to break larger problems into smaller problems whenever possible.

3.2 PVlib POA evaluation

Before using the pvlib POA simulations for parameter estimation, the simulated estimates should first be evaluated. The figure 3.1 indicates that in clear sky conditions the pvlib irradiance model is following real world measurements closely with a few exceptions. During cloudy days, the measurements can be seen to exceed the power generation estimated by the model. The cause for this increase could be reasoned to be the additional sunlight reflected from clouds towards solar panels in partly cloudy weather conditions. Regardless the reason, this shows that the noise introduced in measurements can be positive as well as negative. There is also some deviation visible in the power values of first and last non-zero hours which may prove to be an issue in later stages.

The same figure shows the importance of finding clear sky days as the only characteristic in the measurement plots which seems to be consistent between the cloud free and cloudy days seems to be the timing of the first and last non zero minutes. In the 240th day for example, most of the measurements are either higher or lower than the estimated values, but the first and last non-zero minutes seem to align with the first and last non-zero minutes of POA simulations. On the 70th and 150th day, the difference between the modeled first and last non-zero minutes is visible on the graph. Whether this difference is significant will depend on the algorithms applied.

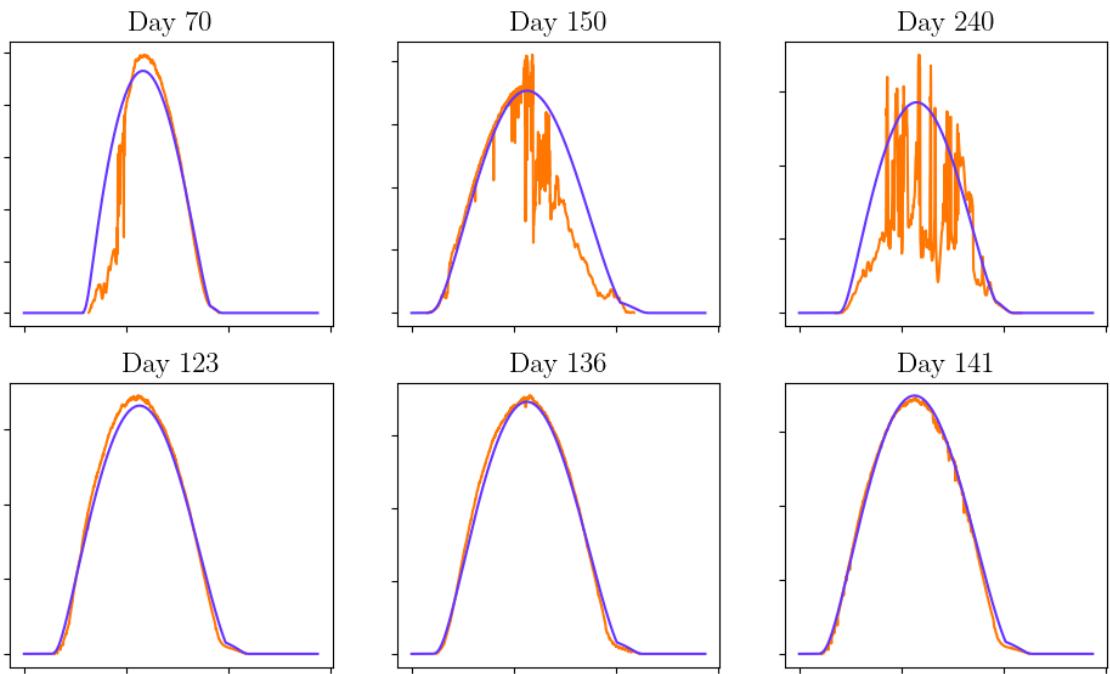


Figure 3.1: Power output of FMI Kumpula PV installation and the pvlib POA simulation computed with the parameters 2.2. Horizontal axis on the graphs corresponds to time and vertical axis marks the estimated power values. The purpose of the graphs is to display the different shapes and deviations from POA models and thus axis names and numbers were left out. Upper row contains randomly selected days while as the lower row has days chosen by a clear sky algorithm mentioned in chapter 2.2.1. Measurements are from 2017. POA irradiance values were multiplied by 20 in order to match the curves values on power axis.

3.3 Influence of different parameters on the PVlib poa model

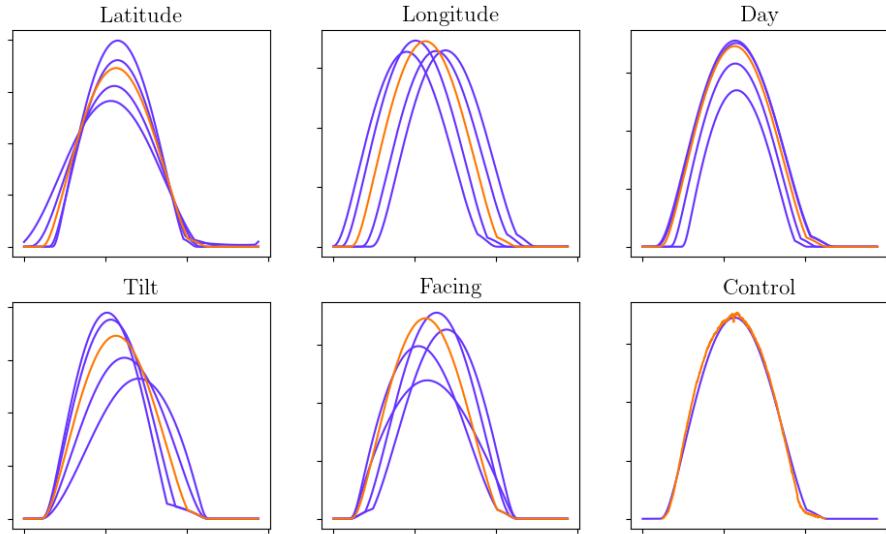


Figure 3.2: POA irradiance simulations with different parameters. The control contains a curve modeled after FMI Helsinki 2.2 PV installation and other graphs show how the POA curves change when POA parameters latitude, longitude, day, tilt or facing(azimuth) are adjusted. Adjustments were done in steps of 8 for latitude, longitude 15, day 25, tilt 15 and azimuth 60.

In the best case scenario each of the simulation function inputs would affect one measurable property in the irradiance plots and their relationship would be bijective. To give an example, if the peak power minute was isolated from all other parameters than the longitude and the relationship between longitude and peak power minute was linear, it would be possible to solve the peak power minute to longitude function with just a few plane of array irradiance simulations.

In the exact opposite case where every measurable property of irradiance plots is affected by every input parameter, solving the parameters would be much harder or even impossible due to limited computational resources. For example if all of the parameters influenced the same traits to different extents and the system was not bijective, multiple parameter combinations could result in the same simulated power graph. In a such system there would not be a single solution but rather a set of possible solutions.

The problem of solving installation parameters lies somewhere in between the

two extremes. The longitude parameter would seem to shift the curve along the time axis where as tilt and facing(azimuth) parameters do not affect the first or last non-zero minutes but they do affect the shape of the curve. Observations of parameter to trait interactions are listed on table 3.2.

Parameter	Traits affected
Latitude	Shape, first and last minute times
Longitude	First and last minute times
Tilt	Shape
Azimuth	Shape

Table 3.2: Function input to observed trait table.

3.3.1 Influence of different longitudes

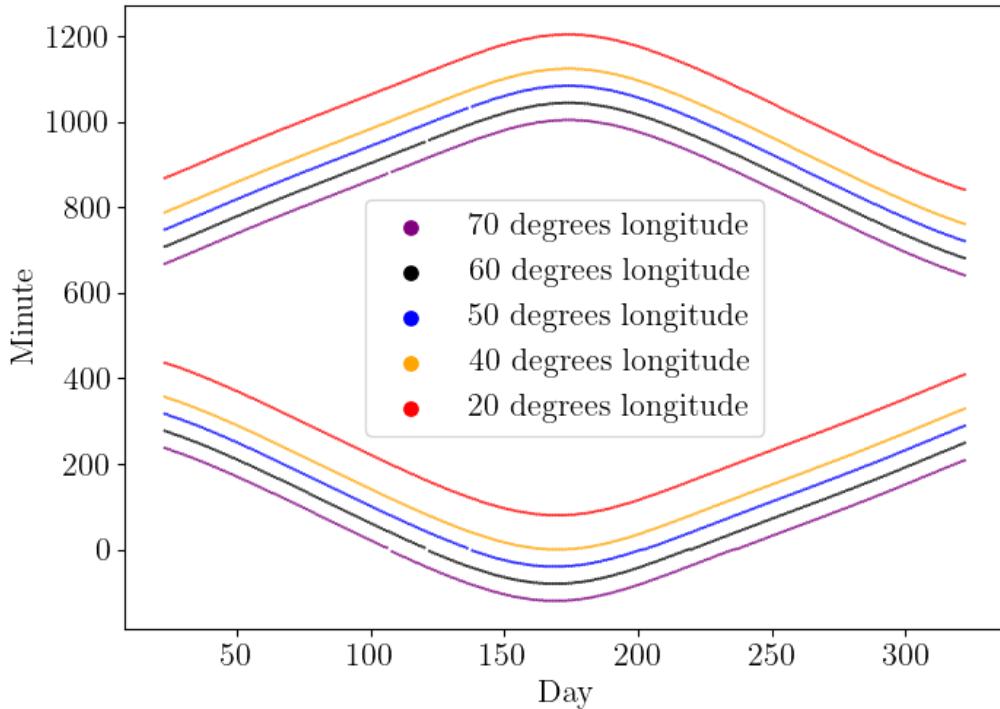


Figure 3.3: First and last non-zero minutes of each day from year long simulations at different longitudes.

Based on earlier observations listed in table 3.2, solving the longitude of installations would seem like a sensible starting point. The figure comparing the effects of different parameters seemed to suggest that the relationship between longitude and significant minute times is very close to linear and the same is seen here in figure 3.3. In Hagdadi 2017 [1] and in Williams 2012 [2] this relationship was used in order to determine the geographic longitude. The algorithms used by both of the articles relies on calculating an approximation for the time of the solar noon based on the average of the first and last minutes, this solar noon minute is then translated into a geographic longitude coordinate.

3.3.2 Influence of different latitudes

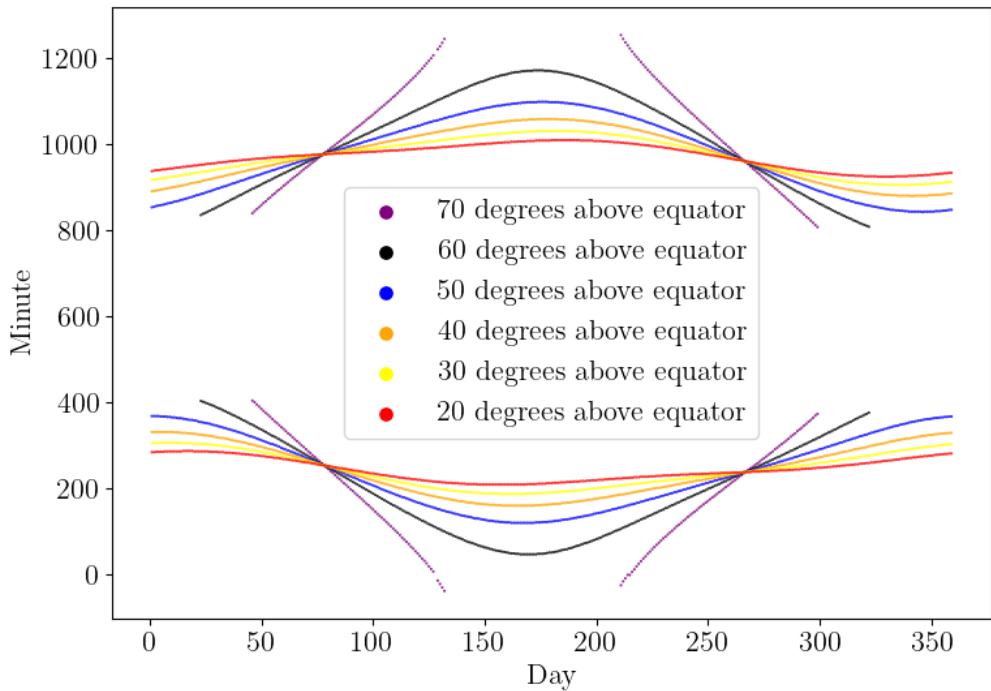


Figure 3.4: First and last non-zero power minutes of each day from year long simulations at different latitudes

The latitude simulations show that the day length stays fairly consistent for locations close to the equator, but with latitudes of 50° and higher, the day to day variation is significant. These POA simulations would imply that the region around equinoxes to be ideal for day length based analysis as there day length is always well defined and the rate of change can be measured.

CHAPTER IV

Estimating geographic location

In order to evaluate the performance of longitude and latitude estimation functions, it may prove useful to be able to translate the error values from degrees to kilometers. The following two equations 4.1 and 4.2 can be used to approximate the degree deltas of longitude and latitude estimation functions in kilometers. Note that these functions can be off by several percents as they rely on the assumption that the Earth is a perfect sphere and not an irregular ellipsoid.

Latitudinal distance to kilometers(Distance on North-South axis)

$$Distance_{latitudinal}(lat_d) = (40000km/360^\circ) * lat_d \quad (4.1)$$

Where *lat_d* is the distance between two points in degrees latitude and 40000km is an approximation for Earth's circumference.

Longitudinal distance to kilometers at given latitude(Distance on East-West axis)

$$Distance_{longitudinal}(lon_d, lat) = (40000km/360^\circ) * \cos(lat) * lon_d \quad (4.2)$$

Where *lon_d* is the distance in degrees longitude and *lat* is the latitude for which the distance is calculated.

As long as the deviations are not significant and highly accurate error values are not needed, the total error in absolute terms can be estimated by using the latitudinal and longitudinal distances as the x and y coordinates on a cartesian plane and computing the euclidean distance between the origin and resulting point.

4.1 Estimating geographic longitude

As mentioned in sections 3.3.2 and 3.3.1, the geographic location of a PV system has a strong connection to the timing of the first and last non-zero measurements of each day whereas the influence of tilt and facing parameters seems to be nonexistent. The relationship would seem to be so clear that without further analysis it would be tempting to use fairly simplistic mathematical models for these estimations. The following longitude estimation function 4.3 can be derived with the use of two basic assumptions. These assumptions are that solar noon occurs at 12:00 or 720 minutes at longitude 0 each day and at 6:00 or 360 minutes at 90 degrees. Rest of the values can then be linearly interpolated. Note that here solar noon refers to the midpoint between the first and last non-zero minute which is different from astronomical solar noon which occurs nearly at the same time.

Naive solar noon to longitude equation

$$\text{Longitude}(sn) = 180^\circ - \frac{360^\circ}{1440} * sn \quad (4.3)$$

Where sn is the approximated solar noon minute calculated by taking the average of first and last non-zero power generation minute of a day.

The simplicity of 4.3 makes the equation appealing, but the assumption of solar noon occurring at 720 minutes should still be verified. In the figure 4.1 solar noons can be seen to occur at around 720 minutes but they can also be observed occurring 15 minutes earlier or later than that. This 15 minute delta would translate into an error range of ± 3.75 degrees or approximately $\pm 200\text{km}$ at the latitudes of Helsinki according to the equation 4.1.

Knowing that the PV installation is within a 400 kilometer wide slice should be in most cases be accurate enough for determining the country in which the PV installation is located in, but for most other purposes this level of accuracy is unlikely to be valuable. Fortunately the naive model can be improved upon by taking the solar noon timing variation into account.

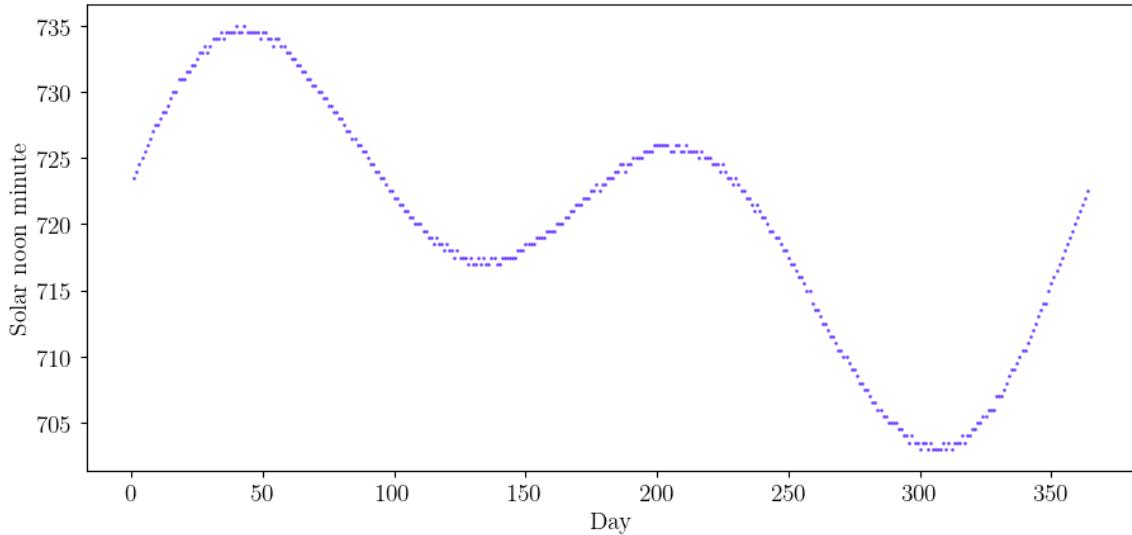


Figure 4.1: Approximations of solar noon minutes based on PVlib POA function at longitude 0° for year 2023. This pattern is caused by the Earth's axial tilt and elliptical orbit around the Sun [4].

Improved longitude estimation function

$$Longitude(sn) = \frac{360}{1440}(sn_{poa} - sn) \quad (4.4)$$

Where sn is the solar noon estimate based on measurement data and sn_{poa} is the simulated solar noon at 0 degrees longitude. The new function parameter sn_{poa} compensates for the variation seen in 4.1.

The improved algorithm—just like the original—can be used on a single day of data.

However noise present in power generation measurements may result in significant errors in estimates based on individual days. These errors can be mitigated by using the algorithm on larger sections of data and averaging the results, or alternatively the algorithm could be applied only on selected cloud free days where the expected errors are likely to be smaller. If the unfiltered multi-day approach is used, choosing the right day range is crucial. If the range is too narrow, a single outlier value can distort the results significantly, however if the whole year is used, certain periods of the year may contain more noise than others and thus their use could decrease the accuracy of the results. The two scatterplots in figure 4.2 show that

the data quality from the very first and last days of the year seem to be significantly worse than the data from the longest days of the year. The same graph would also seem to suggest that overall data quality decreases the further north the installation is. Based on these visualizations, days outside the range of 100th to 280th would seem unsuitable for first and last minute based analysis between latitudes $60^{\circ}N$ and $63^{\circ}N$.

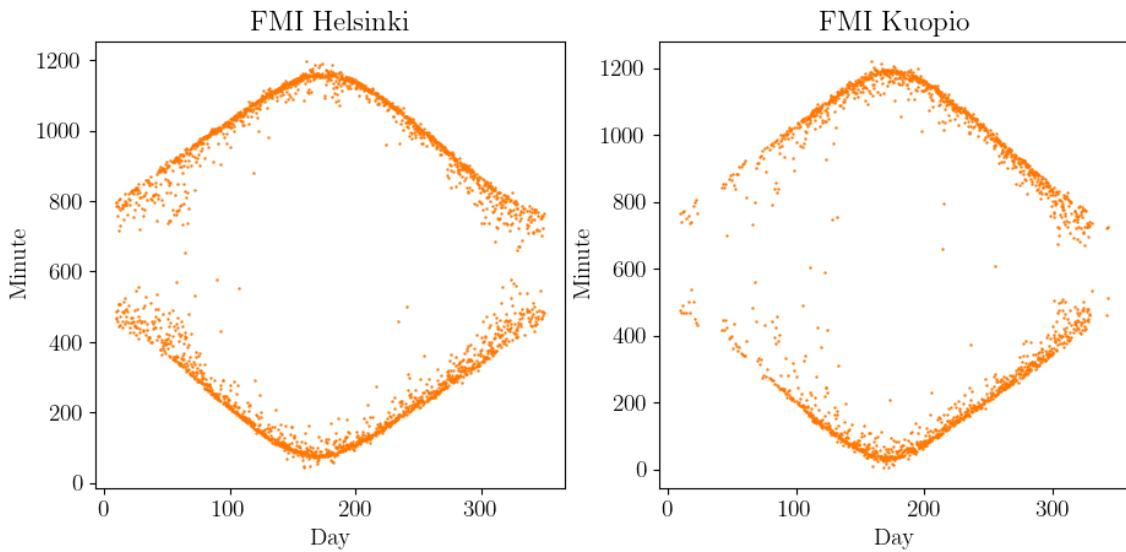


Figure 4.2: First and last non-zero power minutes of each day during years 2017 to 2021 from FMI Helsinki and Kuopio datasets.

4.1.1 Longitude estimation results

The improved algorithm was tested on the day range of 125th to 250th of each year from both FMI datasets and the results can be seen on the table 4.1. For the Helsinki installations these estimates are all off by less than 0.3° while as the Kuopio installation deltas are a bit higher with max of just over 1° . More impressively, the mean delta of multiple years for the Helsinki dataset is just 0.07° and 0.457° for Kuopio. In kilometers, the mean deltas can be approximated to 4 and 28 kilometers respectively. The lower accuracy of the Kuopio estimations could be due to multitude of factors ranging from differences in local climate or lower elevation of the installation among others.

Year	Longitude Helsinki	Δ°	Longitude Kuopio	Δ°
2021	25.115°	0.154°	26.625°	-1.009°
2020	25.029°	0.068°	27.691°	0.057°
2019	24.944°	-0.017°	27.411°	-0.223°
2018	25.243°	0.282°	26.862°	-0.772°
2017	24.836°	-0.125°	27.297°	-0.337°
mean	25.031°	0.07°	27.177°	-0.457°

Table 4.1: Means of multi-day longitude estimations from 125th to 250th day of each year.

4.1.2 Possible issues and further development ideas

While experimenting with the solar minute estimation functions, a curious trait was found. In figure 4.3, the average of the first and last minute is approximately the same at different latitudes as long as the latitude is below 50 degrees. As the latitude is increased, the solar noon estimates begin to deviate significantly, becoming strongly skewed after 70 degrees. At first this behavior seems strange as astronomical solar noon should occur happen at the same time when longitude and the day are the same regardless of latitude. However as the solar noon estimates are calculated based on the first and last non-zero irradiance minute of the day, it would make sense that the estimations could be off by significant amount during equinoxes due to rapid changes in day lengths. Figuring out how significantly this affects longitude estimations is challenging. In theory, if the same bias occurs in both the measurements and the model, no corrections would be needed. The effect should be also lessened by using longer day ranges for predicting longitudes or by making sure that the intervals include an equal amount of days from both halves of the year.

Improvements in the algorithm accuracy could also be achieved via by increasing the sampling interval of the irradiance simulations. PVlib POA simulations include a parameter for sampling frequency which is currently set to 1-per-minute in order to match the measuring frequency of FMI datasets. This could be increased to 1-per-second and the added resolution could help in determining more accurate estimates for solar noon times, resulting in possible gains in algorithm accuracy.

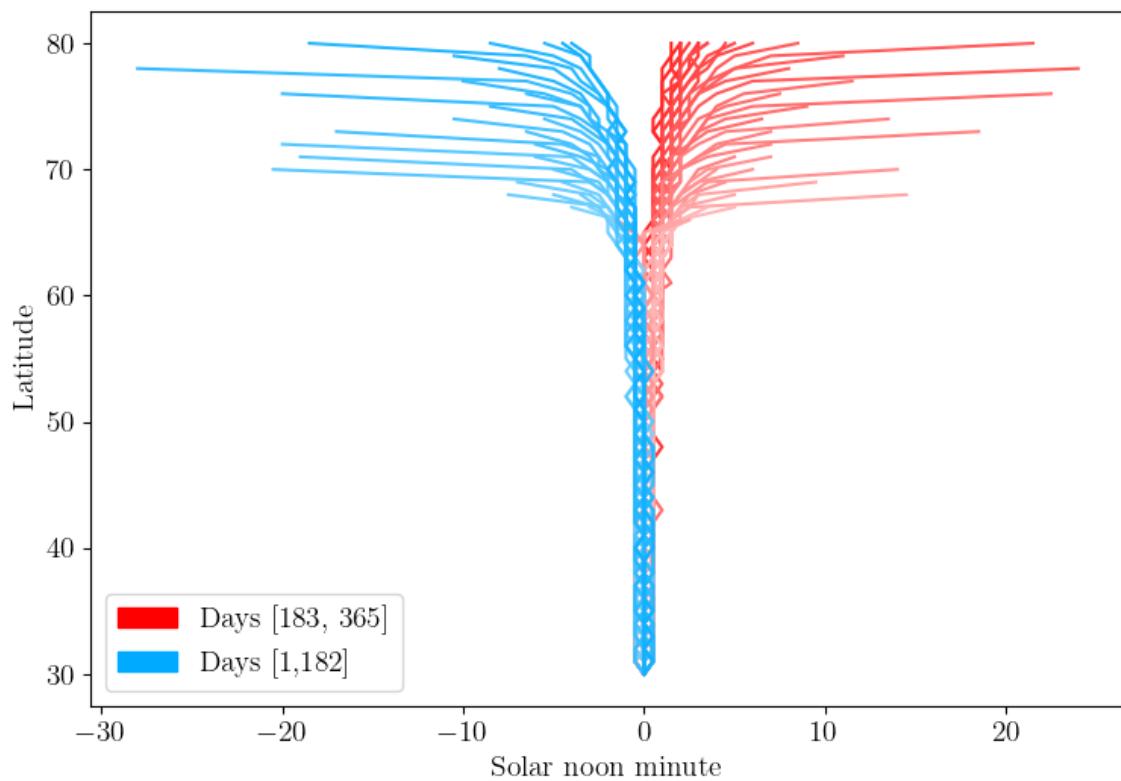


Figure 4.3: Relationship between latitude parameter and estimated solar noon time. Each line represents a different day of the year and x-axis values are normalized so that each line begins at 0 deviation. Lines with darker colors mark days which are further away from spring and fall equinoxes.

4.2 Estimating geographic latitude

Similarly to the longitude, the latitude of an installation is strongly connected to the timing of the first and last non-zero minutes of the day. In figure 3.4, the simulated first and last minutes can be seen to change day by day at varying rates based on the latitude. In mathematical terms it could be said that the derivative of the day-to-first-minute function is determined by the latitude of the installation. And for the days around equinoxes, and at higher latitudes of 50° to 70° , this relationship would seem to be bijective as per earlier figure 3.4. Based on these observations, the obvious method for solving latitudes would be the following algorithm.

4.2.1 Latitude algorithm

1. Simulate first non-zero minutes over a specific day range at latitude l .
2. Fit a linear equation to the simulated day to first minute pairs from step 1.
3. Repeat steps 1 and 2 for every relevant latitude, graph linear equation slopes from step 2 as X-axis values and latitudes as Y axis values.
4. Fit an n-degree polynomial equation to the graph from step 3.
5. If the day length change can be inferred from PV installation power output, this rate of change can then be used as the input for the polynomial from step 4 and the value of the polynomial should be the geographic latitude.

Notes: Due to seasonal differences in data quality, polar winters and the midnight sun, the range of days chosen for the algorithm is important. If the range is short, individual outliers in measurements can result in large errors. Whereas if the range is too long, it will be harder to choose the range while avoiding low data quality sections. In the algorithm visualization figure 4.4, the range of 250th to 300th seems to result in acceptable slope to latitude curve smoothness.

4.2.2 Latitude algorithm visualization

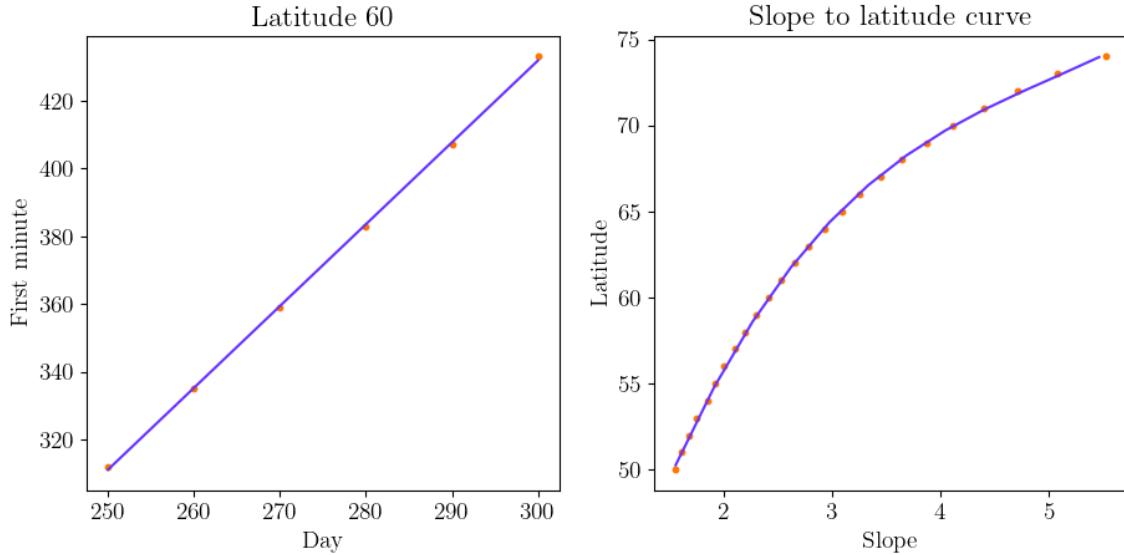


Figure 4.4: Graph on the left shows the almost linear relationship between day and simulated first minutes. Second graph shows the relationship between the slope angle and latitude.

FMI Kumpula		
Year	Predicted latitude	Error
2021	61.365°	1.161°
2020	64.493°	4.289°
2019	63.121°	2.917°
2018	61.190°	0.986°
2017	57.515°	-2.789°

Table 4.2: Results from estimating the latitude of FMI Kumpula PV installation with the preceeding algorithm. Day range of 250th to 300th was used.

4.2.3 Improving the algorithm

The results of the algorithm shown in table 4.2 are somewhere in the correct range, but the delta of over 4° in the 2020 estimate is significant and much higher than

the error of the longitude estimation algorithm. The first step in improving the algorithm would be the use of last non-zero days as well as the first non-zero days. This doubles the amount of outputs from the algorithm and while doubling the amount of outputs does not directly increase the accuracy of the algorithm, it can provide additional insights into the performance of the algorithm. This is especially valuable as the available datasets are small.

FMI Kumpula				
Year	First min. p.	Error	Last min. p.	Error
2021	61.365°	1.161°	63.685°	3.481°
2020	64.493°	4.289°	64.288°	4.084°
2019	63.121°	2.917°	66.762°	6.558°
2018	61.190°	0.986°	60.230°	0.026°
2017	57.515°	-2.789°	62.256°	2.052°

Table 4.3: Latitude algorithm with added output for last minutes based prediction.

The second step in improving the algorithm is choosing the best possible day range for latitude estimation. One way of choosing the day ranges would be by testing multiple day ranges and choosing the range which results in the lowest average absolute error from the known latitude. While this would result in a circular proof, by using multiple datasets from different geographic regions, the method could be used to find universally well behaving day ranges which could then be used for datasets with unverified or unknown coordinates.

Standard deviation minimization is the second option for automated day range selection. As there are two estimated latitude values per year, datasets with n years of data would provide $n * 2$ estimated latitude values. Standard deviation of these values could be expected to be small if the day interval does not contain days with bad data quality and this means that the interval selection can be automated. However exhaustively searching the day interval space with reasonable intervals can be slow and as seen in the figure 4.5, low standard deviation intervals are common in the parameter space. This same figure indicates that an arbitrarily chosen long interval is likely to perform almost as well as the very best algorithmically chosen interval.

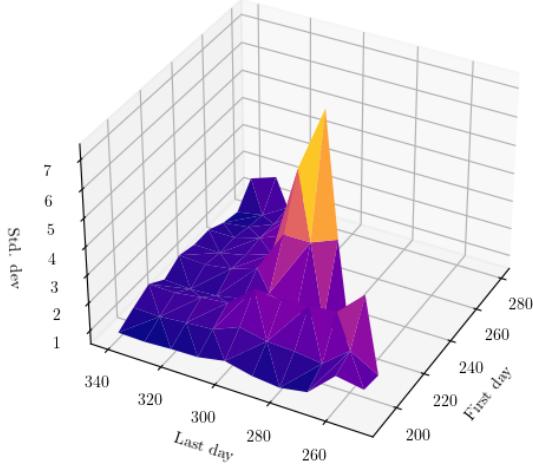


Figure 4.5: 3D -surface where X and Y -axis correspond to the interval star and end days and Z -axis marks the standard deviation of latitude estimations when the interval is used to predict latitudes from FMI Kuopio dataset.

4.2.4 Latitude estimation results

The following two tables contain examples of the results of the latitude estimation algorithm. Results of the latitude estimation algorithm are not as good as the longitude estimations, but for now they will suffice. The predictions follow a similar pattern as the previous longitude estimations in that predictions for the Helsinki installation are grouped tighter and their errors are lower than those of the Kuopio installation.

FMI Helsinki latitude estimation results				
Year	First min. p.	Error	Last min. p.	Error
2021	59.792°	-0.677°	60.186°	-0.334°
2020	59.792°	-0.412°	60.186°	-0.018°
2019	59.896°	-0.308°	59.558°	-0.646°
2018	59.945°	-0.259°	59.463°	-0.741°
2017	60.577°	0.373°	60.008°	-0.196°

Table 4.4: Estimated latitudes for FMI Helsinki Kumpula dataset with day range of 190th to 250th

FMI Kuopio latitude estimation results				
Year	First min. p.	Error	Last min. p.	Error
2021	62.626°	-0.266°	63.197°	0.305°
2020	62.259°	-0.633°	61.895°	-0.997°
2019	62.983°	0.091°	62.708°	-0.184°
2018	62.722°	-0.170°	62.874°	-0.018°
2017	61.669°	-1.223°	61.152°	-1.740°

Table 4.5: Estimated latitudes for FMI Kuopio Kumpula dataset with day range of 190th to 280th.

4.2.5 Possible issues and further development ideas

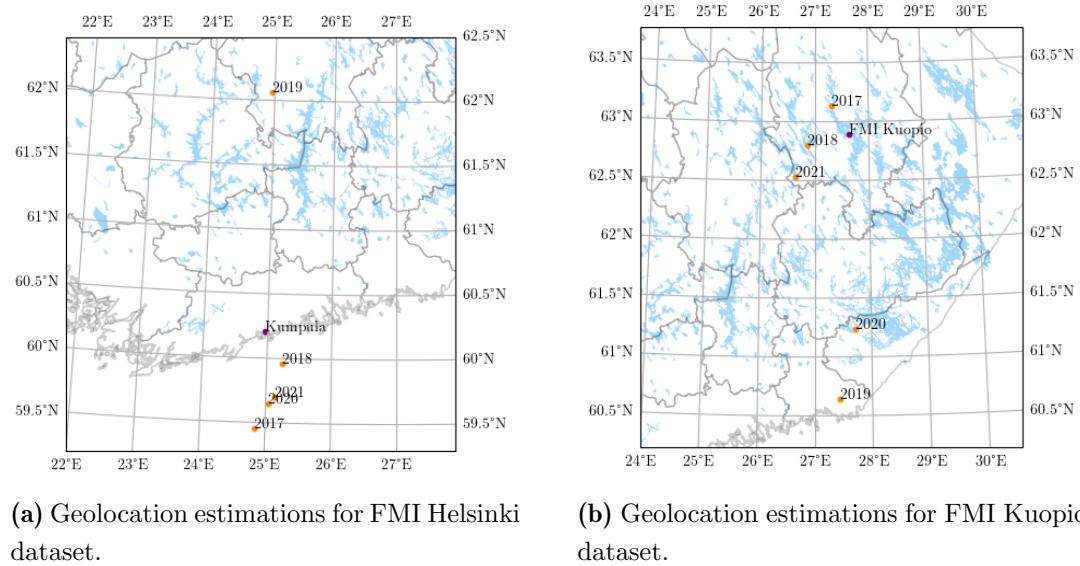
PVlib POA based first and last minute estimations are slow to compute and thus exhaustively searching the day interval space can take up to several hours of computing time. As only the first and last minute times are needed, the use of simpler computational methods could improve the computation speed significantly, allowing for the use of brute force day range selection algorithms.

Different methods could also be used. In Hagdadi 2017 [1] latitude estimations are done by fitting solar irradiance models with 3 unknown parameters to power generation measurement data. The latitude deltas of 1.65 to 3.42 degrees in the 2017 article are higher than those achieved by this paper, however as the datasets, geographical regions and algorithms are different, direct comparison can not be made.

In earlier figure 4.4 the slope to latitude fitting can be seen to be slightly off. This is because the polynomial used is of 2nd degree and higher degree polynomials may result in a closer fit. Similarly a piecewise linear interpolation based fitting could result in a more accurate model.

4.3 Combined latitude and longitude estimations

As it is unlikely that the longitude and latitude estimation algorithms are used in isolation from one another, their results should be examined together. This can be done by plotting the estimated locations on a map. Here the two installations in Helsinki and Kuopio and their predicted locations per year are plotted side by side.



In the Helsinki predictions figure, the estimated geolocations are grouped together with the exception of the year 2019 and a similar pattern is visible in Kuopio predictions where 2017, 2018 and 2021 are grouped together while 2019 and 2020 deviate. One degree on the latitude axis is approximately 110 km regardless of latitude and longitude, one degree of longitude is 56km at 60° N and 50 km at 63° N. As the deviation is strongest on the latitude axis, it is likely that the latitude prediction algorithm is more sensitive to variations in the data and further development should be focused on more accurate latitude prediction.

CHAPTER V

Estimating panel angles

Solar panel installation angles are a large factor in deciding the energy output of a PV system. If panel angles can be freely chosen during planning and installation phases, it can make sense to either optimize for total power generation or power generation during peak consumption hours. This means that even if installation angles could be freely chosen, installation angles are unlikely to be the same for every system in the same geographical region. Panel angles may also be restricted by installation sites and mounting types.

Panel angles can be difficult to measure accurately. The tilt angle of the panels or the angle between the panel normal and zenith(the point directly above) can easily be measured with an angle ruler and a bubble level, but the azimuth angle of the panels is much harder to measure with the same degree of accuracy. If an accurate compass is used and the difference between the magnetic north and the geographic north is taken into account, metal structures and electrical systems nearby can still distort local magnetic fields enough to cause errors in measurements. The challenges in taking accurate measurements are not insurmountable, but they may contribute to the inaccuracies and the lack of available information in PV installation parameter metadata.

The space of possible panel installation angles can be thought as a half unit sphere in a spherical coordinate system where each point on the surface represents a direction to which the normal of the solar panels could be directed towards. A visualization of parameter space in 3D and 2D is shown in [5.1a](#) and [5.1b](#). The 3 dots in the subfigure b) mark the zenith for which azimuth is not well defined(red), the installation angles of FMI Helsinki installation azimuth 135° tilt 15° (blue) and

a close to power generation maximized installation with directly south facing panels with the tilt of 45°(black).

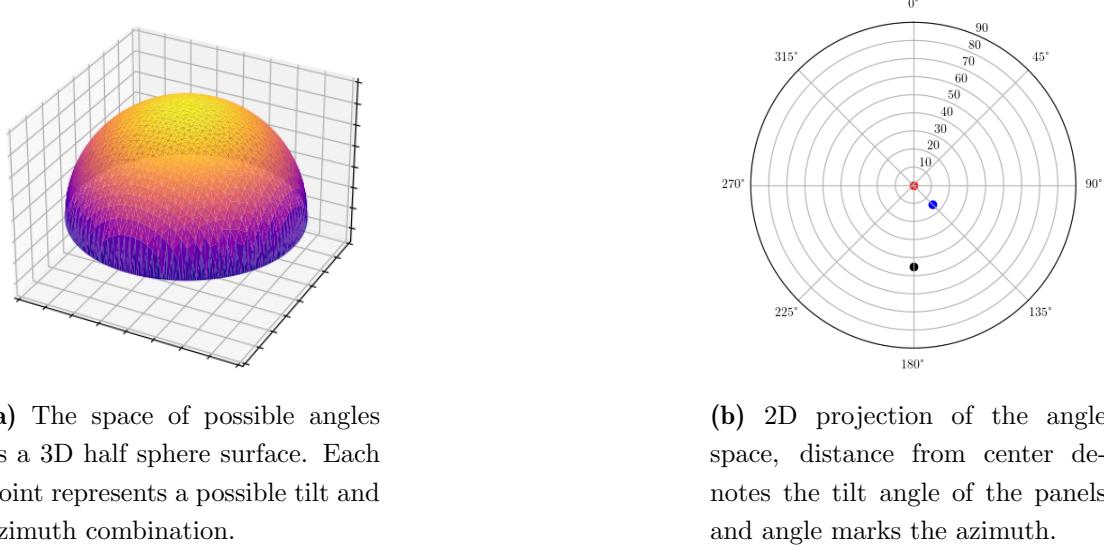


Figure 5.1: Angle space visualizations.

Estimating panel installation angles requires the use of multiple functions, each of which can be defined in multiple ways. These functions are defined in the following sections.

- Prediction error function for quantifying how good a prediction was when the correct panel parameters are known.
- Model error function for measuring the error between simulated power values and measured power values.
- Multiplier matching function for matching the magnitude of simulated power values with the magnitude of measurements. This does not change the shape of simulations.
- Angle space discretization function for discretizing the angle space into n discrete points which can then be tested with model error function.

5.1 Prediction error function

In this thesis, the proposed error estimation method combines the tilt and azimuth delta values into one error angle value, the angular distance between two points on a spherical surface. The goal is then to develop a panel angle estimation function which results in the lowest angle error value with the available datasets.

Alternative approaches can also be chosen as the function or functions for measuring the distance between two points in angle space can be defined in multiple ways. The simplest way is to use the delta of known tilt and azimuth angles as two separate error values without normalizing in any way. This method was used in Haggadi's 2017 article but such values are not directly comparable between installations as the significance of azimuth delta depends on tilt angle.

Deriving angle space distance equation

Let $v = [v_1, v_2]$ and $k = [k_1, k_2]$ be two component angle-space vectors so that $v_1, k_1 \in [0, 90]$ and $v_2, k_2 \in [0, 360]$. These vectors represent points on the surface of a unit sphere and their components are the angles of spherical coordinate system. The cartesian coordinates of these points are:

$$x_v = \sin(v_1)\cos(v_2) \quad (5.1)$$

$$y_v = \sin(v_1)\sin(v_2) \quad (5.2)$$

$$z_v = \cos(v_1) \quad (5.3)$$

And

$$x_k = \sin(v_1)\cos(v_2) \quad (5.4)$$

$$y_k = \sin(v_1)\sin(v_2) \quad (5.5)$$

$$z_k = \cos(v_1) \quad (5.6)$$

And the cartesian distance between these two points can be calculated with the following equation:

$$d = \sqrt{(x_v - x_k)^2 + (y_v - y_k)^2 + (z_v - z_k)^2} \quad (5.7)$$

The two points and the origin form an isosceles triangle with the sides from the origin to the vector end points having the length of 1 while the distance between the vector end points is the same as d .

As the lengths of three sides are known, the angles of the triangle can be calculated with the cosine rule.

$$a^2 = b^2 + c^2 - 2bc \cos(A) \quad (5.8)$$

Where

a = Side opposing the angle A, same as earlier value d

b = Side opposing angle B, value is 1

c = Side opposing angle C, value is 1

Substituting known values into the cosine equation.

$$a^2 = b^2 + c^2 - 2bc \cos(A) \quad (5.9)$$

$$d^2 = 1^2 + 1^2 - 2 \cos(A) \quad (5.10)$$

$$d^2 = 2 - 2 \cos(A) \quad (5.11)$$

Solving for angle A

$$d^2 = 2 - 2 \cos(A) \quad (5.12)$$

$$2 \cos(A) = 2 - d^2 \quad (5.13)$$

$$\cos(A) = \frac{2 - d^2}{2} \quad (5.14)$$

$$A = \cos^{-1}\left(\frac{2 - d^2}{2}\right) \quad (5.15)$$

Renaming A as *Error*.

$$Error = \cos^{-1}\left(\frac{2 - d^2}{2}\right) \quad (5.16)$$

By first calculating the distance between the vectors by using equations 5.1-5.7 and then substituting the distance into equation 5.16, the resulting angle can then be used as an error value between two panel angle measurements.

5.2 Simulation error function

Simulation error function measures how much the predicted power generation values vary from the measured power generation values. The purpose of the simulation error function is to be able to generate a single numerical value which describes how well a certain parameter combination models the measurements. By then testing out multiple parameter combinations, the combination with the lowest simulation error function value should be the best fit and the parameters used for the simulation should be within a small error of the physical parameters of the solar PV installation. In 5.2 the error between a cloud free day and randomly chosen set of wrong simulation parameters is visualized.

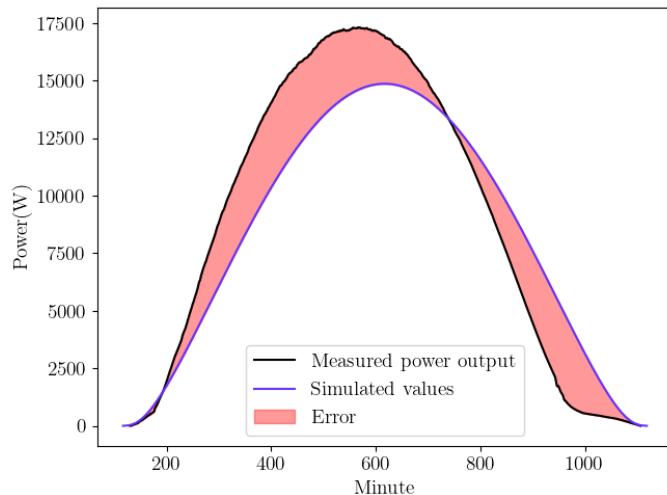


Figure 5.2: Error area between measured power values and simulated irradiance values with different panel installation angles.

5.2.1 Area based error function

The following function can be used in order to get a numerical value which signifies the area between simulated and measured power values.

$$Error = \sum_{t=0}^{1439} |m(t) - p(t)| \quad (5.17)$$

Where $m(t)$ is the measured power at minute t and $p(t)$ is simulated power at minute t .

The error values can also be normalized so that the error value reflects the average percentual deviation between the measured and modeled values.

$$Error = \frac{\sum_{t=a}^b m(t)/p(t)}{b - a} \quad (5.18)$$

Where $m(t)$ is the measured power at minute t and $p(t)$ is simulated power at minute t .

5.2.2 Alternative simulation error functions

Other error estimation methods could be used as well. Skew values, peak power generation minutes and other similar characteristics could be measured and matched. The benefit of such characteristics matching based error algorithms is that characteristics errors have a direction as well as magnitude and these could be used in order to estimate the proximity and direction of the best fit in angle space.

The downsides of using characteristics are that measurement data contains noise which can distort the estimated values of characteristics and having a direction and a magnitude is of limited value unless additional functions are created for turning these error values into directions in parameter space. These relationships between traits, their error directions and magnitudes could prove to be difficult to solve.

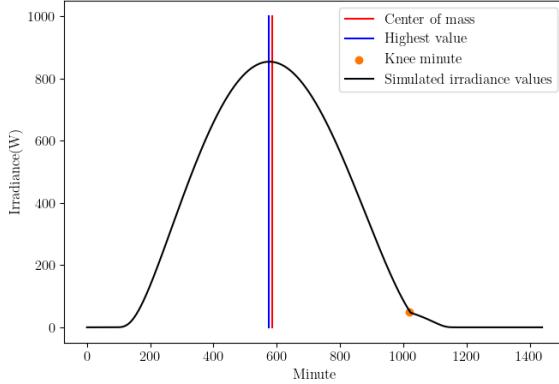


Figure 5.3: Center of mass, highest value and knee minute marked on a simulated irradiance figure.

5.3 Multiplier matching

As the plane of array irradiance model models the solar radiation towards a virtual 1 m^2 sized panel, using the values to simulate the power generation of a solar power installation requires the use of a multiplier. This multiplier is related to the efficiency and the surface area of the solar PV installation.

5.3.1 Area based multiplier matching

The multiplier value can be solved by making the assumption that a plane of array irradiance curve $p(t)$ with correct simulation parameters matches the measurements curve $m(t)$ in its shape but not magnitude. By calculating the sum of simulated and measured power values, their ratio can then be used as a multiplier. After scaling $p(t)$ with a multiplier to match $m(t)$ the two curves should be nearly indistinguishable.

$$M_{multiplier} = \frac{\sum_{t=0}^{1439} m(t)}{\sum_{t=0}^{1439} p(t)} \quad (5.19)$$

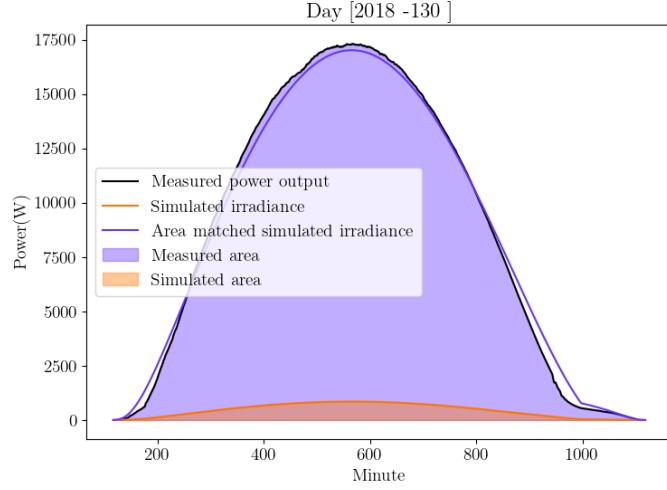


Figure 5.4: Visualization of automated multiplier matching. Measurement data from FMI Helsinki dataset, plane of array irradiance simulation was computed with FMI Kumpula coordinates and installation angle parameters.

5.3.2 Segmented multiplier matching

The earlier multiplier matching method falls apart if the measurements used for multiplier solving contain deviations caused by clouds, trees or other sources. If the energy production during a certain section of the day varies from the expected, then this abnormal segment will affect the resulting multiplier value. If these deviating segments could be avoided, then partially cloudy days could still be used for multiplier matching. The process of segmentation can be done with multiple different methods. Here the interval between the first and the last non-zero power minute of the day was split into 10 segments of equivalent length.

Now that the data is split into segments, the earlier area based multiplier algorithm can be used to compute a multiplier value for each segment. A mathematical representation for the segments S_i multiplier M_i could be as follows.

$$M_i = \frac{\sum_{t=a_i}^{b_i} m(t)}{\sum_{t=a_i}^{b_i} p(t)} \quad (5.20)$$

Where a_i and b_i are the first and last minutes of the interval i .

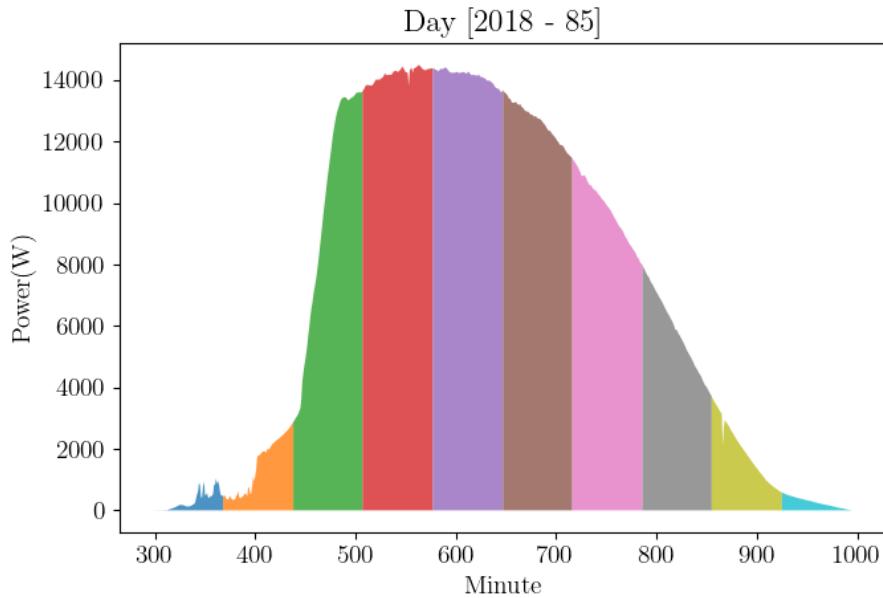


Figure 5.5: Partially cloudy day split into 10 segments.

Interval	Multiplier
[299, 367]	1.26
[368, 437]	3.22
[438, 506]	16.56
[507, 576]	21.68
[577, 646]	21.58
[647, 715]	21.52
[716, 785]	21.23
[786, 854]	19.80
[855, 924]	16.08
[925, 994]	16.80

Table 5.1: Segment based multiplier matching intervals and resulting multipliers. FMI Kumpula dataset day [2018 - 85] was used with irradiance simulation parameters listed in [2.2](#).

In [5.1](#) the segments 4 to 7 are all tightly grouped and their average is 21.50. These closely grouped segments, commonly referred to as clusters in the field of data science, can be identified algorithmically by locating a window that encom-

passes a relatively high number of values within it. For example a $\pm 5\%$ window $[M_i * 0.95, M_i * 1.05]$ would contain the cluster if any of the 4 cluster values was used as as M_i .

A possible issue with the clustering algorithm could rise from random chance. The last multiplier values 16.08 and 16.80 are close enough to the early 16.56 that due to random variation in the data, the algorithm could in some cases choose the wrong multiplier value cluster. The low values are unexpected as the plot appears to be free from cloud interference and it is possible that the deviation from the expected values occurs due to increased reflectivity of the solar panels resulting from a higher angle of incidence. This issue could be mitigated to some extent by using a more advanced solar power generation simulation algorithm.

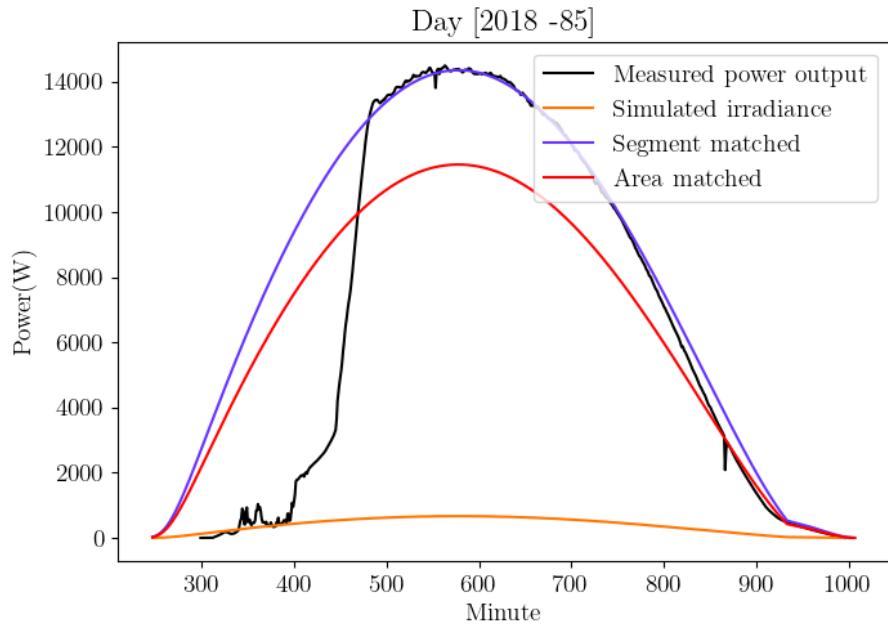


Figure 5.6: Comparison of area and segmented area based multiplier matching algorithms on a partially cloudy day.

5.3.3 Translating multiplier values to PV system rated power values

As the simulated irradiance values are an estimation of power radiated towards a single square meter sized imaginary panel, the scaling multiplier can be used for estimating the power rating of a solar PV installation. The power ratings of PV installations describe the expected power generation in watts that could be expected to be generated during peak power generation hours if the panels were oriented optimally.

With the help of the rated power value, the surface area of the panels can be estimated. The precise estimation is difficult as different panel types and panel age affect the efficiency of PV systems, but having some frame of reference for installation sizes could prove to be useful in other studies where lidar data or satellite images are available.

Rated power estimation equation

$$\text{RatedPower} = MP \quad (5.21)$$

Where M is the multiplier required to match the POA simulated values with measured values and P is an estimation for solar irradiance per square meter or approximately 1000w.

By using 5.21 and the clustered multiplier values from 5.1, the power rating of the Helsinki installation would be estimated as 21.5KW which is within close proximity the reported 21KW in 2.2.

Panel area estimation equation

$$\text{PanelArea} = \text{RatePower} * \frac{1m^2}{\eta} \quad (5.22)$$

Where η is the efficiency of solar panels, typically in the range of 0.15 to 0.20. The efficiency coefficient varies according to panel type, age, variance in manufacturing and other factors.

5.4 Angle space discretization

The next step is angle space discretization. The panel angles are denoted with a doublet of tilt and azimuth values, ranging from 0 to 90 and 0 to 360 respectively. If the tilt and azimuth axes are discretized individually in steps of 5 so that tilt is [0, 5, 10, 15... 90] and azimuth [0, 5, 10, 15... 355], the permutations of these tilt and azimuth values create an even grid in the euclidean projection of angle space where $x = \text{tilt}$, $y = \text{azimuth}$. However as the physical phenomena represented by the angle values is not a point on a plane but a point on a half-sphere surface, this results in an uneven discretization 5.7a. A better option is to use Fibonacci lattice [5] for a more even distribution of points on a spherical surface 5.7b.

Fibonacci lattice point n of k equation

$$s = n + 0.5 \quad (5.23)$$

$$\phi = a\cos(1 - 2s/k) \quad (5.24)$$

$$\theta = \pi s(1 + \sqrt{5}) \quad (5.25)$$

Where n is the point number, k is the amount of points, ϕ is the panel tilt angle and θ is the azimuth angle.

$$x = \cos(\theta)\sin(\phi) \quad (5.26)$$

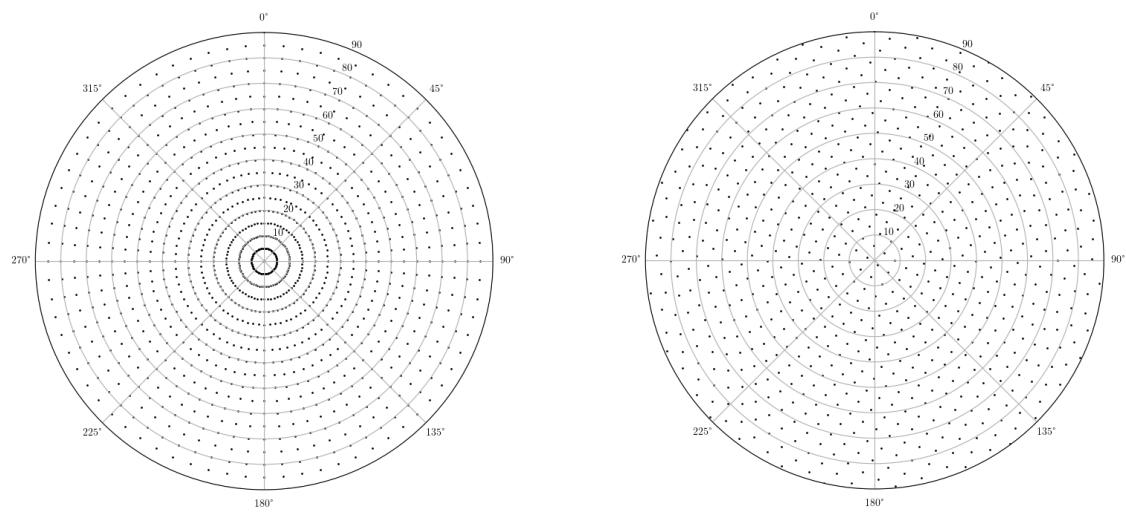
$$y = \sin(\theta)\sin(\phi) \quad (5.27)$$

$$z = \cos(\phi) \quad (5.28)$$

x , y and z are the corresponding cartesian coordinates.

5.4.1 Importance of lattice density

The density of lattices is an important measurable characteristic which proves useful during the optimization of angle estimation algorithm. The most useful metric would be the sphere center angle distance between neighboring points. This would be useful as it can be used for determining whether errors in predictions are lattice or function fitting related. For example, if the lattice neighbors are approximately 1 degree away from one another and the predicted angle is 5 degrees off from the known installation angle, then the error is caused by model fitting and not lattice density as there must



(a) In steps of 5 discretization with 1296 points

(b) Fibonacci lattice-based discretization with 756 points.

Figure 5.7: Comparison of two different discretization patterns. Fibonacci lattice based discretization on right shows a more even distribution of points than the latitude-longitude lattice. The minimum density is approximately the same in both graphs despite the difference in point counts.

have been multiple lattice points closer to the known angle point than the discovered best fit. However if grid density is near to or lower than angle estimation error, the lattice is likely to be a contributing to angle estimation errors.

Calculating neighbor center angle distances for both *in-steps-of-n* and Fibonacci lattices is somewhat challenging. In *in-steps-of-n*, the value n can be used as an estimate for max center angle distance as n will always be the tilt distance to the nearest neighbor with different tilt angle. With Fibonacci lattices the easiest way of estimating center angle distances is taking the coordinates of the first two lattice points and calculating their center angle distance with earlier error equation 5.16. These first two points should be used as Fibonacci lattice points are distributed on a single arm spiral pattern, resulting in later sequential points being further from one another.

Another method for calculating Fibonacci lattice point distances is dividing the surface area of the angle space by the amount of calculated lattice points. This area-per-point value could then be used in order to estimate how far points are from one another on average. In later sections, the first two points derived distance will be used.

5.5 Solving panel angles

Now that the geographic location and multiplier value of installation are known to be solvable and error functions have been defined, the last step is to solve the panel installation angles. The chosen method relies on splitting angle space into n discrete points and evaluating each of their fitness by calculating an error value. Here the angle space was split into 10 discrete points with fibonacci lattice 5.4 and the fitness of each point was evaluated with area error 5.17 and multiplier matching 5.19 functions.

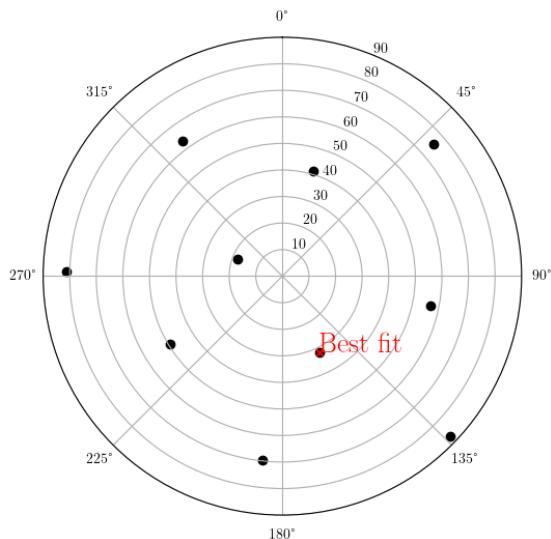


Figure 5.8: Polar plot of test points for a single day of data from FMI Kumpula dataset.

Tilt	Azimuth	Error
18.19	291.25	3785693.23
31.79	153.74	265843.23
41.41	16.23	4331163.39
49.46	238.72	4861757.21
56.63	101.22	3811371.79
63.26	323.71	7196389.58
69.51	186.2	1795822.66
75.52	48.69	5985516.23
81.37	271.18	7403298.37
87.13	133.68	2957569.38

Table 5.2: Tilt, azimuth and error table for single day.

Now that the method can be seen to work, it is time to improve the results. This can be done by generating larger lattices and thus by evaluating higher amount of datapoints, the algorithm has a higher chance of finding the global minimum error point. As per 5.4.1, Fibonacci lattice of 1000 points would have the angular resolution of approximately 4 degrees where as 10000 points would be near to 1.5 degrees. The performance can also be improved by evaluating best fits for multiple days at once. The resulting point cloud of best fits can then be used for averaging out noise in the predictions.

The plot 5.9 is a result of using the angle finding algorithm on 41 days from FMI Kumpula dataset with a Fibonacci lattice of 1000 points. The two darker spots near

the center of the graph are the two most common best fits, $[17.0^\circ, 138.4^\circ]$ with 17 and $[22.7^\circ, 143.1^\circ]$ with 16 out of 41 days. These groupings are as close to the known installation angles of $[15^\circ, 135^\circ]$ as could be expected from a 1000 point lattice. The next step is tightening the cluster, this can be done by adjusting the smoothness requirement of the cloud free day algorithm or by restricting the day range. In 5.10 the tightening was accomplished with day range restrictions and 22 days were accepted by the algorithm.

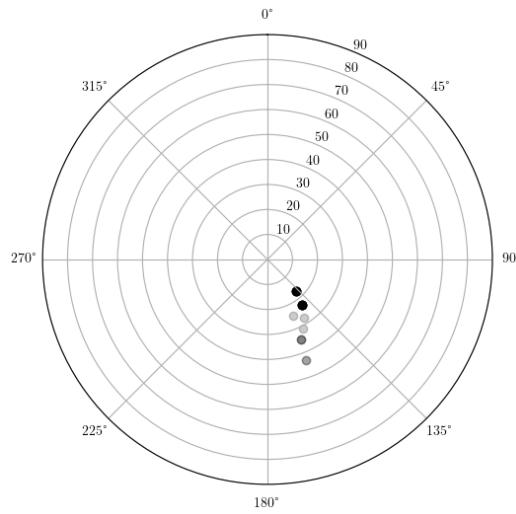


Figure 5.9: Best fits for multiple cloud free days from FMI Kumpula dataset.

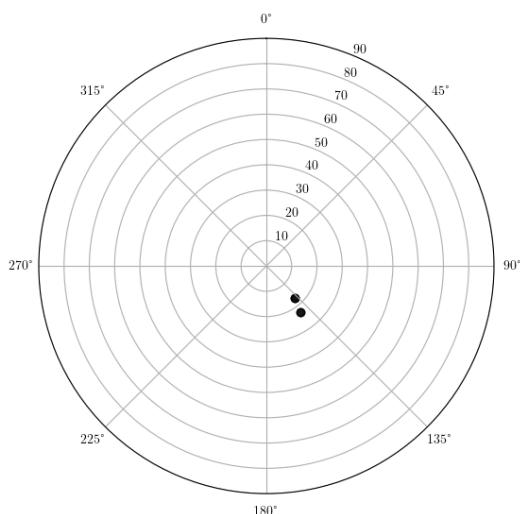


Figure 5.10: Best fits when day range is restricted to days 140-220.

As all of the estimates converge on two neighboring points, the final step is increasing the lattice resolution further. With 10 000 point lattice, the cluster tightened further 5.11. Out of the 22 days, 7/22 or 32% had best fit at $[19.34, 136.87]$ with error of 4.38 degrees and 6/22 or 27% at $[17.74, 135.06]$ with error of 2.74 degrees. Rest of the best fits were distributed in smaller clusters near these points. The lowest angle distance best fit was $[17.09, 130.32]$ with angle distance of 2.46 degrees. As the angle distance errors are higher than the discretization resolution and as the predicted angles are systematically biased, it would seem that the error is caused by the solar irradiance model or model fitting and not angle space discretization.

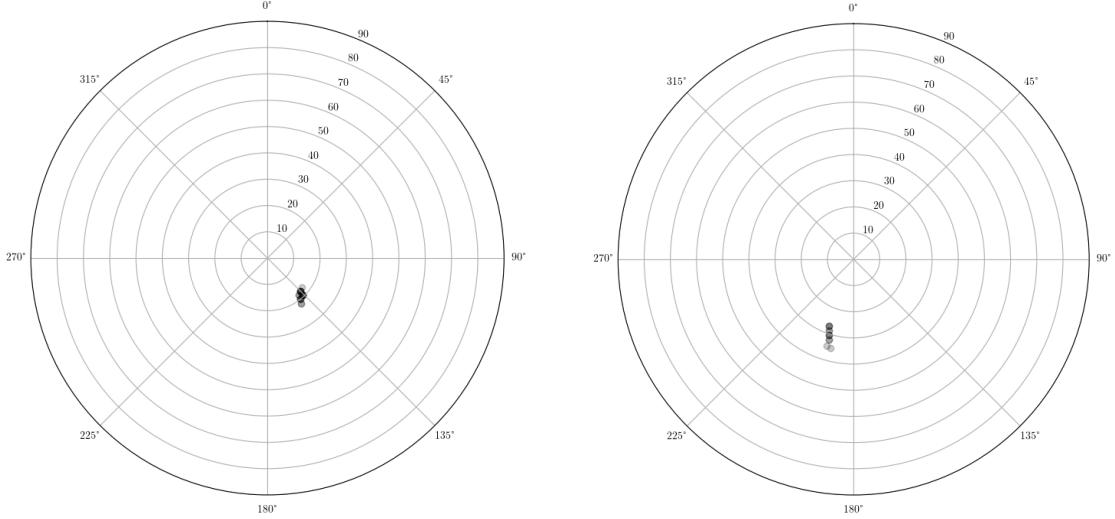


Figure 5.11: Best fits with 10 000 sample Fibonacci lattice and 22 days from FMI Kumpula dataset. Multiple years were used, day range restricted to 140-220. Actual installation angles were 15 degrees tilt, 135 azimuth.

Figure 5.12: 10 000 sample Fibonacci lattice and 16 days from FMI Kuopio dataset. Multiple years were used, day range restricted to 140-220. Actual installation angles were 15 degrees tilt, 217 azimuth.

5.5.1 Evaluation of results

The estimated installation angles installations are fairly good. A delta of less than 4.5° as was achieved with FMI Kumpula is small enough to be a result of measurement or rounding error. The estimates for FMI Kuopio are off by more than 10 degrees which is less encouraging. As the reported angles for FMI Kuopio were 15° and 217° , it would seem like the angle measurements were rounded to nearest degree. This would eliminate the reporting accuracy as a plausible cause for the estimation errors and thus either there has been a reporting error or that the installation angle estimation algorithms are not performing as well for FMI Kuopio dataset.

Figures 5.13 and 5.14 shows that the models based on best found fits are better fits than simulations done with the known parameters. This is true for both Helsinki and Kuopio datasets. This would suggest that the model fitting works as intended and that either the model is inaccurate or there is an error in reported panel angles. The more likely cause of the two is the the solar irradiance model and for some undetermined reason the error of the model is more significant for the Kuopio

FMI Kumpula			
n	Tilt	Azimuth	Error
7	19.34°	136.87°	4.38°
6	17.74°	135.06°	2.74°
4	19.92°	141.60°	5.30°
2	21.37°	143.41°	6.87°
1	21.37°	143.41°	6.87°
1	17.09°	130.32°	2.46°

Table 5.3: Estimation results table for FMI Kumpula.

FMI Kuopio			
n	Tilt	Azimuth	Error
3	27.07°	200.24°	13.37°
3	30.50°	198.01°	16.96°
2	28.83°	199.13°	15.21°
2	32.09°	196.89°	18.65°
1	34.52°	197.58°	20.90°
1	34.52°	197.58°	20.90°
1	35.07°	194.65°	21.86°

Table 5.4: Estimation results table for FMI Kuopio.

installation. Possible causes could be related to lower sun angles resulting in higher reflective losses or shadowing during last production hours which would also explain the uneven structure visible in the last non-zero hours in [5.14](#).

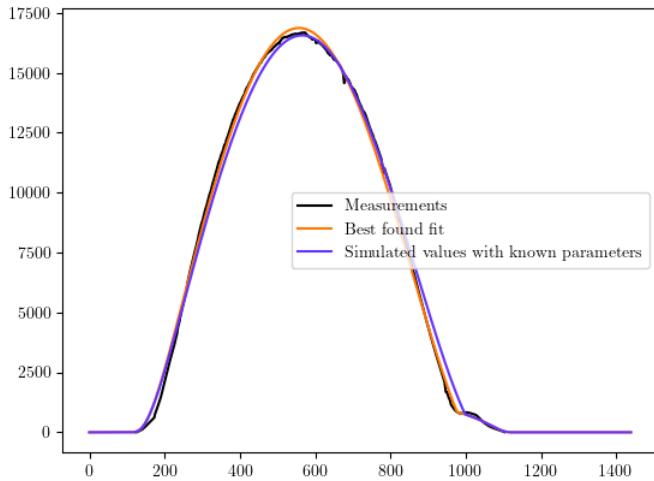


Figure 5.13: Comparison between a single day of measurements and two simulations. One with known installation angles and one with best found fit. Day is from FMI Kumpula dataset with reported installation angles of 15° and 135° degrees. Best fit at 19.34° , 136.87°

5.5.2 Possible issues and further development ideas

The algorithm steps of cloud free day detection and lattice generation are computationally fast but evaluating a high density lattice for multiple days can take several minutes. For example, the time required for the generation of 5.10 consist of 9 seconds of data loading and preprocessing, 1 second of cloud free day detection and 11 minutes of angle pair evaluation. With 22 days and 1000 points per day, the resulting 22 000 evaluations were done at the average speed of 33 evaluations per second. This comparably long processing time is due to inefficient code but the algorithms speed can also be improved by more intelligent latticing. The general location of the best fit could be solved with a low density lattice and a local high density lattice could then be used to estimate a higher accuracy angle pair. Combining code optimizations and localized angle space lattices could reduce the computation time significantly.

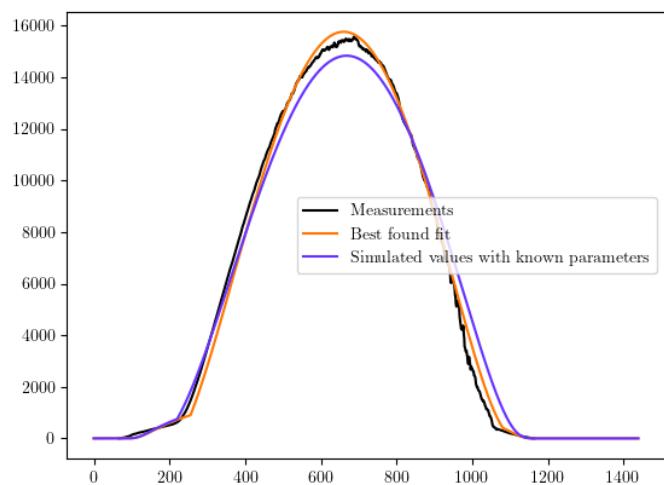


Figure 5.14: Comparison for FMI Kuopio installation. Panel angles are 15° and 217° degrees. Best found fit at 19.34° , 136.87°

5.6 Final remarks

The initial goal of this thesis was to find a simple way of estimating parameters of solar PV installations and this goal has been accomplished with moderate success. Some of the algorithms are thousands of lines of long, but the underlying mathematics was still kept simple. As a result, the code can be understood and modified by a wider audience. This is in particular contrast with AI and machine learning based approaches which often provide good results but which tend to be less insightful.

From the perspectives of mathematics and programming, model fitting problems are not particularly difficult. In this thesis, the challenges rose from optimization, understanding patterns in the data and discovering where the limits of the estimation algorithms come from. The insights gained while tackling these issues may be more valuable to other researchers than the final estimation algorithms. Examples of such insights are the center angle error function, Fibonacci-lattice based angle space discretization and angle space resolution estimates.

REFERENCES

- [1] N. Haghdadi, J. Copper, A. Bruce, and I. MacGill, “A method to estimate the location and orientation of distributed photovoltaic systems from their generation output data,” *Renew. Energy* **108**, 390–400 (2017).
- [2] M. K. Williams, S. L. Kerrigan, and A. Thornton, “Automatic detection of PV system configuration,” *Proceedings of World Renewable Energy Forum* (2012).
- [3] H. Böök., Poikonen, A., Aarva, A., Mielonen, T., Pitkänen, M.R.A., Lindfors, and A.V., “Photovoltaic system modeling: A validation study at high latitudes with implementation of a novel DNI quality control method.,” *Sol Energy* **204**, 316–329 (2020).
- [4] V. Vajpai, “Analemma: The Sun’s Journey In The Sky,” <https://ysjournal.com/astrophysics/analemma-the-suns-journey-in-the-sky/> (2021).
- [5] M. Roberts, “How to evenly distribute points on a sphere more effectively than the canonical Fibonacci Lattice,” <https://extremelearning.com.au/how-to-evenly-distribute-points-on-a-sphere-more-effectively-than-the-canonical-fibonacci-lattice/> (2020).