

Novel Smartphone Based Energy Calculation of a PV Array

Using Hemispherical Images, a Python Script and LTSpice Circuit Simulation.

Björn Fridqvist Nimvik
May 18, 2022

AEBM05, Master Thesis in Energy and Building Design
Faculty of Engineering | Lund University



Student: Björn Fridqvist Nimvik

Examiner: Jouri Kanters (Energy and Building Design, Lund University)

Supervisor: Henrik Davidsson (Energy and Building Design, Lund University)

Abstract

The problem with conventional solar energy calculation for photovoltaic (PV from now on) systems is that often only solar irradiation is analysed and local shading is omitted. This thesis will develop a method that includes the influence of local shading such as chimneys, trees or buildings by performing image analysis of the sky view on site, as well as solar irradiation. By including local shading, the energy calculation will be more precise and provide a better decision base for costumers.

The goal of this degree project was to first design, build and validate a circuit simulation model of a PV array. After that to write a complementary Python script that could assign unique irradiation data to each cell in the circuit model, run a simulation for any given time period of the year and calculate the energy output. The ability to assign irradiation data to each cell is necessary in order to overcome the problem with local shading that could affect some cells but not all at any given time.

Another goal was to investigate how increasing the amount of images used to represent the array affected the energy calculation. The array model was limited to 3×3 modules, and only three significant dates were analysed (spring equinox, summer solstice and winter solstice). Several field tests were set up where 135 hemispherical images of the sky were captured in an area representing the PV array. The hemispherical images were fed into the Python script where local shading for each image were calculated and solar irradiation for every hour of the year returned. The solar irradiation for each image were applied it its corresponding cell in the circuit model, a simulation ran for three significant days and the energy for each day could be read. Using 135 images to represent the array is defined as the reference case.

To compare the difference when decreasing number of photos used, the python script was re-run in two new cases. One only using the four corner images and one only using the middle most image of the array.

The results showed that when the array was placed at 0 meters away from a 1 meter high wall a large error occurred when using either four or one image compared to the reference case, but four images gave less error. When the array was placed at 1 meter away from the wall the error was less than 5% using either 4 images or 1 image. At 2 meters away from the wall the error was less than 1% using either 4 images or 1 image.

Acknowledgements

I would like to give my biggest gratitude to my supervisor Henrik Davidsson and thank him for his enormous patience of me during this thesis. Thank you for all the help and for all the interesting conversations we had every Monday morning, discussing everything from e-mail providers to crypto-currencies, and sometimes PV array energy calculations if we had the time. I would also like to thank my dog Nikki for keeping me company at home during the pandemic, and my girlfriend Linnea helping me out whenever i ran into a problem I couldn't solve myself.

Table of content

1	Introduction	6
1.1	Background	7
1.2	Research Question	7
1.2.1	Subquestions	7
1.3	Goal	7
1.4	Limitations	8
2	Theory	9
2.1	Bypass Diode	10
2.2	PV cell circuit model	11
2.3	Ideality factor, n	13
2.4	Solar irradiation	13
2.5	Temperature	14
2.6	PV module - Windon Premium 265W	15
2.7	Typical Meteorological Year (TMY)	15
3	Method	16
3.1	Circuit model	16
3.1.1	Designing and creating PV-array circuit model	16
3.1.2	Verifying the circuit model	18
3.2	Python script	19
3.3	Field tests	22
3.3.1	Image selection settings	23
3.3.2	Locations	25
4	Results	28
5	Discussion	33
5.1	Further research	34
5.1.1	Larger arrays	34
5.1.2	Different image selection setting	34
5.1.3	Simplify circuit model	34
5.1.4	Bypass diodes in circuit model	35
5.1.5	Image interpolation	35
6	Conclusion	36
	References	37

1 Introduction

One big concerns regarding global warming is unsustainable energy production. According to International Energy Agency (IEA), less than 14 % of global energy production came from renewable sources in year 2018. (1). The incentive to invest in renewable energy sources, such as wind- or solar power, must be improved. Sweden has an ambitious energy goal that 100 % of all energy production year 2040 must be from renewable sources. To reach this goal current energy sources, mainly nuclear energy production, must be replaced by a combination of wind- and solar energy production (2).

Solar power is today only a small fraction of Swedish energy production. According to Swedish Energy Agency, only 0.4 % of produced electricity 2020 was through solar power. Wind was 12 %, hydro and nuclear was 39 % each. The remaining 10 % is combustion based production (2). There are lots of potential for growth in solar energy production, and as the price keeps decreasing more are looking to invest in a PV system (3). One way to increase the incentive to invest in solar energy production is to offer a way to accurately estimate yearly energy production before installment. That way the costumer could calculate the return of investment fairly accurately and feel secure when investing.

The one of the biggest problems with solar energy production is its sensitivity to shade. Shading on a PV array can drastically decrease its power output and in turn lower the profits of an installation for the owner. As such conventional methods of only analysing irradiation (while omitting local shade conditions) will overestimate the performance of the array and the profitability of it. Another problem is that most PV installations today are sold with a guarantee of its top effect, rather than an estimation of its full year energy production. This makes it hard for costumers to estimate how much energy they will generate over a year and if the investment will be worth the cost. A new method for calculating the energy output of a PV array, that takes local shade condition into consideration, is needed. The system should only require a smartphone, a cheap clip on fish eye lens and the software this project aims to create. A system like this will be beneficial from to all from small privately owned arrays to large scale industrial installations since the profitability could be determined before installment.

The first part of the project was created by a former student at Lund University, where he developed a method for calculating the solar irradiation, with local shade conditions accounted for, on a small area using just one hemispherical image captured on a smartphone with a fish eye lens attached (4). However if we have a large array, can one image represent the whole array? Could we instead take one image in every corner, or do we need to take one image for every cell on the array to accurately represent the actual irradiation over the array?

The goal for this thesis was to create a Python script that could take hemispherical images as input, and calculate energy output of a photovoltaic array as output using circuit simulation.

Another goal was to analyse how increasing the amount of images, and the placement of captured images, affect the energy output calculation.

1.1 Background

The price of solar PV has dropped drastically for many years and is considered, together with wind power, to be the cheapest source of energy. (3). As a result all from home owner to big scale projects are looking to generate their electricity from solar PV. However before investing in a PV-system it is crucial to determine if the investment is profitable or not and how long the payback time will be.

Many providers of PV systems will analyse location, roof tilt, roof azimuth and roof height when determining the performance. However while this gives a general idea it is still a best case scenario. Even a small shadow, such as a tree or nearby chimney could very well decrease productivity significantly. This is why a local shade analysis is important to perform before a solar PV system is installed. The problem is that the local shade analysis methods available requires special knowledge or specialized tools that ordinary people do not have, so a easy to use and more available method is needed.

The final method proposed in this project will only utilise a cheap $>180^\circ$ clip-on fish eye lens and a smartphone.

1.2 Research Question

The research question for this thesis is: Is it possible to use one or more hemispherical images captured a smartphone to calculate solar energy potential for an array of PV modules?

1.2.1 Subquestions

- Where should the sample images be captured in the grid?
- How is the energy output affected when increasing the amount of input images?
- Is it possible to use an electrical circuit simulation model together with a Python script to accurately estimate power output?

1.3 Goal

The goal for this continuation of the project is to build and verify a PV energy calculation script in programming language Python using a circuit model built in the electrical circuit simulation program LTspice. The script should be able to target every single cell in the model so that different irradiation can be applied to each cell depending on shading conditions. The input should be one, or more, hemispherical images of the sky and the output should be an approximation of power generated by the array.

1.4 Limitations

A 3×3 array of PV modules was chosen as the array as it resulted in a large enough area while still being manageable. Nine modules resulted in 540 PV cells that needed to be handled individually in Python and LTspice. Instead of capturing 540 images, one for each cell, it was decided that 135 images would be sufficient in order to save computational power when performing the shade analysis on the images.

2 Theory

The photovoltaic cell, when put together in series or in parallel, is what makes up the module and several modules makes an array, see figure (2.1). PV cells are made from layers of silicon, where one side is N-doped, with an excess of electrons, and the other side is P-doped, with a deficiency of electrons. This design is similar to transistors and diodes. When put together they become what is called a "PN-junction" that is sensitive to photon-emitting sources, e.g. the sun or a flashlight (5). A PV cell behaves practically just like a PN-junction, where hole-electron pairs are separated when the surface is hit by photons, creating a current source when they travel to recombine. Normally a cell produces around 0.6 V when exposed to an intense light source, and the current is proportional to the amount of photons that hit it. (6)

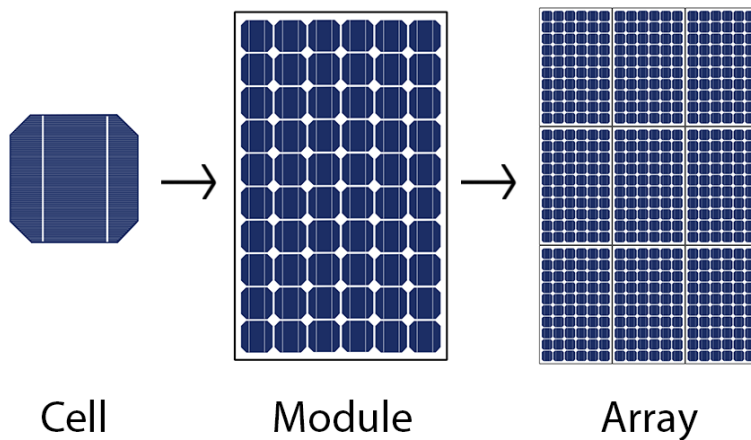


Figure 2.1: PV- cell, module and array.

Modules come in a wide verity of output ranges and the power output is determined by how many cells are in the module. The cells that are connected in series is called a string and modules are often arranged with two or more strings with a bypass diode parallel with the string. See figure (2.2).

To increase the power output to the level demanded by the user, many modules are put in series or parallel until requirements are met. The power output is determined by how many modules are in the array, not necessarily the way they are connected to each other.

2.1 Bypass Diode

The current in a string is determined by the worst performing cell, which will be the case when a cell is shaded or broken. To overcome this problem a bypass diode can be included in the circuit, which allows the current to circumvent the blockage.

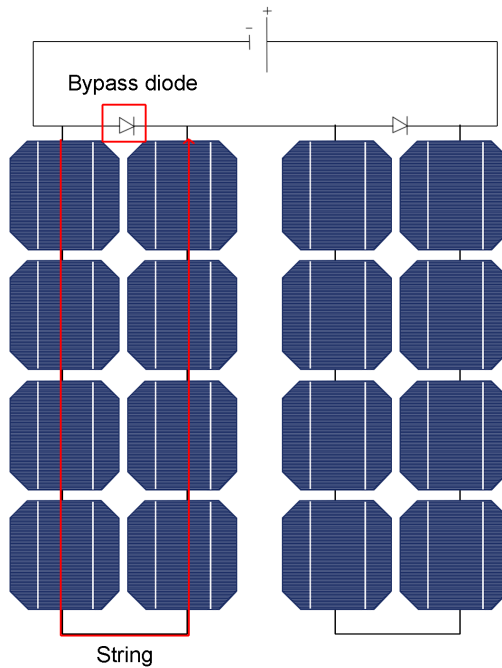


Figure 2.2: Example of a string and a bypass diode.

Ideally the bypass diode should be placed between every single cell, but due to practical and economical reasons that is never the case. The diode is instead placed between a string in order to protect other strings.

2.2 PV cell circuit model

The PV cell is modeled with a single diode, current source, shunt- and series resistance and a load, as shown in figure (2.3). The parallel resistance comes from internal error in the silicon crystals and should be very large for a functioning cell. The series resistance is the resistance in wires and connections and should be very small. The current source and the diode is determined by the materials used and the size of the cell. The load is decided by the user and could be anything from a lamp to a battery. (7).

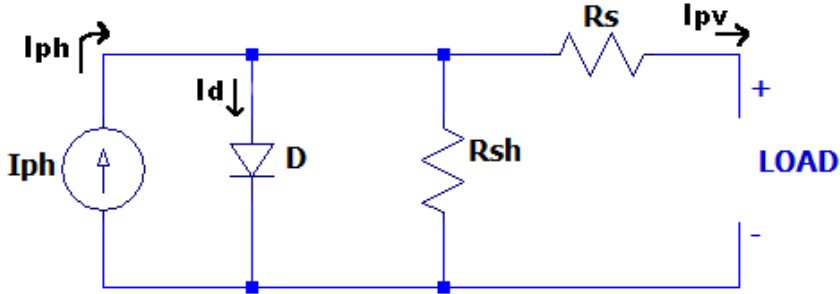


Figure 2.3: Equivalent circuit of a PV cell. ((8))

The output current, I_{pv} , of an ideal PV cell can be expressed as equation (2.1). This is the current that passes through the load. Note that the current passing through the shunt resistance has been omitted in these equations due to it being in the μA -region.

$$I_{pv} = I_{ph} - I_D \quad (\text{eq. 2.1})$$

Where I_{ph} is the photo generated current. I_D is the current running through the diode which can be replaced by Shockley's diode equation (9). See equation (2.2)

$$I_{pv} = I_{ph} - I_0 \left(e^{\frac{qV_D}{nkT}} - 1 \right) \quad (\text{eq. 2.2})$$

Where I_0 is the reverse bias saturation current in the diode, q is the electron charge, V_D is the voltage over the diode, k is Boltzmann's equation, T is the temperature and n is the ideality factor.

The reverse bias saturation current for the diode, I_0 , is also determined during testing under standard test conditions (STC), and is calculated when I_{SC} and V_{OC} is acquired, where I_{SC} is the short circuit current and V_{OC} is the open circuit voltage. See equation (2.3) (10; 7).

$$I_0 = \left[\frac{I_{SC}}{\left(e^{\frac{q \cdot V_{OC}}{n \cdot k_b \cdot T_r}} - 1 \right)} \right] \quad (eq. 2.3)$$

The photo generated current, I_{ph} , can be found out from equation 2.1 by shorting the circuit under STC, so that cell temperature is 25 °C and 1 000W shone on the cell. Then the current through the diode $I_D = 0$ and the photo generated current becomes the short circuit current, see equation (2.4) (9).

$$I_{ph} \approx I_{SC} \quad (eq. 2.4)$$

Since the photo generated current is proportionally affected by two outer factors, temperature and irradiance, equation (2.4) must be rewritten to be true outside STC, see equation (2.5).

$$I_{ph} = (I_{SC} + \left(\frac{\partial I_{SC}}{100} \cdot I_{SC} \cdot (T - T_r) \right)) \cdot \frac{S}{1000} \quad (eq. 2.5)$$

Where T is cell temperature, T_r is rated temperature for the cell and S is irradiance. ∂I_{SC} is the temperature coefficient of the current (11).

When photo generated current, diode current and load are known it is possible to perform a circuit analysis and equation (2.1) can be rewritten as equation (2.6)

$$I_{pv} = \left((I_{SC} + \left(\frac{\partial I_{SC}}{100} * I_{SC} * (T - T_r) \right)) * \frac{S}{1000} \right) - \left(\left[\frac{I_{SC}}{\left(e^{\frac{q \cdot V_{OC}}{n \cdot k \cdot T_r}} - 1 \right)} \right] \left(e^{\frac{q \cdot V_D}{n \cdot k \cdot T}} - 1 \right) \right) \quad (eq. 2.6)$$

2.3 Ideality factor, n

The ideality factor measures how closely the diode follows the ideal diode equation, and will be adjusted in the circuit model to better fit experimental data. $n=1$ is an ideal diode, or an ideal PV-cell. Note that a higher n lowers the performance of the cell, see figure (2.4). (12)

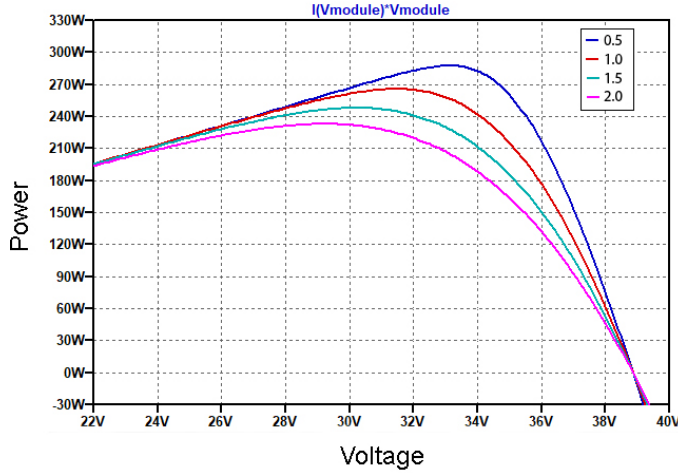


Figure 2.4: Different ideality factor n , simulated on module *Winda Premium 265 Multi*.

2.4 Solar irradiation

The short circuit current I_{SC} for every cell model is determined by the manufacturer during STC and is 8.9 A for the model used in this thesis. In figure (2.5) it is shown how the current in the cell is proportional to the irradiance. The current increases linearly with the irradiance.

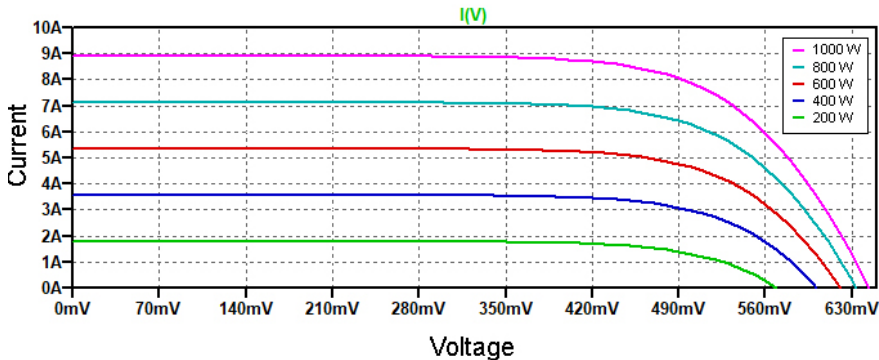


Figure 2.5: Different solar irradiance levels on a single cell

2.5 Temperature

The temperature of a cell also affects its performance, although not as much as irradiation. Increases in temperature will reduce the bandgap of the semiconductor which slightly increases I_{SC} , but primarily lowers the open circuit voltage V_{OC} , see figure (2.6). Note that all simulations in this method will assume a cell temperature of 25 °C, which is the rated temperature of the cell.(13)

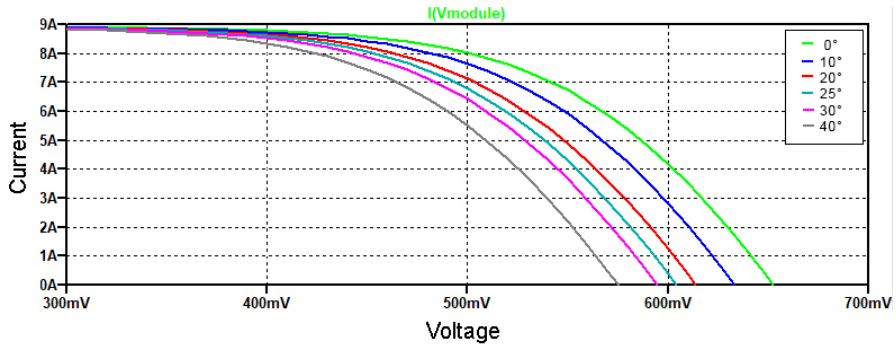


Figure 2.6: Different temperatures on a single cell. 1 000 W irradiance

2.6 PV module - Windon Premium 265W

The module model used in the field tests were *Windon Premium 265 Multi*. It was chosen because experimental data of its performance in laboratory settings was available in a report by Gallardo-Saavedra S. (11) and thus the step of verifying the model could be done purely in circuit simulation software LTspice (14) by comparing the results of different irradiation settings. Windon 265 consists of 60 cells in total, with three bypass diodes for ever string of 20 cells.

Table 2.1: Specifications for Windon Premium 265 Multi

Nominal Power (P_{mpp})	265 W
Open circuit Voltage (V_{oc})	38.6 V
Short circuit current (I_{sc})	8.9 A
Efficiency	18.4 %
Temperature coefficient (∂V_{oc})	-0.320 %/K
Temperature coefficient (∂I_{sc})	-0.77 %/K

See appendix for data sheet (15).

2.7 Typical Meteorological Year (TMY)

The irradiance database used in this method is pulled from European Commission's Typical Meteorological Year generator. The database is continuously updated and consists of selected weather data for a specific location. Solar irradiation is presented for every hour of the year and is generated from a data bank, at least 12 years in duration The data is selected to present a range of weather phenomena of the selected location while still giving the annual averages. (16)

3 Method

The method consists of three sections. The first part is the designing and making a PV-array circuit model in LTSpice. The second part is building a Python script that can take hemispherical images as input, feed them into the circuit model, and estimate energy as output for any given time period of the year. Last is the field tests where the script was tested by capturing images at three different locations.

3.1 Circuit model

The circuit simulation model was created in LTSpice (14). LTSpice is a free analog electronic circuit simulator computer software.

3.1.1 Designing and creating PV-array circuit model

First a circuit model of the cell was created using the widely acceptable model of a PV cell, 1M5P (single mechanism five parameters), see figure (3.1). The five parameters are the current source I_{ph} , diode D_{cell} , parallel and series resistance R_p and R_s and the load voltage V . All parameters are needed prior to the simulation.

All parameter specifications could be found in the manufacturer's data sheet (see appendix 6.1), except for the reverse saturation current in D_{cell} , which was calculated using equation 2.3.

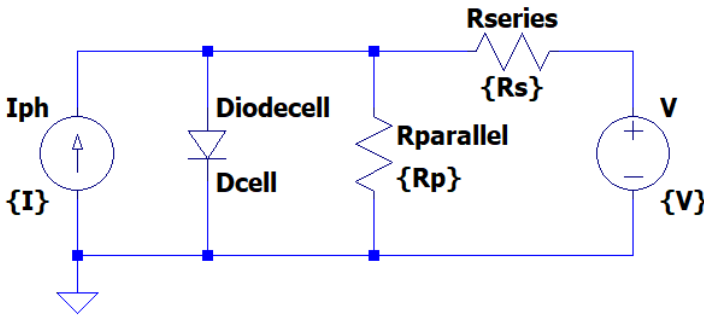


Figure 3.1: LTSpice model of a PV cell.

Then the cells were put together in groups of 20, a so called string, with a bypass diode connected over each string. Three strings were connected after each other so that the module consisted of 60 cells total, see figure (3.2).

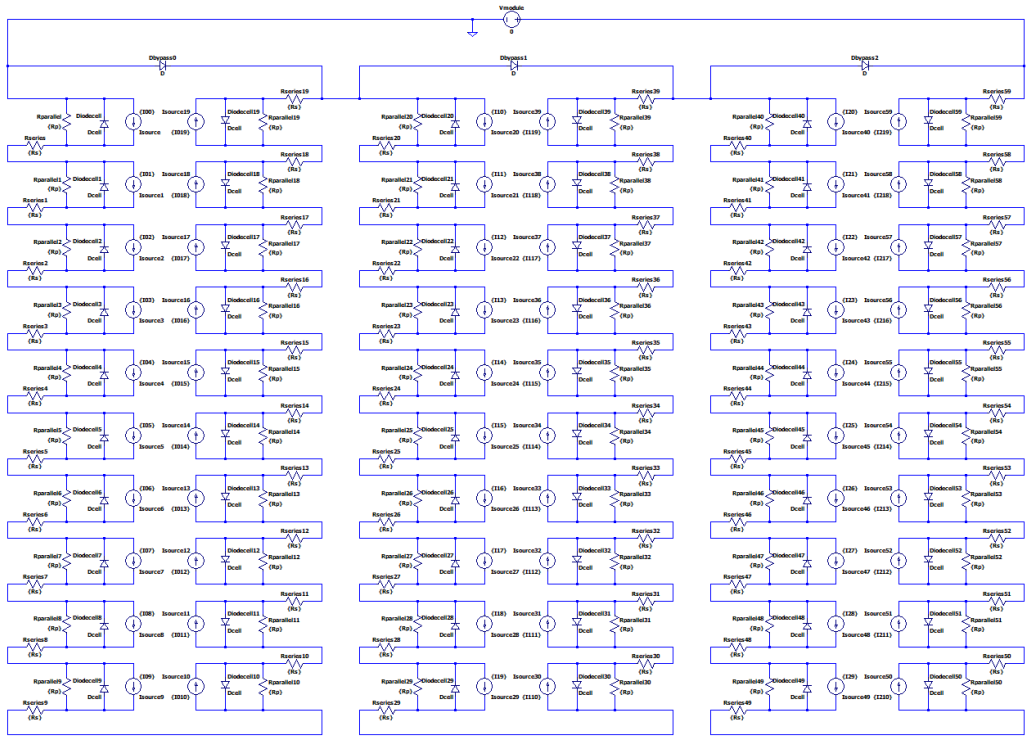


Figure 3.2: LTspice model of a PV module. (8)

At this point the circuit model of the module had to be verified before creating an array. That was done by comparing simulation results to experimental data and adjusting the ideality factor, n . The process will be explained in the next section.

Once the model was confirmed to be accurate, defined in this case as less than 1 % difference from the experimental data, the array circuit model could be build. The array consists of nine modules, three lines with three modules each. Each module on a line is connected in series and each line is connected in parallel, see figure (3.3). No new bypass diodes are introduced between any modules.



Figure 3.3: LTspice model of a PV array. (8)

Each of the 540 current sources were given unique names so that the python script could assign irradiation data individually to them.

3.1.2 Verifying the circuit model

To be certain the module circuit model was indeed behaving like its physical counterpart, a validation of the model was done. It was done by performing different shading cases in the simulation and comparing the results to experimental data presented in an article by Gallardo-Saavedra S. (11). The results from the verification process can be seen in the appendix (6.1)

Some necessary adjustments were made to mimic the experimental data. Studying equation (2.6) shows that the only variable that is adjustable without compromising the model is the ideality factor, n . The ideality factor n was adjusted from 1.0 to 1.8 (yielding a lower power output) so that the circuit model gave the same results as the experimental data.

3.2 Python script

The python script consists of three sections, so called modules (not to be confused with PV modules), all working together and a main module to run the script from. The modules are explained below, see figure (3.4) for visual representation. The goal was to create a script that uses a folder of 135 hemispherical images (representing 540 real PV cells) as input, and the estimated energy production as the output. The reason 135 images was chosen instead of 540 were mainly due to computational power. Performing shade analysis on a image is the most resource demanding part of the script, so using 135 and letting one image represent four cells will drastically improve performance.

Main The main module takes all user input, such as calibration of the lens, image matrix dimensions, module tilt, longitude and latitude. The results from simulations are also returned here so that plots can be made.

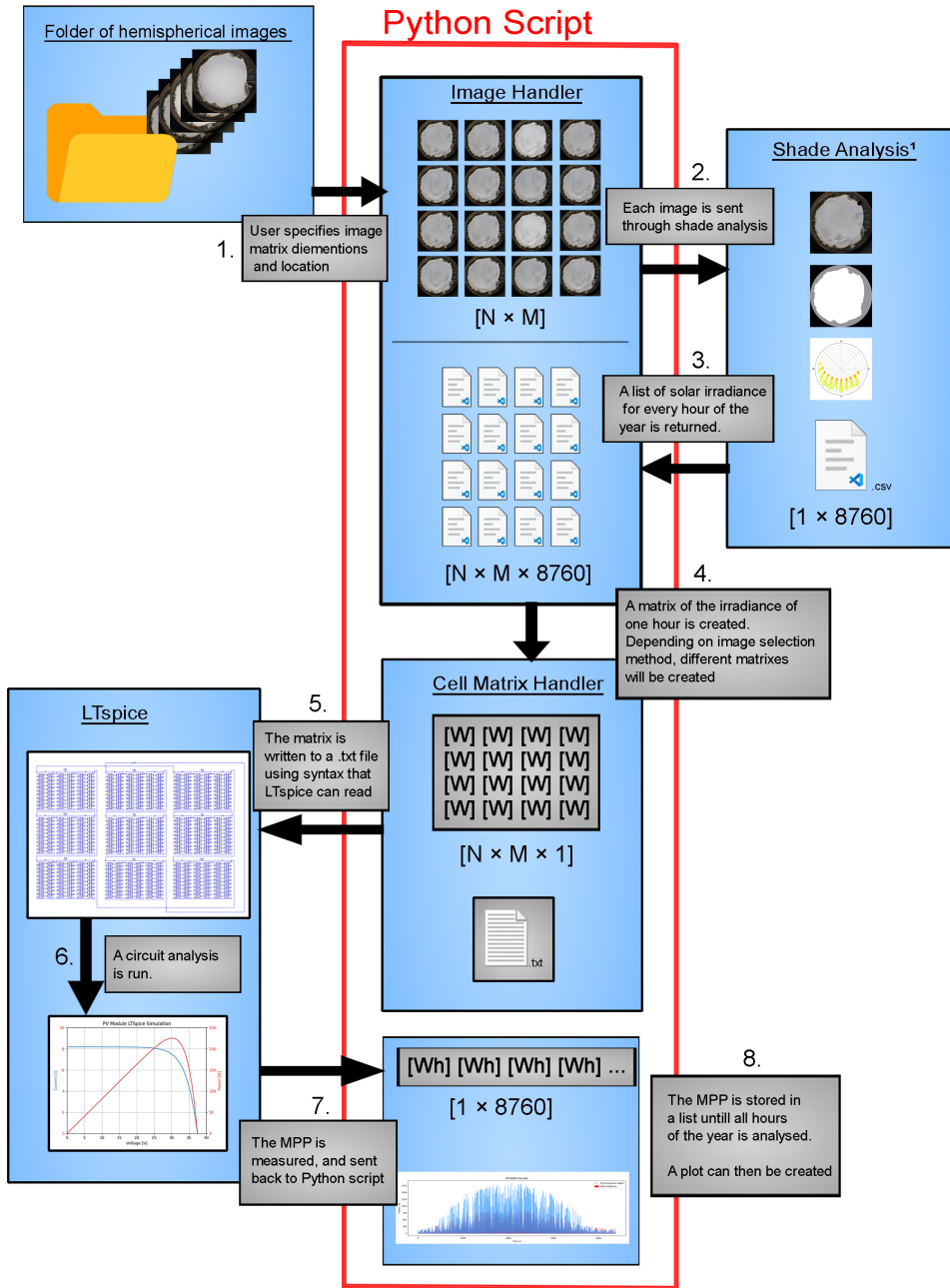
LTspice Command Responsible for communication between Python to LTspice. This module will prepare, run LTspice simulations and feed the results back to Python.

Image handler Takes a folder of hemispherical images as input. The images must be alphabetically organized by the user starting with the most north-west image, going in rows, and ending at the most south-east. Its main purpose is to organize images in a matrix so that the images are organized in a way that represents the actual PV array. There are three alternatives for the user to choose from. Either to use all 135 captured images, or to only use the four corner images to represent the whole array, or to only use the middle most image to represent the whole array. These image selection settings are explained further in subsection (3.3.1)

Cell matrix handler Its purpose is to prepare a snapshot of the irradiance relating to each image in the image matrix for a specified hour, and assign that data to a PV cell in the circuit model. It will then write the data in a format that LTSpice can read.

A simplified flowchart of a full year analysis is presented in figure (3.4), with numbers relating to the text below.

1. The user feeds a folder of alphabetically organised hemispherical images into the script. The images must represent the physical area where the PV array would be, and must be organised starting at the most north-western image going in rows to the most south-eastern image. At this step the user must also specify the image matrix dimensions (how many rows of images, and how many columns of images that were captured), module tilt, azimuth angle, longitude and latitude.
2. Each image, one at a time, goes to the Python script written by Andersson J. (4), where image analysis is performed on the image. All sky obscuring obstacles are detected and a shade mask is created. The actual irradiation data is pulled from the European Commission's Typical Meteorological Year generator (17) and is applied to the shade mask together with the sun's position to determine direct and diffuse irradiation. This is done for every image and a list of global irradiation (direct, diffuse and reflected) for every hour of the year is return to the main script.
3. The lists are organised so they keep the dimensions of the image matrix.
4. A snapshot of the current irradiance matrix for the hour to be analysed is created. At this stage the user must specify how many, and which, of the images should be used. The different image selection settings are explained in section (3.3.1). Depending on the setting, different snapshot matrix will be created.
5. LTSpice uses its own syntax so the snapshot matrix containing irradiance data points is rewritten in a text document that LTSpice can read. The snapshot matrix is at this stage fitted so that every cell in the circuit simulation gets its corresponding irradiance data.
6. A circuit analysis is run.
7. The maximum power point is read and sent back to the main script.
8. When all hours of the year, or selected time period, have been analysed and their respective maximum power point is collected, energy calculation data is ready.



*Andersson J. Smartphone-based shade analysis using hemispherical fisheye imaging for local solar energy potential(2020)

Figure 3.4: Flowchart of the Python script

3.3 Field tests

The goal of the field tests are to determine how a decreasing number of captured images affect the energy calculation.

The module that was used for field tests was *Windon Premium 265W*, see section (2.6). *Windon Premium 265W* consists of six columns of cells and ten rows, in total 60 cells. When put together in a 3×3 array there will be 18 columns of cells and 30 rows. In total 540 cells, see figure (3.5).

Three image selection methods are used. First a reference case, where one image is captured for every four cells. Secondly only one image is captured at the center of the array. Lastly only images in corners are captured, one at each corner of the array.

The tests were performed at three different locations, first at Ulrikedal (55.70, 13.19), Lund, Sweden. Since the script does not recognise a difference in a real image or computer graphics, the other two field tests were performed in a computer generated environment, using web tool *Dynamic Overshadowing* by (18). This was done so that controllable shading conditions could be created.

At every location three significant days were analysed, instead of a full year analysis. They are spring equinox (20/3-21), summer solstice (21/6-21) and winter solstice (21/12-21), and chosen because they offer the largest difference in sun angles between each other. The benefit of choosing the three dates is that the error in image selection settings might be bigger or smaller during any of these dates than the others, which would not be noted if using the average error from a full year analysis.

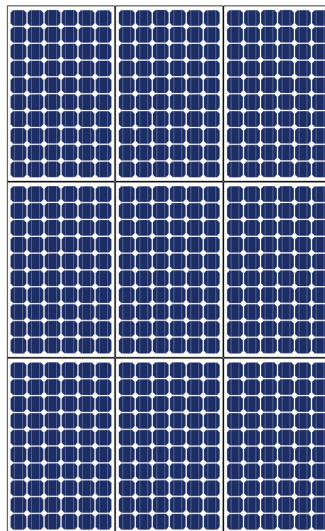


Figure 3.5: Visual representation of the PV array used in field tests. (8)

3.3.1 Image selection settings

In order to determine the significance of the camera placement and how many images that are necessary for an accurate calculation, three image selection settings were introduced. They are *reference*, *one image (center)*, and *corners*. The different settings determine which cells will be assigned which irradiance data in the simulation.

Reference The reference setting is what the other two settings will be compared to. Since it is not practically viable to take one image for every PV cell, one image was captured for every four cell in order to save computational power, see figure (3.6)

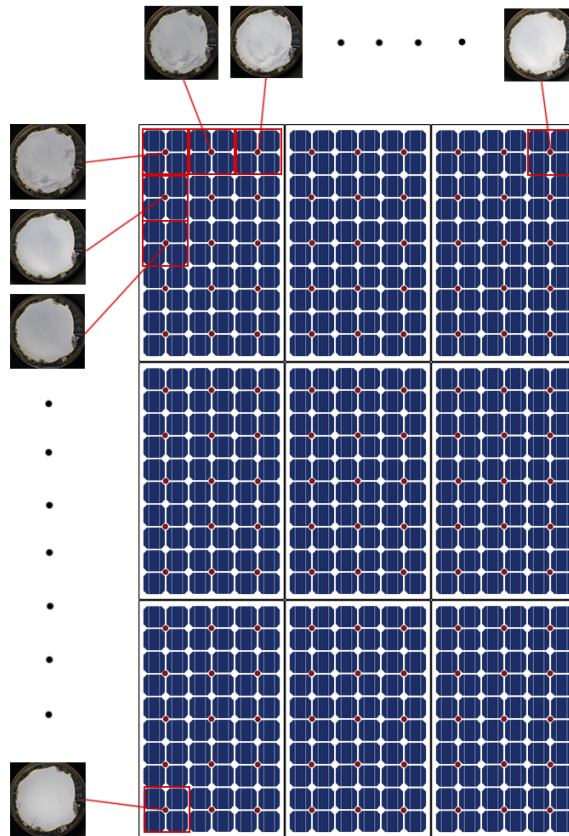


Figure 3.6: The images are captured at every red dot. One image represents to four cells. ((8))

The array used in this method has 540 cells, which means 135 images needs to be captured for the reference case.

One image (center) The One image (center) setting uses only one image that will represent all PV cells, see figure (3.7). The image is captured at the middle of the array and the script assigns the irradiation data from this one image to all cells.

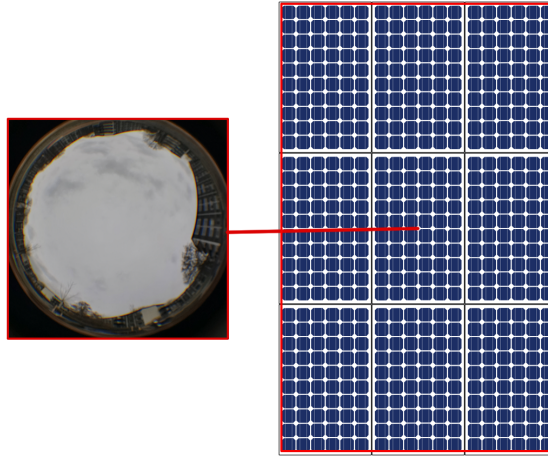


Figure 3.7: One image taken at middle of module, assigned to every cell. ((8))

Corners Images from each of the four corners of the array are captured and put into the script. The script will assign the image to represent $\frac{1}{4}$ of the array, i.e. the quadrant in which the image was captured, see figure (3.8).

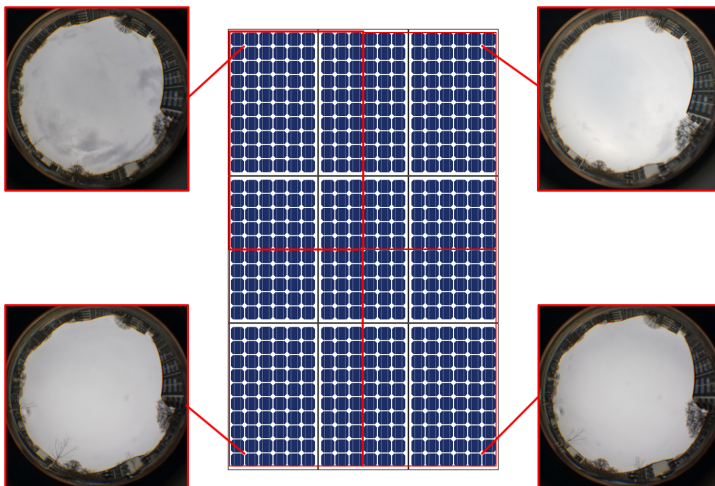


Figure 3.8: Four images, one taking at every corner, representing one quarter of the total cells. ((8))

3.3.2 Locations

The field tests were set up at three locations. One real location, Ulrikedal (55.70, 13.19), Lund, Sweden, and two computer generated locations, *South wall* and *West wall* which will be explained below.

Ulrikedal was surrounded by high multi level buildings in all directions and placed on the ground rather than on a roof in order to observe the influence of local shade, see figure (3.9).

To more clearly observe the difference in the different settings, the other two locations were computer generated to get a controllable environment (18). In each of the generated locations an infinitely long, one meter high wall was placed. One going east-west, see figure (3.10) , and one going north-south, see figure figure (3.11). The imagined arrays were then placed at zero meter, one meter and two meters away from each wall, pointing north.

Ulrikedal The first field test was done at Ulrikedal, Lund, Sweden. A grid was laid out in a grid on the ground pointing north, so that one square in the grid represented four cells in the array, see figure (3.9). A camera rig including a weight to point directly under the camera, a camera stand, a cellphone and a $>180^\circ$ degree fish-eye lens was necessary to capture images. All images are captured at 0° tilt, 0° azimuth (north) and at 1.5 m height. The images were then captured at every instance where the thread intersected with another creating a 9×15 image matrix. Starting at the northwestern corner and then capturing row-wise, with the final image being the most south-east. The order of the images were important to keep track of as they needed to be correctly labeled when fed into the script, it was continuously checked that the images captured correlated to the placement in the grid. In total 135 images were captured, representing 540 PV cells.

Simulations were then run for every significant day, using all three image selection methods



Figure 3.9: Visual representation of image capturing at site Ulrikedal. ((8))

Computer generated locations In order to perform the analysis in a controlled environment the computer generated locations were created in the web app *Dynamic Overshadowing*. (18). Two locations were created, *South Wall*, see figure (3.10) and *West Wall*, see figure (3.11). In each location an one meter high, infinitely long, wall was introduced. The array was placed at zero meter, one meter and two meters away from the one meter high wall. Hemispherical images were captured in a 9×15 grid, correctly labeled and saved to the computer for each distance and location. Simulations were then run for every significant day, using all three image selection methods

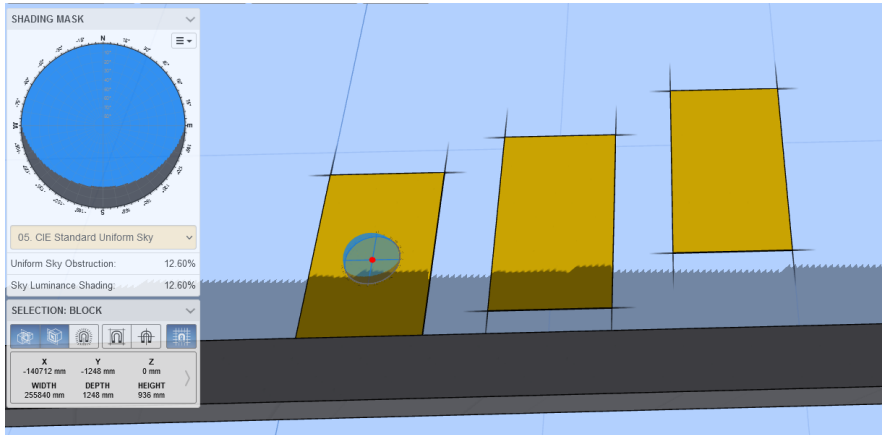


Figure 3.10: Image capturing at computer generated location, South Wall. The 3×3 array is represented by the brown area, and the wall by the gray area.

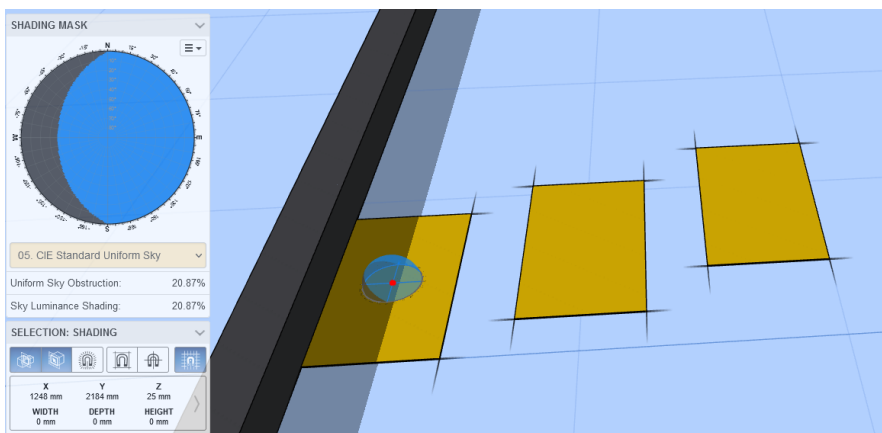


Figure 3.11: Image capturing at computer generated location, West Wall. The 3×3 array is represented by the brown area, and the wall by the gray area.

4 Results

Ulrikedal In figure (4.1), there is no visible difference in the three image selection settings for any significant day. It is likely that there is not enough difference in the input images captured in the field test to produce different outputs using the different image selection methods. As expected, the power output is largest during summer and lowest in winter. In summer there are two peaks as opposed to only one that occur during spring and winter. This is not caused by local shade, then we would also see it in the other dates. It is most likely caused by a variance the irradiance data used for summer solstice, such as large clouds.

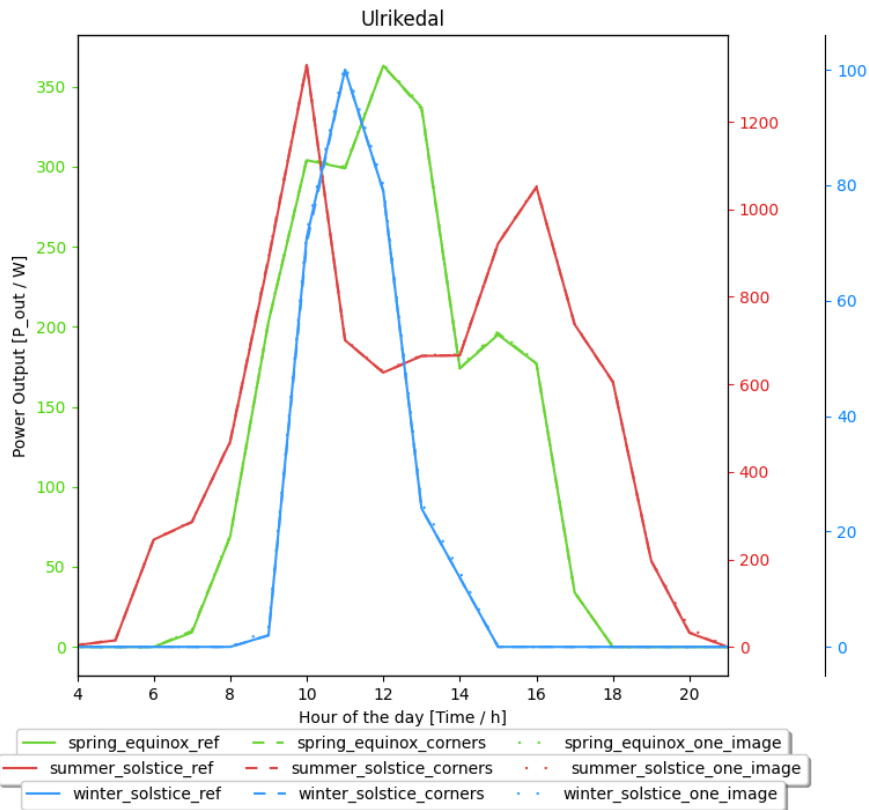


Figure 4.1: Ulrikedal power curve for spring equinox (20/3-21), summer solstice (21/6-21) and winter solstice (21/12-21) and three image selection settings.

The energy calculations presented in table 4.1 shows the energy calculated from the power output presented in figure (4.1). The relative error, which is the percentage difference from the reference case, is displayed in parenthesis. Note that larger value, regardless of

positive and negative, is worse.

The error is generally slightly higher when using one image (center) setting, rather than the corner setting.

Table 4.1: Energy calculation for Ulrikedal.

Significant day	Reference [kWh]	One image (Center) [kWh]	Corners [kWh]
Winter solstice	0.288	0.294 (+ 2%)	0.287 (- 0.3%)
Spring equinox	2.164	2.174 (+ 0.5%)	2.166 (+ 0.1%)
Summer solstice	9.437	9.473(+ 0.1%)	9.444 (+ 0%)

South Wall The difference between the different image selection settings are obvious when the array is placed 0 m away from the one meter high wall, see figure (4.2). The difference from reference case, or error, gets drastically lower at 1 m and is not visible at 2 m away.

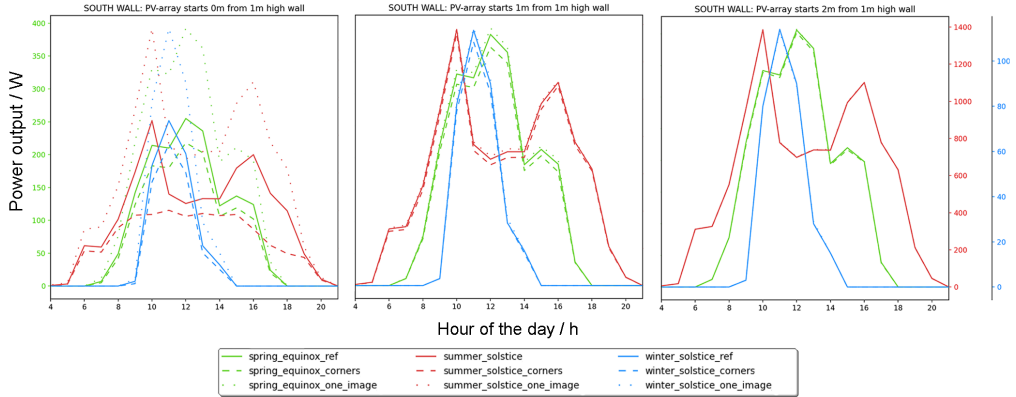


Figure 4.2: South wall PV curve for three significant days and three image selection settings.

Table (4.2) shows that the error is largest for One image (center) at 0 m is at about 54% for all significant days. The error decreases to about 2-5% at 1 m and the error is almost zero at 2 m away. The error is significantly larger when using one image at 0 m than when using four images.

Table 4.2: Energy calculation for South wall.

0 meter			
Significant date	Reference [kWh]	One image (Center) [kWh]	Corners [kWh]
Winter solstice	0.205	0.317 (+ 54.6%)	0.174 (-15.1%)
Spring equinox	1.520	2.343 (+ 54.1%)	1.302 (-14.3%)
Summer Solstice	6.548	10.017 (+ 52.9%)	4.229 (-35.4%)
1 meter			
Winter solstice	0.320	0.327 (+ 2%)	0.307 (-4%)
Spring equinox	2.350	2.409 (+ 2.5%)	2.237 (-4.8%)
Summer Solstice	10.084	10.215 (+ 1.3%)	9.769 (-3.1%)
2 meter			
Winter solstice	0.330	0.329 (-0.3%)	0.330 (0%)
Spring equinox	2.430	2.433 (+ 0.1%)	2.403 (-1.2%)
Summer Solstice	10.268	10.294 (+0.2%)	10.268 (0%)

West wall Like the results form *South Wall*, the error is largest at 0 m and decreases as the array is moved away from the wall, see figure (4.3). However looking at table (4.3) the Corners setting now produces the largest error at both 0 m and 1m. In general the relative errors are lower in *West wall* location.

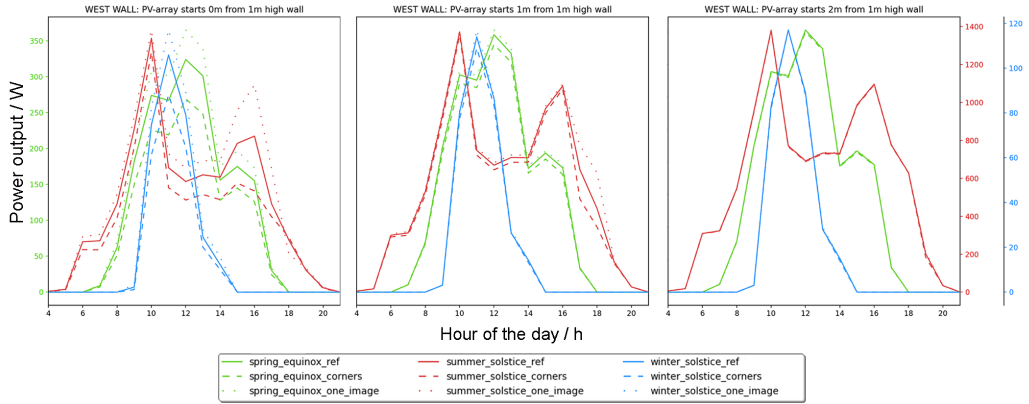


Figure 4.3: West wall PV curve for three significant days and three image selection settings.

Table 4.3: Energy calculation for West wall.

0 meter			
Significant date	Reference [kWh]	One image (Center) [kWh]	Corners [kWh]
Winter Solstice	0.268	0.301 (+ 12.3%)	0.220 (-17.9%)
Spring Equinox	1.935	2.174 (+ 12.3%)	1.593 (-18.0%)
Summer Solstice	7.871	8.929 (+ 13.4%)	6.601 (-16.1%)
1 meter			
Winter Solstice 1m	0.319	0.325(+ 1.2%)	0.307 (-3.7%)
Spring Equinox 1m	9.540	9.980(+ 4.6%)	9.068 (-4.9%)
Summer Solstice 1m	2.300	2.348 (+ 2.1%)	2.212 (-3.8%)
2 meter			
Winter Solstice 2m	0.335	0.337(+ 0.6%)	0.332 (-0.9%)
Spring Equinox 2m	2.408	2.415 (+ 0.3%)	2.393 (-0.6%)
Summer Solstice 2m	10.190	10.220 (+ 0.3%)	10.133 (-0.6%)

A summary of table 4.3 and table 4.3 is presented in figure 4.4.

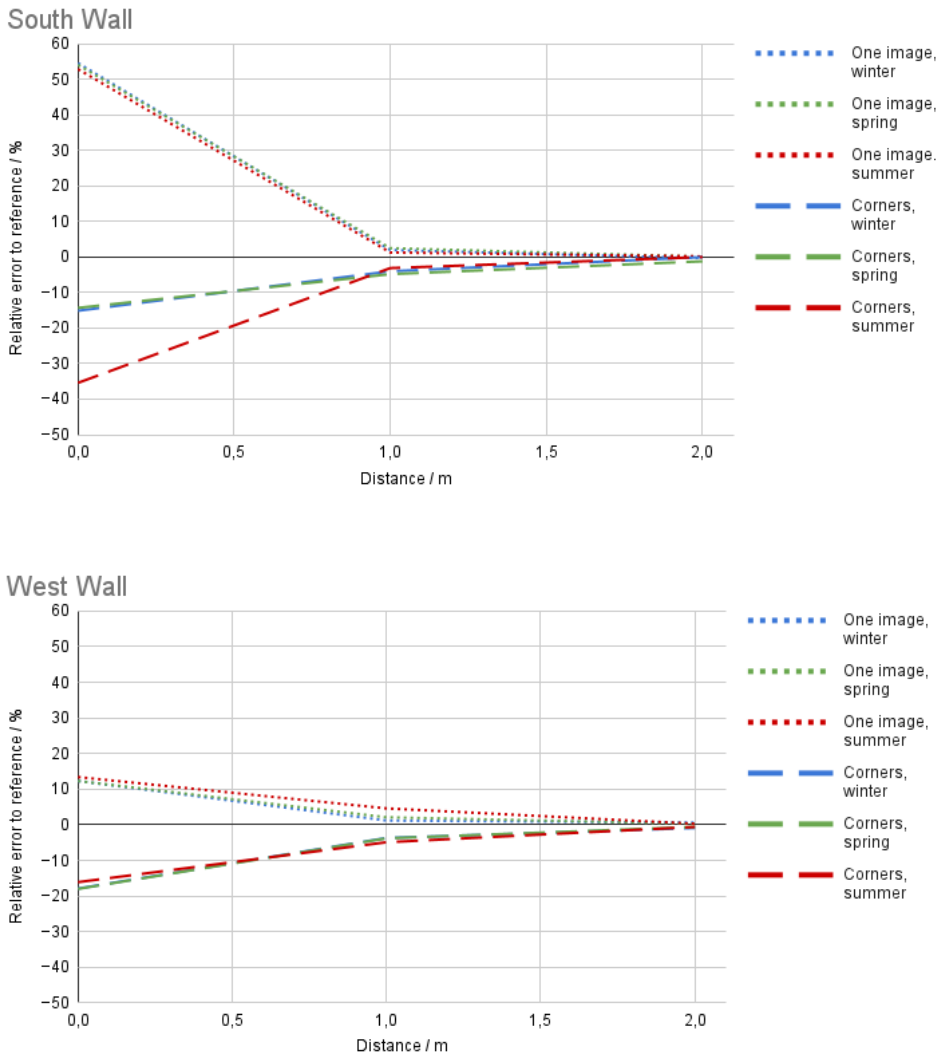


Figure 4.4: Relative error from reference when using the two alternative image selection settings, four images (corners) and one image (center).

5 Discussion

The results from the image selection settings at **Ulrikedal** did not offer enough difference in the captured images, i.e. the area of the array was too small to offer a difference in local shading for the program to recognise. Because of that, it is not possible to draw any conclusions of the effectiveness of the image selection settings. In this case one image is enough as 1 image gave the same result as 135 images.

The Ulrikedal field test showed that the method worked as intended, that it was indeed possible to capture several hemispherical images, feed them into the script and estimate the energy generation.

As presumed in the **Computer generated locations**, the error of the different image selection settings decrease as the array is moved away from the one meter high wall, regardless of wall direction, see figure (4.4). Already at 1 meter away from the 1 meter high wall the error is less than 5 % and less than 1 % at 2 meters, regardless of wall direction and significant date.

However at 0 meters away from the one meter high wall the error was large. The worst case was when using one image and the wall to the south, which yielded in a 55% error. When using four images the error was between -15% and -35%. This tells us that when the wall is placed to the south, four images gives a better approximation than one. It also tells us that during the summer when irradiance is stronger, the error will be larger.

When the wall was placed to the west the error for both alternative image selection settings were similar, around +/- 15%. This tells us that when the wall is placed to the west, four images gives the same results as one image, with no regards to time of the year.

From these results we can draw the conclusion that when the array is placed far away from buildings, such as the Ulrikedal case, one image is sufficient to accurately represent the whole array. When the array is placed 2 meters away from a 1 meter high wall the using 4 images gives the same result as when using 1, less than 1% error on the approximation. At 1 meter away from the wall the error will be less than 5% when either using 4 images or 1 image. At 0 meters away the error will be very large when using 4 images or 1 image. The error will be largest if the wall is placed south of the array and only one image is used.

Because the sun spends more time in the south direction than it does in either east or west direction, it makes sense that the array will be more sensitive to error when the wall is placed to the south. The results are similar to what one might expect, that is that more sample images are necessary when the sun obscuring obstacle is placed to the south of an array, and that the error diminishes as the array is moved further away from the obstacle. Since the error is very large between 0m and 1m away from the wall in these results a better way of approximating irradiation is necessary in that region.

5.1 Further research

The end goal of this project is to have a system that is available and easy to use. At this moment more work is required to reach that goal.

5.1.1 Larger arrays

At the moment the script only works for a 3×3 array since that is the way the circuit model was build. In the future the script needs to be able to run in any configuration set up by the user. A solution may be to pre build many circuit models and let the script chose the one that fits the user's specification.

5.1.2 Different image selection setting

In this thesis three image selections settings were analysed. Perhaps there is a better way represent all cells a different set up of images. Perhaps five images, one in each corner and one in the center, would be better. Or maybe one image for each module. See figure (5.1)

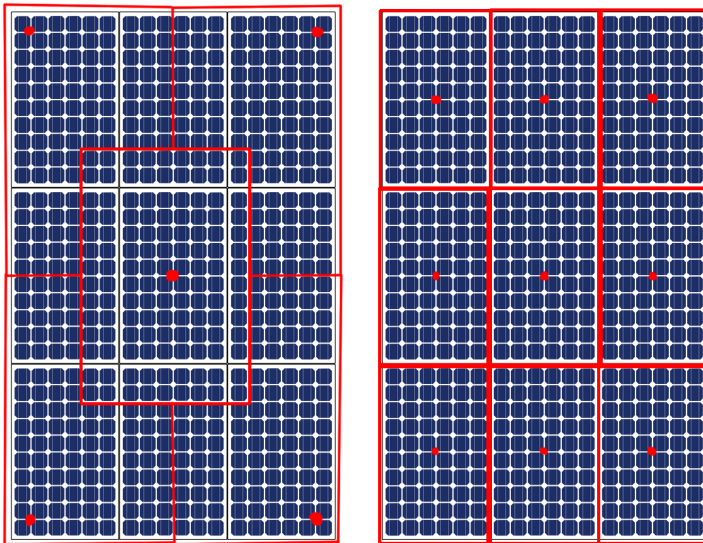


Figure 5.1: Possible new image selection settings.

5.1.3 Simplify circuit model

The LTspice circuit model as of now consists of nine modules, or 540 cells (or over 2500 components). As LTspice is specifically created to run simulations of smaller filters to analyse transients, this model has really pushed LTspice to its limit. When building the array model the program became very slow and hard to work in. It is recommended that in

order to improve performance and to simulate more than nine modules the model should be simplified. For example, when only using one image as input, all the cells in the model will get the same irradiance data. Therefore the 540 cells could be merged into one cell that is 540 times more powerful.

5.1.4 Bypass diodes in circuit model

The bypass diodes used in the circuit model are not the same as the ones in Windon Premium 265W, since they were not specified in the data sheet. Instead the default diode model in LTSpice was used instead and may impact the resemblance of the model to its physical counterpart.

5.1.5 Image interpolation

An early idea was, since the single image shade analysis script written by Andersson J. (4) generates a local shade mask of the sky view, to always take four images (one in each corner) and to interpolate the would be images between them. A couple of weeks of research and testing showed that it would not be possible to do within the time frame of this thesis, but a method using *Structure from Motion* was found.

Structure from Motion COLMAP is a free software developed as a PhD project and is a general purpose Structure-from-Motion (SfM) program. It uses several images as input, and is able to estimate a 3D environment form the movements of the photos. However the program is resource-heavy and may not be able to run on older hardware.

6 Conclusion

A novel photo voltaic energy calculation system using a smartphone, a fish-eye clip on lens, a circuit simulation program and a Python script was created, validated and tested in field. Three image selection settings were build into the Python script to be able to analyse each setting's impact on the energy calculation.

The first field test was performed using the intended method by capturing hemispherical images at a real location. However the difference between the three difference image selection settings were to small to draw any conclusion. Two other field tests were simulated in a computer generated environment where the array was placed next to a one meter high wall, one in south direction and one in west direction, at different distances.

The results showed that for any image selection setting and wall direction, when the array was placed at 1 meter away from the 1 meter high wall the error was less than 5 % and less than 1 % at 2 meters. When array was placed at 0 meter away from the wall the error ranged from -15 % to +55 %. In general the error was less when using four images instead of one.

While the script worked as intended, a folder of hemispherical photos were used as input, and energy estimated as output, the script is far from done. As of now it only works with the three by three array model created in LTSpice. It is suggested by the author that the next step of this project is to continue to develop the script so that it can work for larger arrays and other image selection settings.

References

- [1] IEA, “Key world energy statistics 2020.” <https://www.iea.org/reports/key-world-energy-statistics-2020>. Accessed: 2022-05-19.
- [2] *Energiläget 2021*. ET 2021:10, Energimyndigheten, 2021.
- [3] Lazard, “Levelized cost of energy analysis - version 13.0.” <https://couleenergy.com/standard-test-conditions-of-pv-module/>, 2019.
- [4] J. Andersson, “Smartphone-based shade analysis using hemispherical fisheye imaging for local solar energy potential,” 2020. Student Paper.
- [5] Gil Knier, “How do photovoltaics work?.” <https://science.nasa.gov/science-news/science-at-nasa/2002/solarcells/>. Accessed: 2022-03-11.
- [6] “Bypass Diodes in Solar Panels.” <https://www.electronics-tutorials.ws/diode/bypass-diodes.html>. Accessed: 2022-01-09.
- [7] N. A. Zainal and A. Apen, “Modelling of photovoltaic module using matlab simulink,” *IOP Conference Series: Materials Science and Engineering*, vol. 114, p. 012137, 02 2016.
- [8] B. F. Nimvik, 2022.
- [9] F. Rasool, M. Drieberg, N. Badruddin, and B. S. Mahinder Singh, “Pv panel modeling with improved parameter extraction technique,” *Solar Energy*, vol. 153, pp. 519–530, 2017.
- [10] PV Education, “Standard test conditions of pv module.”, 2020. Accessed: 2021-12-05.
- [11] B. K. Sara Gallardo-Saavedra, “Simulation, validation and analysis of shading effects on a pv system,” *Solar Energy*, vol. 170, pp. 828–839, 2018.
- [12] PVEducation, “Ideality factor.” <https://www.pveducation.org/pvcdrom/solar-cell-operation/ideality-factor>, 2020. Accessed: 2022-03-04.
- [13] PV Education, “Effect of temperature.” <https://www.pveducation.org/pvcdrom/solar-cell-operation/effect-of-temperature>, 2022. Accessed: 2022-04-03.
- [14] “LTSpiceXVII.” <https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html>. Accessed: 2021-12-03.
- [15] “Datablad Premium 265 SWE.” <https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html>. Accessed: 2021-12-03.
- [16] PV Education, “Typical meteorological year.” <https://www.pveducation.org/pvcdrom/properties-of-sunlight/typical-meteorological-year-data-tmy>. Accessed: 2022-03-11.
- [17] EU Science Hub, “Tmy generator.” https://joint-research-centre.ec.europa.eu/pvgis-photovoltaic-geographical-information-system/pvgis-tools/tmy-generator_en. Accessed: 2022-04-03.
- [18] D. A. J. Marsh, “Dynamic overshadowing,” 2020.

Appendix



Data på Windon Premium 265 Multi

Nominell Effekt (Pmpp)	265 Wp
Max producerad spänning V mpp (V)	31,2 VDC
Max producerad ström I mpp (A)	8,4 A
Max spänning olastad V oc (V)	38,6 VDC
Kortslutningsström I oc (A)	8,9 A
Celleffektivitet (%)	18,4 %
Effekt tolerans mpp (%)	-1/+3%

Generell data på modulen

Dimension (L x W x H):	1665x991x35 mm
Modul yta:	1,65m ²
Vikt:	19 Kg
Cell typ:	156x156 crystalline
Max systemspänning:	1000 VDC
Antal celler:	60st
Celler / bypassdiod:	20st
Antal bypassdioder:	3st
Antal strängar:	3st (4st ribbon/sträng)
Front glas:	Högtransparent glas med låg järnhalt och etsbehandlad antireflexyta.
Glastjocklek:	3,2 mm
Kopplingsbox:	ETE, svetsad, flamsäker med membran.
Ram:	Anodiserad och pulverlackad ALU
Dränering:	Homogena hörn, dränering 7 x 12 mm
Anlitad modulbyggare:	X-disk Polen / SPS Italien
Tillverkare/producent:	Windon AB, Sverige



Elektrisk data

Bypassdiode ström:	15 (A)
Max säkring i serie:	12 (A)
Max omvänt ström:	20 (A)
Modulens resistans:	305 (mΩ)

Temperature raitings (uppmätt av TUV)

Nominell arbetstemperatur (NOCT):	47,5 (°C)
Temperaturkoefficient Pmpp:	-0,382 (%/K)
Temperaturkoefficient Voc:	-0,320 (%/K)
Temperaturkoefficient Isc:	-0,077 (%/K)

Beskrivning av Windons modul:

Windons modul är i första hand producerad för att hålla bra i våra nordiska väder. Vi har lagt energi på de viktiga materialen för att bibehålla en produkt som håller väldigt länge i nordens väder. Vi har en ram som är både anodiserad och pulverlackad och vi har homogena maskinpressade ALU-hörn i ramprofilen för att klara problemen från kondens och frostsprängning. Vi har också en kopplingsbox med 100% svetsade anslutningar och har ett membran för att hålla kondens borta. Detta är ett antal val vi gjort för att hålla en säker och hållbar produkt. Vi har utöver kvalitet tänkt på att få en bra produktion i nordiskt väder och har då valt ett dyrare etsat glas för att öka vår årsproduktion med 12-18% samtidigt som vi då får ett helt självvregörande glas. Smutshalten når bara ca 2% på vårt glas mot ca 25% på ett vanligt coatat AR-glas. Allt detta är vad som gör vår produkt helt unik.

Garanti:

15 års material och utförande, fabriksgaranti.
25 års linjär effektgaranti ned till 85%. Minsta värdet första året är 97%.
50 års garanterad effektgenerering.

Certifikat:

TUV fabriksinspektion, MCS

Producent:

Windon AB
Häija 5
59022 Väderstad, Sverige
+46 144 301110, service@windon.se

Återförsäljare/montör:



Figure 6.1: Data sheet for Windon-Premium-265-Multi PV module

Table 6.1: The seven shading configurations used to validate circuit model.

Case	Setting
0	No shading on the module, reference case.
1	One cell 50 % shaded.
2	One cell 75 % shaded.
3	Two cells 50 % shaded (cells covered by the same bypass diode).
4	Two cells 50 % shaded (cells covered by different bypass diodes).
5	Three cells 50 % shaded (cells covered by different bypass diodes).
6	One row 50% shaded.
7a	First bypass circuit: 70 %, Second bypass circuit: 58 %, Third bypass circuit: 0 %
7b	First bypass circuit: 80 %, Second bypass circuit: 68 %, Third bypass circuit: 0 %
7c	First bypass circuit: 90 %, Second bypass circuit: 78 %, Third bypass circuit: 0 %

Verification of module model Cases refer to the cases presented in table (6.1), and the experimental data to refers to data acquired from (11). Note that the verification presented in table (6.2) was done on a single module model, that was later used to build the array model.

Case	Module Circuit Model $P_{mpp}[W]$	Experimental data $P_{mpp}[W]$
0	238.6	240.2
1	156.1	156.6
2	152.0	153.1
3	154.8	155.4
4	154.9	155.4
5	153.8	154.3
6	150.7	151.1
7a	81.3	97.6
7b	65.5	66.0
7c	65.4	66.0

Table 6.2: Comparison of the simulated values to experimental data.

For complete IV-, PV-curve of the cases, see appendix 6.2.

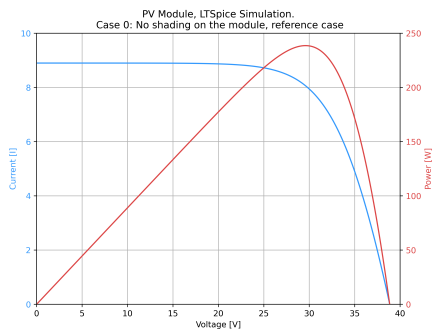


Figure 6.2a: case 0

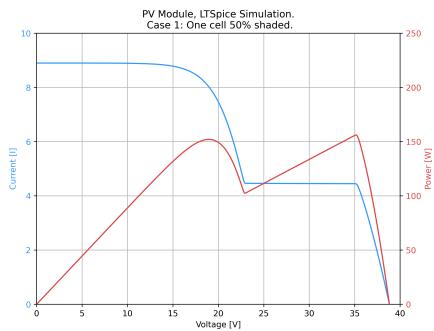


Figure 6.2b: case 1

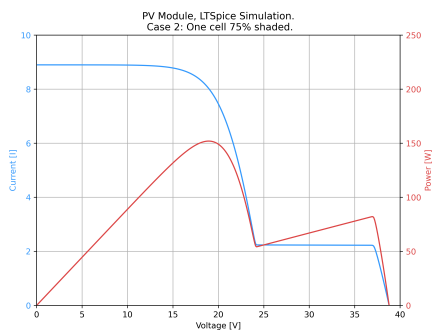


Figure 6.2c: case 2

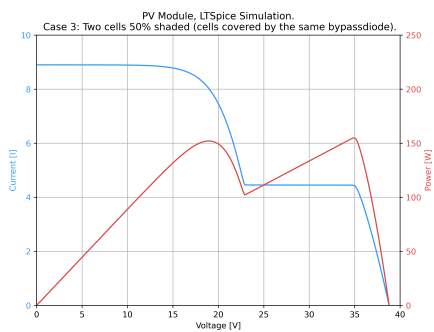


Figure 6.2d: case 3

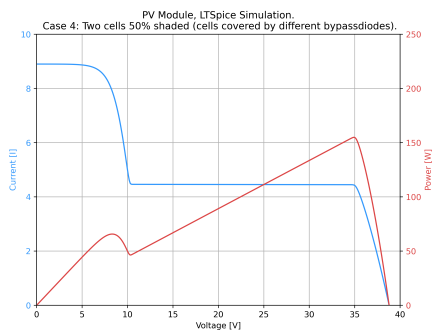


Figure 6.2e: case 4

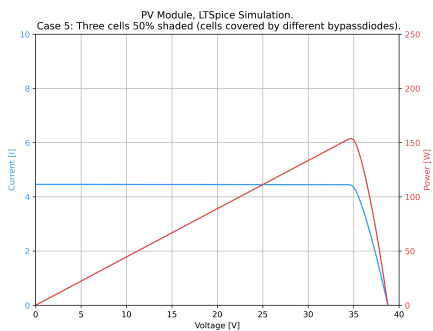


Figure 6.2f: case 5

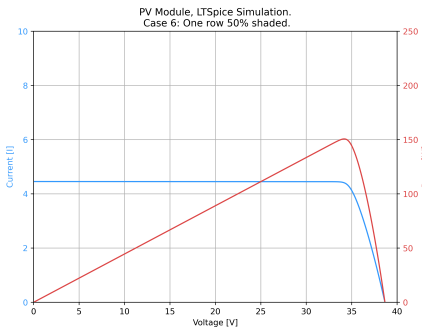


Figure 6.2g: case 6

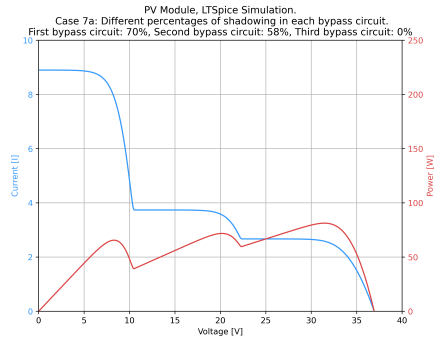


Figure 6.2h: case 7a

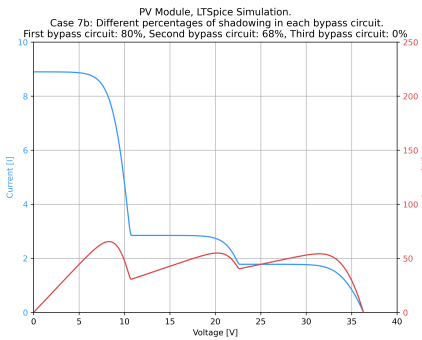


Figure 6.2i: case 7b

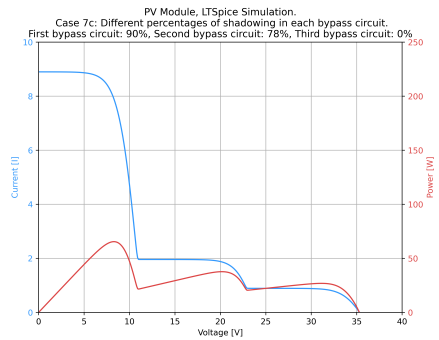


Figure 6.2j: case 7c

Figure 6.2: all cases



LUND UNIVERSITY

Dept of Architecture and Built Environment: Division of Energy and Building Design
Dept of Building and Environmental Technology: Divisions of Building Physics and Building Services