Using ray tracing to model agri-PV greenhouse energy production and PAR levels

Taeo Laue

Master’s Thesis for M.Sc. in
Renewable Energy Systems
30 credits

Institute of Technology Systems
Faculty of Mathematics and Natural Sciences
UNIVERSITY OF OSLO

June 2022

Using ray tracing to model agri-PV greenhouse energy production and PAR levels

2022

Using ray tracing to model agri-PV greenhouse energy production and PAR levels

Taeo Laue

http://www.duo.uio.no/

Trykk: Reprosentralen, Universitetet i Oslo

Abstract

In order to further the sustainability of renewable energy generation, the way land use is allocated must be considered. In the case of solar energy, land that is best used for photovoltaic power plants is often also best used for agriculture. Agri-PV, also referred to as agrivoltaics, APV, agriphotovoltaics, and solar sharing, is the combining of both agriculture and PV energy generation on the same piece of land and is one solution to this conflict of interests. However, in order to convince farmers, governing bodies, and the general public that agri-PV is a worthwhile investment, comprehensive modelling is required. In this paper, ray tracing with Radiance and bifacial_radiance is employed to create 6 models of agri-PV greenhouses located in Kjeller, Norway, consisting of 3 PV-array topologies and each with two different diffusive greenhouse covering materials. From these models energy yield is calculated, shadowing from the modules is mapped and measured, and the reduction in PAR (photosynthetic active radiation) in comparison to greenhouse grown tomatoes is calculated. In comparing the irradiance at ground level inside the simulated greenhouses, it is found that the transmission and diffusivity of the greenhouse covering has the highest influence on the amount of light entering the greenhouse. The effect of shading from the modules is also reduced due to the presence of a diffusive cover, while the amount of light reflected to the rear cells of the simulated bifacial modules are affected by both the diffusive cover and the tilt of the modules, with PV-array tilted at a summer-favoring 45° receiving the highest yearly insolation and producing the most energy out of the 3 tested topologies.

Preface

I would like to thank, acknowledge, and remember several people who inspired me and helped me get to where I am today.

First, my loving wife Martha without whom I would have never started this journey. What a joy it is to be a part of your life!

Second, my dear friend Matthew who passed away from Covid at the beginning of writing this paper. Your friendship and fireside conversations meant the world to me. Thank you for inspiring me to follow my passions and always being willing to offer a fresh perspective on any of my life’s questions.

Third, and finally, I want to thank Erik Stensrud Marstein and Heine Nygard Riise, my supervisors. Without your help, I wouldn’t have come half as far. You were both a constant source of support throughout this whole process and I really cannot thank you enough. So, here’s at least one more: THANK YOU!

Contents

1. Introduction

As modern society moves towards electrification and employing ever increasing amounts of renewable energy, many questions are raised concerning the logistics of the space required for renewable energy generators. In the case of solar energy, photovoltaic (PV) power plants large enough to supply electricity to high demand consumers will require large amounts of land. According to a report from NREL, a PV plant large enough to generate all of the electricity demanded by 1000 homes would use 3.2 acres, or .129 square kilometers [1], [2]. Often the land that is best for PV power generation is also land that could be best used for agriculture. Large open fields or equator-facing slopes that receive the highest amount of yearly irradiation are generally in high demand for farmers and industrial agricultural producers as well as PV-plant builders. Initially, this appears to result in a conflict of interest: How should a governmental agency zone land use? Should renewable PV energy be a priority, or agriculture? In some instances, one option or the other makes the most sense. For example, a plot of land's may not be viable for agriculture in which case a PV-only power plant could be a better option. An example of this are the large spaces of arid desert land in the Sahara. In other instances, where the land is desirable for both PV power and agriculture, an alternate solution presents itself: agri-PV. "Agri-PV" is short for agricultural photovoltaics, which is essentially the clever combination of using a plot of land for both agriculture and PV power generation. "Agri-PV" has also been referred to as "agrivoltaics", "agrovoltaics" "solar sharing" and "agri-photovoltaics" in various projects and literatures [3]–[6]. Agri-PV as a concept was first introduced by Adolf Goetzberger and Armin Zastrow in 1981, as a solution to the competing interests surrounding the use of arable land for either agriculture or PV energy generation [4]. In 2004, the first prototypes of agri-PV were developed by Akira Nagashima in Japan, where it is known as "solar-sharing" [7]. This system of sharing sunlight between both a PV energy generator and an agricultural crop incorporates well-known concept in biology called the "light saturation curve", which refers to the relationship between the amount of Photosynthetically Active Radiation (PAR) a plant receives and the amount of photosynthesis that plant is able to perform. The reason this relationship is known as a "curve" is because there is a peak level of PAR that corresponds to a peak level of photosynthesis in each species of plant. This means that any level of PAR over that "light saturation point" cannot be used effectively by the plant [8]–[10]. Therefore, depending on the species of crop, there is a potential for a crop to be grown in a semi-shaded or completely shaded area underneath an array of PV modules with little to no reduction in photosynthetic production [11].

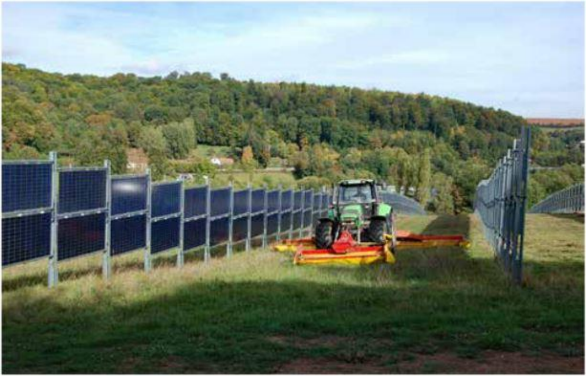


Figure 1- Vertical bifacial modules found at Eppelborn-Dirmingen solar park, Saarlan [6]

While the concept of agri-PV seems somewhat simple, the execution is not. The very nature of agri-PV requires an intimate knowledge of a broad range of subjects. On the physics side, this required knowledge includes photovoltaic semi-conductor physics, PV system analysis [12], the relationship between solar irradiation and PAR (photosynthetic active radiation) [13], and the physics of light diffusion [14].

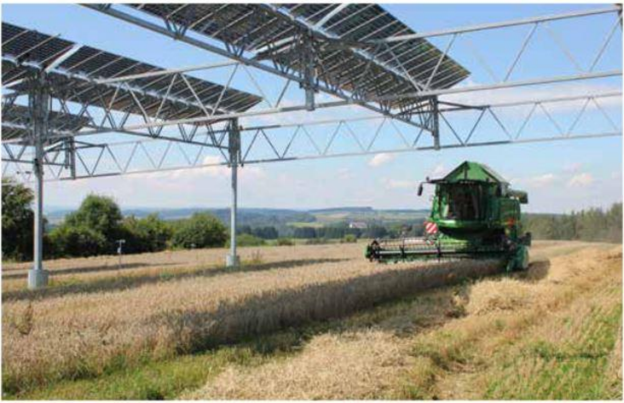


Figure 2 – vaulted PV modules with combiner underneath [6]

On the biological side, the required knowledge incudes the light saturation and compensation points of crops [15], optimal CO_2 concentration and temperature for a greenhouse crop [16], and (if possible) the potential growth model and crop yield of a greenhouse crop based on the aforementioned variables. Therefore, it seems logical that the topology of the agri-PV array must be tailored to the needs of the crop that will be grown underneath it.

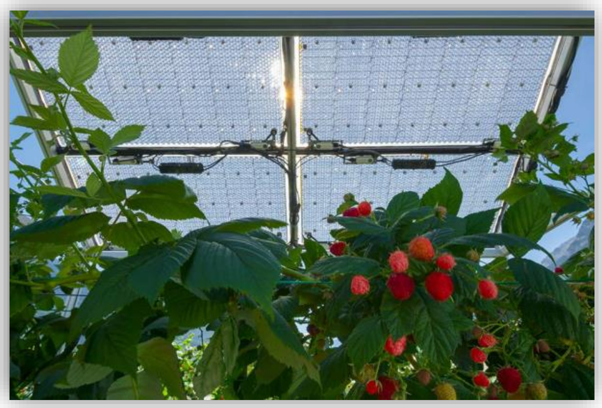


Figure 3 - Raspberries being grown underneath a thin-cell type solar array in Switzerland [17]

There have been a number of pilot projects that have used a wide variety of topologies to create an agri-PV system, including vertical bifacial panels that allow for a crop to grow between them (Figure 1) [6], vaulted PV modules that are mounted on structures that allow large farm vehicles to pass underneath them (Figure 2) [6], and smaller coverings that are simultaneously used for shade and cover from extreme weather (Figure 3) [17]. The variety of PV array topologies required creates a challenge since, unlike a standard PV power plant, there is no “one-size-fits-all” option that every farmer can choose from. However, this challenge could also be seen as an opportunity since such a wide variety of solutions for agri-PV systems means there will likely be a wide variety of players in the agri-PV market, meaning healthy amounts of competition and research. Also, it would put many farmers minds at rest since no two farms are ever the same, and the higher amount of variety means that they would be able to choose from a larger selection of options that are tailored to meet the needs of their farms.

Another factor is location. Even if the crop is the same between two different agri-PV systems, the system topology can still end up being vastly different depending on the location of the farm. If the farm is closer to the equator, an east-west vertical bifacial system may be a better option since the solar altitude angle changes much less throughout the day on average (meaning the sun is directly overhead for more hours of the day). If a system is built in a location further away from the equator, such as in Norway, the geometry and orientation of the agri-PV system's modules will benefit more from being oriented in a direction that faces the equator and has a tilt perpendicular to the sun's maximum elevation angle at solar noon.

This paper will focus on a single type of agri-PV application, namely using building-integration PV methods to conceptually mount bifacial PV modules on the roofs of simulated greenhouses. Greenhouses have an interesting potential for being transformed into agri-PV systems. Often for agri-PV systems, the mounting/support structure account for a significant part of the system costs, which greenhouses are able to reduce since the same support/structure is already required for the greenhouse itself.

The question then becomes one of determining just how much of a reduction in PAR a greenhouse owner can expect after installing PV panels over their greenhouse, as well as the amount of solar energy output that can be expected. To do this effectively, a robust form of modelling is required. In the case of this study, ray tracing is selected to model the incoming solar irradiance to ascertain both the expected amount of solar irradiation reaching the PV cells and the amount reaching the ground inside the greenhouse. The software selected to perform this simulation is Radiance via bifacial_radiance [18], a Python wrapper used to create Radiance objects of solar modules, structures, and sensors within the Radiance rendering engine.

This aim of this study is to present a scalable model of an agri-PV greenhouse system that can be manipulated to reflect various materials, particularly those used as outer greenhouse covering for diffusing light, as well as various PV array topologies. From these models several metrics are simulated that can assist in creating a dual yield for this type of agri-PV system. First, the model simulates the reduced PAR levels due to PV-module shadowing inside the agri-PV greenhouse when compared to a standard greenhouse, which is vital for calculating a crop yield. This reduction in irradiance can also be compared to the reduction in PAR with the consideration of the light saturation point (LSP) of a crop when a crop yield will be modelled. Second, an estimated energy yield is calculated for the simulated installed PV modules. Third, a comparison of these metrics is made between the different PV-array topologies and the two simulated greenhouse cover materials.

The structure of this study consists of a literature review on other agri-PV studies of similar nature, the impact of varying levels of PAR on a greenhouse crop and the modelling of crop yields, the impact of different types of greenhouse cover materials on the irradiance levels inside the greenhouse, and various methods of using optical modelling for simulating agri-PV systems. Following this, there is an explanation of how bifacial_radiance, a Python wrapper, is used to create a Radiance model in order to simulate incoming solar irradiation on both the bifacial PV modules and the ground inside the greenhouse. Modifications on module-type, array layout and placement, and greenhouse cover material types are conducted as well in order to ascertain how the ground irradiance within the greenhouse is affected by these variables. Finally, an energy yield on a monthly and yearly timescale are modeled, and an hourly comparison between a standard greenhouse and an agri-PV greenhouse is made for the duration of the tomato growing season.

2. Literature review

The literature review conducted for this paper consists of articles found with keyword searches on Oria and Google Scholar throughout the period of January to May of 2022. The searches were combinations of keywords included “agrivoltaics” “agri-PV” “solar sharing” “greenhouse” “tomato” “bifacial” “model” “ray tracing” and “Norway”. Papers that were referenced in the ones found via keyword search were also read and included if relevant.

2.1 Modelling technique selection and ray tracing

There are many ways to model energy output from PV modules. Some of the more commonly used models are PC1D [19], PVsyst [20], and pvlib [21]. For the solar irradiance input these models use either irradiance data provided by the user, reference EPW/TMY files created with ground irradiance measurements at weather monitoring stations, or pull satellite irradiance data from sources such as SolCast [22] to find the necessary GHI (global horizontal irradiance), DNI (direct normal irradiance) and DHI (diffuse horizontal irradiance) for a given location base on latitude and longitude. However, while the data mentioned is enough for modeling and approximating the energy output from PV panels, there is a deficit in the modeling of shadows created by the panels which is necessary for accurately modelling agri-PV systems. For this type of modelling, ray tracing becomes a viable candidate.

As show in Figure 4 (adapted from [23]), ray tracing is a form of image rendering that utilizes millions of simulated rays to create a 2D image of a 3D object. It does this by sending these simulated rays from a chosen 2D plane (typically the individual pixels that make up a viewing screen) at the 3D scene and calculates the color of each pixel by the number of objects that ray bounces off before terminating at the source of light for the image. This is similar to how the human eye works, except ray tracing does this in reverse with the ray originating from the eye/observer/display screen. By doing this, not only can more realistic shadows be rendered, but also additional light sources that are the result of light reflections bouncing off objects in the 3D scene. In typical ray

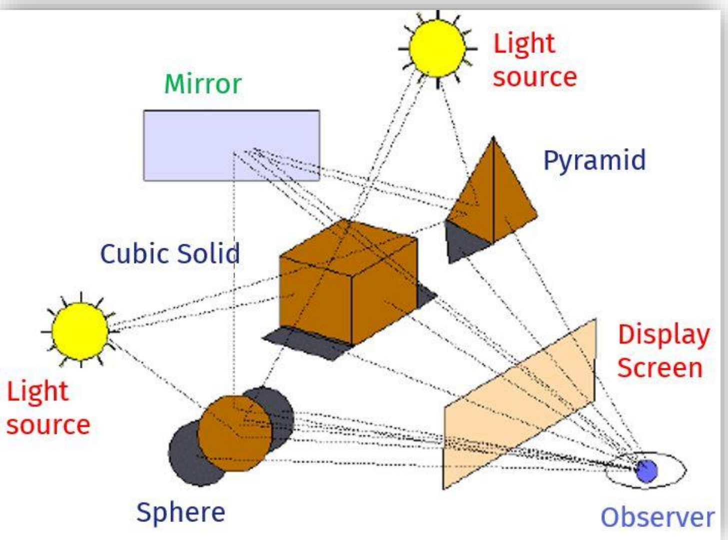


Figure 4 - Diagram showing how ray tracing works, adapted from [23]

tracing that is used for real-time rendering, such as that used in the video game engine Unreal Engine 4 (UE4), this allows for incredibly photorealistic images and virtual environments. UE4 is the original choice of rendering engine for this project for this reason. However, newer rending engines typically have one flaw when it comes to accurate models: they typically do not prioritize physical sunlight irradiance accuracy. Photorealism, while key for artistic mediums such as film and video games, is often not based on the physical photon wavelengths of real light sources, whether that be sunlight or artificial light. There are, however, ray tracing programs that provide models based on sunlight irradiance accuracy. One that has been in use in the architecture industry since 1985 is Radiance, developed by Greg Ward [24]. In terms of generating validated results, Radiance is a physically-based light simulation tool which integrates sky data and produces results comparable to real-life scenarios and has had its output validated against real-world irradiation data many times. It is a trusted rendering engine still used to this day for projects requiring results that are as close to real world measurements as possible. This ray tracing software's base unit for its rays is W m^{-2} steradian^{-1}, all of which is based on physical photon wavelengths. There are several other advantages and disadvantages to using engines like UE4 (and now Unreal Engine 5 aka UE5) or Radiance to model these sorts of projects, which will be discussed in the discussion section 6.1. Additionally, some tests of UE4 have shown that the video game engine is able to produce irradiance measurements comparable to the measurements produced by Radiance [25], [26].

2.2 PAR, Light Saturation Curve, and Crop Yield

Photosynthetic active radiation (PAR) refers to the spectrum of light that can be utilized by plants for photosynthesis and primary production. This spectrum consists of light with wavelengths of 400 to 700 nm and falls within the visible range of light [27], [28]. Most terrestrial ecosystem models use global PAR, or PAR received from all directions. Since the simulated models of this paper use diffusive covers, and therefore the irradiance measured at ground level within the greenhouse is diffused irradiance, the effect of diffusion on PAR should be understood and accounted for. Also, since solar irradiance consists of wavelengths outside of the PAR spectrum, an understanding of the percentage of solar irradiance that makes up PAR is necessary. There are several studies that have examined this relationship between irradiance and PAR and have determined a conversion factor that suitably relates them [13], [29]. This is known as a quanta-to-energy ratio since PAR is often measured in quantum units (photosynthetic photon flux density, PPFD, µmol m^{-2} s^{-1}) and irradiance is measured in energy units (W m^{-2}). Several of these studies have concluded that a conversion factor of 4.56 µmol J^{-1} between PAR (µmol m^{-2} s^{-1}) and irradiance (W m^{-2}, J m^{-2} s^{-1}) is relatively insensitive to cloud scattering and diffusion from atmospheric conditions [13], [29], [30]. One study [13] on the effects of diffusion on global PAR found that as the percentage of diffuse PAR increases, the conversion factor slowly increases as well, from a low of 4.23 µmol J^{-1} at percentages of diffuse PAR 23% to 24.5% to 4.54 µmol J^{-1} at percentages of diffuse PAR between 46% to 49%. Higher levels of diffusion were not studied in the aforementioned paper. However, for this study and the models simulated, the global PAR conversion factor of 4.56 µmol J^{-1} seemed sufficient for estimating the conversion between PAR and irradiance.

There have been many studies that examine the correlation between the amount of PAR (photosynthetically active radiation) a crop receives and the change in a marker used to measure net photosynthesis in that crop, usually net CO_2 utilization (measured in µmol CO_2 m^{-2} s^{-1}) [31] [9], [11], [14], [32]. As mentioned earlier, the relationship between the level of PAR a plant receives and the amount of photosynthesis is able to perform is known as the light saturation curve and shows both the light compensation point (LCP) and the light saturation point (LSP) for that plant. As shown in Figure 5 (adapted from [6]), the LSP and LCP of each plant is unique to the crop species. These points can also be influenced by environmental factors such as temperature, humidity and the level of direct vs diffuse PAR. A study on the effects of diffusive covers on greenhouse grown tomatoes found that higher levels of diffusion (20% haze factor vs 29% haze factor) in the covers resulted in tomatoes with a higher growth yield in both stem and fruit, as well higher amount of photosynthesis by decreasing the leaf and air temperatures during the hotter summer months [14], [33].

It is important to understand how particular crop yield studies and predictions are. Crop yield models are nearly always specific to a location, growing method (outdoor, indoor, greenhouse, etc.), light source (full sun, LED, growing lights, diffuse light, etc.), ambient CO_2 concentration, and temperature. While these are generally the main factors found in a crop yield model, there can be additional factors accounted for such as soil type, nutrient makeup and concentration, and crop phenotypes and mutations. Since the primary focus of this study is using ray tracing to model the effects of module shading on the light levels inside a greenhouse, a complete and comprehensive crop yield analysis is determined to be outside the scope of this paper. However, calculating the level of PAR inside agri-PV greenhouses is certainly a

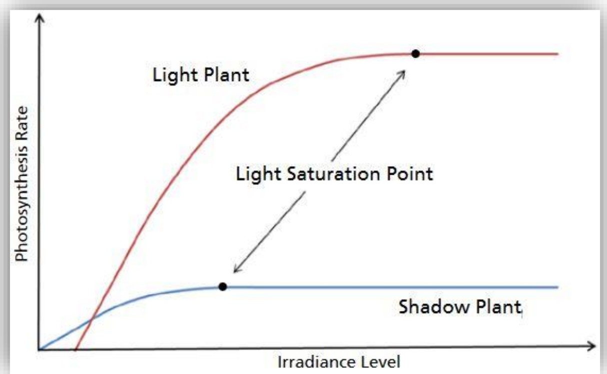


Figure 5 - Simplified diagram of light saturation curves for a shade tolerant plant (shadow plant) and a shade intolerant plant (light plant)

key measurement that is required when conducting a full crop yield estimation. It is with this in mind that the study focused on showing the expected reduction in PAR when comparing a standard greenhouse to an agri-PV greenhouse, as well as showing an example of a typical Norwegian greenhouse crop's (tomato) LSP and LCP [27] in comparison to the levels of PAR inside these greenhouses. Using the aforementioned conversion factor of 4.56 µmol J^{-1} the tomato's light saturation point of 754.5 µmol m^{-2} s^{-1} PAR is converted to 164.7 W m^{-2} irradiance and the light compensation point of 38.9 µmol m^{-2} s^{-1} PAR is converted to 8.53 W m^{-2} irradiance. It is the hope that if these measurements can be simulated accurately, they would be able to contribute to a final crop yield analysis that incorporates all the required input including greenhouse humidity, temperature, growing structures, nutrient concentration, and others.

2.3 Examples of other Agri-PV modelling projects

Other studies similar to this paper were also studied throughout the literature review in order to gain some insight into what has already been researched concerning the use of ray tracing or other types of simulations for modelling agri-PV systems and what methods were used [33]–[35]. Some studies of particular note also used ray tracing with Radiance to perform their simulations e.g., with a focus on measuring changes caused by differences in albedo [34], mounting height and structure [17] or with a focus on evaluating the performance of a different topologies of vertically oriented bifacial modules [15]. From these studies, inspiration was drawn for which metrics to observe and measurements to take, such as finding the difference between the amount of ground irradiance received between the different models, testing the simulations of different greenhouse cover materials, testing different PV array topologies, and which hourly, monthly and yearly time-series to make for examining the effects of changes to these variables. In the case of [15], material parameters for a semi-transparent diffusive greenhouse cover in Radiance were also used and further studies referenced in the paper were used to approximate the parameters necessary for the second greenhouse cover, as is discussed in section 3.1.3.

3. Building the greenhouse model

3.1 The Radiance engine and bifacial_radiance

As mentioned earlier, the original engine of choice was Unreal Engine4 (UE4). This is for several reasons. First, UE4 has a robust community of users. There are plenty of forums and tutorials on how to get started and excel with UE4. As a result, the learning curve isn't as steep as some other 3D modelling software and engines. Another reason for using UE4 is its ability to utilize a GPU (graphics processing unit) to perform ray tracing quickly and efficiently in real-time. This allows for a user to not only be able to interact with their model as they build it, but it also allows for a user to walk through their finalized model in real time, something that cannot be done in bifacial_radiance and Radiance. The ability to create and walk through a photorealistic model in UE4 is appealing since it not only adds an extra level of functionality and interactivity, but it also adds an aesthetic appeal to the model that could be valuable when a modeled agri-PV system is shown to a potential client.

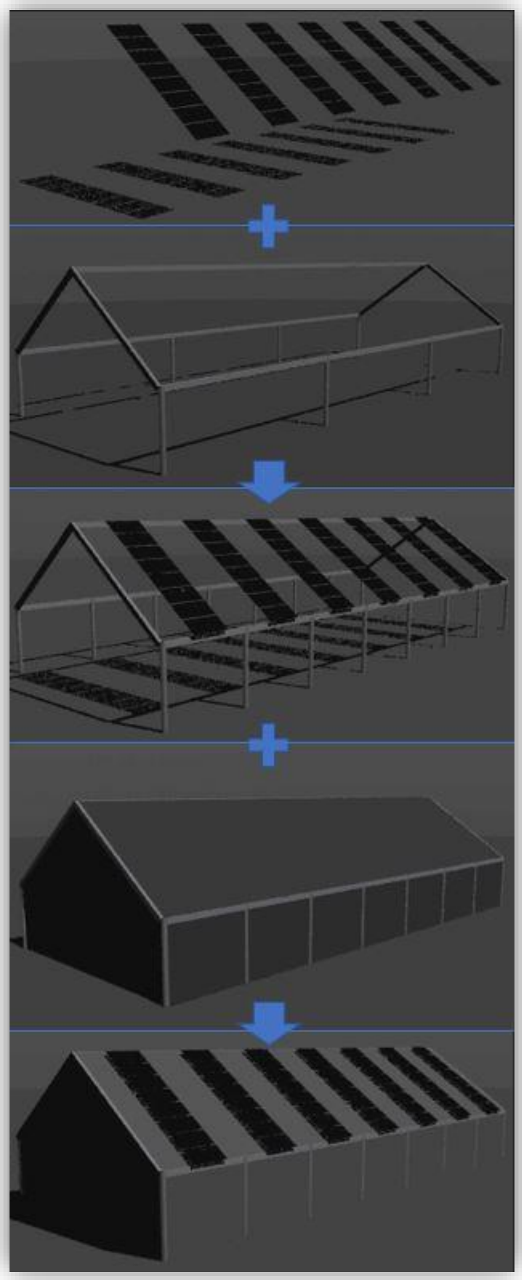


Figure 6 - The model building process and the different parts of the model. The model starts with the modules alone, then the beams are added, and then the greenhouse walls, roof, and floor (not visible) are added for the final complete model.

There is one major drawback from using UE4: it is difficult to accurately assess solar irradiance received by the PV-panels and the ground underneath since creating light irradiance sensors is not something that the engine does natively. The validity of the irradiance data from UE4 is also in question. One study on the validity of irradiance data from UE4 found that when setting the number of ray traced bounces to higher levels (7 bounces per ray instead of 3 per ray), the irradiance data simulated is very close to the validated irradiance data from Radiance [25].

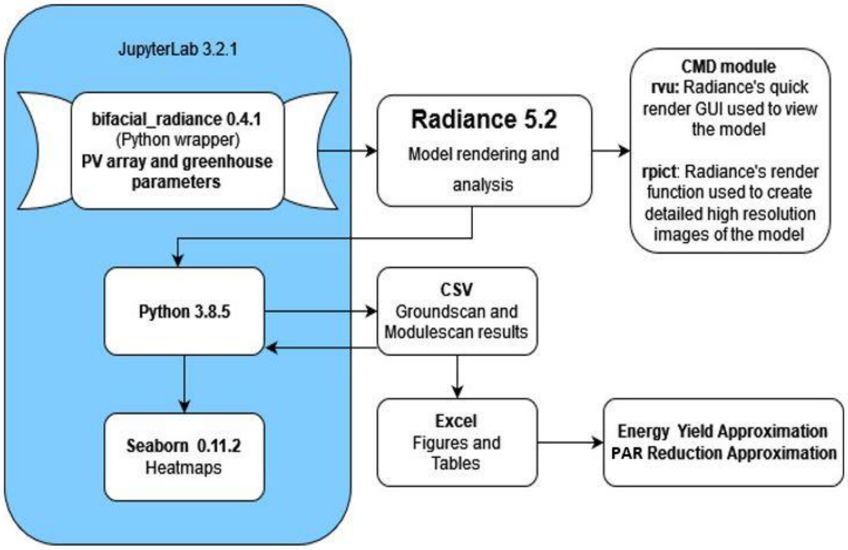

JupyterLab 3.2.1
bifacial_radiance 0.4.1
(Python wrapper)
PV array and greenhouse parameters
Radiance 5.2
Model rendering and analysis
CMD module
rvu: Radiance's quick render GUI used to view the model
rpict: Radiance's render function used to create detailed high resolution images of the model
Python 3.8.5
CSV
Groundscan and Modulescan results
Seaborn 0.11.2
Heatmaps
Excel
Figures and Tables
Energy Yield Approximation
PAR Reduction Approximation

Figure 7 *– Ray traced model input/output flow chart*

The Radiance engine, while not having the advantage of real-time interaction and real-time photorealistic ray tracing, does one thing extremely well: it gives physically accurate and validated measurements of solar irradiance. For this reason, as well as those discussed in Section 2.1, Radiance is ultimately selected as the best choice for conducting these simulations. The next step is choosing the best way to create the model within Radiance. There are many options for creating models in Radiance, depending on the application. For example, Rhino3D [36] has several plugins that make it possible to export a 3D object from Rhino3D or another CAD modelling software to Radiance [33]. However, this would require additional programming and 3D modeling to correctly model the PV cells, modules, and array. Bifacial_radiance is a Python wrapper used to create the specific lines of code required by Radiance to build a PV system model and perform an analysis of the PV model at the cell level [18]. Bifacial_radiance does a great job of not only building cell-level specific PV modules, but also placing virtual sensors that will give analysis feedback on how much irradiance the PV modules receive. This is one of the core functions of bifacial_radiance and is the reason why the PV modules are the first part of the model built as seen in Figure 6 where the modules are alone, floating in mid-air. As seen in Figure 7, the bifacial_radiance wrapper is the starting point for the entire model. This wrapper is freely available on GitHub. Anaconda 2.1.1 is used as the launch platform for JupyterLab 3.2.1. All model simulations are built and ran in JupyterLab with Python 3.8.5. With bifacial_radiance, the parameters for the PV array are set, and the code necessary to build and analyze the model with Radiance is generated by the wrapper. This is explained in detail in Section 3.1.4 below.

3.1.1 Selecting the greenhouse model's site, shape and dimensions

The site selected is a section of rural open farmland in Kjeller, Norway. The reason for this selection is locality to the researcher (Kjeller, Norway) where there are several greenhouses of approximately the same size and form as the model created in Radiance with bifacial_radiance. The greenhouse model's shape and dimensions are based on an approximation of the same local industrial greenhouse in Kjeller, Norway. This greenhouse has a floor space of approximately 540 m^2 and is a little less than 10 meters tall at its highest point.

3.1.2 Selecting the panel size and type

The simulated modules are built to be 72 cell PERC monocrystalline bifacial modules. There are many companies that make modules like this, including Heliene, Longi, Trina Solar, etc. There are options to have bifacial glass-glass modules where both the front and rear encapsulating sheets are made of glass allowing some light to pass through the module between the cells where it will be absorbed by the rear cells after bouncing off any surface or object behind the module. For this simulation, the simulated module is based on the Heliene 72M G1, 405 W$_p$ module with slightly different dimensions [37]. Most notably, a larger gap of 2cm between cells. The module's efficiency, bifaciality factor, and cell size are used when calculating the estimated energy output each array would have throughout the year, as shown in Table 1.

3.1.3 Creating the model

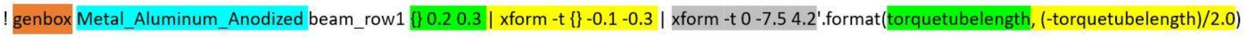
Creating the agri-PV greenhouse model in bifacial_radiance requires a bit of knowledge in both Python and creating objects and materials in Radiance. Even though it does not have the same robust user network and tutorials that UE4 has, the bifacial_radiance GitHub [38] offers many helpful tutorials for setting up module parameters, defining the required inputs for spacing the modules in the PV array, the azimuth angle of the PV array, and the latitude and longitude of the model's location, and even a tutorial on how to create additional structures and force the sensors to be placed at ground level during an analysis in order to get ground irradiance measurements in an agri-PV system. Bifacial_radiance also allows for creating a single axis tracking array and has a built-in function that creates a torquetube that the modules would rotate on. Since all arrays modeled for this study are fixed-tilted roof-mounted arrays, this option is not explored.

Table 1 - Simulated module details

Cell size	15.6cm * 15.6cm
Gap between cells	2 cm
Module orientation	12x6 - Landscape
Module size	2.1m * 1.1m
Module material	Glass-glass, frameless
Module efficiency	20.38%
Bifaciality factor	0.7 (70%)
Total area of cells	98.12 m^2

For all of the ray traced models, the PV module settings (shown in Table 1) are the same with the dimensions being mostly based on the aforementioned Heliene bifacial module. All modules in all models had an azimuth angle of 180°, which for bifacial_radiance means facing due south.

After setting the parameters for the PV array, bifacial_radiance uses the latitude and longitude provided by the users to pull an EPW (EnergyPlus Weather) file from the NREL (National Renewable Energy Laboratory) GitHub that contains TMY (Typical Meteorological Year) [39] data that the model will use for determining the DNI (direct normal irradiance), DHI (diffuse horizontal irradiance) and GHI (global horizontal irradiance) for each hour throughout the year in the given location. The EPW data is pulled from the closest weather monitoring station that NREL has access to. In the case of this study, the Oslo Fornebu location is used. After this EPW file is pulled, bifacial_radiance uses one of two functions to build the model's 'sky': *gencumsky* or *gendaylit*. *Gendaylit* is a Radiance-standard function that creates a scene with irradiance levels based on a single hour. This is what's used for all of the simulations used to generate hourly results, like the heatmaps and the hourly time-series of the tomato-growing season found in the results section. *Gencumsky* is a bifacial_radiance function that creates a 'cumulative sky' by adding all of the hourly irradiance levels for a selected time period (from hours to a full year) and outputs the insolation in Wh m^{-2} for that period in a single analysis, saving processing time.



```
! genbox Metal_Aluminum_Anodized beam_row1 {} 0.2 0.3 | xform -t {} -0.1 -0.3 | xform -t 0 -7.5 4.2'.format(torquetubelength, (-torquetubelength)/2.0)
```

Figure 8 - Example of a line of code used to generate an object; code is found inside a .rad file used to build the final .oct file

From here, the rest of the model's structure needs to be created. Radiance uses a specific command line format to generate objects in different shapes and sizes. For the models create in this study, all the various beams, pillars and roof support structures of the greenhouse need to be generated by a specific line of code that creates a *.rad* file which is then used by bifacial_radiance to create the final *.oct* file. An example of this code can be seen in Figure 8, which contains the object geometry command '*genbox*' (orange) explained in Table 2, the object's material name (light blue) which can be found in the materials file generated by bifacial_radiance, the object dimension parameters (green), the initial movement (-t, transform) in the x y and z direction (yellow) and the secondary movement in x, y, and z (grey). All values are based in meters, so 0.1 is equivalent to 10 cm. The model of the greenhouse required separate *.rad* files for each object in the

Table 2 - List of commands used to generate various geometrical shapes in Radiance

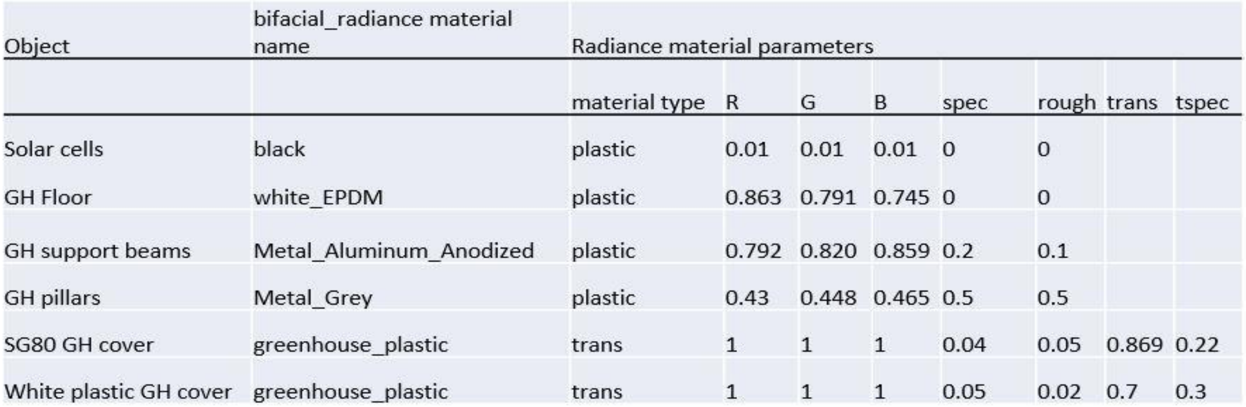
genbox	*generate a RADIANCE description of a box*
genprism	*generate a RADIANCE description of a prism*
genrev	*generate a RADIANCE description of surface of revolution*
gensky	*generate a RADIANCE description of the sky*
gensurf	*generate a RADIANCE description of a functional surface*
genworm	*generate a RADIANCE description of a functional worm*

model, which included greenhouse support beams, pillars, walls, roofs and a special prism object for each of the east and west facing walls. Most of these objects are based on cubic geometry, except for the pillars which used *genrev* to create cylinder objects, and the gaps between the rectangular parts of the eastern and western walls and the roof, which required the use of *genprism* to create a 3-dimensional polygon that fit the space. After all of the objects are generated, bifacial_radiance will build an .oct file that will contain the model objects, the generated sky (based on the sun location, DNI, DHI, and GHI provided by the EPW file), and the material list, all of which are .*rad* files [40].

One thing to note about the way that bifacial_radiance generates the location of the PV array: it places the center of the array at the (0,0,0) origin point of the Radiance scene. This is necessary to know since all other objects are generated from the origin point in whatever x, y, or z direction specified, and therefore must be offset (xform -t, transformed) by one half of their length in the western direction in order for them to be centered with the PV array. This is why in Figure 8, the second section in yellow that specifies the secondary transformation of the object (-torquetubelength/2.0) exists.

It should also be noted that this part of the process, the building of the model, took the longest to complete. There were ample amounts of debugging and hours of searching for answers on how to create objects in Radiance, as well as a lot of time looking into just how bifacial_radiance generates *.rad* files and how it places sensors for analysis.

Table 3 - List of materials used and their Radiance parameters



Object	bifacial_radiance material name	Radiance material parameters							
		material type	R	G	B	spec	rough	trans	tspec
Solar cells	black	plastic	0.01	0.01	0.01	0	0		
GH Floor	white_EPDM	plastic	0.863	0.791	0.745	0	0		
GH support beams	Metal_Aluminum_Anodized	plastic	0.792	0.820	0.859	0.2	0.1		
GH pillars	Metal_Grey	plastic	0.43	0.448	0.465	0.5	0.5		
SG80 GH cover	greenhouse_plastic	trans	1	1	1	0.04	0.05	0.869	0.22
White plastic GH cover	greenhouse_plastic	trans	1	1	1	0.05	0.02	0.7	0.3

3.1.4 Selecting model materials and creating a transparent material in Radiance

All materials used for creation of the models are in Table 3. Most of the materials' parameters needed for the Radiance objects' that are built in bifacial_radiance are provided the first time the Python wrapper runs. It creates a file called *ground.rad* in the *materials* folder of the *testfolder* that the user has specified. The first four materials listed in Table 3 are all materials created in this fashion. These "plastic" materials require 5 parameters. The first three are R, G, and B values ranging from 0 (black) to 1 (white).

A custom color can be created by using varying amounts of each value e.g., for a white material the combination of values is simply 1, 1, 1 and for a purely red material the values are 1, 0, 0. The next two values refer to the material's specularity (reflectance) and roughness. In Table 3, these values are 'spec' and 'rough'. The acceptable values for specularity range from 0 (matte) to 1 (satin) but the suggested maximum value is 0.02. For roughness, the acceptable values also range from 0 (polished) to 1 (low gloss), but the suggested maximum for this value is 0.07 [24].

In order to accurately simulate a diffusive greenhouse cover, bifacial_radiance (and Radiance) requires the creation of a custom material that reflects some of the real-world properties of a diffusive cover material. A "trans" material, as it's called in Radiance, requires seven parameters. The first five parameters are the same as a "plastic" material. The final two parameters are transmission ('trans' in Table 3) and transmitted specularity ('tspec'). Transmission refers to the fraction of light that travels through the material. The value ranges from 0 (opaque) to 1 (100% transparent). The final parameter is transmitted specularity. This is the fraction of the transmitted light that is not diffusely scattered. The values can range from 0 (completely diffuse) to 1 (clear). This value is essentially the opposite of the haze factor. For example, the SG80 material is a glass-based taken from [33] that has a haze factor of 78%, so the value for tspec is 0.22, meaning only 22% of the light transmitted through the material is not scattered do to diffusion. For the theoretical greenhouse plastic material, labeled at "white plastic GH cover" in Table 3, the haze factor is set to 70% and the transmission value is also set to 70%.

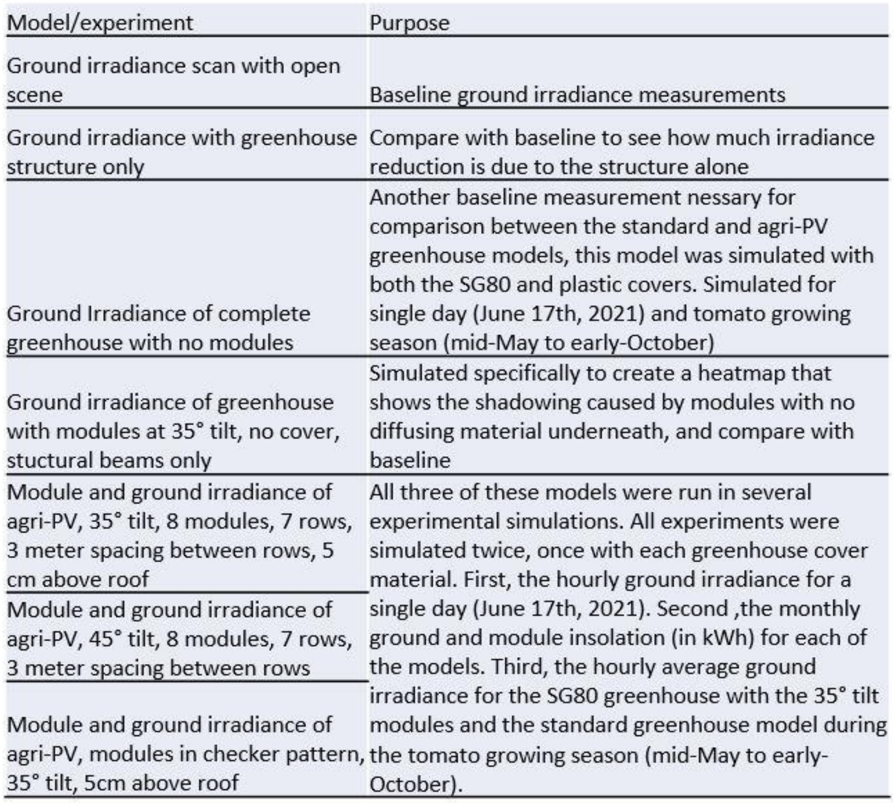
3.2 Analysis of the model

Where bifacial_radiance excels is in creating *.rad* files for simulations of PV modules from user specifications and creating sensors within the Radiance scene in order to generate measurements irradiance data from an analysis. The ray tracing analysis functions similar to how Figure 4 depicts ray tracing, except instead of the focal point being the 'observer' or the 'display screen', the focal point is one of the many sensors placed within the scene. At its current build (0.4.1), bifacial_radiance doesn't fully support native sensor placement on user-specified cell-level modules. What this means is that instead of each cell having its own sensor, the sensors are placed in either a single vertical line in the middle of the PV array or in a diagonal line across the array (if the user specifies this). However, it is possible to create a loop in Python that will place as many vertical lines of sensors as specified, as seen in Figure 9. This process takes a much longer time, but outputs a much more detailed map of irradiance values. After the analysis is complete, a dataframe is generated containing the amount of irradiance received at the front and back of the module, as well as the material that the scan originated from. This allows the user to run a 'clean results' function that sets all measurements originating from non-PV-module parts of the model to NaN (not a number) so that final results and summations are not contaminated with ‘bad’ data.

3.3 Selecting parameters to test

Once a working model is created, there are many different parameters that need to be tested. The list of experiments simulated can be seen in Table 4. First is getting ground-level irradiance data for an open scene with no objects or structures. This gives a baseline measurement for irradiance values generated by Radiance and allows for a comparison to real-world values and the subsequent experiments.

Table 4 - List of models created and experiments simulated



Model/experiment	Purpose
Ground irradiance scan with open scene	Baseline ground irradiance measurements
Ground irradiance with greenhouse structure only	Compare with baseline to see how much irradiance reduction is due to the structure alone
Ground Irradiance of complete greenhouse with no modules	Another baseline measurement nessary for comparison between the standard and agri-PV greenhouse models, this model was simulated with both the SG80 and plastic covers. Simulated for single day (June 17th, 2021) and tomato growing season (mid-May to early-October)
Ground irradiance of greenhouse with modules at 35° tilt, no cover, stuctural beams only	Simulated specifically to create a heatmap that shows the shadowing caused by modules with no diffusing material underneath, and compare with baseline
Module and ground irradiance of agri-PV, 35° tilt, 8 modules, 7 rows, 3 meter spacing between rows, 5 cm above roof	All three of these models were run in several experimental simulations. All experiments were simulated twice, once with each greenhouse cover material. First, the hourly ground irradiance for a single day (June 17th, 2021). Second ,the monthly ground and module insolation (in kWh) for each of the models. Third, the hourly average ground irradiance for the SG80 greenhouse with the 35° tilt modules and the standard greenhouse model during the tomato growing season (mid-May to early-October).
Module and ground irradiance of agri-PV, 45° tilt, 8 modules, 7 rows, 3 meter spacing between rows	
Module and ground irradiance of agri-PV, modules in checker pattern, 35° tilt, 5cm above roof	

Next, a greenhouse structure no modules and no diffusive cover is simulated to see how much the ground irradiance is reduced by the greenhouse structural beams alone. If the reduction is significant, this could be necessary to account for when making comparisons between the different models. If not, then it can be safely ignored.

Next, a complete standard greenhouse with diffusive cover is modeled and the ground irradiance is measured. Both greenhouse (SG80 and plastic) cover materials are tested and compared. This standard greenhouse model (with the SG80 cover) is also used for getting average hourly ground irradiance data during the tomato growing season, as shown in section 5.7.

Finally, the completed agri-PV models are simulated. These included 6 different models, consisting of 3 PV array topologies and 2 diffusive cover materials. The models are used in several experiments, including the single day hourly ground irradiance (June 17, 2021) which are used to generate the heatmaps found in section 5.1-5-5, and the hourly ground irradiance measurements simulated during the tomato growing season, as seen in section 5.7.

This final experiment is conducted, and the ground irradiance levels are compared to the light saturation point (LSP) and light compensation point (LCP) of a greenhouse grown tomato crop.

4. Energy Yield and Light Saturation Models

4.1 Energy Yield Calculations

For the energy yield, the models are used to simulate the monthly insolation falling on the panels (Wh m^{-2}). The equation used to calculate the approximate energy yield is the following:

$$TotalMonthlyEnergy = A*Eff*PR*(Ins(Front)+(Ins(Rear)*BF)) \tag{1}$$

Where *A* is the total area of the PV cells, *Eff* is the module efficiency (shown in Table 1), *PR* is the performance ratio for each month (shown in Table 5), *Ins* is the average insolation (in kWh m^{-2}) received by the front or rear cells each month, and *BF* is the bifaciality factor of the module (shown in Table 1). The performance ratio values were adapted from a research paper on the performance ratios of 15 different PV systems in Norway [41]. The average of the monthly performance ratios across the 15 systems was used for this study's monthly performance ratios, shown in Table 5. It should be noted that any performance ratios of zero in the paper referenced were excluded since it was likely that the system was non-functional for that month.

Table 5 - Table of monthly performance ratios adapted from [41]

Month	Performance Ratio
Jan	0.247
Feb	0.53
Mar	0.588
Apr	0.885
May	0.946
Jun	0.936
Jul	0.957
Aug	0.954
Sep	0.922
Oct	0.846
Nov	0.641
Dec	0.289

4.2 Converting solar irradiance to PAR

As discussed in section 2.2, there have been several studies on the conversion of irradiance (in W m^{-2}) to PAR (in µmol m^{-2} s^{-1}). One of the main issues with converting between the two values is that W m^{-2} is often measured as a unit of energy and PAR data typically uses quanta units. The other issue is that irradiance accounts for a larger part of the light spectrum than PAR (400-700 nm) and the amount of PAR a plant receives is also influenced by the level of diffusivity in the atmosphere, whether that be from clouds or aerosols or, in the case of a greenhouse, a diffusive cover [13]. It has been found, however, that a conversion rate of 4.56 µmol J^{-1} is sufficient for converting between irradiance and PAR with minimal or zero error. Since it's a simple conversion, in this paper the term 'irradiance' is often used in place of PAR.

4.3 How crop yield studies are conducted

As discussed in section 2.2 of the literature review, crop yield studies are nearly always specific to both a single crop and climate. This means that it is necessary to have a crop yield study specific not only to the species of crop being grown, but also to the location and environment of the crop. In the case of crop yield models performed in greenhouses, the temperature, light quality, humidity and several other measurements are also necessary when creating the model. The original intention of this paper was to have a complete crop yield model alongside the energy yield model. However, as the research progressed, it quickly became apparent that this was outside the scope of the thesis and that it would be better to concentrate on creating accurate measurements of one of the key metrics used in crop yield models: PAR levels received by the crop. It is for this reason that the crop yield model was changed to a PAR level model with comparisons to an example crop's LSP and LCP, a greenhouse grown tomato.

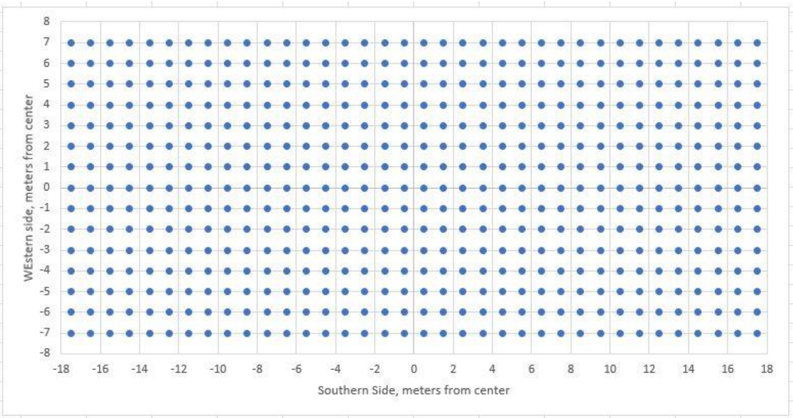


Figure 9 - Sensor placement at ground level for heatmaps

4.4 Modelling PAR levels in comparison to light saturation points

The inside of a greenhouse is a semi-controllable microclimate. Depending on the complexity of a greenhouse's design, temperature, light diffusivity, nutrient concentration, and light cycles can all be controlled. PAR, the amount of photosynthetically active radiation a plant receives every second, plays a significant role in the total photosynthesis a plant performs [27], [31].

For this reason, three different types of ground irradiance models are simulated. First, hourly heatmaps for a single day in June are created in order to show the effects of diffusive greenhouse cover materials on both the levels of PAR inside the greenhouse and on shadowing patterns. The sensor placement for the simulations to create the heatmaps can be seen in Figure 9.

Second, average monthly ground insolation (measured in Wh m^{-2}) is simulated in order to compare between the standard and agri-PV greenhouse models and show how much of a reduction in insolation can be expected for each month. This simulation is conducted with only the SG80 covered greenhouses

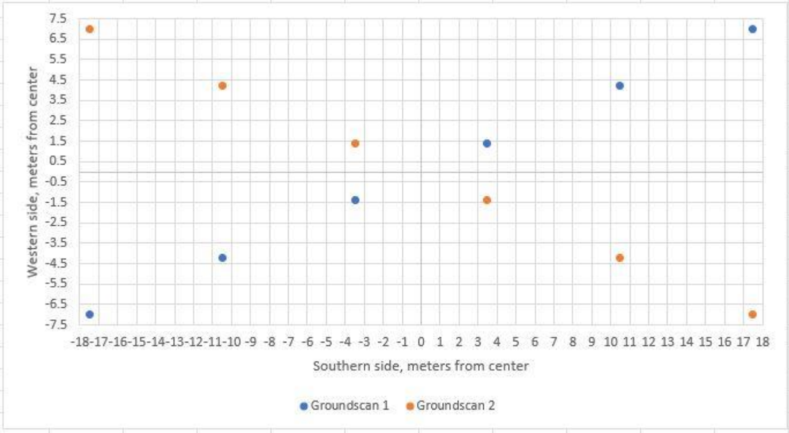


Figure 10 - Sensor placement for hourly average ground irradiance measurements

and with only the 35° tilt modules since it had been discovered that all models with the same cover material had nearly identical average levels of ground irradiance.

Finally, a simulation of the average hourly ground irradiance for the growing season of greenhouse grown tomatoes (mid-May to early-October) is conducted [16]. The models used are the standard greenhouse with no modules installed and the SG80 cover, and the 35° tilted agri-PV greenhouse with the same SG80 cover. Since these hourly irradiance simulations take a very long time to process, the total number of sensors is reduced to 12. The placement of the sensors at ground level inside the greenhouse for these measurements can be seen in Figure 10. The sensors are placed this way to get the most evenly distributed measurements for an average ground irradiance measurement. For the results of this final simulation, the LSP and LCP of tomatoes (164.7 and 8.53 W m^{-2} respectively, calculated from 754.5 and 38.9 µmol m^{-2} s^{-1} respectively [27]) was used as a maximum and minimum value for the ground irradiance, since values above the LSP wouldn't lead to additional growth and values below the LCP wouldn't lead to reduction in growth. With this in mind, the percentage of PAR reduced when the light levels in the agri-PV greenhouse were calculated, and an example of a single day's PAR reduction was plotted (Figure 29). In order to do this, the irradiance values for both of the models were bound between the LSP and LCP, meaning all ground irradiance values above the LSP were set to 164.7 W m^{-2} and all irradiance values below the LCP were set to 8.53 W m^{-2}. The percentage of irradiance lost was calculated with the following equation:

$$\frac{GH - M}{LSP} * 100 = PerentageIrradianceLost \qquad (2)$$

where *GH* Is the average hourly ground irradiance inside the standard greenhouse, *M* is the average ground irradiance for the same hours inside the agri-PV greenhouse with modules, and *LSP* is the light saturation point for tomatoes.

5. Results

Sections 5.1-5.5 will contain ground irradiance heatmaps showing the shadowing caused by the modules. All heatmaps are based on a bird's-eye view of the greenhouse floor, with the bottom of the heatmap corresponding to the southern side of the greenhouse. Each section will contain a description and photo of the rendered greenhouse model itself followed by the gathered data and a short explanation. For all heatmaps shown, the timestamp of 11am on June 17, 2021 is used. This is an hour with a high amount of irradiance while the GHI is not at its peak for the day, meaning that shadowing would be more visible in the heatmaps since the light at peak hours tends to wash out the shadows caused by the modules. June 17 is selected as a day with relatively high irradiance but still partially cloudy so that additional diffusion caused by cloudiness could be observed. The hours of 5 to 20 were simulated for all models as well, and the example of a single day's heatmaps can be seen in Appendix A.

5.1 Model with greenhouse structure (no diffusive glass or plastic cover) and modules only

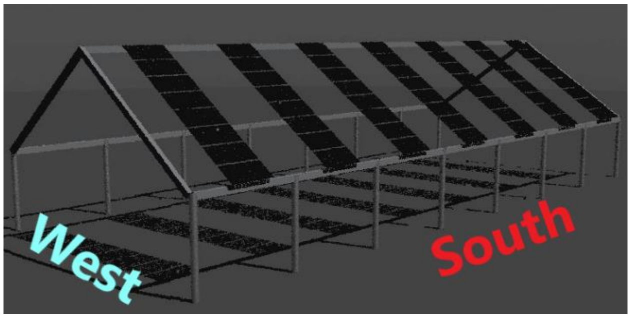


Figure 11 - Low resolution render of the model with greenhouse structure and modules only – Southern and Western sides labeled

The model shown here (Figure 11) is created in order to show the amount of ground irradiance that would be simulated when there is no diffusive cover. In order to simplify the process of creating this model, and all the agri-PV greenhouse models, mounting rails for the modules themselves are omitted. This first model is also created in order to observe how the ray traced simulation would handle shadowing and the amount of irradiation that would still be present in the shadows of cast by the modules and the greenhouse support structure. This creates a baseline for the simulated environment, allowing for a comparison with real-world irradiation data. Satellite irradiation data obtained from Solcast [22] indicates that the global horizontal irradiance at the 11am on the 17th of June, 2021 is 757 W m^{-2}, which is close to the simulated irradiance. In

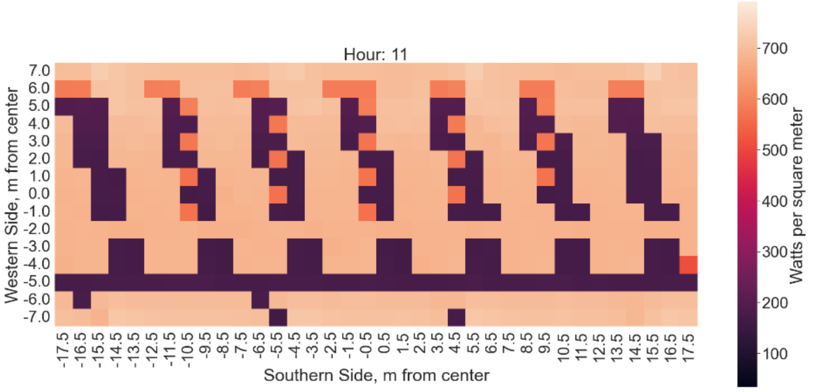


Figure 12 – Heatmap of Figure 11 ground irradiance at 11am on 2021-06-17 Irradiance range: 33.5 to 799.0 W m^{-2}

the simulated data used to create the heatmap seen below, full sun irradiance levels between 680-740 W m^{-2} are observed.

From the ray traced model shown in Figure 11, the following heatmap of the ground irradiance is created (Figure 12). As explained in section 4.4, this is done by changing the location of the sensors in bifacial_radiance that are normally placed on the front of the modules. The heatmap shown here show both the ground irradiation at full sun and the shadows created by the structure and the modules. The virtual sensors used in the model to create this heatmap cover the entire area of the ground inside the greenhouse (the white section of ground in Figure 11). Each square corresponds to a one square meter section of the ground. For Figure 12 the maximum daily irradiance is 790.3 W m^{-2} and the minimum צ is 33.5 W m^{-2}. These maximum and minimum irradiance levels are used to normalize heatmaps made for the hours from 5AM to 8PM. These additional figures are included in Appendix A.

5.2 Complete greenhouse and no modules

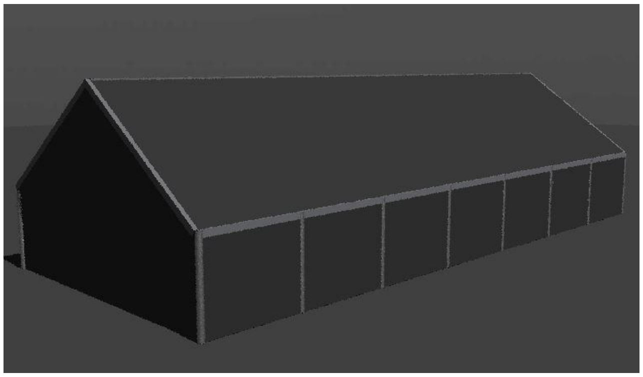


Figure 13 – Low resolution render of complete greenhouse with SG80 cover and no PV modules

The next model created is one of only a greenhouse, complete with diffusive cover. Two types of covers are tested, one based on a diffusive glass material called SG80 [33] and another based somewhat loosely on the diffusive properties of a standard greenhouse plastic covering. The main purpose of this model is to create a baseline for comparison between the ground irradiance of a normal greenhouse and the models with modules. Also, comparing between the heatmaps of the previous model (Figure 12) and this model (Figure 15) will show the reduction in irradiation due to only the greenhouse's diffusive cover. Since the focus of this paper is agri-PV greenhouses, this comparison is important. The assumption is that any crops being grown in a greenhouse will already be subjected to a reduction in irradiance caused by the diffusive cover. It is important to note that the individual light saturation curves of greenhouse crops are influenced by the greenhouse microclimate (temperature, humidity, CO_2 content, etc.) and amount of diffusive light in a greenhouse [42]. However, since the assumption is also that the greenhouse owner will have already created the most optimal greenhouse microclimate for their selected crop, one of the main factors that will have an affect between a greenhouse with PV modules and a normal greenhouse is a reduction in

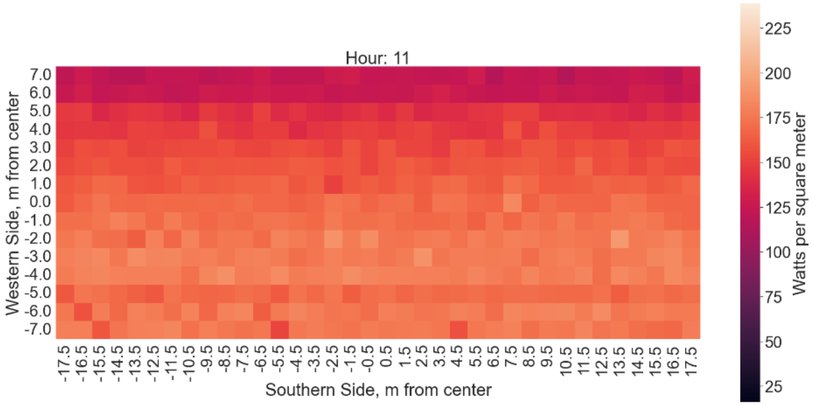


Figure 14 - Heatmap of the floor of the same greenhouse model as the one seen in Figure 13, but with a diffusive cover material based on white greenhouse plastic - Irradiance range: 19.6 to 238.6 W m^{-2}

irradiance reaching the crops. Another assumption made for the sake of simplicity is that the modelled greenhouses will be empty since any structures or crops growing inside the greenhouse will have the same overall effect on the amount of PAR reaching the self-shaded crops. Also, as discussed in section 2.2, several studies have shown that the relationship between solar irradiance (in W m^{-2}) and PAR (in µmol m^{-2} s^{-1}) is a somewhat straightforward multiplication of the solar irradiance by a factor of 4.56 µmol J^{-1} with little to no error [13]. For this reason, as stated earlier in section 4.2, the concepts of PAR and irradiance are treated as interchangeable when concerning the amount of irradiance measured inside the greenhouse models.

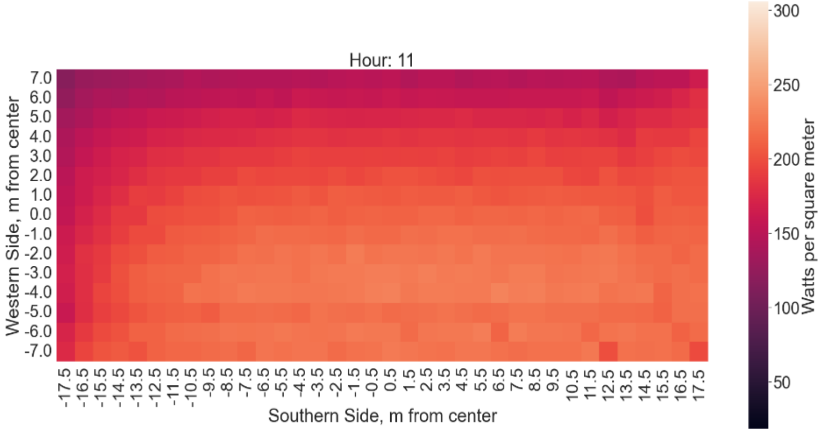


Figure 15 - Heatmap of Figure 13 ground irradiance at 11 a.m. on 2021-06-17 with SG80 diffusive greenhouse glass – Irradiance range: 18.2 to 304.8 W m^{-2}

One of the interesting aspects of the heatmaps generated from Figure 13's model is that it shows how the amount of light reaching different parts of the greenhouse floor is uneven. The northern side of the greenhouse floor receives less light than the southern. This makes sense since the sun in the southern sky, meaning the incident angle of the sun casts a larger amount of light on the front side of the greenhouse. In addition to this, the diffusive cover causes the incoming solar irradiation to scatter at many different angles as it passes into the greenhouse. This leads to a lower concentration of light at the side of the greenhouse further from the sun [32].

In addition to the uneven nature of the light, small shadows cast by the southern support pillars of the greenhouse can be seen at the southern side of the greenhouse, while the shadows of the main supporting beams are no longer visible. This is due to the angle of the incoming light and the diffusive greenhouse cover. As mentioned earlier, two different covering materials are considered as well. The first is a glass-like material referred to as SG80 and modelled in Radiance [33]. The second is a material based on a normal greenhouse plastic as described earlier in Table 3. The SG80 material has an overall higher transmittance of light, with a higher transmittance factor, but higher haze factor as well. This means that a higher amount of direct irradiation passes through this SG80 glass-based material, even though the higher haze factor means more of that light is diffused than the plastic material. This can be a good or bad aspect of this material, depending on the crop being grown inside the greenhouse. In the case of tomatoes, for example, it has been shown that tomatoes grown in higher levels of diffuse light tend to have an increase in flowering and fruit production.

5.3 Complete agri-PV greenhouse with modules at 5cm above the greenhouse roof, same tilt as roof

Now comes the first agri-PV greenhouse model (Figure 16). This is the original design considered for an agri-PV greenhouse for two main reasons: simplicity of design and the potential for a low-cost installation of modules, similar to the cost for a roof-mounted PV system. The mounts required for this design would be minimal and straightforward since there is only 5cm between the roof and the modules. The geometry of the array is also simple and made up of 7 rows of 8 modules. While it is outside the scope of this paper to create a detailed analysis of the costs for installing a system like this, it can be assumed to some degree that the more complex the topography of a roof mounted system, such as those found in Figure 19 and Figure 22 below, the higher the cost will be for installation. This increase in cost may not be significant enough to prevent those designs from being viable, but it seemed pertinent when considering the first agri-PV greenhouse design to go with something simple and seemingly cost effective.

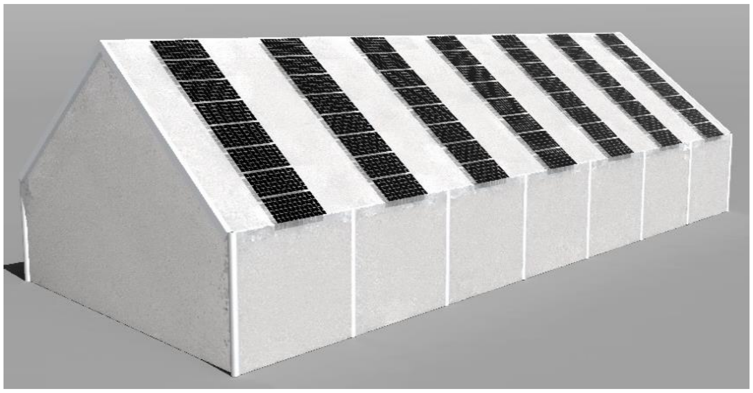


Figure 16 - High resolution render of a complete agri-PV greenhouse with modules

For this model and all other models with modules, the amount of irradiance received by the bifacial modules is also simulated. This allows for a comparison between the amount of irradiation received by the front and rear sides of the module, an estimation of the gain from the energy produced by the rear side of the bifacial modules, and a comparison between energy generated by the other array topologies simulated.

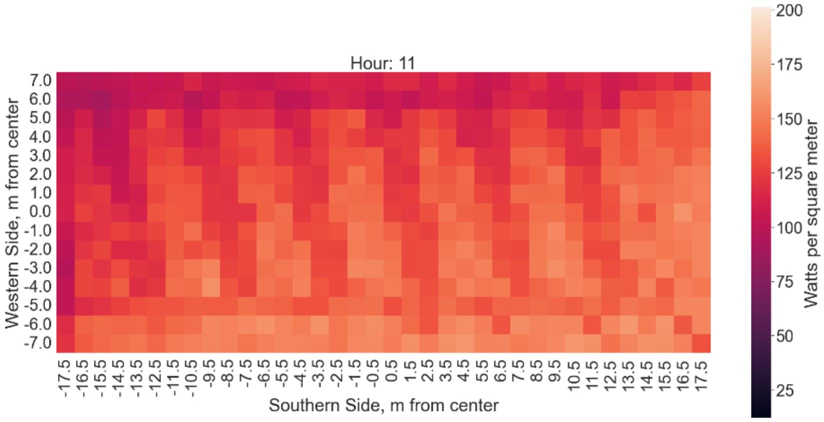


Figure 17 - Heatmap of Figure 16 ground irradiance - plastic greenhouse cover material - Irradiance range: 12.1 to 201.2 W m^{-2}

For the heatmaps in Figure 18 and Figure 17, it can be seen that there is a reduction in the amount of irradiance entering the greenhouse when compared to the heatmaps in Figure 15 and Figure 14. The amount of irradiance reduced is dependent on the simulated cover material type. In the case of the SG80 covering, the maximum ground irradiance is reduced by 50 W m^{-2} in the highest lit areas, with a further reduction in the areas shaded by the modules. However, the amount of irradiance reduction in the shaded areas is much lower than the amount when there is no diffusive cover, as seen in Figure 12. There is a stark contrast between the lit and shaded areas of the ground when there is only a vaulted module and no diffusive cover. The diffusion created by the cover allows for a more even light profile throughout the day, which is something many crops benefit from. This is because the light compensation points of these crops aren't reached during the day, which is possible when the crop is in an area of prolonged shade for an extended period. In the example of tomatoes, they benefit from some amount of shading, since it lowers the ambient and lead temperature, and increases the light use efficiency.

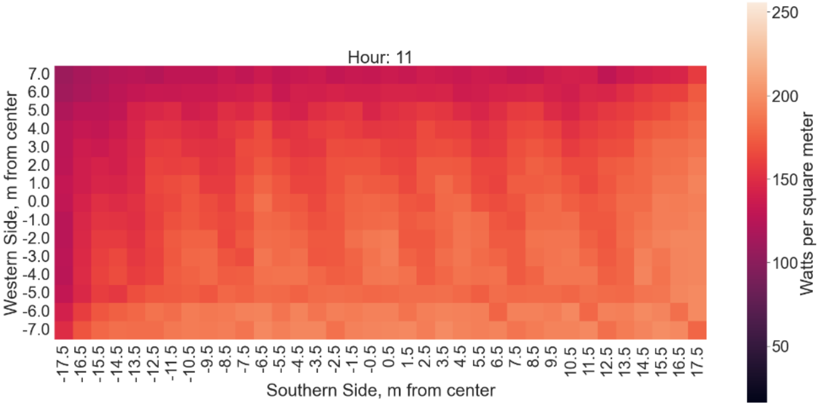


Figure 18 - Heatmap of the ground irradiance for the model seen in Figure 16 with SG80 diffusive glass - Irradiance range: 16.2 to 255.6 W m^{-2}

In the case of the plastic greenhouse covering, there is a further reduction of around 50 W m^{-2} in maximum ground irradiance in comparison to the SG80 greenhouse, as shown in the heatmap in Figure 17. However, the difference in the reduction in maximum irradiance between the normal greenhouse and the agri-PV greenhouse with the plastic greenhouse covering is only around 40 W m^{-2}. It is interesting to note, however, that there is a higher level of contrast between the shaded and lit areas in the plastic covered greenhouse. This is likely due to the lower haze factor of the material, meaning more direct light transmits through the plastic material than the SG80 material.

5.4 Agri-PV greenhouse with modules at a 45-degree tilt to favor summer solar altitude angle

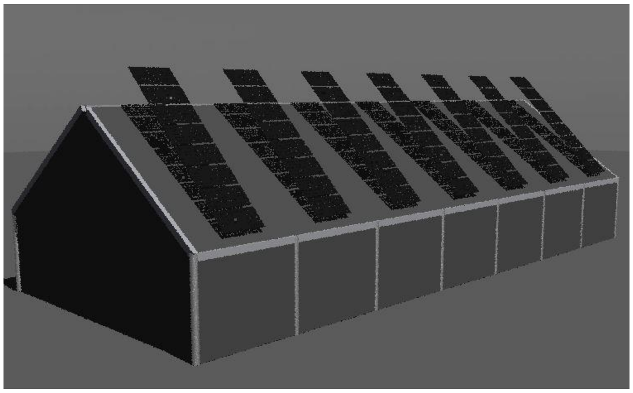


Figure 19 - Low resolution render of a complete agri-PV greenhouse with modules at a 45-degree tilt

For this model, the modules are given a 45-degree tilt. This is done for several reasons. First, at the location in Kjeller, this tilt angle is one that favors the summer months. Since Norway has little to no light in the winter, a summertime-oriented tilt makes sense. The second reason is that the extra space between the roof and modules allows for the rear sides of the bifacial modules to receive more light, increasing overall energy yield.

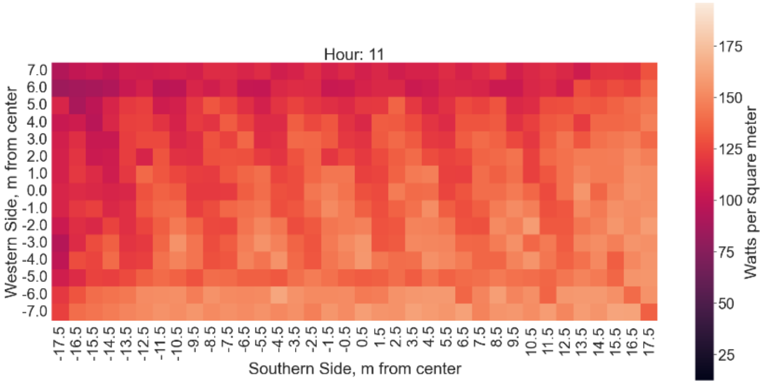


Figure 20 - Heatmap of the ground irradiance of Figure 19 - plastic greenhouse cover - Irradiance range: 12.5 to 195.8 W m^{-2}

Another reason for this module topology is that it allows for a vent to be placed underneath it. This may not be entirely necessary since a vent could be placed on the opposite side of the greenhouse not covered by modules, but the possibility is there if the design of the greenhouse requires it or already has a vent installed on that side before the greenhouse is transformed into an agri-PV greenhouse. For the plastic greenhouse cover, while there is still a reduction in overall irradiance when compared to the previous model's heatmap (Figure 17), the reduction is very minimal. The light/shade contrast of Figure 18 and Figure 20 heatmaps are nearly identical, indicating that the reduction in irradiance between the

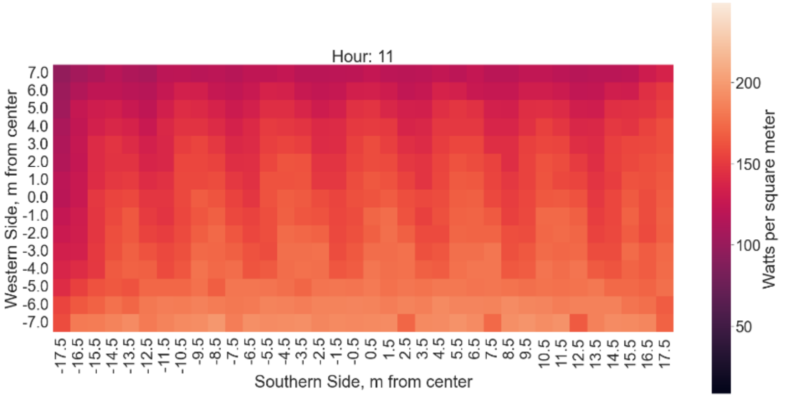


Figure 21- Heatmap of ground irradiance for Figure 19 - SG80 cover – Irradiance range: 8.4 to 249.0 W m^{-2}

two is uniform. This is possibly due to the optimized tilt angle of the panels causing the PV cells to absorb more of the solar photon flux than the 35-degree tilted panels, which makes sense since the cells are closer to being perpendicular to the incoming solar rays at a 45-degree tilt in June. Also, it's interesting to note the further reduction in contrasting shadows with the 45-degree tilt, which is beneficial for some crops, including tomatoes.

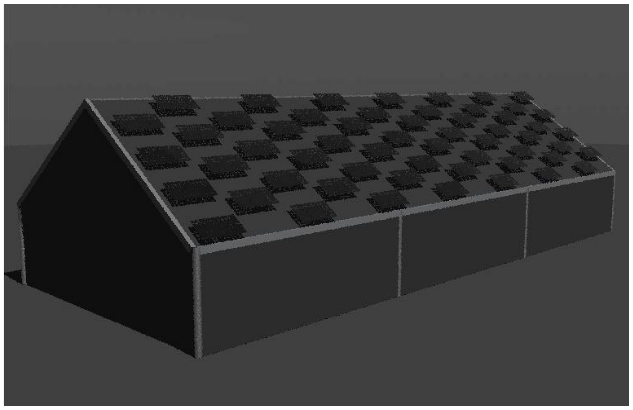


Figure 22 - Low resolution model of an agri-PV greenhouse with modules in a checkerboard pattern

5.5 Agri-PV greenhouse with modules in checker pattern

As with the previous models, this model is created in order to observe if putting the modules in a checker pattern had a major influence on both the average ground irradiance and the light/shade contrast pattern at ground level in the greenhouse. The idea here is that if a crop is extra sensitive to changes in shading throughout the day, a more even shading pattern might be beneficial. It can be seen in the Figure 24 heatmap with the SG80 material that the contrast between light and shade is very minimal and that there is almost no reduction in overall ground irradiance when compared to the previous 35-degree tilt array. However, the model with the same checker PV array topology and the plastic greenhouse covering shows a much higher level of contrasting shadows. The lighting at ground level for this PV array topology is still the most even of the 3 topologies simulated, but it appears that the overall reduction in maximum ground irradiance is the highest.

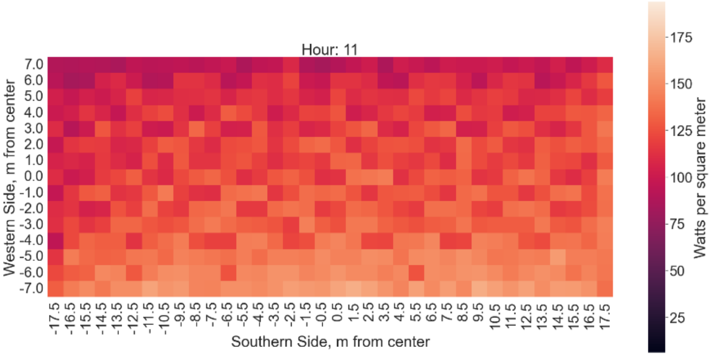


Figure 23 – Ground irradiance heatmap with plastic cover and checker pattern array – Irradiance range: 12.18 – 193.71 W m^{-2}

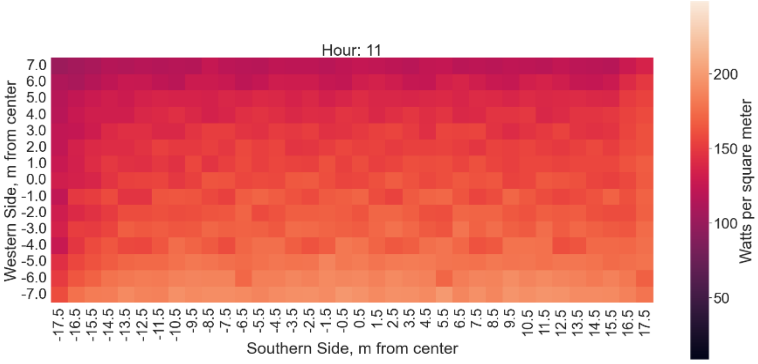
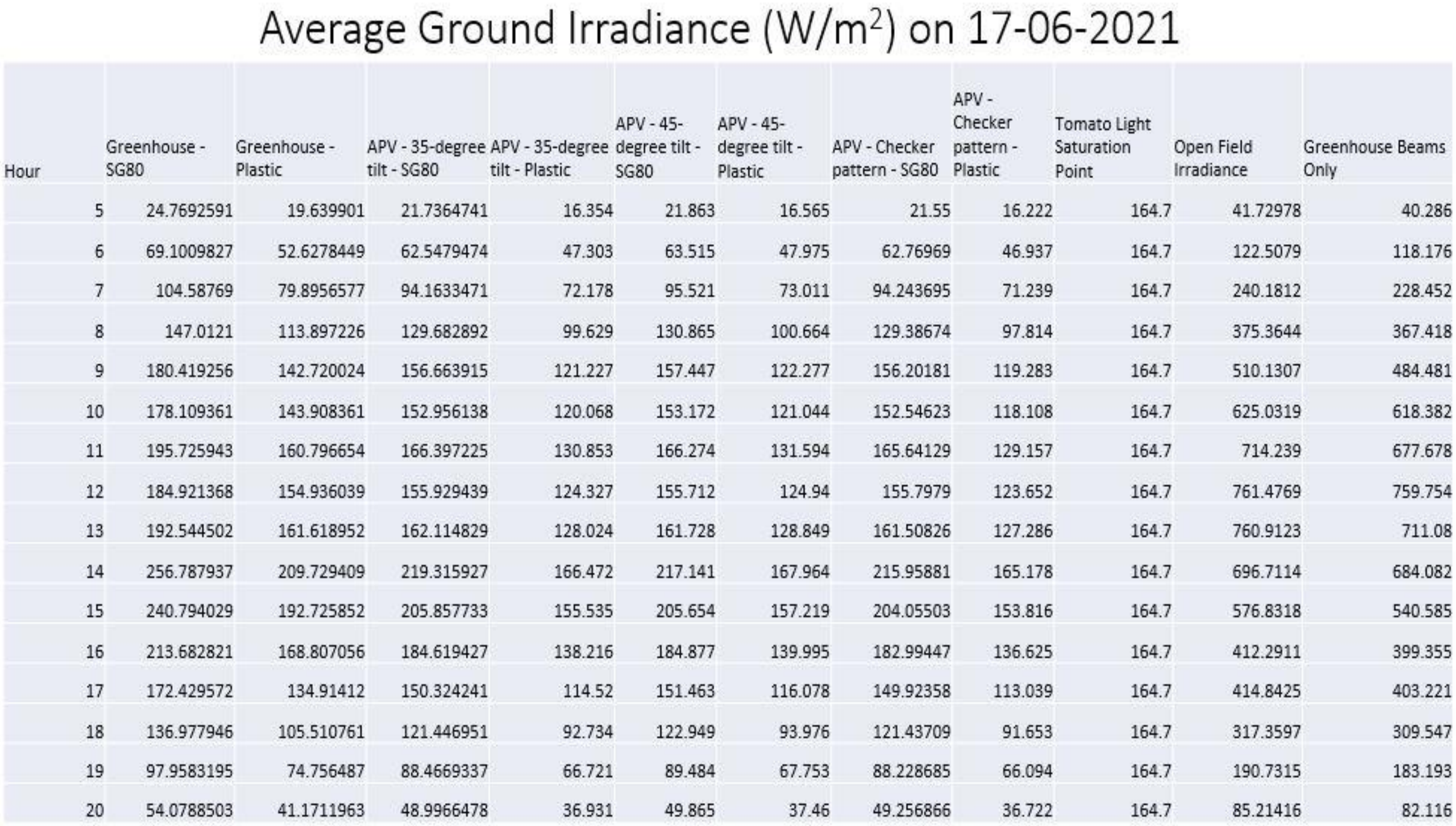


Figure 24 – Ground irradiance heatmap of Figure 22 with SG80 cover and checker pattern array – Irradiance range: 8.6 to 248.6 W m^{-2}



Average Ground Irradiance (W/m^2) on 17-06-2021

Hour	Greenhouse - SG80	Greenhouse - Plastic	APV - 35-degree tilt - SG80	APV - 35-degree tilt - Plastic	APV - 45-degree tilt - SG80	APV - 45-degree tilt - Plastic	APV - Checker pattern - SG80	APV - Checker pattern - Plastic	Tomato Light Saturation Point	Open Field Irradiance	Greenhouse Beams Only
5	24.7692591	19.639901	21.7364741	16.354	21.863	16.565	21.55	16.222	164.7	41.72978	40.286
6	69.1009827	52.6278449	62.5479474	47.303	63.515	47.975	62.76969	46.937	164.7	122.5079	118.176
7	104.58769	79.8956577	94.1633471	72.178	95.521	73.011	94.243695	71.239	164.7	240.1812	228.452
8	147.0121	113.897226	129.682892	99.629	130.865	100.664	129.38674	97.814	164.7	375.3644	367.418
9	180.419256	142.720024	156.663915	121.227	157.447	122.277	156.20181	119.283	164.7	510.1307	484.481
10	178.109361	143.908361	152.956138	120.068	153.172	121.044	152.54623	118.108	164.7	625.0319	618.382
11	195.725943	160.796654	166.397225	130.853	166.274	131.594	165.64129	129.157	164.7	714.239	677.678
12	184.921368	154.936039	155.929439	124.327	155.712	124.94	155.7979	123.652	164.7	761.4769	759.754
13	192.544502	161.618952	162.114829	128.024	161.728	128.849	161.50826	127.286	164.7	760.9123	711.08
14	256.787937	209.729409	219.315927	166.472	217.141	167.964	215.95881	165.178	164.7	696.7114	684.082
15	240.794029	192.725852	205.857733	155.535	205.654	157.219	204.05503	153.816	164.7	576.8318	540.585
16	213.682821	168.807056	184.619427	138.216	184.877	139.995	182.99447	136.625	164.7	412.2911	399.355
17	172.429572	134.91412	150.324241	114.52	151.463	116.078	149.92358	113.039	164.7	414.8425	403.221
18	136.977946	105.510761	121.446951	92.734	122.949	93.976	121.43709	91.653	164.7	317.3597	309.547
19	97.9583195	74.756487	88.4669337	66.721	89.484	67.753	88.228685	66.094	164.7	190.7315	183.193
20	54.0788503	41.1711963	48.9966478	36.931	49.865	37.46	49.256866	36.722	164.7	85.21416	82.116

Table 6 – List of average ground irradiance measurements from all daylight hours on 17-06-2021, in W m^{-2}

5.6 Comparison of average ground irradiance between models with example light saturation point (greenhouse grown tomato)

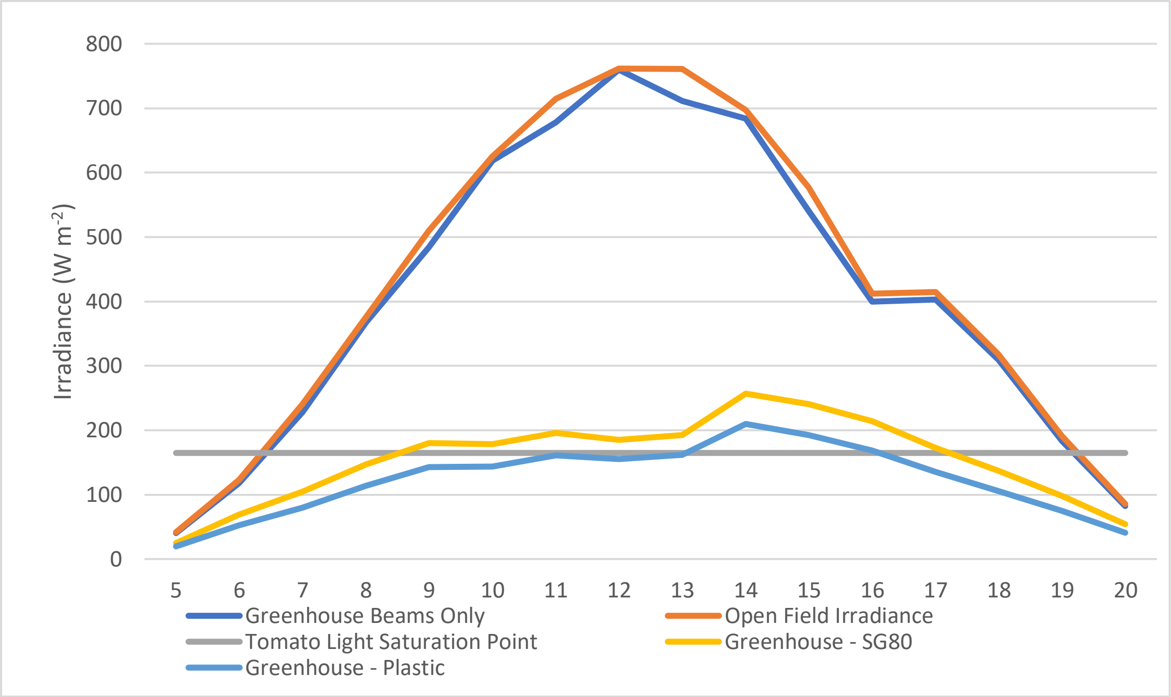


Figure 25 – The Irradiance levels of a model of an open field (no objects), a model with the greenhouse structural beams only, and the complete greenhouse models (no PV modules), all on 17-06-2021, compared to the light saturation point of tomatoes (164.7 W m^{-2})

In order to accurately access the amount of ground irradiance being reduced by the greenhouse covers, it is necessary to create a baseline measurement of the ground irradiance when there is only an open field with no simulated objects. This allows not only for a comparison to the ground irradiance of the other models, but also to real-world data such as satellite-based irradiance data provided by Solcast [22]. As discussed in section 3.1.3, it should be noted that bifacial_radiance sources their DHI, DNI and GHI values from EPW (EnergyPlus Weather) files that use TMY (Typical Meteorological Year) data instead of real-world historical data measurements for each year simulated. This means that the irradiance levels simulated may be close to, but not exactly, the actual measured irradiance values found in satellite-based irradiance data from sources like SolCast. Still, the point of the TMY is to create a time-series that based on multiyear historical data and presents annual average values of weather data that

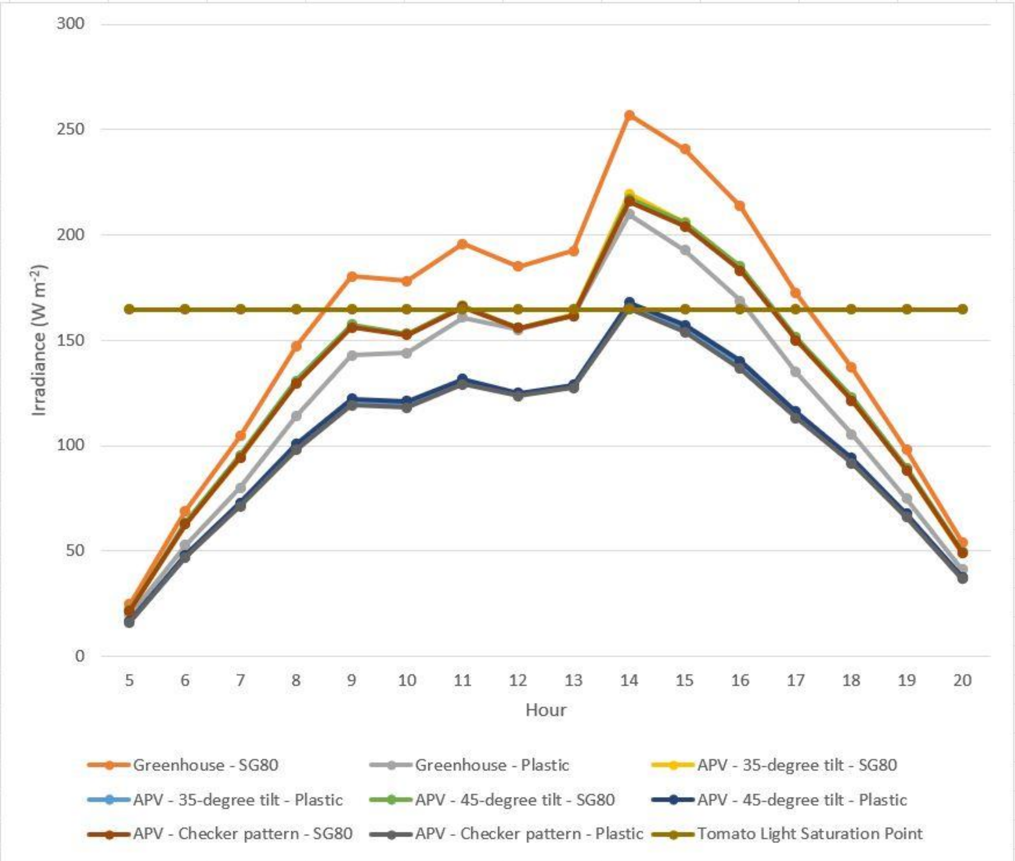


Figure 26 - Comparison of all simulated greenhouse models, standard and agri-PV, with the LSP of tomatoes

are consistent with long term annual averages so that researchers can create energy system simulations without the need for processing many years of meteorological data [39].

Figure 26 shows a comparison of the average ground irradiance of all simulated models for the hours of 5am to 8pm on June 17th, 2021. Table 6 shows the data used to create this figure, including the line that signifies the tomato light saturation point. An interesting observation here is that the average ground irradiance values of all 3 of the plastic cover greenhouses are nearly identical for the models with PV modules installed. The same can be said for the irradiance values of the SG80 models. This makes some sense since the total PV-cell area of all 3 PV-array topologies is the same (98.1 m^2). However, it is absolutely clear that the SG80 material allows more light to enter the greenhouse and is likely to benefit the tomatoes since there are more hours when the light is at or above the LSP. It could also be argued that the higher light levels allowed by the SG80 material could result in higher temperatures within the greenhouse and that this could have a negative effect on the plant growth. This is another reason why the decision is made for this study to present only the percentage of PAR lost when compared to a plant's LSP instead of a full crop yield.

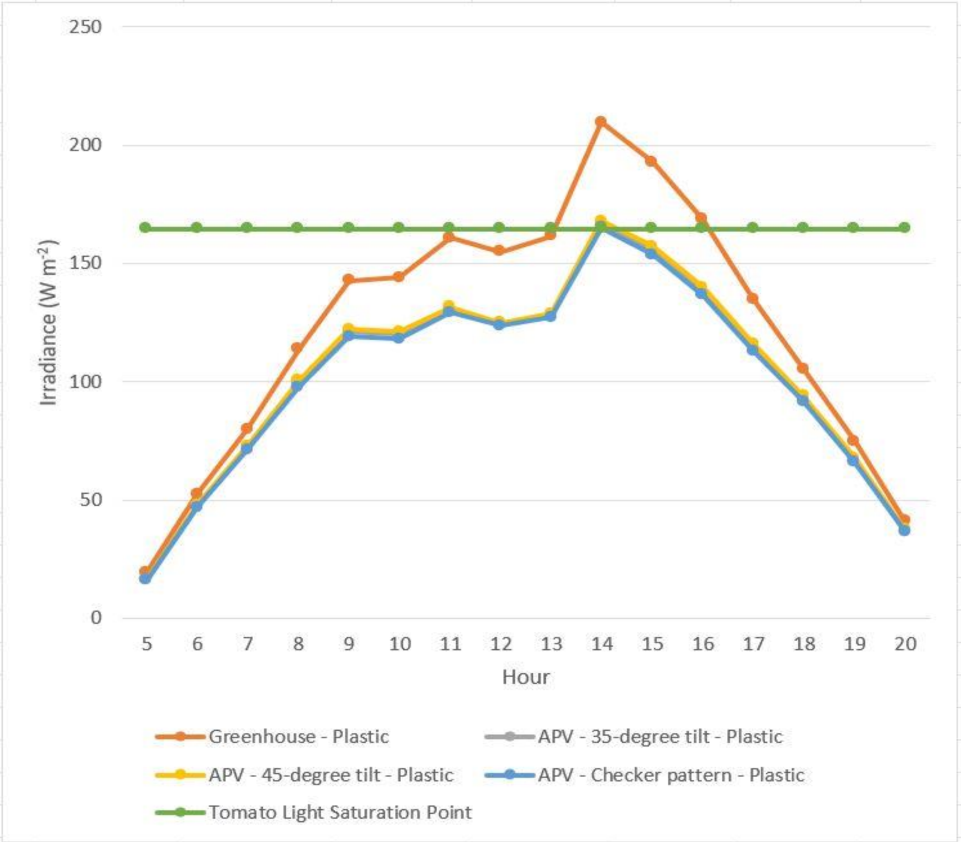
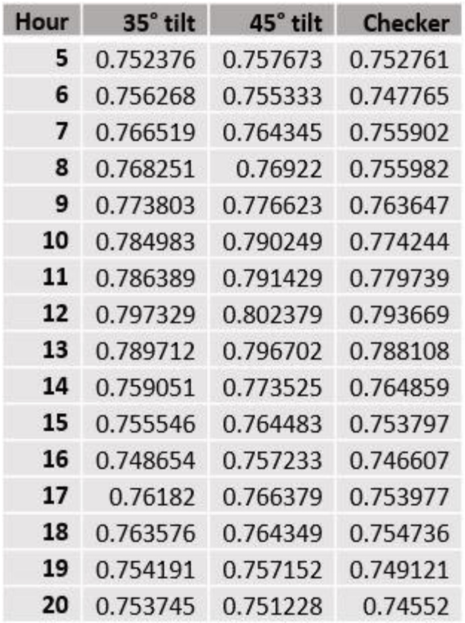


Figure 27 – Average ground irradiance (W m^{-2}) on 17-06-2021 – all models with plastic greenhouse cover

Table 7 – Table of the percentage of average ground irradiance measured on 17-06-21 in the plastic covered greenhouses in comparison to the SG80 covered greenhouses for each PV-array topology



Hour	35° tilt	45° tilt	Checker
5	0.752376	0.757673	0.752761
6	0.756268	0.755333	0.747765
7	0.766519	0.764345	0.755902
8	0.768251	0.76922	0.755982
9	0.773803	0.776623	0.763647
10	0.784983	0.790249	0.774244
11	0.786389	0.791429	0.779739
12	0.797329	0.802379	0.793669
13	0.789712	0.796702	0.788108
14	0.759051	0.773525	0.764859
15	0.755546	0.764483	0.753797
16	0.748654	0.757233	0.746607
17	0.76182	0.766379	0.753977
18	0.763576	0.764349	0.754736
19	0.754191	0.757152	0.749121
20	0.753745	0.751228	0.74552

Figure 27 and Figure 28 show the average ground irradiance on June 17th, 2021 for the plastic covered greenhouse models and the SG80 covered greenhouse models, respectively. The main difference to note here is that the SG80 models have a higher overall average ground irradiance throughout the day, with more hours above the LSP. It can also be seen that the curves for all agri-PV greenhouses are nearly identical, the only difference being that the plastic covering lets in a consistently lower percentage of light at every hour simulated. This can also be seen in Figure 26. It is for this reason that the final tomato season hourly ground irradiance experiment considers only the SG80 greenhouse cover since it seems on average the plastic greenhouse irradiance values are always a set percentage below the results from the SG80 material, and therefore could be calculated from the SG80 results with little to no error. By dividing the hourly average irradiance

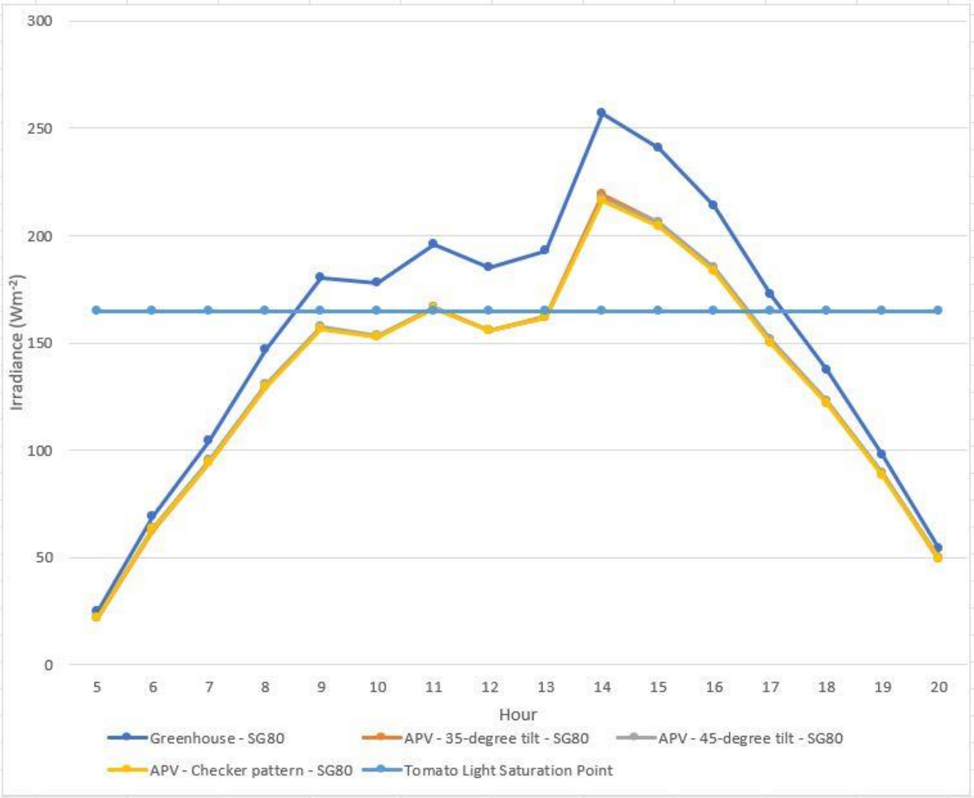


Figure 28 – Average ground irradiance (W m^{-2}) on 17-06-2021 – all models with SG80 greenhouse cover

of the plastic greenhouse cover models by the average hourly irradiance of the SG80 models of the same type, it is discovered that the irradiance values of the plastic models are on average 76.6% (reduced by 23.4%) the values of the SG80 irradiance values, with the range in the percentages of the difference in average irradiance being between 74.5 to 80.2% across all models, as can be seen in Table 7. It could be that the average should be weighted to account for the way the percentage difference increases at hours of high irradiance, or that the percentage change could be averaged for each hour. However, it still seems safe to assume that any simulations of ground irradiance values for models with the SG80 material can simply have their irradiance values reduced by approximately 23.4% to obtain the approximate values of the plastic greenhouse cover models. Therefore, to save processing time in the hourly simulations across the entire tomato growing season of May to October, only the SG80 covered model is considered.

5.7 Comparison of average hourly ground irradiance between a normal greenhouse and an agri-PV greenhouse on all of the days of a yearly tomato growing season

For this experiment, all hours considered to be part of the tomato growing season are simulated and then bound between the light saturation point (LSP) and light concentration point (LCP) of a greenhouse grown tomato crop. The LSP of the tomatoes used for this simulation is 754.5 µmol m^{-2} s^{-1} PAR, which is equivalent to approximately 167.4 W m^{-2} [16], [27]. The percentage of PAR reduced by shading of the modules in the agri-PV greenhouse is calculated and plotted. Figure 29 shows this comparison for a single day in June, while Figure 30 shows the complete ranges of hours simulated. Figure 32 shows the percentage of PAR reduced that is less that the LSP and greater than the LCP. As discussed earlier, it's important to consider these values instead of total irradiance reduction since irradiance levels above a plant's LSP do not result in an increase in photosynthesis. Therefore Figure 31 shows a total of the monthly reduction in ground insolation with the consideration of LSP and LCP. Essentially, it is the monthly sum of the reductions in PAR shown below in Figure 29 and Figure 32, sourced from the data used to create Figure 30. Note that for the months simulated, the percentage of PAR reduced increases as the months get closer to winter. This is possibly due to the overall decrease in PAR available below the LSP as the months progress, meaning that shadowing has a greater affect during low light months once LSP is considered.

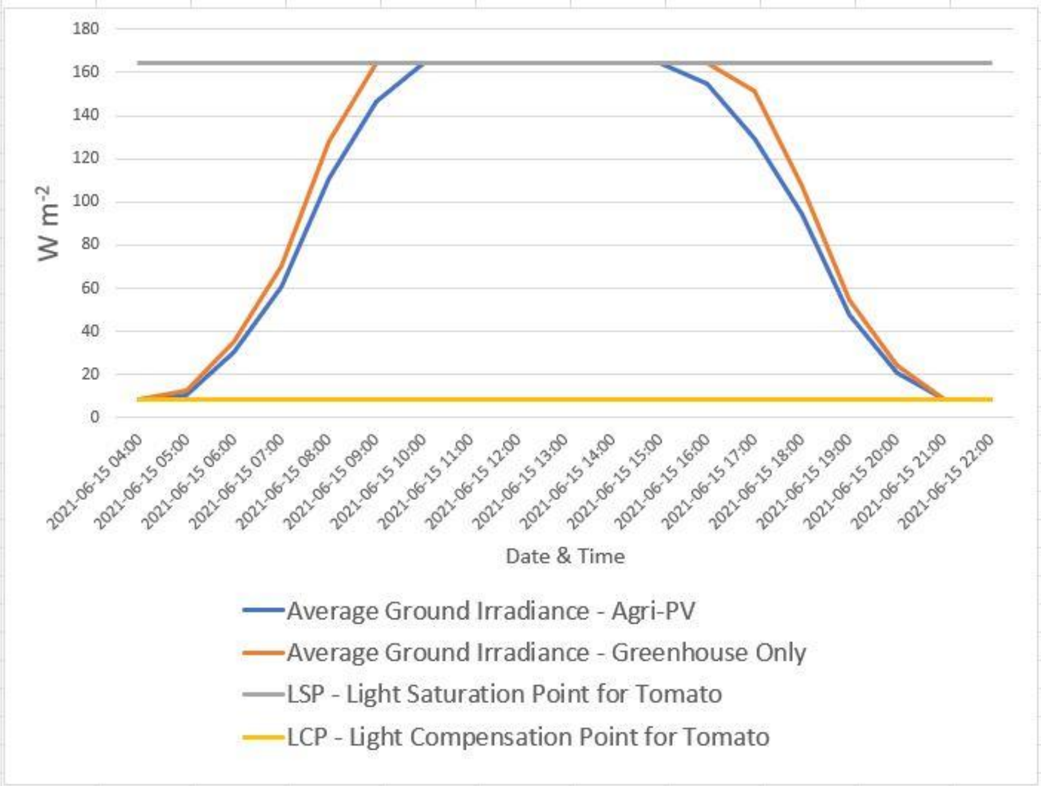


Figure 29 – Example of a single day (15-06-21) taken from the complete average ground irradiance comparison (Figure 30) with the ground irradiance values bound between the LCP and LSP for tomatoes, as described in section 4.4

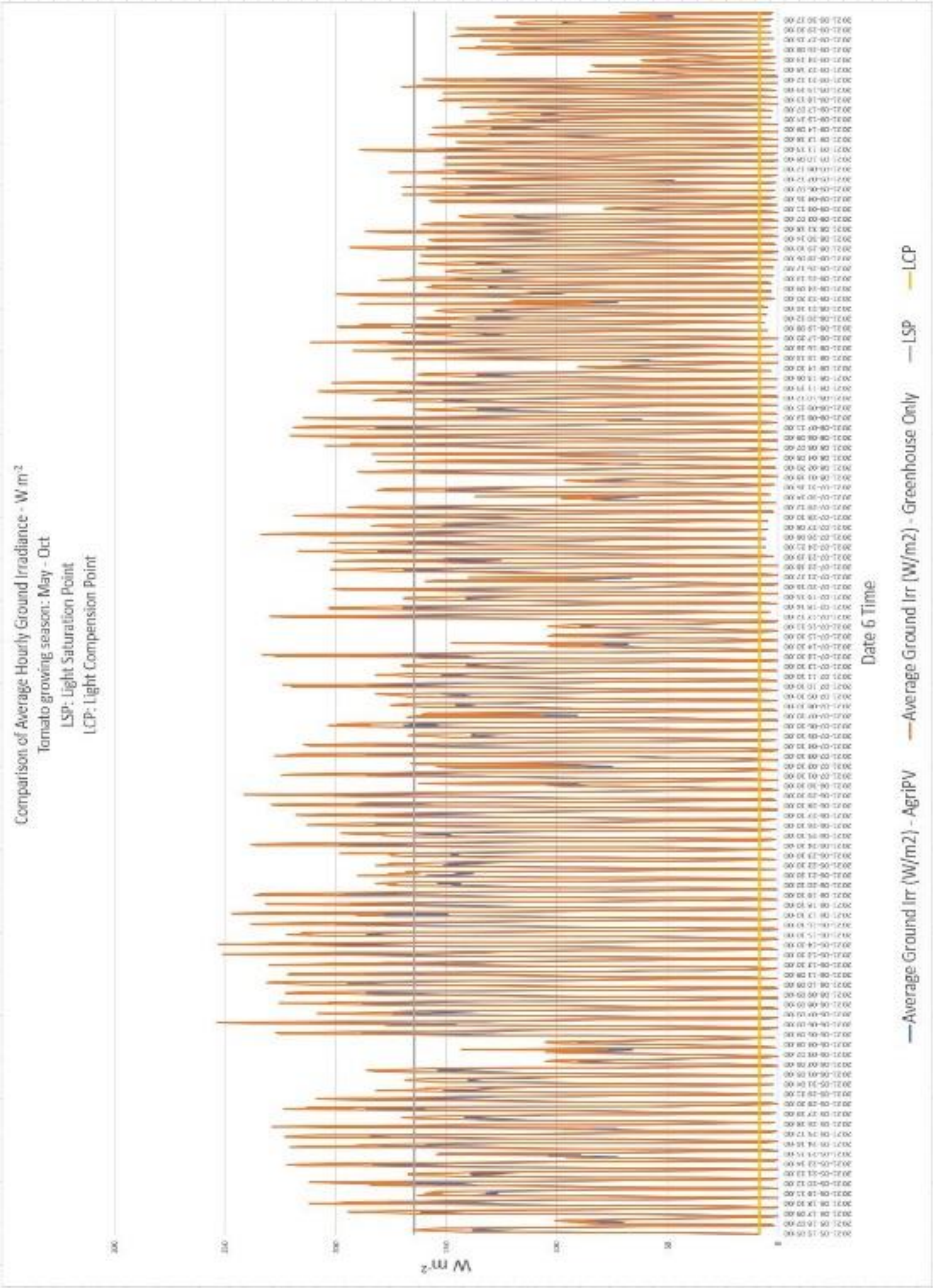


Figure 30 – All hours simulated (daylight hours only) and compared to the LSP and LCP of tomatoes

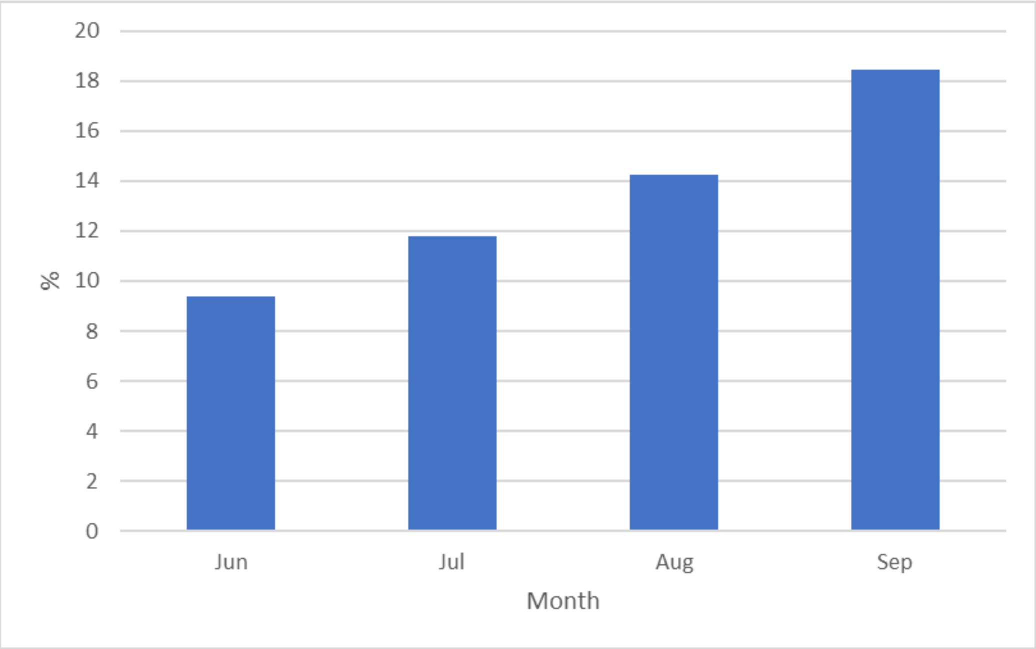


Figure 31 – Average percentage of monthly insolation reduced when ground irradiation values are upper and lower bound by LSP and LCP

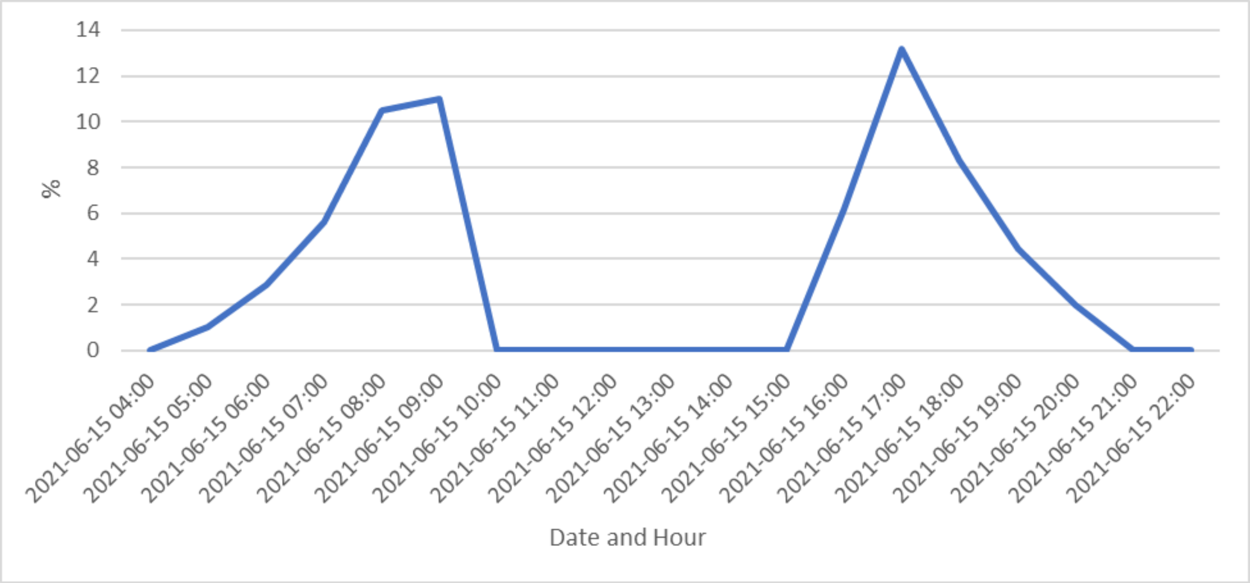


Figure 32 – Percentage of PAR reduced (below tomato LSP and above LCP) in agri-PV greenhouse on 15-06-21

5.8 Comparison of estimated yearly energy output between models with modules

Table 8 – List of estimated monthly energy yield's (in kWh) as calculated by equation (1) in section 4.1

Month		SG80 -35 tilt	SG80 - 45 tilt	SG80 - Checker	Plastic - 35 tilt	Plastic - 45 tilt	Plastic - Checker
Jan		100.363	114.558	101.802	101.616	115.156	89.215
Feb		379.864	417.852	379.183	388.59	421.956	335.366
Mar		988.336	1044.576	976.03	992.555	1080.792	1005.093
Apr		2058.35	2111.773	2040.868	2082.224	2164.076	2076.411
May		3217.603	3241.307	3194.053	3259.847	3326.701	3258.815
Jun		3270.091	3265.489	3251.459	3307.181	3356.195	3285.974
Jul		3275.583	3247.999	3234.012	3279.682	3318.718	3282.518
Aug		2665.823	2693.205	2628.471	2667.046	2766.158	2662.811
Sep		1610.552	1679.732	1598.644	1631.406	1723.705	1628.815
Oct		878.149	940.237	872.141	888.072	965.875	887.224
Nov		259.278	288.012	259.413	265.853	296.129	266.701
Dec		57.714	64.653	57.649	58.705	66.933	50.948
Total (MWh)		18.762	19.109	18.594	18.923	19.602	18.83

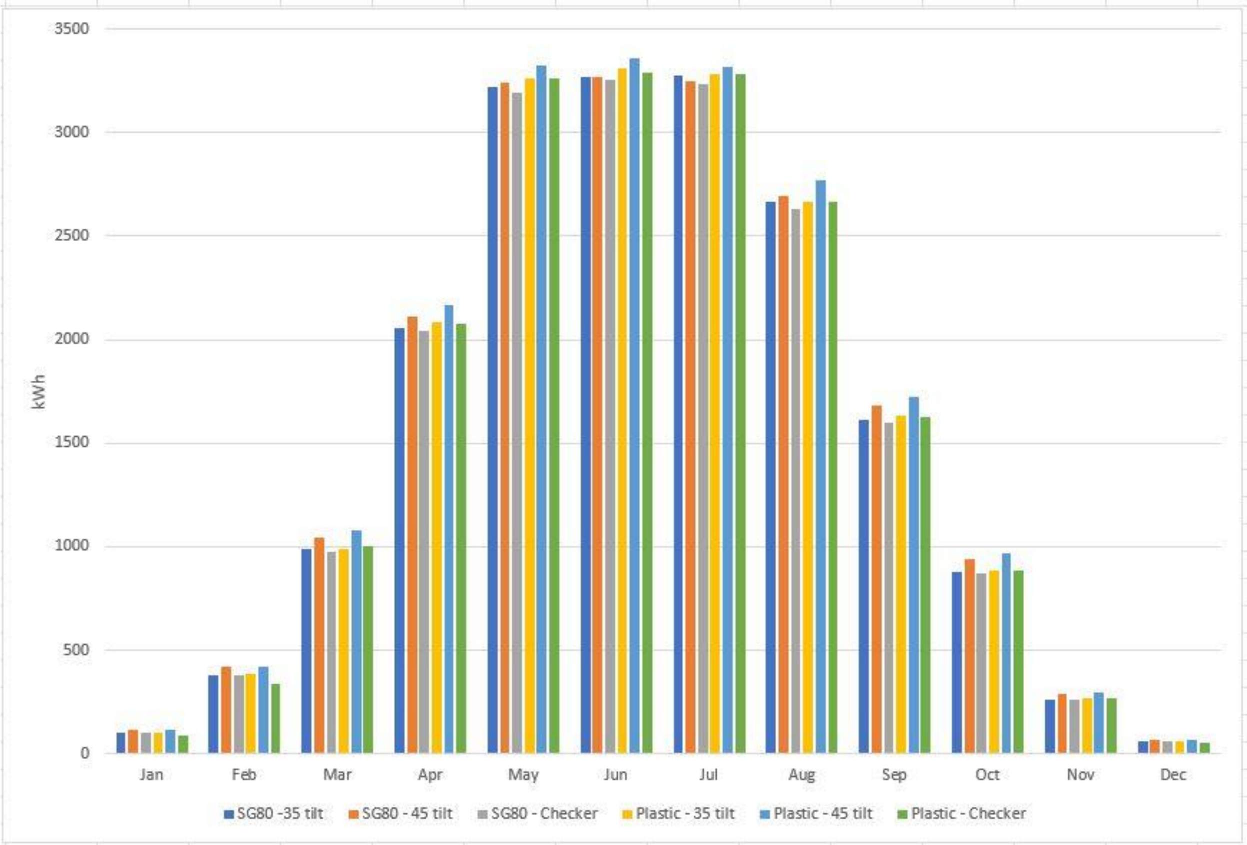


Figure 33 – Graph of monthly energy yields by model based on the data found in Table 8

For the final comparison, the monthly energy yield for each model is calculated using the process discussed in section 4.1 and can be seen in Table 8 and Figure 33. Table 8 shows both the estimated monthly energy yields (in kWh) as well as the yearly total for each model (in MWh). The models with the 45° tilt performed the best, while the models with the checker-styled PV array performed the worst of all three topologies in both plastic and SG80 material model categories. What's more interesting is that the models with the simulated plastic greenhouse cover outperformed their respective SG80 models for all PV-array topologies. The greatest difference in yearly output is a little over 1 MWh between the 45° tilt plastic cover model and the checker styled SG80 cover model. However, upon reflection these results do make sense. The 45° tilt PV-arrays benefit not only from a higher amount of frontside irradiation during the summer, but the larger gap between them and the roof of the greenhouse means the backside cells will receive a larger amount of reflected light as well. Also, the higher production of the PV-array mounted on the plastic cover greenhouse makes sense too since this material allows less light to pass through, meaning a larger percentage of light is reflected back at the backside PV cells on the bifacial modules.

6. Discussion

6.1 Limitations and advantages of Radiance and bifacial_radiance

As the research for this thesis progressed, there were many insights into how bifacial_radiance could be improved, as well as what it exceled at. The ability to quickly define a PV-array's topology and create modules with cell specific parameters is extremely useful. Also, automatic sensor placement that considers the tilt of the modules is very nice to have. However, there were several features missing from bifacial radiance that made the simulations a bit more difficult to conduct. First was automatic placement of sensors on cell-level customized PV modules. Unfortunately, at the time of this paper, the way that sensors were placed on the modules was somewhat haphazard. The user can specify the number of sensors they wish to place, but these sensors will all be placed in essentially a straight line through the center of the PV-array. It is possible to specify that each sensor be placed in a direction of x, y, and z from the previous sensor, but this means that the sensor placement will always be in some form of a straight line through a 3-dimensional space. If the user wants to have a plane of sensors, a *for* loop that places many straight lines of sensors is required. This is how the sensor placement in Figure 9 was performed.

Another feature missing is any sort of 3D CAD implementation. Most engineers and architects prefer to use 3D modelling software for creating models. There are even programs that allow for CAD models to be imported and analyzed in Radiance, such as Rhino 3D [33]. What bifacial_radiance does that these programs do not is provide an easy way to quickly generate modules, and a way to generate cumulative skies based on hourly irradiance data. The integration of being able to import a 3D CAD model and having some way to automatically sense the location of the CAD model in the oct file so modules and sensors could be placed for analysis would be highly beneficial.

Finally, the advantages and limitations of Radiance itself. Many of these were already discussed in section 2.1 of the literature review of this paper. The inability to interact with the model and implement changes in real-time is somewhat frustrating. Another big issue not discussed in the literature review section is the fact that Radiance does not run on GPUs that are specifically engineered to handle ray tracing extremely well. A single presentation from 2011 on accelerating Radiance using OpenCL on a modern GPU was found, but little to nothing else [4]. This means that all ray tracing simulations performed by Radiance were done on the CPU instead, leading to extremely long processing times. Some of the simulations were abandoned after approximations of the processing time were calculated to be several months. For all the accuracy of Radiance as a physically-based rendering engine, this is one of its main drawbacks for use in daylighting simulations.

6.2 Accuracy and error in energy yields from simulations

When considering how best to calculate the energy yield of the PV modules in these agri-PV models, it was necessary to examine how bifacial_radiance performs its simulations. As explained briefly in section 3.1.3, there are two main functions for creating a sky in Radiance that bifacial_radiance uses: *gencumsky* and *gendaylit*. *Gendaylit* simulates a single hour based on sun location, DNI, DHI, and GHI data obtained from a localized EPW. This is what was used for both the hourly heatmaps and the hourly ground irradiance measurements for the tomato growing season, and the processing time can be extensive depending on the number of sensors and the requirements needed for sensor placement. *Gencumsky* creates what bifacial_radiance documentation refers to as a "cumulative sky" with all hourly irradiance data for each part of the sky being generated simultaneously and cumulatively. This is then used for the insolation simulations of the model and allows for these measurements to be simulated much quicker. This means that hourly insolation measurements (in Wh m^{-2}) are simply based on a single irradiance level (in W m^{-2}) measured at each hour with the assumption being that the irradiance is constant throughout each hour. While hourly resolutions for measurements like these are somewhat common in energy simulations and calculations, it should still be considered as it could result in some small amounts of error.

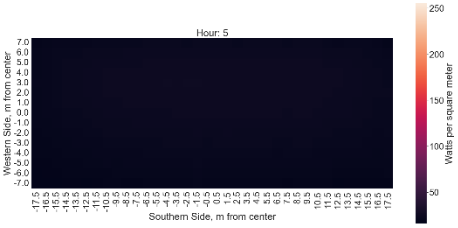
Additionally, the speed of *gencumsky* comes at a price: it is impossible to see the irradiance values of single hours or create an hourly time series from these scans. For this, *gendaylit* is necessary. The problem with *gendaylit* simulations is that creating long hourly time-series requires extensive amounts of processing time. Depending on the number of and required placement of the virtual sensors, simulations can take days, weeks and sometimes months to perform. For this reason, the number of sensors for the hourly simulations of the tomato growing season were reduced to 12 and placed in the X-like pattern seen in Figure 10. The reduction in sensors very likely created a higher level of error when calculating the average ground irradiance for those hours. Even so, those simulations each took almost a week to perform. It should be noted that bifacial_radiance always performs scans in two directions for all sensors in order to get both front- and rear-side irradiance measurements for a bifacial module, which can double the time necessary for performing simulations if only one of the directional scans (front or back) is need. The only reason that the simulations performed for the hourly tomato season simulations didn't take longer was because of a change to bifacial_radiance's backscan parameters that flipped the sensor direction to essentially make it a second frontscan. Therefore, there are two colors for the sensors in Figure 10.

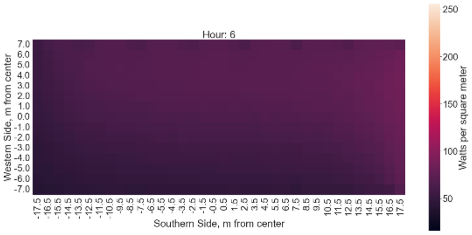
7. Conclusion

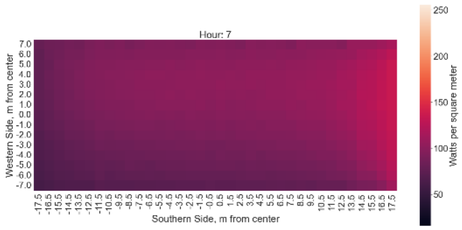
Ray tracing is a powerful tool when it comes to simulating both PV module shadowing and the energy output of agri-PV systems. This method of modelling has great potential for creating accurate and scalable models of both agri-PV systems and standard PV-systems. Not only is it a promising way to model ground irradiance and PV energy output, depending on the complexity of the model it could also be used to calculate PV mismatch caused by shadows falling on the PV modules themselves. Also, as shown in the results of this paper, the shadowing caused by the modules of an agri-PV system can also be calculated as reductions in PAR for use in comprehensive crop growth models. However, ray tracing as a concept is not limited to a single rendering engine. As discussed earlier, it’s been shown that video game engines are not far from being able to produce validated, physically-based daylighting simulations similar to Radiance. This is where ray tracing gets exciting and the potential for a new form of PV and agri-PV modelling software becomes ever more viable. It seems that with a bit more research and experimentation, interactive ray tracing modelling for PV is achievable.

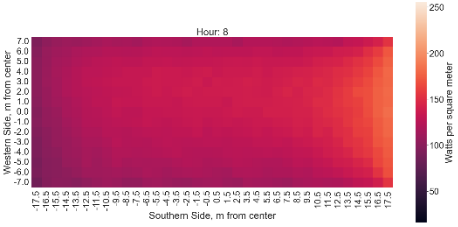
Appendix A

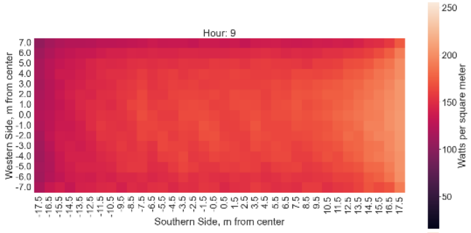
The following is an example of a single day's worth of heatmaps generated, specifically for the model with the SG80 cover material and the modules at a 35° tilt (Figure 16). At the time of this paper, over 850 of these heatmaps were generated for various models and parameters. Some were even turned into animated gifs in order to observe the effects of the shading as the hours progress. If desired, these heatmaps can be provided. Contact Taeo Laue at taeol[at]uio.no or tlaue[at]me.com.

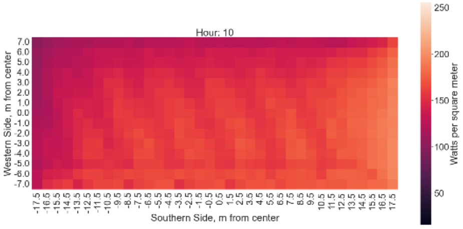


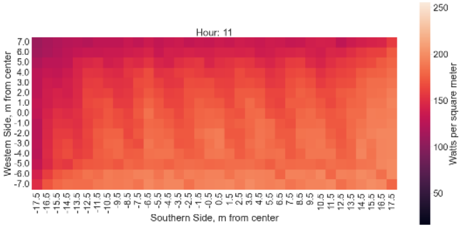


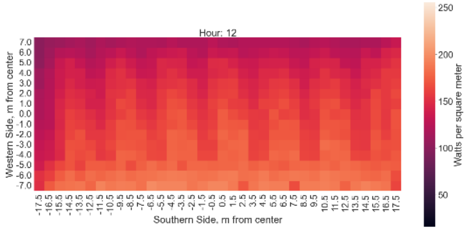


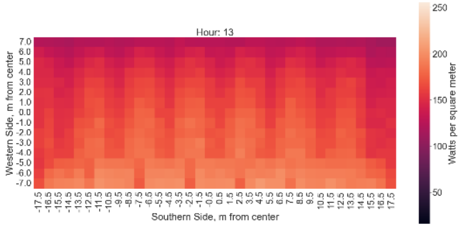


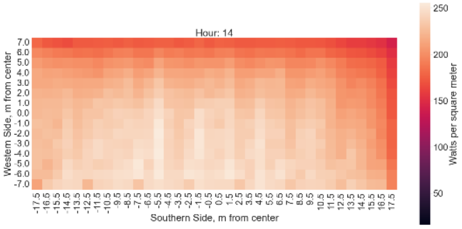


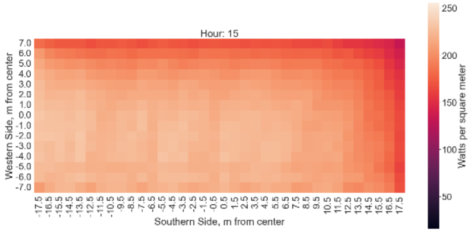


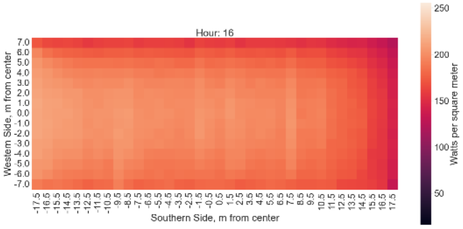


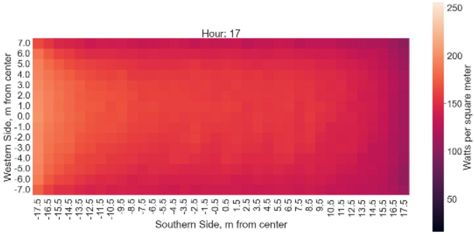


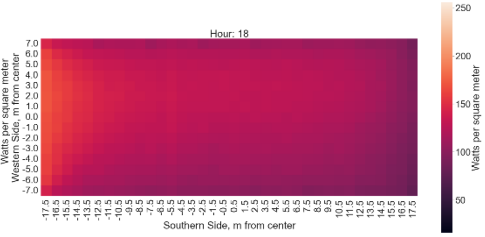


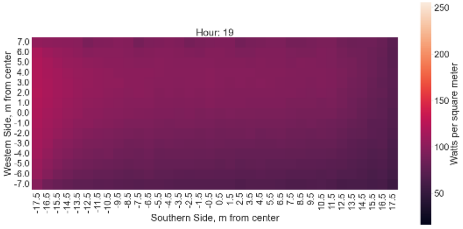


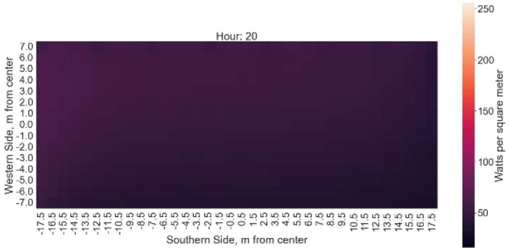












Appendix B

The code below is the main code used for all simulations. However, it should be noted that different simulations required different parameters or operations, and some of them required changes to the code itself. For example, all the hourly heatmaps were created by putting the code below into a giant *for* loop and having that loop run for the range of hours desired for the heatmaps. The monthly insolation calculations didn't require heatmaps and used a function called *gencumsky* to create a cumulative sky based on all selected hours instead of *gendaylit* which created a sky based on a single hour. Should any of the other programs created as modified versions of this program be desired, please contact Taeo Laue at taeol[at]uio.no or tlaue[at]me.com

Optical models of various AgriPV geometries, focusing on BIPV on Greenhouses - Taeo Laue

Steps for this notebook:

- Set module parameters in bifacial_radiance
- Designate current directory as test folder (recommended to move this script to its own folder)
- Create Radiance object
- Create modules and input variables
- Add greenhouse supports
- Add greenhouse roof/walls
- Find irradiance of solar panels
- Hack sensors to calculate ground irradiance
- Analyze results and create heatmaps with Seaborn

```
from bifacial_radiance import *
import os
import numpy as np
import shutil
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt

%matplotlib inline

testfolder = os.getcwd()

print('This simulation will be stored in %s' % testfolder)
```

Flush objects folder (if necessary)

```
#os.makedirs(testfolder, exist_ok = True) #create testfolder if it does not already exist

dir = "objects"

objectsfolder = os.path.join(testfolder, dir)

os.makedirs(objectsfolder, exist_ok = True) #creates objects folder the first time, ignores if
 it already exists
```

```
shutil.rmtree(objectsfolder) #deletes old objects in objects folder

os.makedirs(objectsfolder) #creates blank objects folder each time

print('Radiance objects have been flushed from %s' % objectsfolder)
```

Flush results folder (if necessary)

```
dir = "results"

resultsfolder = os.path.join(testfolder, dir)

os.makedirs(resultsfolder, exist_ok = True)

shutil.rmtree(resultsfolder)

os.makedirs(resultsfolder)

print('Results from previous linescans have been flushed from %s' % resultsfolder)
```

Set module and location parameters for bifacial_radiance

```
#timestamp = 4020 # Noon, June 17th.
simulationname = 'AgriPV'

exp_name = 'plastic_gdl_checker_kjeller_5am'
#Location:
lat = 59.97861601327838   # Kjeller, Norge
lon = 11.0481239196028  # Kjeller, Norge
# MakeModule Parameters
module ='AgriPVglassglass'
numpanels = 8 #4 #8 #number of modules per column of modules
x = 1.95
y = 0.95
xgap = 3.0 #0.444 #4.872 # Leaving 300 centimeters between modules on x direction
ygap = 0.1 #1.15 # Leaving 10 centimeters between modules on y direction
zgap = 0.2 # 20 cm gap to torquetube
sensorsy = 6*numpanels  # this will give 6 sensors per module, 1 per cell
sensorsy_modules = sensorsy

# Other default values:

# TorqueTube Parameters
axisofrotationTorqueTube = False
torqueTube = False
cellLevelModule = True

numcellsx = 12
numcellsy = 6
xcell = 0.156 #0.00000001 #0.156
ycell = 0.156 #0.00000001 #0.156
xcellgap = 0.02
ycellgap = 0.02
glass = False

cellModule = {'numcellsx': numcellsx, 'numcellsy':numcellsy,
              'xcell': xcell, 'ycell': ycell, 'xcellgap': xcellgap, 'ycellgap': ycellgap}
```

```
# SceneDict Parameters
pitch = 15   # m (distance between rows, cannot be zero, doesn't matter if there is only 1 row
)
albedo = 0.2   # ground albedo
clearance_height = 4.3 #4.6   # m
#clearance_height2 = 5.213
nMods = 7 #14   # 7 number of columns
nRows = 1   # only 1 row, mounted on the south side of the greenhouse

azimuth_ang = 180   # Facing south
tilt = 35 #35  # tilt.



apv = RadianceObj(simulationname, path = testfolder)  # Create a RadianceObj 'object' with sim
ulation name
apv.setGround(albedo) # input albedo number or material name like 'concrete'.  To see options,
 run this without any input.
epwfile = apv.getEPW(lat, lon) # Kjeller lat/lon
#starttime = '2021-06-17_1200'
#endtime = '2021-06-17_1300'
date = '2021-06-17'
hour ='12'
min_sec = ':0:0' #required format for pd.to_datetime but cannot be used as part of a file name
timezone =' +1' #make timezone aware, +1 for CET, space required, no _
date_w_time = str(date)+' '+str(hour)+str(min_sec)+str(timezone)
metdata = apv.readWeatherFile(epwfile, coerce_year=2021)#, starttime=starttime, endtime=endtim
e, coerce_year=2021) # read in the EPW weather data from above
timestamp = metdata.datetime.index(pd.to_datetime(date_w_time))
apv.gendaylit(timestamp)  # Use this to simulate only one hour at a time.
# This allows you to "view" the scene on RVU (see instructions below)
# timestamp 4020 : Noon, June 17th.
#apv.genCumSky() # Use this instead of gendaylit to simulate the whole year or selectred time
period between starttime and endtime
# Making module with all the variables
moduleDict = apv.makeModule(name=module,x=x,y=y,numpanels = numpanels, xgap=xgap, ygap=ygap, c
ellModule = cellModule, glass=glass)
# create a scene with all the variables
sceneDict = {'tilt':tilt,'pitch':pitch,'clearance_height':clearance_height,'azimuth':azimuth_a
ng, 'module_type':module, 'nMods': nMods, 'nRows': nRows, 'originx': 0, 'originy': -4.1} #-1.2
2
#sceneDict = {'tilt':tilt,'pitch':pitch,'clearance_height':clearance_height,'azimuth':azimuth_
ang, 'module_type':module, 'nMods': nMods, 'nRows': nRows, 'originx': -1.37, 'originy': -4.6,
'appendRadfile':True} #-1.22
#sceneDict2 = {'tilt':tilt,'pitch':pitch,'clearance_height':clearance_height2,'azimuth':azimut
h_ang, 'module_type':module, 'nMods': nMods, 'nRows': nRows, 'originx': 1.17, 'originy': -3.65
, 'appendRadfile':True}
scene = apv.makeScene(module, sceneDict) #makeScene creates a .rad file with modules
#scene2 = apv.makeScene(module, sceneDict2)
#scene = apv.makeScene(sceneDict=sceneDict) #creates scene without modules?
octfile = apv.makeOct(apv.getfilelist())  # makeOct combines all of the ground, sky and object
 fil|es into a .oct file.
```

Create moduleDict with module parameters

IMPORTANT - READ ME

Add custom transparent material to ground.rad according to Radiance doc:

https://floyd.lbl.gov/radiance/refer/usman2.pdf

==============================

#greenhouse plastic white void trans greenhouse_plastic 0 0 7 1 1 1 0.2 0.5 0.9 0.1

==============================

The number after the first two zeros is a reference for how many parameters the material has, so make sure it is correct or the model will break.

~~To add this material for the program to reference, open this cell (this will show how the text should look in the ground.rad file), copy the text between the "=====" (not including the "===") to the ground.rad file in the 'materials' folder that has been generated in the testfolder directory. Ultimately, it would be nice to code python to automatically add this material the first time this program runs (since we don't need a ground.rad file that gets appended each time the program runs)~~

~~For now, adding the above text manually is sufficient.~~

UPDATE: The code below searches for and adds the custom material if necessary.

```
with open('materials/ground.rad') as f:
    materialslist = f.readlines()

if any("#greenhouse plastic" in s for s in materialslist):
    print("Material found")
    f.close()
else:
    print("Appending")
    materialslist.append('\n\n#greenhouse plastic white\n') #material description
    materialslist.append('void trans greenhouse_plastic\n') #material properties and call acco
rding to Radiancedocumentation
    materialslist.append('0\n')
    materialslist.append('0\n')
    materialslist.append('7 1 1 1 \n')
    materialslist.append('0.05 0.02 \n') #'0.04 0.05 \n')  #this material was sourced from sim
ilar study in Radiance https://www-sciencedirect-com.ezproxy.uio.no/science/article/pii/S03062
61921016986
    materialslist.append('0.7 0.3') #'0.869 0.22')    # Translucent cover - SG80 - 84% hemisph
erical tranmittance, 78% haze factor

    updateRad = open("materials/ground.rad", "w")
    print("Writing to new file")
    for m in materialslist:
        updateRad.write(m)
    updateRad.close()
    f.close()
```

Define torquetubelength

```
torquetubelength = 35.64 #(moduleDict.scenex)*(nMods) #should be just over 35m currently
torquetubelength
```

Create reference post at (0,0) (skip if this is no longer needed)

```
#name='CenterReferencePost'
#text='! genbox black originMarker 0.2 0.2 15 | xform -t -0.1 -0.1 0' # 20 x 20 cm, 15 meters
tall, offset by 10 cm in each direction
```

```
#customObject = apv.makeCustomObject(name,text)
#apv.appendtoScene(radfile=scene.radfiles, customObject=customObject, text='!xform -rz 0')

#octfile = apv.makeOct()
```

Create greenhouse support structure

```
# torquetube 1
name='Beam1'
text='! genbox Metal_Aluminum_Anodized beam_row1 {} 0.2 0.3 | xform -t {} -0.1 -0.3 | xform -t
 0 -7.5 4.2'.format(torquetubelength, (-torquetubelength)/2.0)
#text='! genbox black cuteBox 10 0.2 0.3 | xform -t -5 -0.1 -0.15 | xform  -t 0 15 4.2'.format
(z2nd, xleft, y2nd)
customObject = apv.makeCustomObject(name,text)
apv.appendtoScene(radfile=scene.radfiles, customObject=customObject, text="!xform -rz 0")

name='Beam2'
text='! genbox Metal_Aluminum_Anodized beam_row2 {} 0.2 0.3 | xform -t {} -0.1 -0.3 | xform -t
 0 7.5 4.2'.format(torquetubelength, (-torquetubelength)/2.0)
customObject = apv.makeCustomObject(name,text)
apv.appendtoScene(radfile=scene.radfiles, customObject=customObject, text="!xform -rz 0")

name='Beam3'
text='! genbox Metal_Aluminum_Anodized beam_roof {} 0.2 0.3 | xform -t {} -0.1 -0.3 | xform -t
 0 0 9.45'.format(torquetubelength, (-torquetubelength)/2.0)
customObject = apv.makeCustomObject(name,text)
apv.appendtoScene(radfile=scene.radfiles, customObject=customObject, text="!xform -rz 0")

name='Beam4'
text='! genbox Metal_Aluminum_Anodized beam_NW 0.2 0.3 9.155 | xform -rz 180 -rx 55 | xform -t
 {} 7.5 4.2'.format(-torquetubelength/2.0) # south facing roof, same length as torquetube
customObject = apv.makeCustomObject(name,text)
apv.appendtoScene(scene.radfiles, customObject, '!xform -rz 0')

name='Beam5'
text='! genbox Metal_Aluminum_Anodized beam_SW 0.2 0.3 9.155 | xform -rx -55 | xform -t {} -7.
5 4.2'.format((-torquetubelength/2.0)-0.2) # south facing roof, same length as torquetube
customObject = apv.makeCustomObject(name,text)
apv.appendtoScene(scene.radfiles, customObject, '!xform -rz 0')

name='Beam6'
text='! genbox Metal_Aluminum_Anodized beam_NE 0.2 0.3 9.155 | xform -rz 180 -rx 55 | xform -t
 {} 7.5 4.2'.format((torquetubelength/2.0)+0.2) # south facing roof, same length as torquetube
customObject = apv.makeCustomObject(name,text)
apv.appendtoScene(scene.radfiles, customObject, '!xform -rz 0')

name='Beam7'
text='! genbox Metal_Aluminum_Anodized beam_SE 0.2 0.3 9.155 | xform -rx -55 | xform -t {} -7.
5 4.2'.format(torquetubelength/2.0) # south facing roof, same length as torquetube
customObject = apv.makeCustomObject(name,text)
apv.appendtoScene(scene.radfiles, customObject, '!xform -rz 0')

name='Posts'
post1x = (torquetubelength)/2.0
postsep = post1x*2.0/(numpanels-1)
#! genrev Metal_Grey tube1 t*1.004 0.05 32 | xform -ry 90 -t -0.502 0 0
text= '! genrev Metal_Grey tube1row1 t*4.2 0.15 32 | xform -t {} -7.5 0'.format(post1x)
text += '\r\n! genrev Metal_Grey tube1row2 t*4.2 0.15 32 | xform -t {} 7.5 0'.format(post1x)

for i in range (1, numpanels):
    text += '\r\n! genrev Metal_Grey tube{}row1 t*4.2 0.15 32 | xform -t {} -7.5 0'.format(i+1
, post1x-postsep*i)
    text += '\r\n! genrev Metal_Grey tube{}row2 t*4.2 0.15 32 | xform -t {} 7.5 0'.format(i+1,
 post1x-postsep*i)

customObject = apv.makeCustomObject(name,text)
apv.appendtoScene(radfile=scene.radfiles, customObject=customObject, text="!xform -rz 0")

octfile = apv.makeOct()  # makeOct combines all of the ground, sky and object files into a .oc
t file.
```

Add roof, walls, floor and gap between east/west walls and roof, and then create the oct file

```
name='GlassGreenhouseWallSouth'
text='! genbox greenhouse_plastic greenhousewall_S {} 0.001 4.2 | xform -t {} -0.1 0 | xform -
t 0 -7.5 0'.format(torquetubelength, (-torquetubelength)/2.0) # 28 m wall, 4.2 meters high
customObject = apv.makeCustomObject(name,text)
apv.appendtoScene(radfile=scene.radfiles, customObject=customObject, text='!xform -rz 0')

name='GlassGreenhouseWallNorth'
text='! genbox greenhouse_plastic greenhousewall_N {} 0.001 4.2 | xform -t {} -0.1 0 | xform -
t 0 7.5 0'.format(torquetubelength, (-torquetubelength)/2.0) # 28 m wall, 4.2 meters high
customObject = apv.makeCustomObject(name,text)
apv.appendtoScene(radfile=scene.radfiles, customObject=customObject, text='!xform -rz 0')

name='GlassGreenhouseWallWest'
text='! genbox greenhouse_plastic greenhousewall_W 0.001 15 4.2 | xform -t {} -0.1 0 | xform -
t 0 -7.5 0'.format((-torquetubelength)/2.0) # 15 m wall, 4.2 meters high
customObject = apv.makeCustomObject(name,text)
apv.appendtoScene(radfile=scene.radfiles, customObject=customObject, text='!xform -rz 0')

name='GlassGreenhouseWallEast'
text='! genbox greenhouse_plastic greenhousewall_E 0.001 15 4.2 | xform -t {} -0.1 0 | xform -
t 0 -7.5 0'.format((torquetubelength)/2.0) # 15 m wall, 4.2 meters high
customObject = apv.makeCustomObject(name,text)
apv.appendtoScene(radfile=scene.radfiles, customObject=customObject, text='!xform -rz 0')

name='WesternGap'
text='! genprism greenhouse_plastic prism_w 3 0 0 7.5 5.25 15 0 -l 0 0 .001 | xform -rz 90 -ry
 90 | xform -t {} -7.5 4.2'.format((-torquetubelength)/2.0)
customObject = apv.makeCustomObject(name,text)
apv.appendtoScene(radfile=scene.radfiles, customObject=customObject, text='!xform -rz 0')

name='EasternGap'
text='! genprism greenhouse_plastic prism_e 3 0 0 7.5 5.25 15 0 -l 0 0 .001 | xform -rz 90 -ry
 90 | xform -t {} -7.5 4.2'.format((torquetubelength)/2.0)
customObject = apv.makeCustomObject(name,text)
apv.appendtoScene(radfile=scene.radfiles, customObject=customObject, text='!xform -rz 0')

name='GlassGreenhouseRoofSouth'
text='! genbox greenhouse_plastic greenhouseroof_S {} 0.001 9.155 | xform -rx -55 | xform -t {
} -7.5 4.2'.format(torquetubelength, (-torquetubelength)/2.0) # south facing roof, same length
 as torquetube
customObject = apv.makeCustomObject(name,text)
apv.appendtoScene(scene.radfiles, customObject, '!xform -rz 0')

name='GlassGreenhouseRoofNorth'
text='! genbox greenhouse_plastic greenhouseroof_N {} 0.001 9.155 | xform -rx 55 | xform -t {}
 7.5 4.2'.format(torquetubelength, (-torquetubelength)/2.0) # south facing roof, same length a
```

```
s torquetube
customObject = apv.makeCustomObject(name,text)
apv.appendtoScene(scene.radfiles, customObject, '!xform -rz 0')

name='Greenhouse_Floor'
text='! genbox white_EPDM CenterPatch {} 15 0.01 | xform -t {} -7.5 0'.format(torquetubelength
,(-torquetubelength)/2.0)
customObject = apv.makeCustomObject(name,text)
apv.appendtoScene(scene.radfiles, customObject, '!xform -rz 0')

octfile = apv.makeOct(apv.getfilelist())  # makeOct combines all of the ground, sky and object
 files into a .oct file
```

This section had the roof and wall material changed to Metal_Gray in order to test if the ray tracing was truly working within the walls of the greenhouse model

Check RVU: rvu -vf views\front.vp -e errors.txt -pe 0.5 -vp -34 -30 10 -vd 0.7 0.7 -0.15 AgriPV.oct

Module analysis

```
analysis = AnalysisObj(octfile, apv.name)  # return an analysis object including the scan dime
nsions for back irradiance
frontscan, backscan = analysis.moduleAnalysis(scene, sensorsy=sensorsy)
analysis.analysis(octfile, simulationname+"_modules_"+str(exp_name), frontscan, backscan) #str
(starttime)+"_to_"+str(endtime), frontscan, backscan) #, accuracy='high')

moduleResultFile = 'results\irr_AgriPV_modules_'+str(exp_name)+'.csv' #str(starttime)+"_to_"+s
tr(endtime)+'.csv'
moduleresults_loaded = load.read1Result(moduleResultFile)
moduleresults_loaded

clean_moduleresults = load.cleanResult(moduleresults_loaded)

cleanedfolder = os.path.join(testfolder, 'cleaned_module_results')

os.makedirs(cleanedfolder, exist_ok = True)

print("Cleaned results folder located at %s" %cleanedfolder)

clean_moduleresults.to_csv('cleaned_module_results/cleaned_irr_modules_'+str(exp_name)+'.csv')
 #+str(starttime)+'_to_'+str(endtime)+'.csv')

clean_moduleresults
```

Ground irradiance analysis at 5 cm above the ground

```
sensorsy = 15
sensorsx = 36

analysis = AnalysisObj(octfile, apv.name)  # return an analysis object including the scan dime
nsions for back irradiance
startgroundsample= -17.5 #(means the scans will start at -17.5m and run to 17.5m, all within t
he greenhouse)
spacingbetweensamples = 1 #one meter between each scan

%%time
i = 0
for i in range (0, sensorsx):
    frontscan, backscan = analysis.moduleAnalysis(scene, sensorsy=sensorsy)
    groundscan = frontscan
    groundscan['zstart'] = 0.05  # setting it 5 cm from the ground.
    groundscan['zinc'] = 0   # no tilt necessary.
    groundscan['yinc'] = 1   # increasing spacing so it covers all distance between rows
    groundscan['ystart'] = -7 #scans start at -7 meters in the y direction (southern side of the greenhouse floor)
    groundscan['xinc'] = 0 #each subsequent sensor should be placed in a straight line, no increments
    groundscan['xstart'] = startgroundsample + i*spacingbetweensamples   # increasing spacing so it covers all distance between rows
    analysis.analysis(octfile, simulationname+'_groundscan_5cm_'+str(i), groundscan, backscan)


import csv

i = 0
exp_name = 'plastic_gdl_checker_kjeller_5am'

arrayWm2Front = []
#arrayWm2Back = []
arrayMatFront = []
#arrayMatBack = []
arrayx = []
arrayy = []
faillist = []

for i in range (0, sensorsx):
    try:
        iresults = 'results\irr_AgriPV_groundscan_5cm_'+str(i)+'.csv'
        resultsDF = load.read1Result(iresults)
        arrayWm2Front.extend(list(resultsDF['Wm2Front']))
        #arrayWm2Back.extend(list(resultsDF['Wm2Back']))
        arrayMatFront.extend(list(resultsDF['mattype']))
        #arrayMatBack.extend(list(resultsDF['rearMat']))
        arrayx.extend(list(resultsDF['x']))
        arrayy.extend(list(resultsDF['y']))
    except:
        print(" FAILED ", i)
        faillist.append(i)

#ground_5cm_result_file = 'results\irr_AgriPV_groundscan_5cm.csv'
#ground_irr_5cm = load.read1Result(ground_5cm_result_file)
#ground_irr_5cm

header = ['Bifacial Gain Percent', 'Average Front Irr (W/m2)', 'Average Back Irr (W/m2)', 'Max Ground Irr (W/m2)', 'Min Ground Irr (W/m2)', 'Average Ground Irr (W/m2)']
data = ["{:.3f}".format((clean_moduleresults['Wm2Back'].sum()*100) / clean_moduleresults['Wm2Front'].sum()), "{:.3f}".format((clean_moduleresults['Wm2Front'].sum()/(sensorsy_modules))), "{:.3f}".format((clean_moduleresults['Wm2Back'].sum()/(sensorsy_modules))), "{:.3f}".format(max(arrayWm2Front)), "{:.3f}".format(min(arrayWm2Front)),"{:.3f}".format(sum(arrayWm2Front)/(sensorsx*sensorsy))]

with open('cleaned_module_results/bf_gain_'+str(exp_name)+'.csv', 'w', newline='') as f:
    writer = csv.writer(f)

    # write the header
    writer.writerow(header)
    writer.writerow(data)
```

Create heatmap of ground irradiance

```
heatmap_df = pd.DataFrame(list(zip(arrayx, arrayy,
                                  arrayWm2Front, arrayMatFront)), #arrayMatBack)), arrayWm2Bac
k,
                         columns = ['x', 'y',
                                    'Wm2Front', 'MatFront']) #, 'MatBack']) 'Wm2Back']

sns.set(font_scale=2, rc = {'figure.figsize':(20,10)})
heatmap_df_piv = heatmap_df.pivot('y', 'x', 'Wm2Front')
#plt.title(str(starttime)+'_to_'+str(endtime))
plt.title(str(exp_name))
sns.set(font_scale=2, rc = {'figure.figsize':(20,10)})
display_heatmap = sns.heatmap(heatmap_df_piv, square=True, cbar_kws={'label': 'Watts per squar
e meter'})
display_heatmap.invert_yaxis()
display_heatmap.set(ylabel="Western Side, m from center", xlabel="Southern Side, m from center
")
plt.show()

heatmapsfolder = os.path.join(testfolder, 'heatmaps')

os.makedirs(heatmapsfolder, exist_ok = True)

print("Heatmaps folder located at %s" %heatmapsfolder)

groundscansfolder = os.path.join(testfolder, 'groundscans')

os.makedirs(groundscansfolder, exist_ok = True)

print("Groundscans folder located at %s" %groundscansfolder)

display_heatmap.figure.savefig("heatmaps/heatmap_"+str(exp_name)+".png", bbox_inches='tight')
heatmap_df.to_csv('groundscans/groundscan_'+str(exp_name)+'.csv')
#display_heatmap.figure.savefig("heatmaps/heatmap_"+str(starttime)+"_to_"+str(endtime)+".png")
#heatmap_df.to_csv('groundscans/groundscan_'+str(starttime)+"_to_"+str(endtime)+'.csv')
```

References

[1] S. Ong, C. Campbell, P. Denholm, R. Margolis, and G. Heath, “Land-Use Requirements for Solar Power Plants in the United States,” NREL/TP-6A20-56290, 1086349, Jun. 2013. doi: 10.2172/1086349.

[2] “Land-use requirements of solar,” *ScienceDaily*. https://www.sciencedaily.com/releases/2013/08/130806145537.htm (accessed May. 04, 2022).

[3] C. Toledo and A. Scognamiglio, “Agrivoltaic Systems Design and Assessment: A Critical Review, and a Descriptive Model towards a Sustainable Landscape Vision (Three-Dimensional Agrivoltaic Patterns),” *Sustainability*, vol. 13, no. 12, Art. no. 12, Jan. 2021, doi: 10.3390/su13126871.

[5] T. S. and A. Nagashima, “Solar Sharing for Both Food and Clean Energy Production: Performance of Agrivoltaic Systems for Corn, A Typical Shade-Intolerant Crop.” https://www.mdpi.com/2076-3298/6/6/65 (accessed Mar. 30, 2022).

[6] “Agrivoltaics: Opportunities for Agriculture and the Energy Transition - Fraunhofer ISE,” Fraunhofer Institute for Solar Energy Systems ISE. https://www.ise.fraunhofer.de/en/publications/studies/agrivoltaics-opportunities-for-agriculture-and-the-energy-transition.html (accessed Apr. 20, 2022).

[7] N. Akira and T. Sekiyama, *Solar Sharing: Changing the world and life*. 2020.

[8] R. C. Costa, E. O. Calvete, J. Schons, and F. H. Reginatto, “CHLOROPHYLL CONTENT IN STRAWBERRY LEAVES PRODUCED UNDER SHADING SCREENS IN GREENHOUSE,” *Acta Hortic.*, no. 926, pp. 321–324, Jan. 2012, doi: 10.17660/ActaHortic.2012.926.44.

[9] K. Król and W. Lech, “EFFECT OF SHORTENING DAYLENGTH ON FLOWER BUD DIFFERENTIATION AND CROWN DEVELOPMENT IN ‘ELSANTA’ STRAWBERRY,” *Acta Hortic.*, no. 926, pp. 307–313, Jan. 2012, doi: 10.17660/ActaHortic.2012.926.42.

[10] Xiaoyan Li, Wei Chen, and Yadong Li, “STUDY ON PHOTOSYNTHETIC CHARACTERISTICS OF BLUEBERRY IN GREENHOUSE,” *Acta Hortic.*, no. 926, pp. 315–319, Jan. 2012, doi: 10.17660/ActaHortic.2012.926.43.

[11] P. Angmo, T. Dolma, N. Phuntsog, O. Chaurasia, and T. Stobdan, “Effect of shading and high temperature amplitude on yield and phenolic contents of greenhouse capsicum (Capsicum annuum L.),” *Open Access Res. J. Biol. Pharm.*, vol. 4, pp. 030–039, Jan. 2022, doi: 10.53022/oarjbp.2022.4.1.0053.

[12] A. G. Imenes, H. G. Beyer, K. Boysen, J. O. Odden, and R. E. Grundt, “Performance of grid-connected PV system in Southern Norway,” in *2015 IEEE 42nd Photovoltaic Specialist Conference (PVSC)*, Jun. 2015, pp. 1–6. doi: 10.1109/PVSC.2015.7355823.

[13] D. G. Dye, “Spectral composition and quanta-to-energy ratio of diffuse photosynthetically active radiation under diverse cloud conditions,” *J. Geophys. Res. Atmospheres*, vol. 109, no. D10, 2004, doi: 10.1029/2003JD004251.

[15] G. T. Bruggink and E. Heuvelink, “Influence of light on the growth of young tomato, cucumber and sweet pepper plants in the greenhouse: Effects on relative growth rate, net assimilation rate and leaf area ratio,” *Sci. Hortic.*, vol. 31, no. 3, pp. 161–174, May 1987, doi: 10.1016/0304-4238(87)90043-4.

[16] M. Naseer, T. Persson, I. Righini, C. Stanghellini, H. Maessen, and M. J. Verheul, “Bio-economic evaluation of greenhouse designs for seasonal tomato production in Norway,” *Biosyst. Eng.*, vol. 212, pp. 413–430, Dec. 2021, doi: 10.1016/j.biosystemseng.2021.11.005.

[17] “LAUNCH OF A MULTI-FUNCTIONAL PILOT AGRIVOLTAIC INSTALLATION BY INSOLIGHT, ROMANDE ENERGIE AND AGROSCOPE – Insolight.” https://insolight.ch/launch-of-a-multi-functional-pilot-agrivoltaic-installation-by-insolight-romande-energie-and-agroscope/ (accessed Apr. 04, 2022).

[18] cdeline *et al.*, *NREL/bifacial_radiance: v0.4.1*. Zenodo, 2022. doi: 10.5281/zenodo.6342434.

[19] “PC1D | PVEducation.” https://www.pveducation.org/pvcdrom/welcome-to-pvcdrom/pc1d (accessed May 09, 2022).

[20] “Home,” *PVsyst*. https://www.pvsyst.com/ (accessed May 09, 2022).

[21] *pvlib: A set of functions and classes for simulating the performance of photovoltaic energy systems*. Accessed: May 07, 2022. [OS Independent]. Available: https://github.com/pvlib/pvlib-python

[22] “Solcast - Solar Forecasting & Solar Irradiance Data,” *Solcast*. https://solcast.com/ (accessed May 08, 2022).

[23] H. Villa-Martinez, “Accelerating Algorithms for Ray Tracing,” Dec. 2006.

[24] “The RADIANCE Lighting Simulation and Rendering System.” https://floyd.lbl.gov/radiance/papers/sg94.1/paper.html (accessed Feb 06, 2022).

[25] M. Hegazy, K. Yasufuku, and H. Abe, “Validating Game Engines as a Quantitative Daylighting Simulation Tool,” Apr. 2021.

[26] M. Hegazy, K. Yasufuku, and H. Abe, “An interactive approach to investigate brightness perception of daylighting in Immersive Virtual Environments: Comparing subjective responses and quantitative metrics,” *Build. Simul.*, vol. 15, no. 1, pp. 41–68, Jan. 2022, doi: 10.1007/s12273-021-0798-3.

[27] K. S. Park, S. K. Kim, S. G. Lee, H. J. Lee, and J. K. Kwon, “Application of plasma lighting for growth and flowering of tomato plants,” *Hortic. Environ. Biotechnol.*, vol. 59, no. 6, pp. 827–833, Dec. 2018, doi: 10.1007/s13580-018-0052-9.

[28] C. R. Brodersen, T. C. Vogelmann, W. E. Williams, and H. L. Gorton, “A new paradigm in leaf-level photosynthesis: direct and diffuse lights are not equal,” *Plant Cell Environ.*, vol. 31, no. 1, pp. 159–164, 2008, doi: 10.1111/j.1365-3040.2007.01751.x.

[29] K. J. McCree, “Test of current definitions of photosynthetically active radiation against leaf photosynthesis data,” *Agric. Meteorol.*, vol. 10, pp. 443–453, Jan. 1972, doi: 10.1016/0002-1571(72)90045-3.

[30] K. J. McCree, “The action spectrum, absorptance and quantum yield of photosynthesis in crop plants,” *Agric. Meteorol.*, vol. 9, pp. 191–216, Jan. 1971, doi: 10.1016/0002-1571(71)90022-7.

[31] M. Sandri, A. Luiz, M. Witter, and D. Tiago, “Effect of shading on tomato plants grow under greenhouse,” *Hortic. Bras.*, vol. 21, Dec. 2003, doi: 10.1590/S0102-05362003000400013.

[33] O. A. Katsikogiannis, H. Ziar, and O. Isabella, “Integration of bifacial photovoltaics in agrivoltaic systems: A synergistic design approach,” *Appl. Energy*, vol. 309, p. 118475, Mar. 2022, doi: 10.1016/j.apenergy.2021.118475.

[34] J. Louw and A. Rix, “Irradiance modelling for bi-facial PV modules using the ray tracing technique,” Mar. 2019. doi: 10.1109/RoboMech.2019.8704817.

[35] M. Evans, J. Langley, F. Shapiro, and G. Jones, “A Validated Model, Scalability, and Plant Growth Results for an Agrivoltaic Greenhouse,” *Sustainability*, vol. 14, p. 6154, May 2022, doi: 10.3390/su14106154.

[36] “Rhino - Rhinoceros 3D.” https://www.rhino3d.com/ (accessed Mar. 09, 2022).

[37] “72M G1 Bifacial Module,” *HELIENE*. https://heliene.com/72m-g1-bifacial-module/ (accessed May 11, 2022).

[38] *bifacial_radiance*. National Renewable Energy Laboratory, 2022. Accessed: May 09, 2022. [Online]. Available: https://github.com/NREL/bifacial_radiance

[39] “EnergyPlus.” https://energyplus.net/weather/sources#IWEC (accessed May. 08, 2022).

[40] S. Crone, “RADIANCE USERS MANUAL,” p. 64. https://www.radiance-online.org/archived/radsite/radiance/refer/usman2.pdf/ (accessed Feb. 15, 2022).

[41] A. Pettersen, “The performance of household PV systems in Southeastern Norway,” *109*, 2019, Accessed: May 06, 2022. [Online]. Available: https://nmbu.brage.unit.no/nmbu-xmlui/handle/11250/2612009

[42] T. Pan *et al.*, “Increased CO $_2$ and light intensity regulate growth and leaf gas exchange in tomato,” *Physiol. Plant.*, vol. 168, no. 3, pp. 694–708, Mar. 2020, doi: 10.1111/ppl.13015.