Bifacial_Radiance Training Part 1

Silvana Ayala Peláez and Chris Deline

NREL Webinar
October 17, 2019

Housekeeping

- This webinar is being recorded.
  - A recording will be made available on the “NREL Learning” YouTube channel and the GitHub & readthedocs page a few weeks after the webinar.
- Audio issues?
  - Try switching from computer to phone audio
  - If problems persist, contact GoToWebinar technical support at (877) 582-7011.
- Have a question?
  - Please submit it using the ”Questions” panel in the GoToWebinar interface at any time.
  - We will answer as many questions as possible at several point during the webinar.

bifacial_radiance

Open Source toolkit for working with RADIANCE for the ray-trace modeling of Bifacial Photovoltaics

Available on: https://github.com/NREL/bifacial_radiance
Documentation: https://bifacial-radiance.readthedocs.io
Installation: https://youtu.be/4A9GocfHKyM
V. 0.3.3.1 (10/16/19)

Radiance

Radiance is a ray-trace software – a suite of tools for performing lighting simulation.

Developed in 1985, by Greg Ward at Lawrence Berkeley National Laboratory

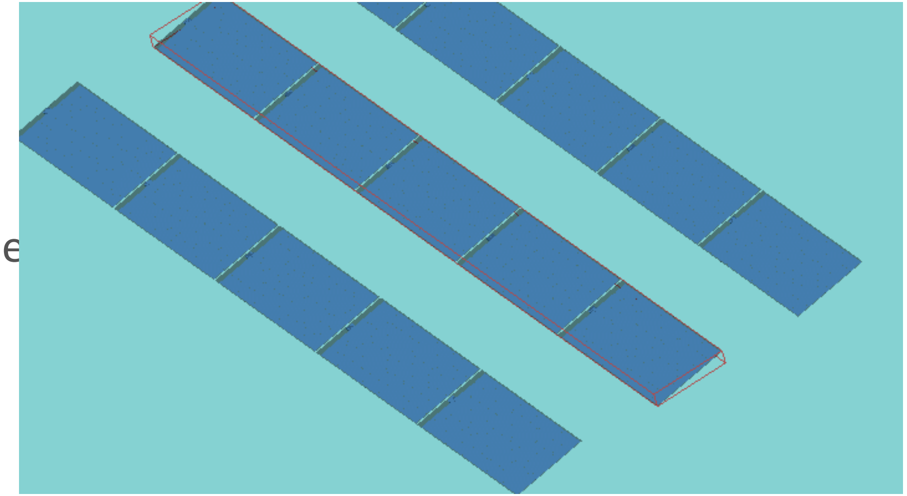
Underlying simulation engine for many packages.

Homepage: https://floyd.lbl.gov/radiance/HOME.html

Modules << Sky

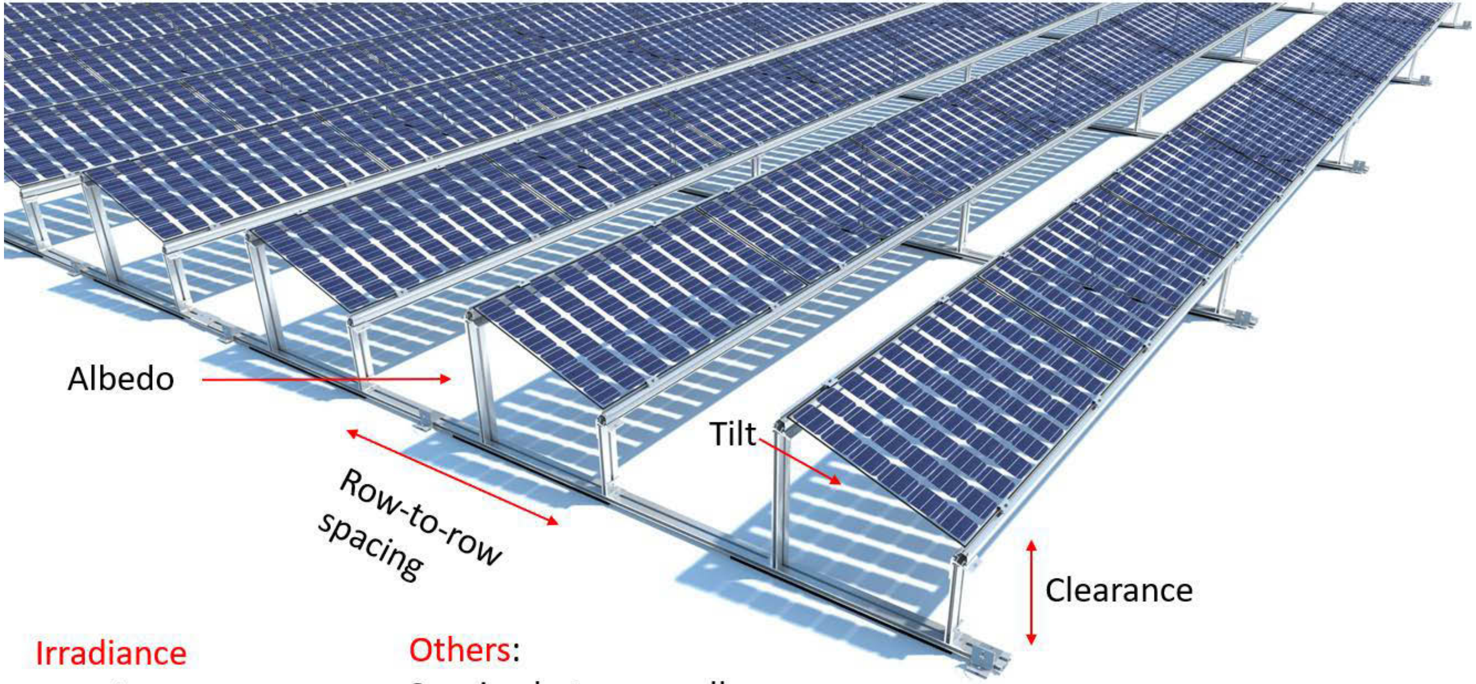
We use backward ray-trace to evaluate the irradiance (W/m^2) at the modules

Reduces complexity and run-time.



Bifacial Challenge: accurate modelling of rear irradiance

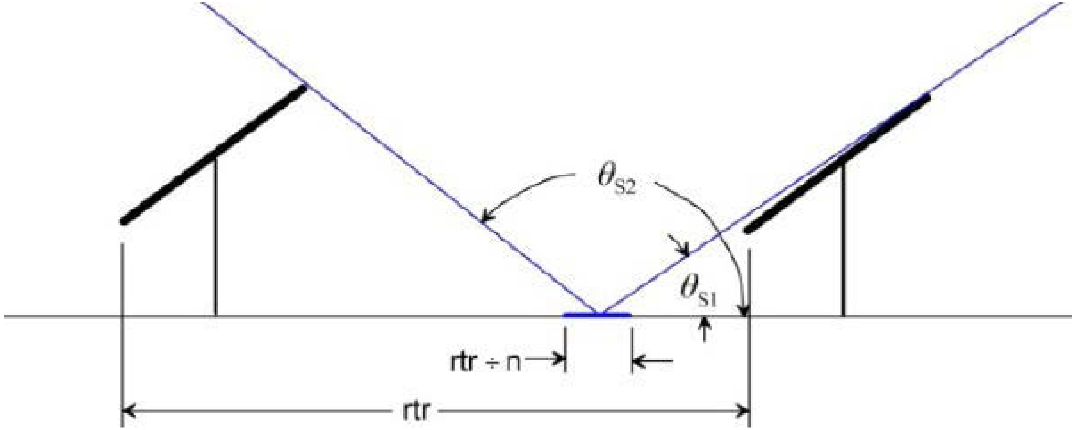
Image courtesy of Opsun trackers, via Francois Gilles-Gagnon



Irradiance
Location
Weather
Sky Diffuse

Others:
Spacing between cells
#rows, #panels
Mounting Structure
Other scene elements

View Factor Model for Rear Irradiance

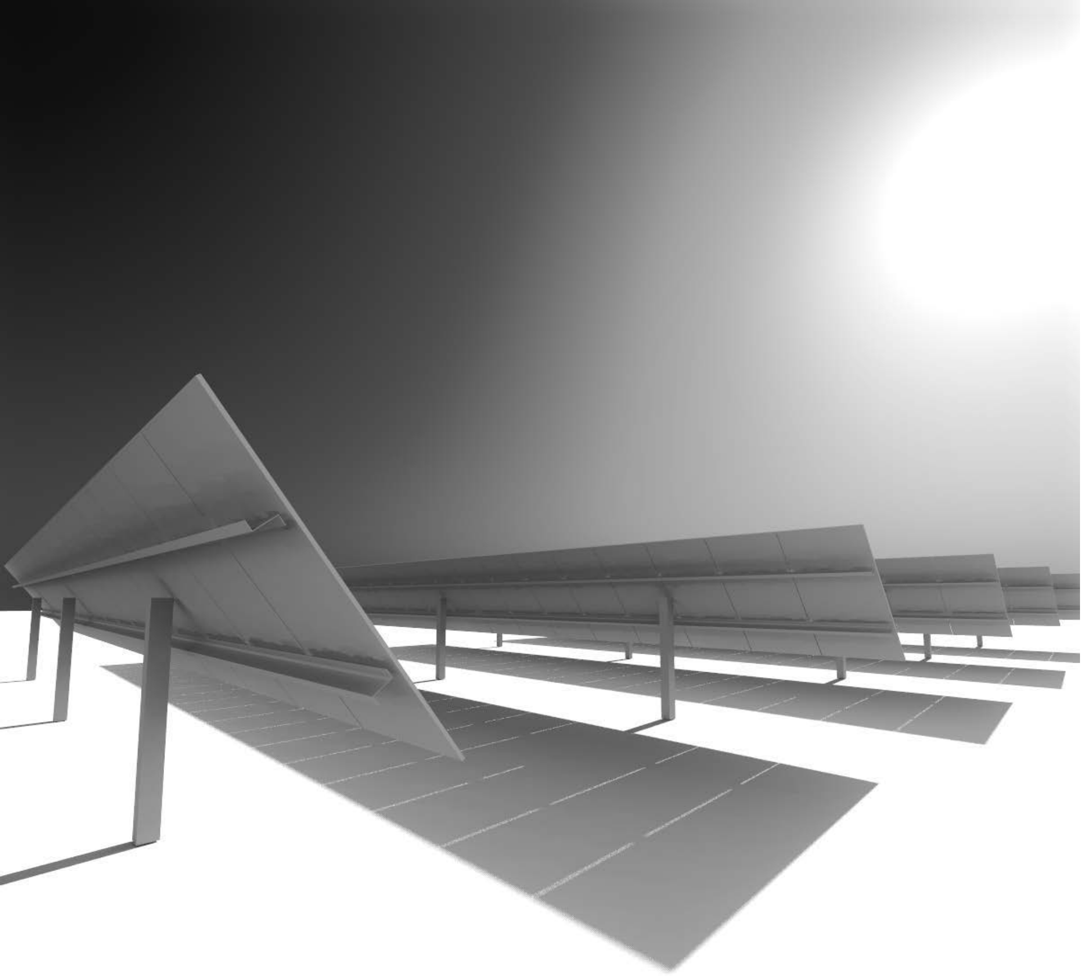


Simple
basic
geometry

Fast
computationally
inexpensive

Common
Behind
SAM, Pvsyst, bifacialvf
and others

Simple
basic
geometry



bifacial_radiance

Courtesy of Jason Alderman

Created with NREL bifacial_radiance https://github.com/NREL/bifacial_radiance

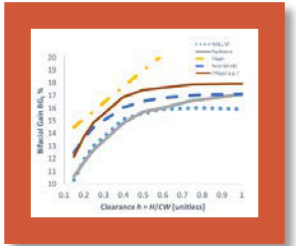
Raytrace benefits:

- Any size array
- Sample any module
- Evaluate edge effects
- Complicated geometries
    - Modules
    - Racking
    - Obstructions
- Evaluate shading
- Evaluate electrical mismatch
- Open source
- Dedicated visual interface
- Validated

Cons:

- Complexity
- Run-times

← Visual interface
← Training
← HPC integration
← Simplified models

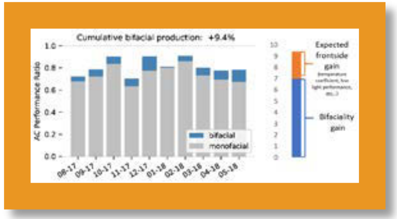


2017

Comparison of models and validation with test-bed data

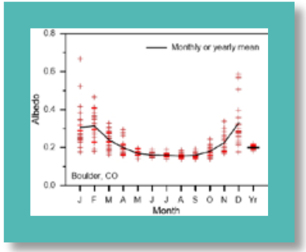
Single-axis tracking model

2018



Framework to calculate bifacial gain with field data

Validation with 2 100kW fields



Albedo assessment and database at Duramat

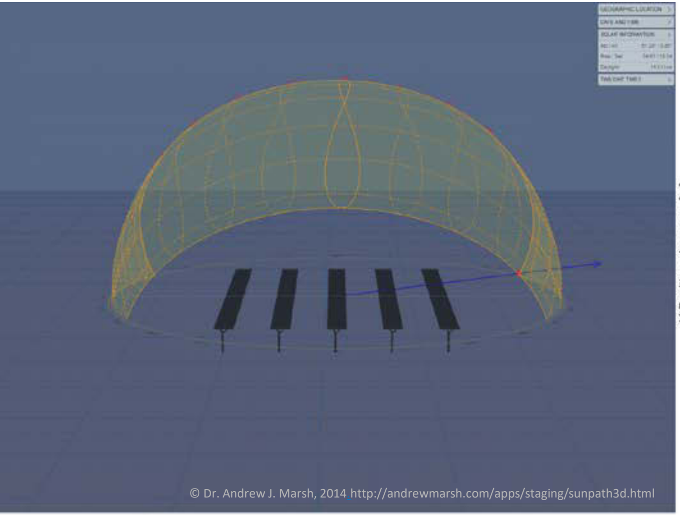
2019

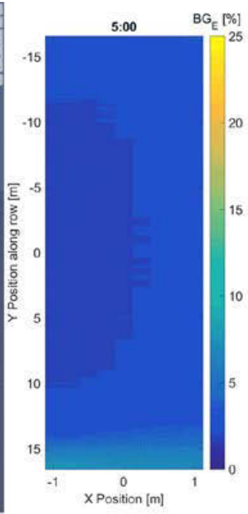
Standards for bifacial rating

Capacity testing

Assess system performance impact from rear irradiance shading and electrical mismatch

Publications





S. Ayala Pelaez, C. Deline, et al "Comparison of bifacial solar irradiance model predictions with field validation," IEEE J. Photovoltaics, vol. 9, no. 1, pp. 82–88, 2019. https://ieeexplore.ieee.org/abstract/document/8534404

S. Ayala Pelaez, C. Deline, et al "Model and Validation of Single-Axis Tracking with Bifacial PV" IEEE J. Photovoltaics, 2019.
https://ieeexplore.ieee.org/abstract/document/8644027
https://www.nrel.gov/docs/fy19osti/72039.pdf

S. Ayala Pelaez, C. Deline, et al "Effect of Torque-Tube Parameters on Rear-Irradiance and Rear-Shading Loss for Bifacial PV Performance on Single-Axis Tracking Systems", 46th PVSC. https://www.nrel.gov/docs/fy20osti/73203.pdf

Deline, C., et al "Bifacial PV Mismatch Loss Estimation and Parameterization". 36th EUPVSEC
https://www.nrel.gov/docs/fy20osti/73541.pdf
*Submitted as extended journal to PinPV

Complete list of publications: https://github.com/NREL/bifacial_radiance/wiki

Bifacial_radiance Training Part 1

1. Why/how we raytrace
2. Geometry
3. Skies
4. Github
5. Demo 1: Tutorial 1 in Jupyter Journal
6. Other Tutorials
7. Demo 2: GUI

20 minutes

10 minutes

10 minutes

Why | How we raytrace?

Components

Sky

Source Properties

"Scene"

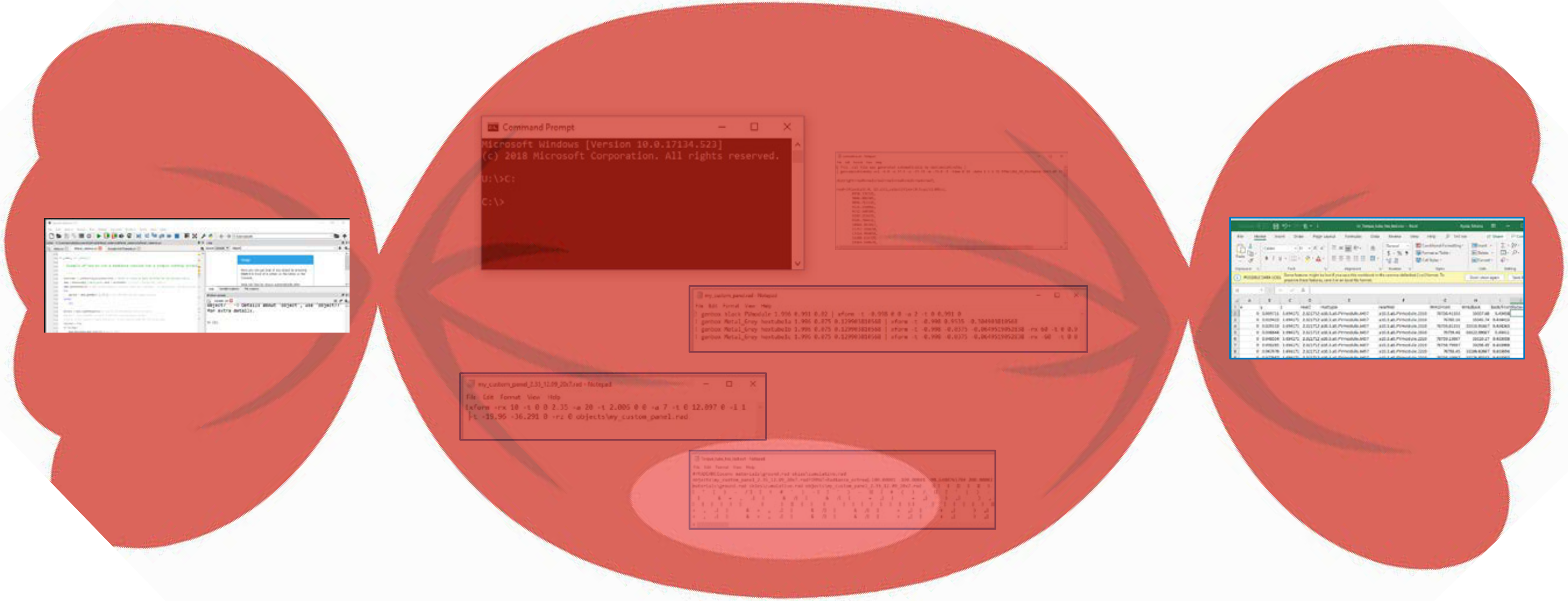
Modules
Racks
Obstructions
etc..

Materials
Properties

What does bifacial_radiance do?

bifacial_radiance

bifacial_radiance is a python wrapper for calling and using Radiance, with specific functions to generate geometry (text files) related to bifacial pv systems



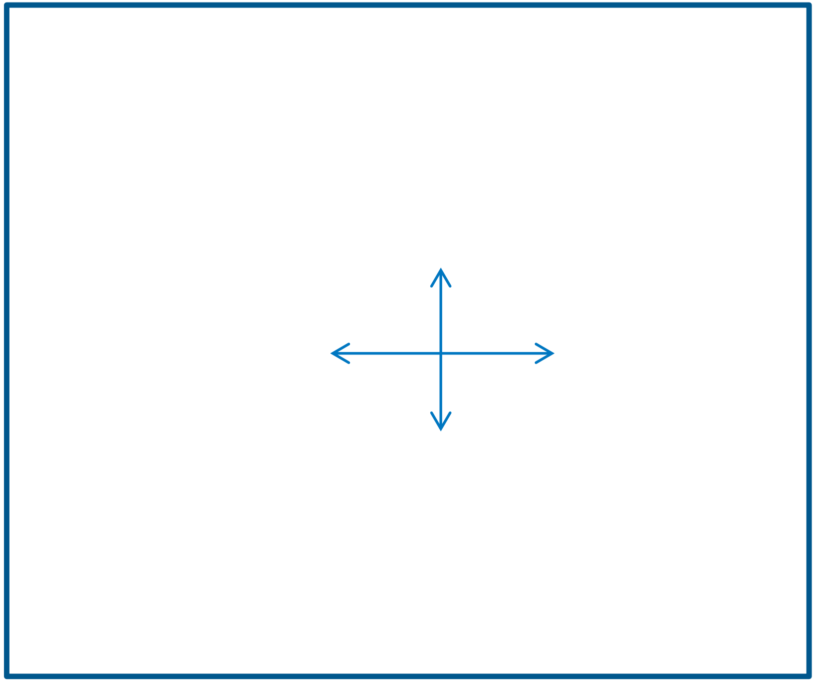
How a full example might look like

In [ ]:

```
from bifacial_radiance import *
demo = RadianceObj()
demo.setGround(0.62)
epwfile = demo.getEPW(37.5, -77.6)
metdata = demo.readWeatherFile(epwfile)
demo.gendaylit(metdata, 4020)
demo.makeModule("My_panel",x=1,y=2)
sceneDict={'tilt':30, 'pitch':3, 'clearance_height':0.5, 'azimuth':180, 'nMods':10, 'nRows':4}
scene = demo.makeScene("My_panel", sceneDict)
octfile=demo.makeOct()
analysis = AnalysisObj(octfile, demo.name)
frontscan, backscan = analysis.moduleAnalysis(scene)
analysis.analysis(octfile, demo.name, frontscan, backscan)
```

MAIN STEPS / INSTRUCTIONS

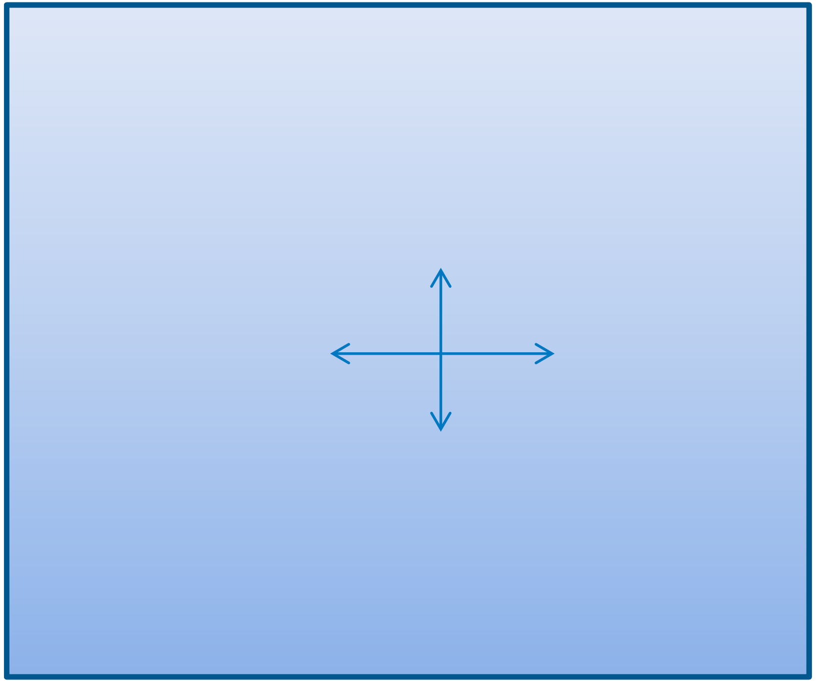
1. Make Radiance Object



MAIN STEPS / INSTRUCTIONS

1. Make Radiance Object

2. Make Sky

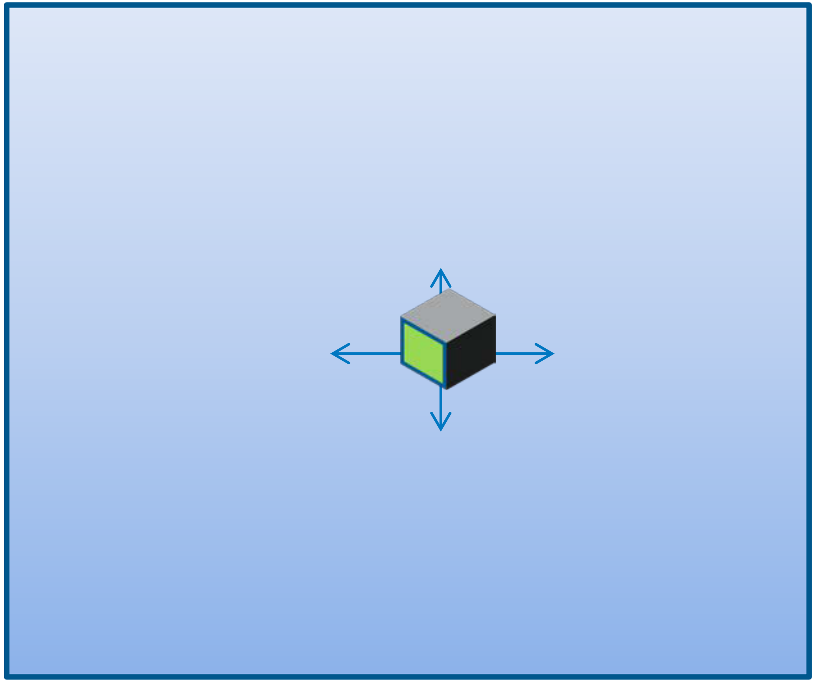


MAIN STEPS / INSTRUCTIONS

1. Make Radiance Object

2. Make Sky

3. Make Module



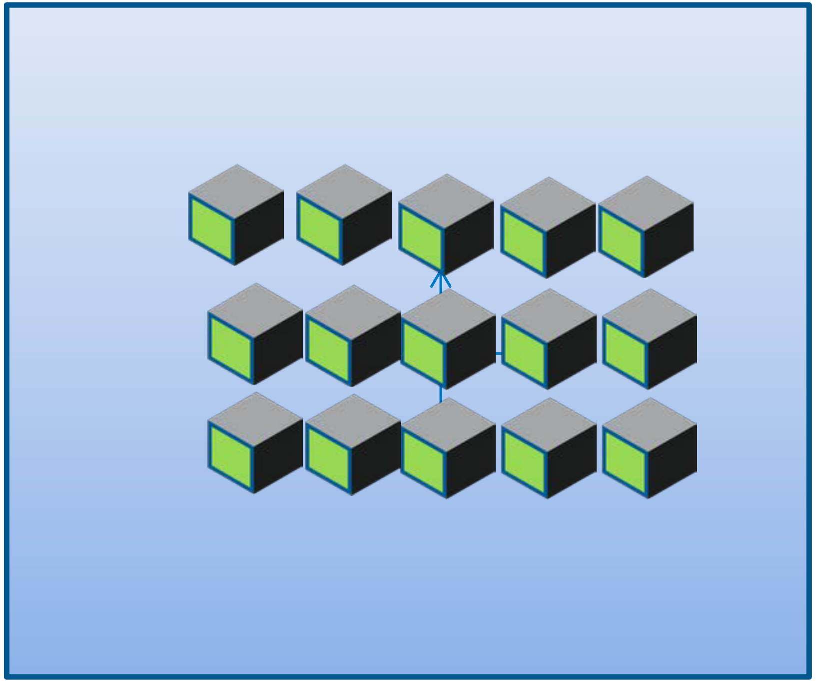
MAIN STEPS / INSTRUCTIONS

1. Make Radiance Object

2. Make Sky

3. Make Module

4. Make Scene



MAIN STEPS / INSTRUCTIONS

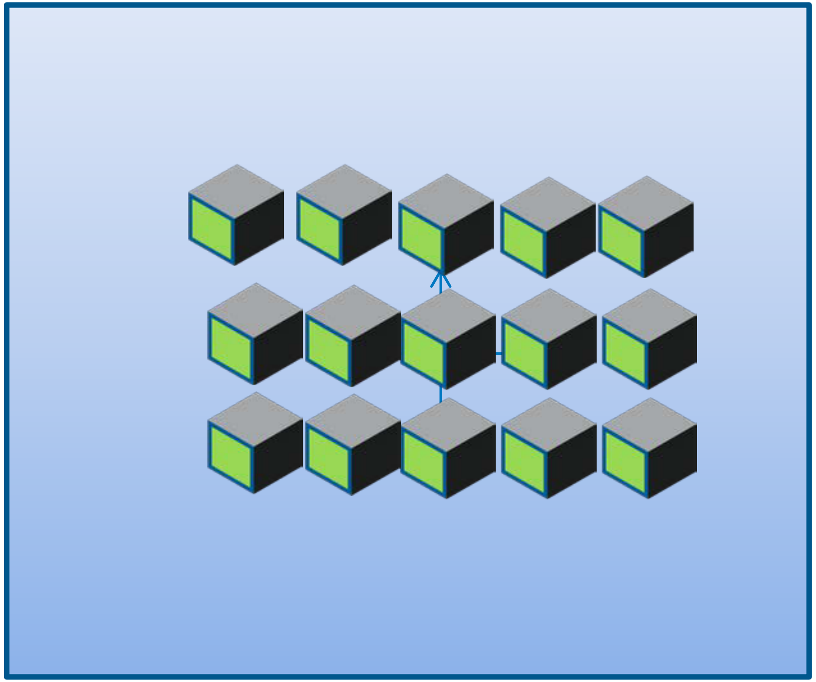
1. Make Radiance Object

2. Make Sky

3. Make Module

4. Make Scene

5. Make Oct



MAIN STEPS / INSTRUCTIONS

1. Make Radiance Object

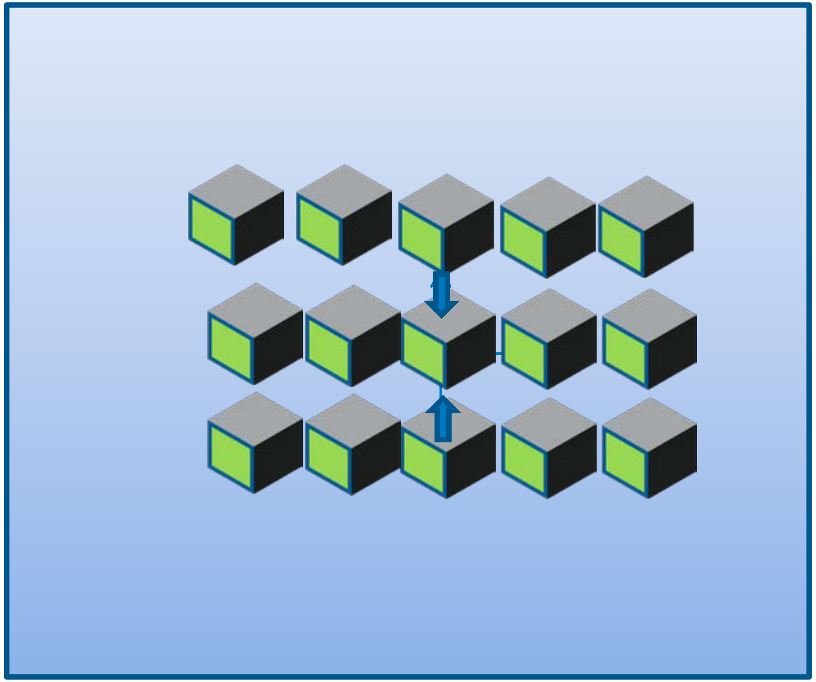
2. Make Sky

3. Make Module

4. Make Scene

5. Make Oct

6. Analysis Obj



MAIN STEPS / INSTRUCTIONS

1. Make Radiance Object

2. Make Sky

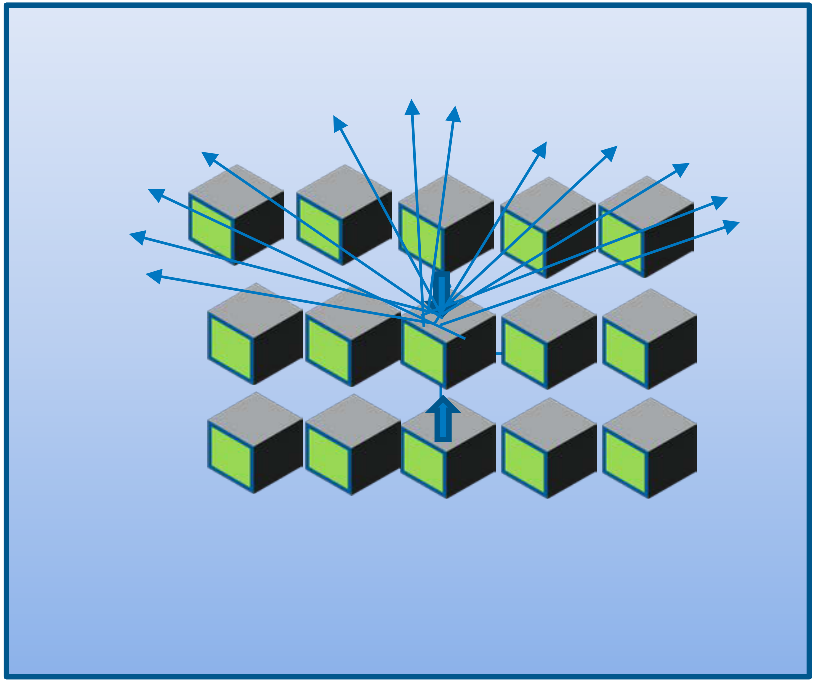
3. Make Module

4. Make Scene

5. Make Oct

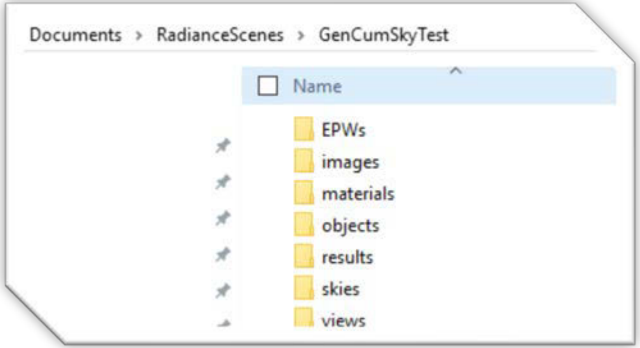
6. Analysis Obj

7. Analysis



MAIN STEPS / INSTRUCTIONS

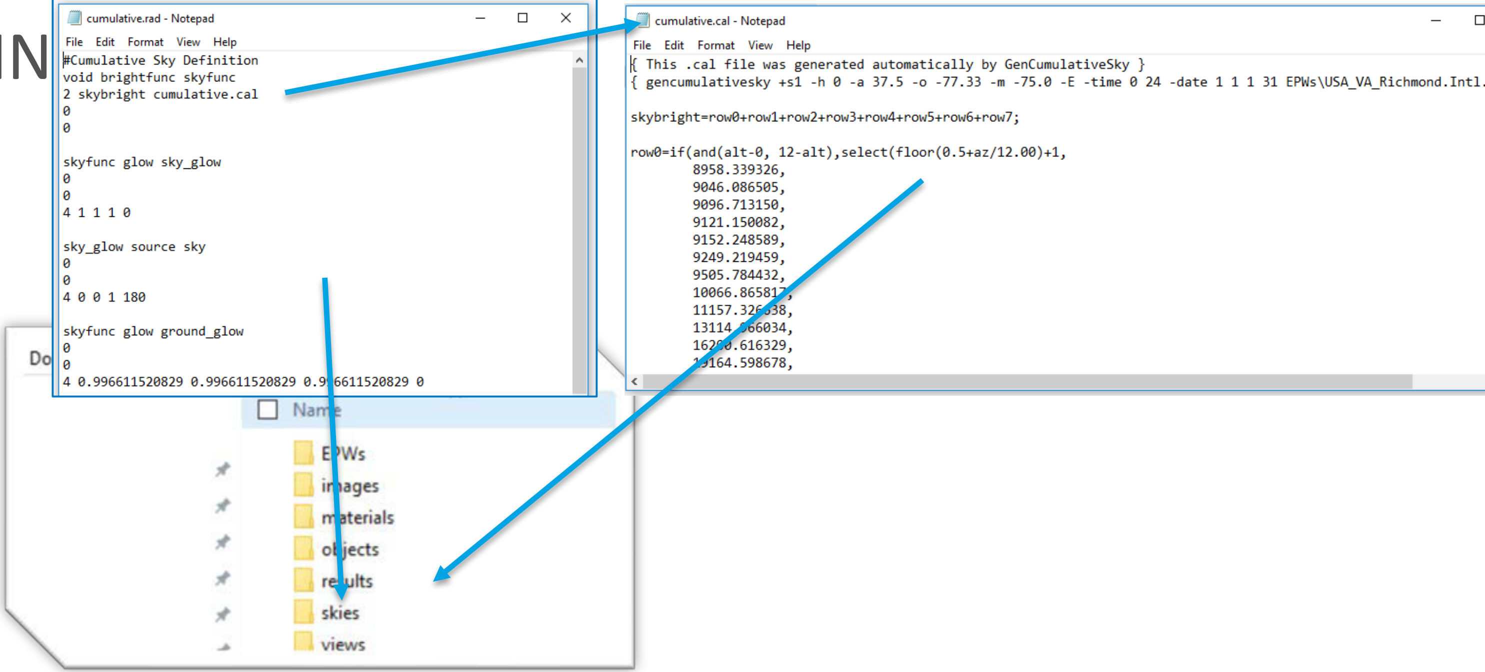
1. Make Radiance Object



MAIN STEPS / IN

1. Make Radiance Object

2. Make Sky

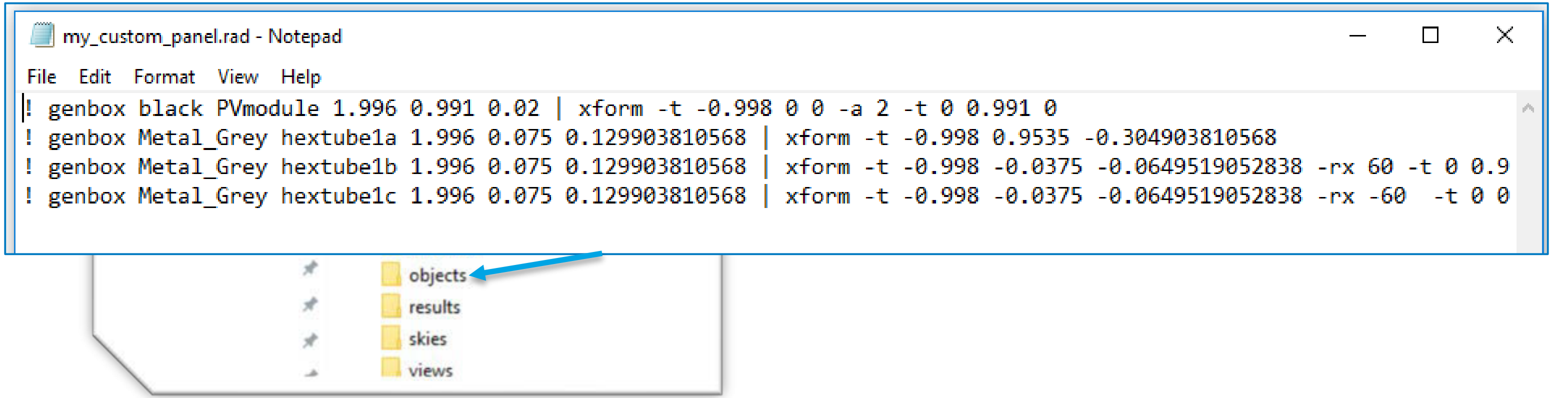


MAIN STEPS / INSTRUCTIONS

1. Make Radiance Object

2. Make Sky

3. Make Module



MAIN STEPS / INSTRUCTIONS

1. Make Radiance Object

2. Make Sky

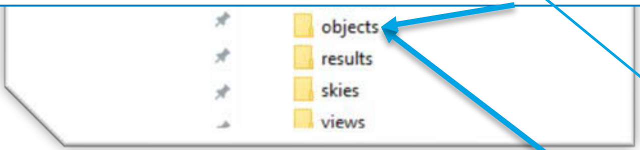
3. Make Module

4. Make Scene

my_custom_panel.rad - Notepad

File Edit Format View Help

```
! genbox black PVmodule 1.996 0.991 0.02 | xform -t -0.998 0 0 -a 2 -t 0 0.991 0
! genbox Metal_Grey hextube1a 1.996 0.075 0.129903810568 | xform -t -0.998 0.9535 -0.304903810568
! genbox Metal_Grey hextube1b 1.996 0.075 0.129903810568 | xform -t -0.998 -0.0375 -0.0649519052838 -rx 60 -t 0 0.9
! genbox Metal_Grey hextube1c 1.996 0.075 0.129903810568 | xform -t -0.998 -0.0375 -0.0649519052838 -rx -60  -t 0 0
```



my_custom_panel_2.35_12.09_20x7.rad - Notepad

File Edit Format View Help

```
!xform -rx 10 -t 0 0 2.35 -a 20 -t 2.006 0 0 -a 7 -t 0 12.097 0 -i 1
 -t -19.96 -36.291 0 -rz 0 objects\my_custom_panel.rad
```

MAIN STEPS / INSTRUCTIONS

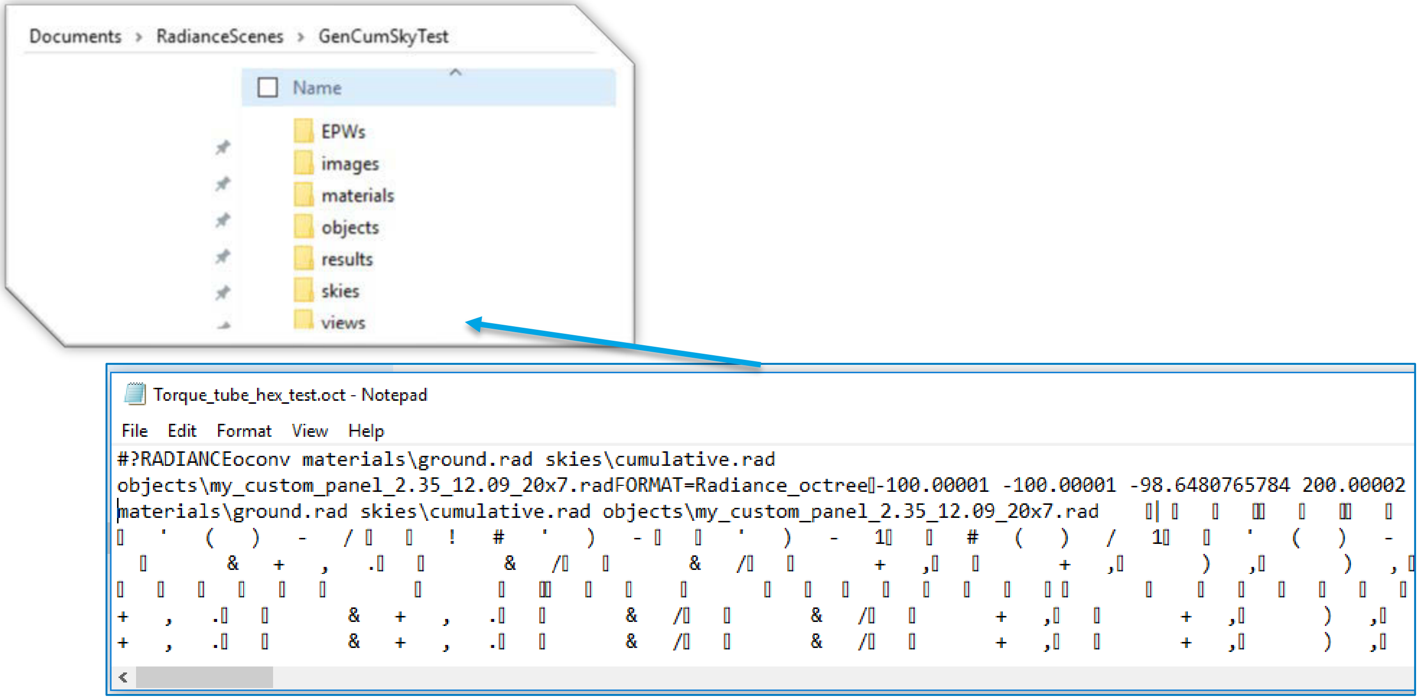
1. Make Radiance Object

2. Make Sky

3. Make Module

4. Make Scene

5. Make Oct



MAIN STEPS / IN

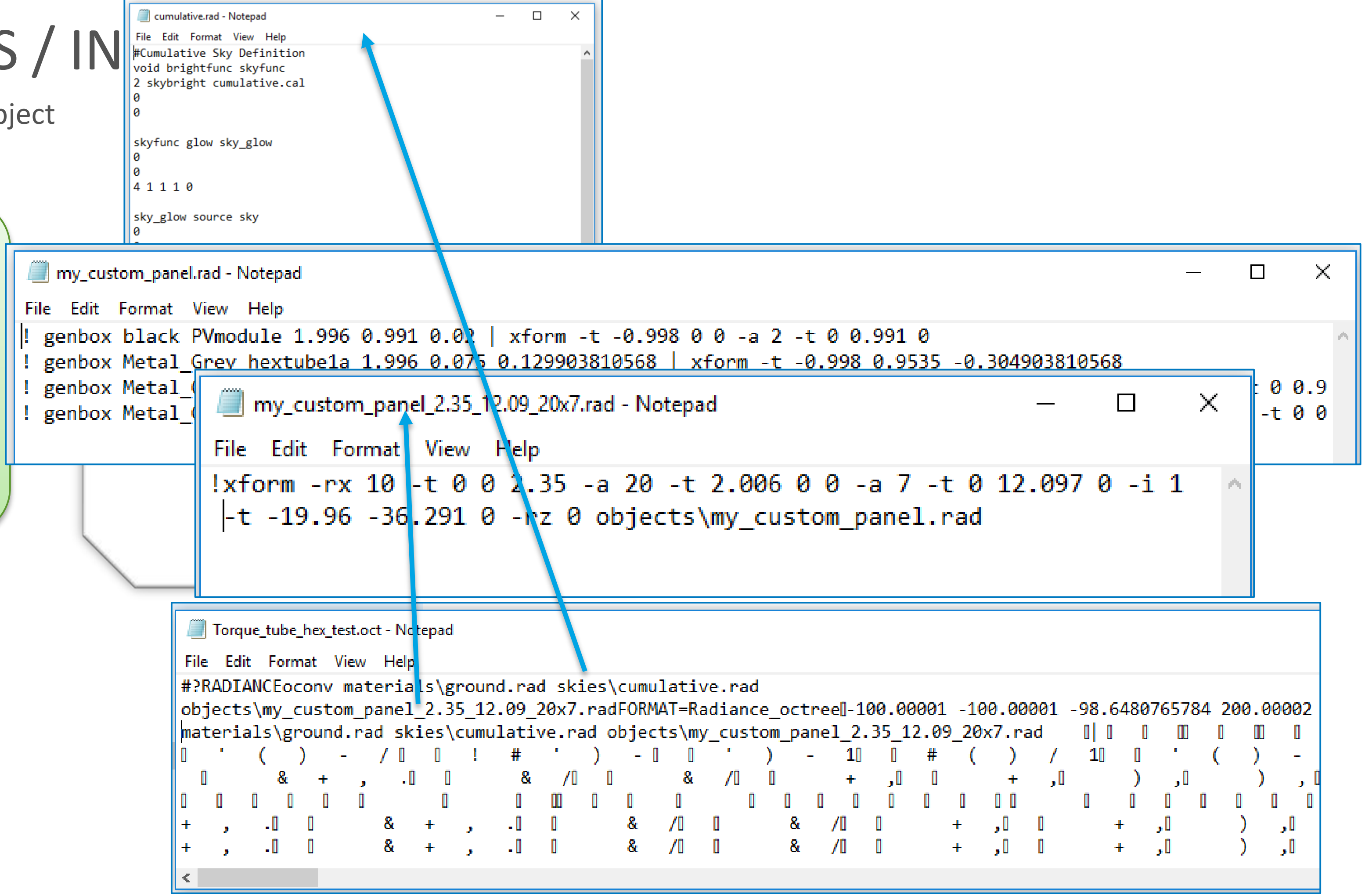
1. Make Radiance Object

2. Make Sky

3. Make Module

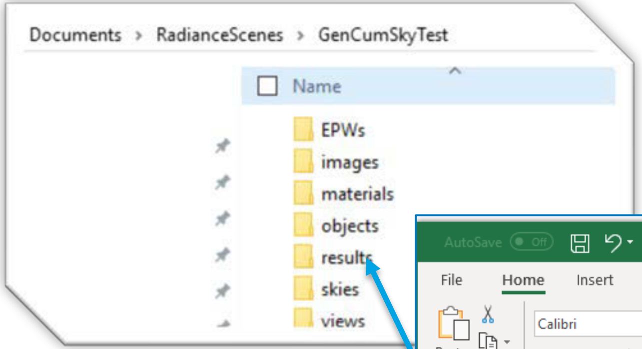
4. Make Scene

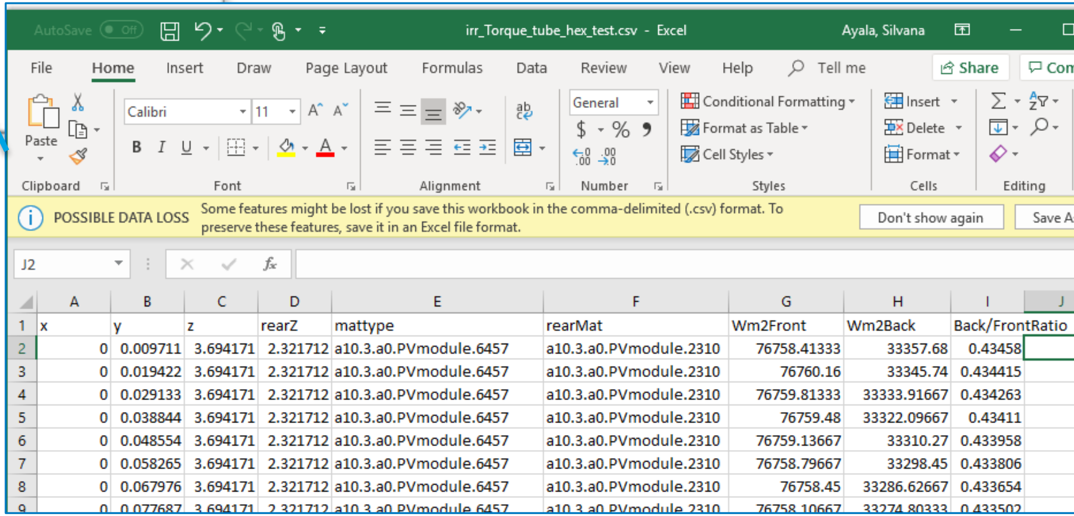
5. Make Oct



MAIN STEPS / INSTRUCTIONS

1. Make Radiance Object
2. Make Sky
3. Make Module
4. Make Scene
5. Make Oct
6. Analysis Obj
7. Analysis





MAIN STEPS / INSTRUCTIONS

1. Make Radiance Object

2. Make Sky

cmd gencumsky

3. Make Module

4. Make Scene

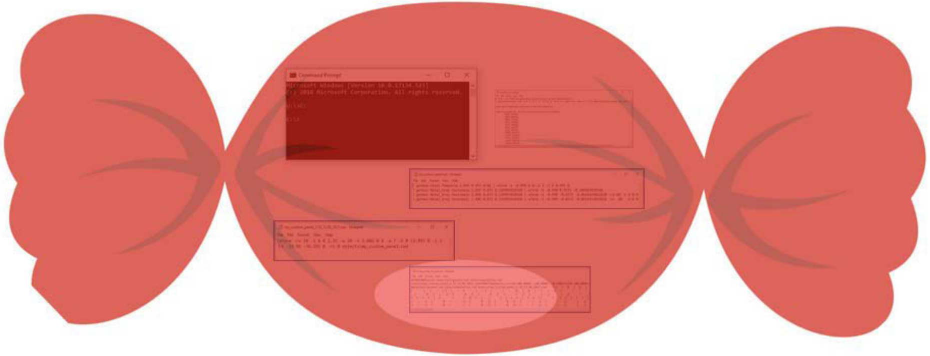
5. Make Oct

cmd oconv

6. Analysis Obj

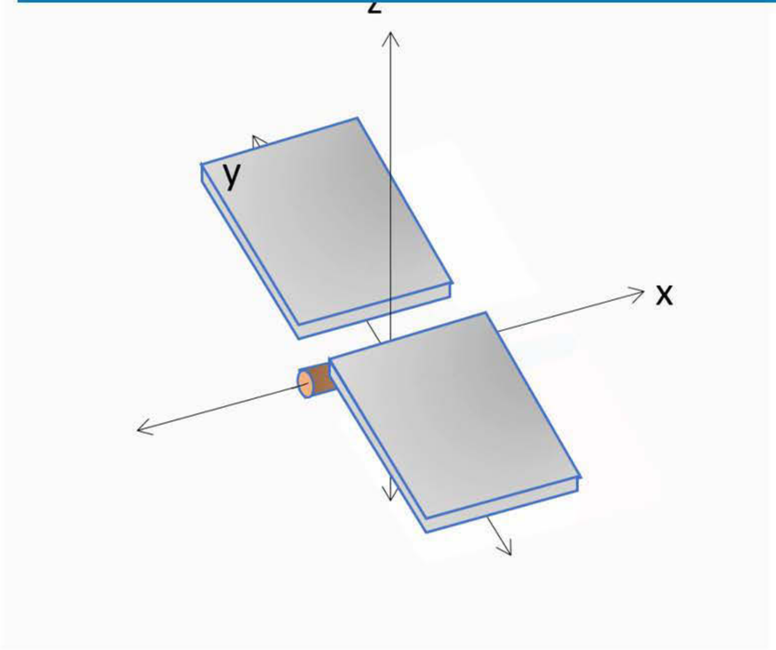
7. Analysis

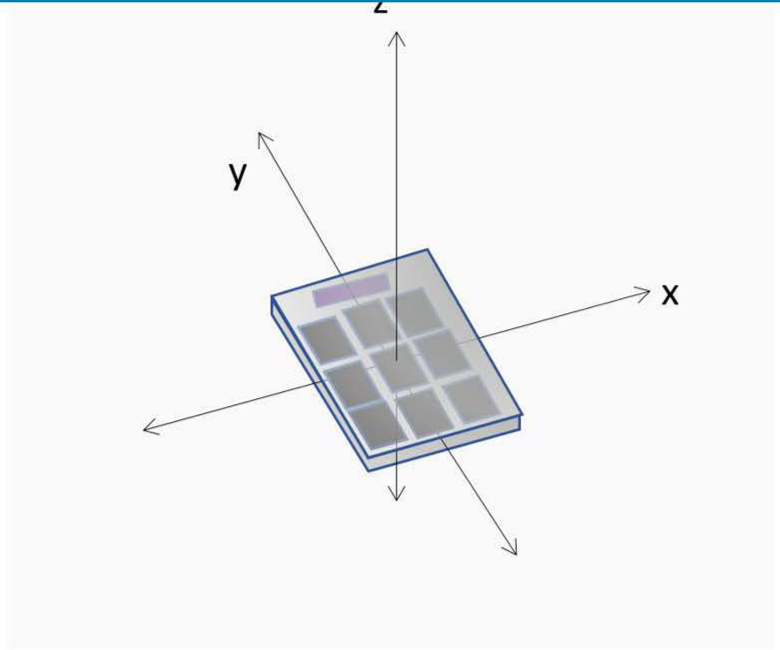
cmd rtrace



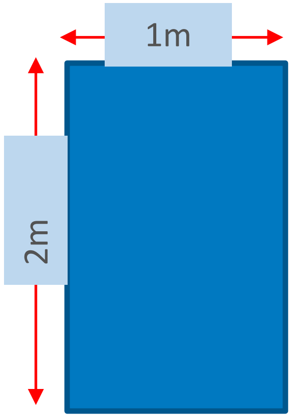
Geometry

makeModule: “module unit” (module, cells, torque tube, etc) to replicate

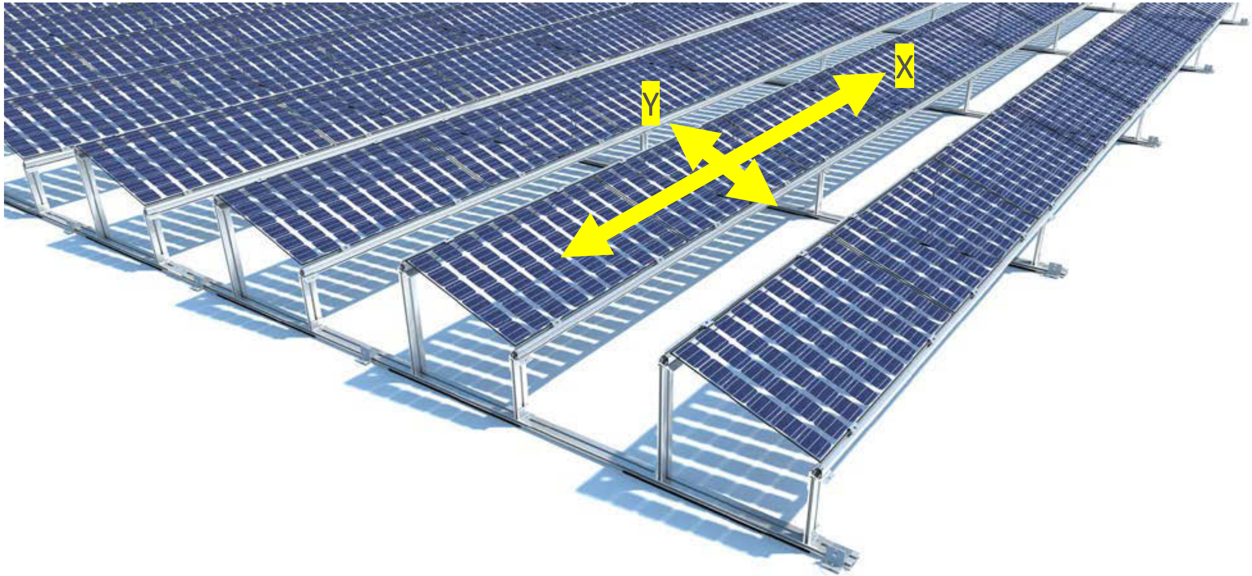




32

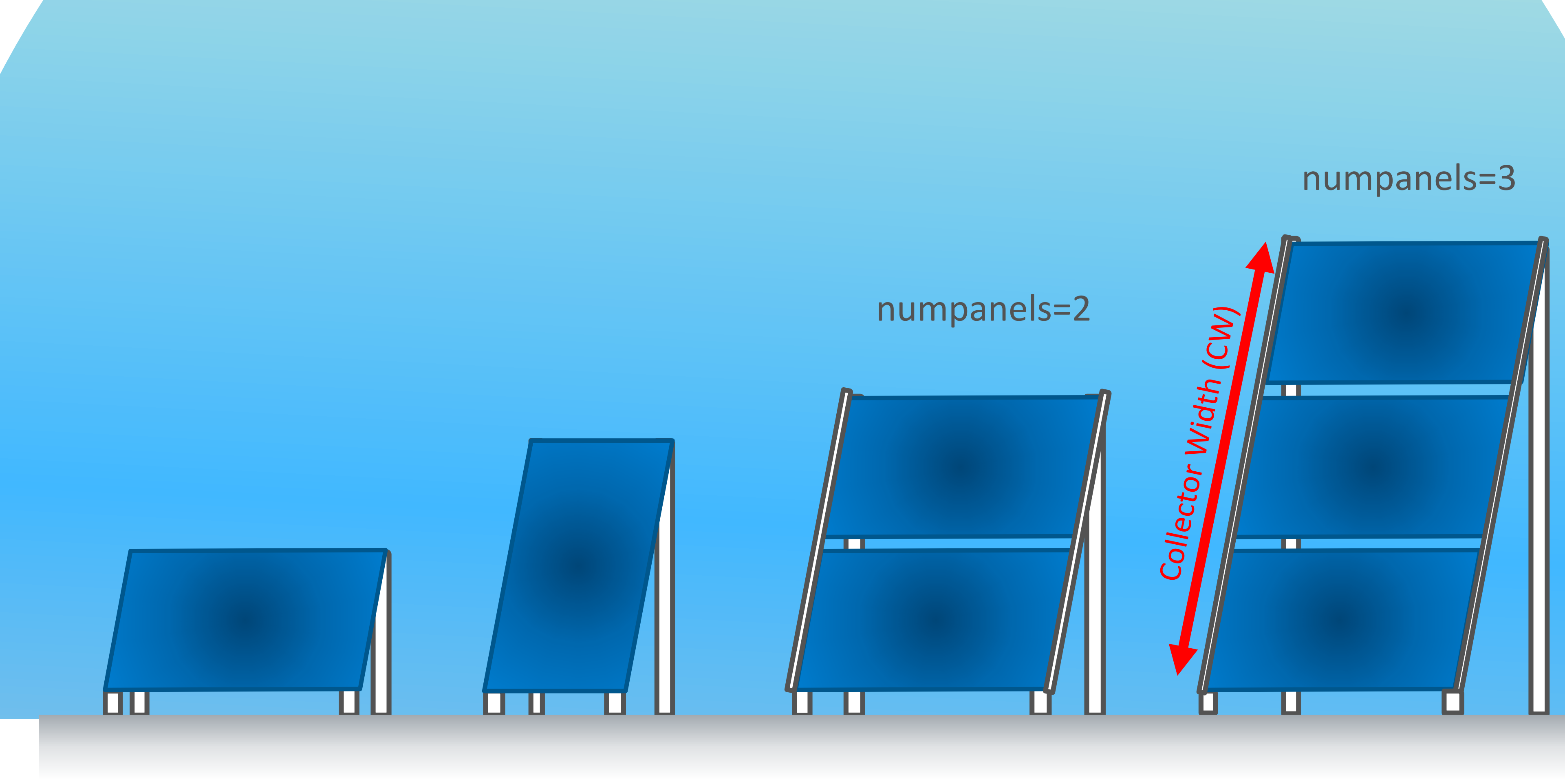


Assign x and y depending on which value should go along the axis / slope



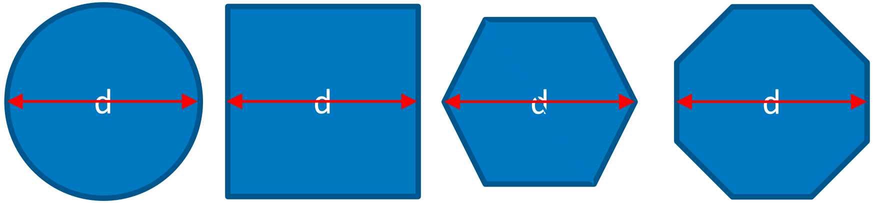
For example:
'PORTRAIT'
y = 2, x = 1

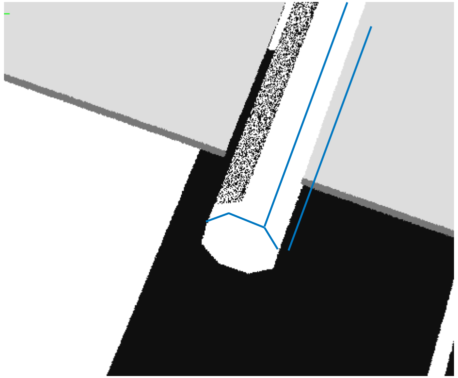
LANDSCAPE
y = 1, x = 2


numpanels=3
numpanels=2
Collector Width (CW)

Torque Tube

All based on "diameter"



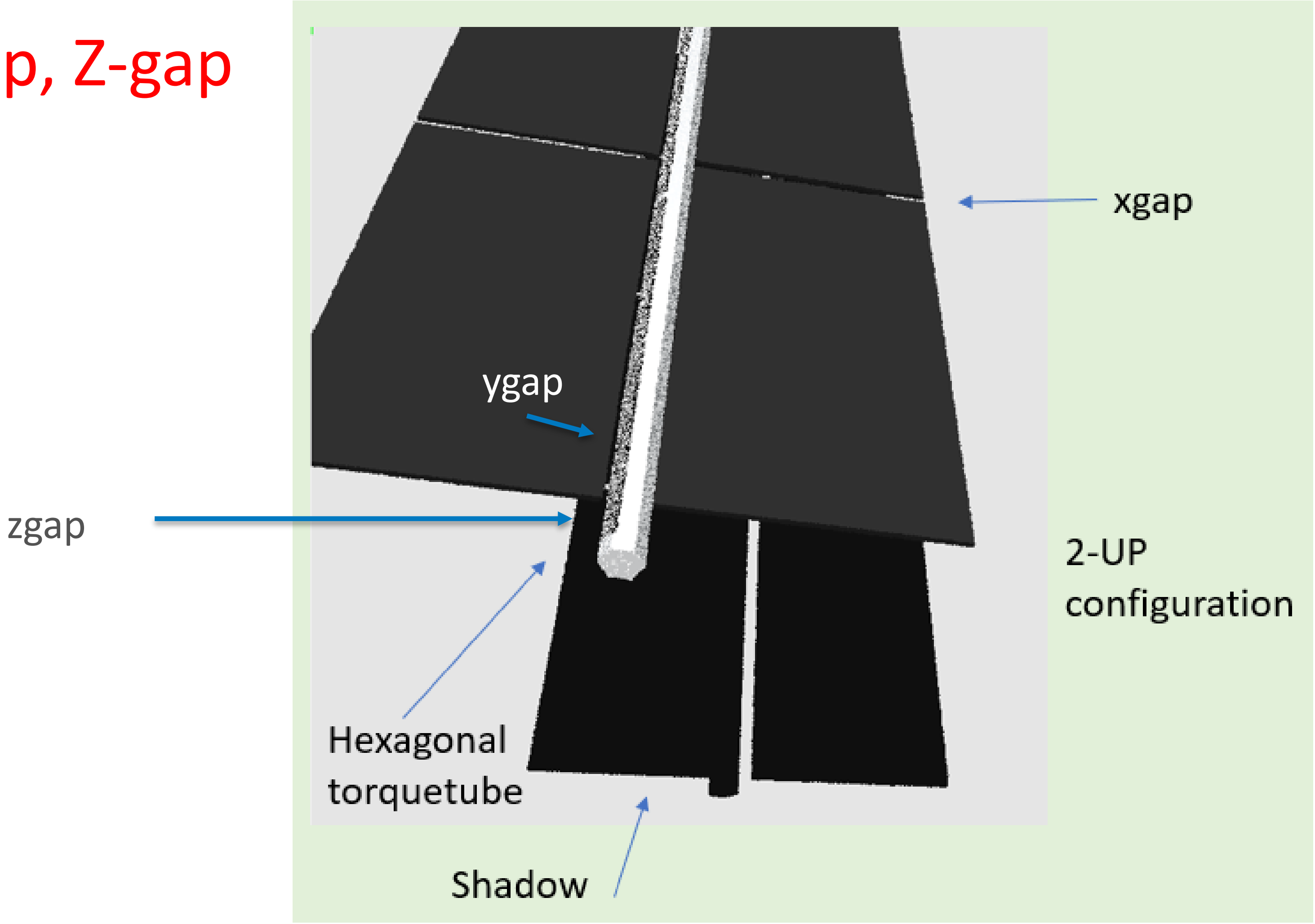


Modifiable material

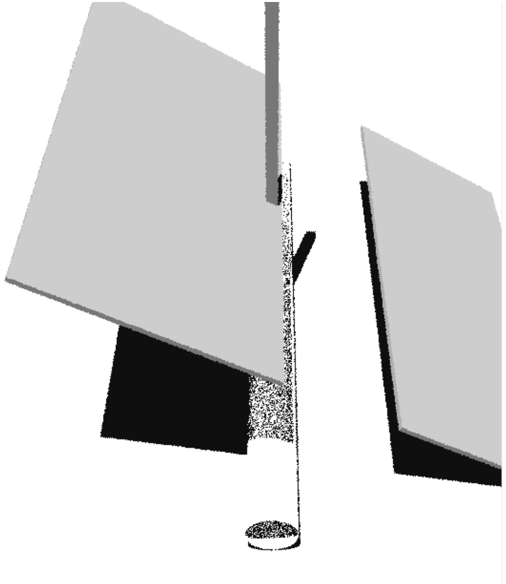
Metal_Grey (GREY NOT GRAY) → 44% reflectivity.

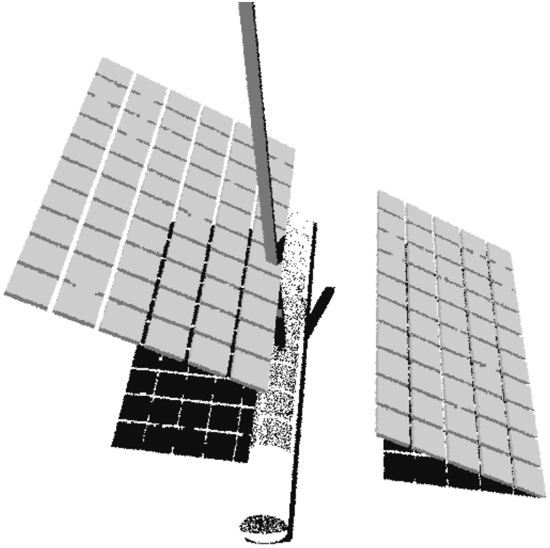
Black → Absorptive.

X-gap, Y-gap, Z-gap



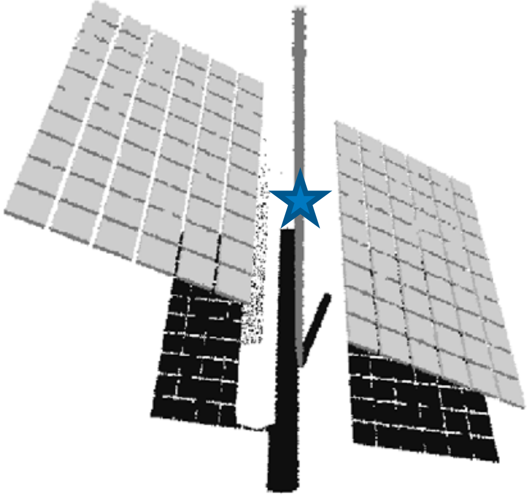
Cell Level Module



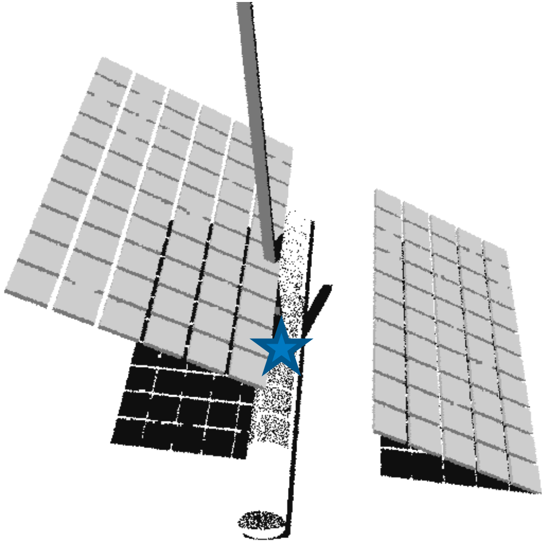


Axis of rotation Torquetube

- FALSE

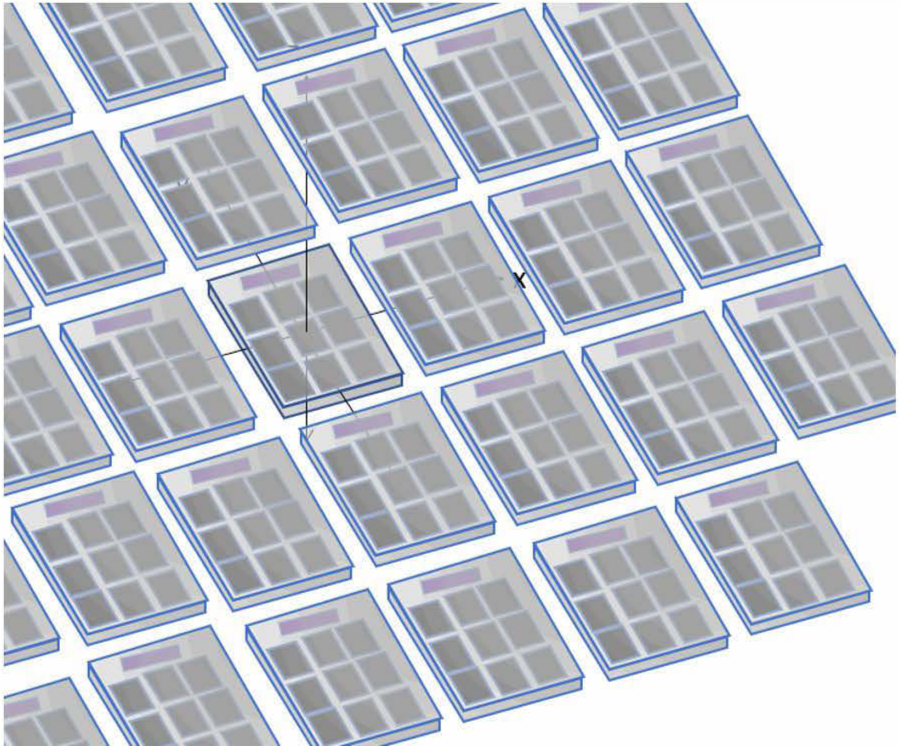


- TRUE



Offset from axis of rotation = zgap + diam/2

makeScene: replicates whatever was defined as the module unit to make the Scene



tilt

clearance height

nRows

Rtr

place center module in 0 (self. scenex * nMods/2)

Azimuth rotation

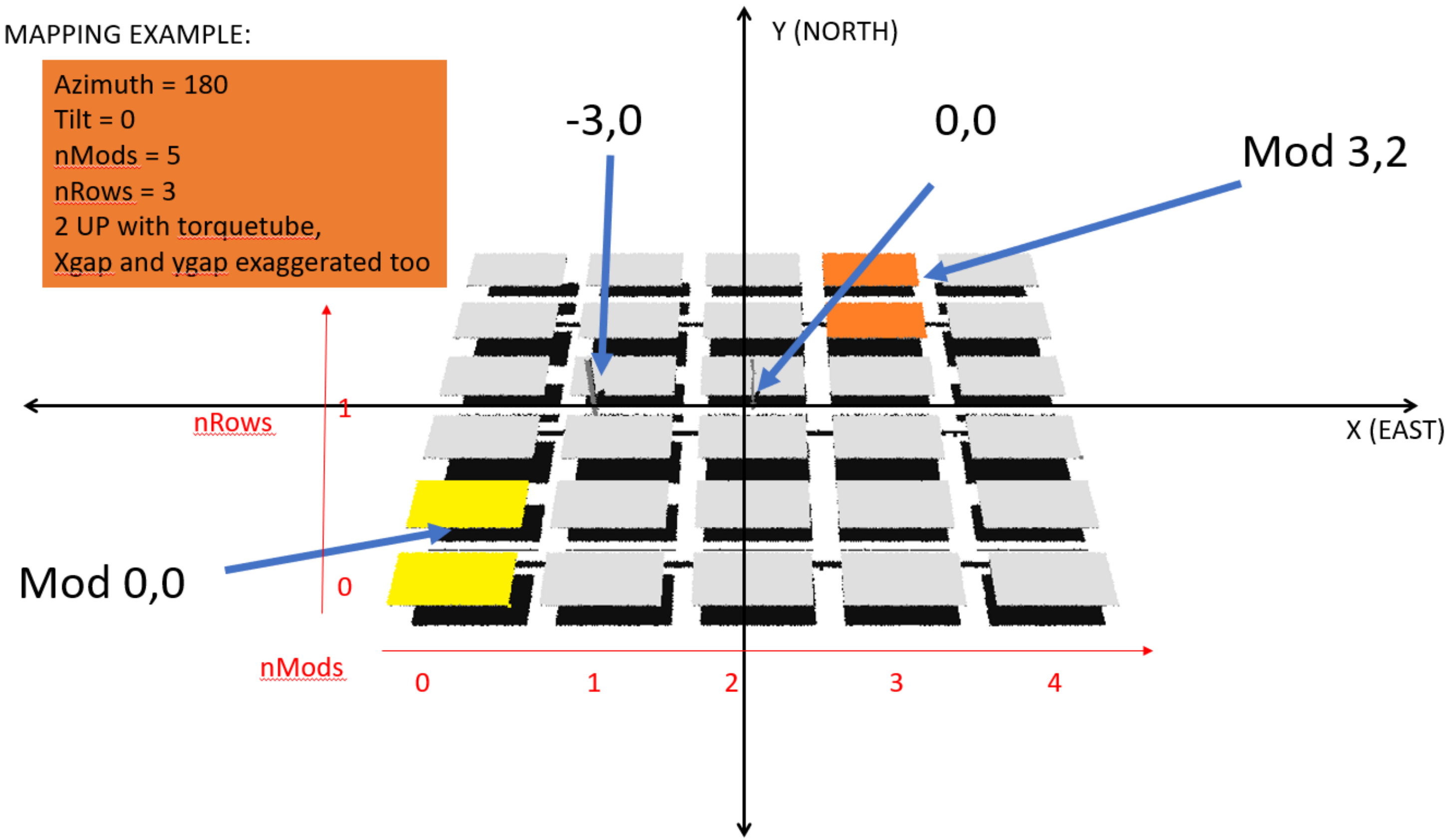
```
!xform -rx 10 -t 0 0 2.35 -a 4 -t 2.046 0 0 -a 1 -t 0 12.552 0 -i 1 -t -4.092 -0.0 0 -rz 90 objects\Panel1.rad
```

nMods

self.scenex

Place center row on 0

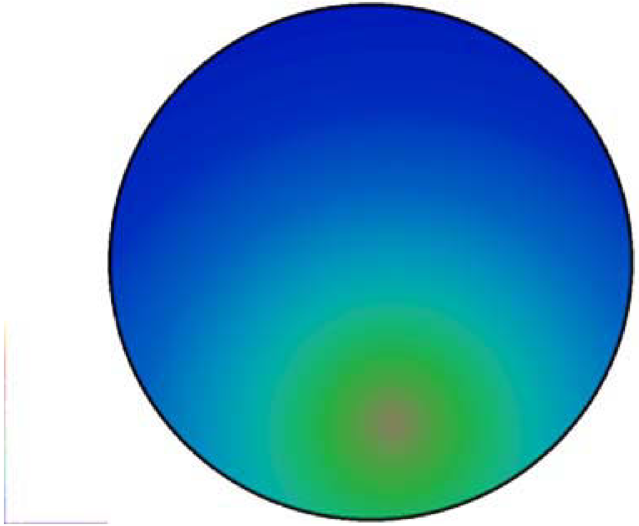
Object to Replicate


MAPPING EXAMPLE:
Azimuth = 180
Tilt = 0
nMods = 5
nRows = 3
2 UP with torquetube,
Xgap and ygap exaggerated too
Y (NORTH)
X (EAST)
-3,0
0,0
Mod 3,2
Mod 0,0
nRows
1
0
nMods
0
1
2
3
4

Sky

Hourly Sky

“Gendaylit”
Hourly distribution
Perez Model
Must know: sun position, DNI, DHI



Pvlib.sun azimuth-180

!gendaylit -ang 29.44 -3.3976 –W 476 149 - g 0.62 -O 1

Pvlib.sun altitude

DNI DHI

ground albedo

W/m^2-sr solar radiation

Richmond, Virginia EPW, timestamp[11] ---- January 1st at Noon

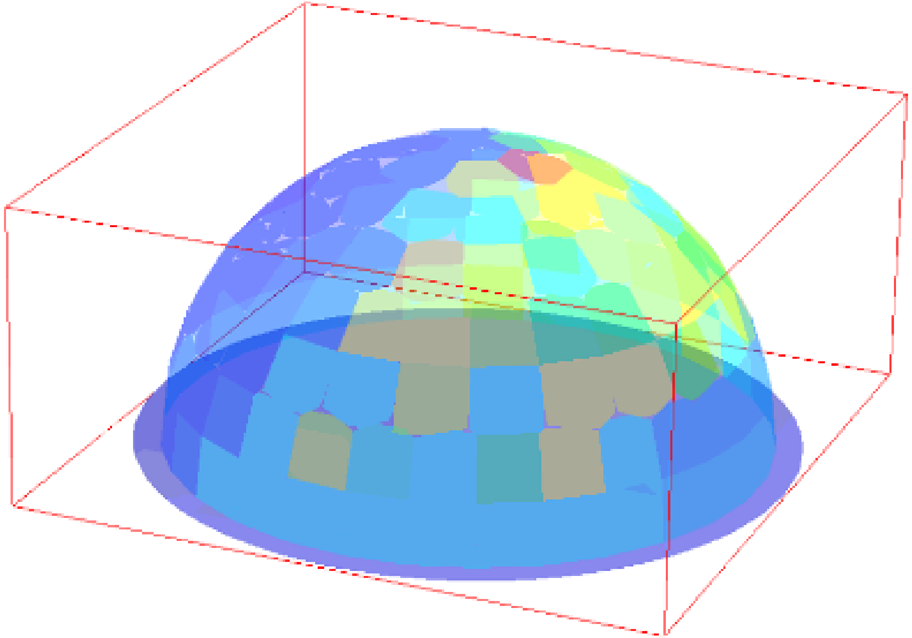
Cumulative sky

"Gencumsky"

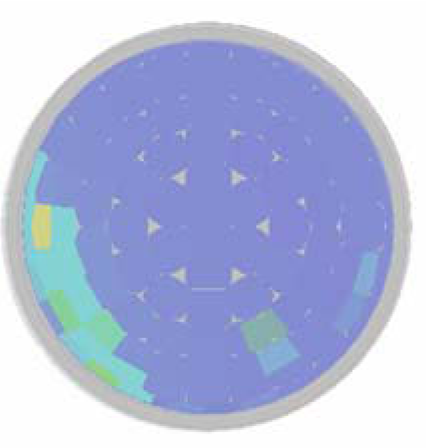
Bins sky into 145 patches of same angular extent

Irradiances for each patch can be cumulative for any period of time.
→ Increase in speed

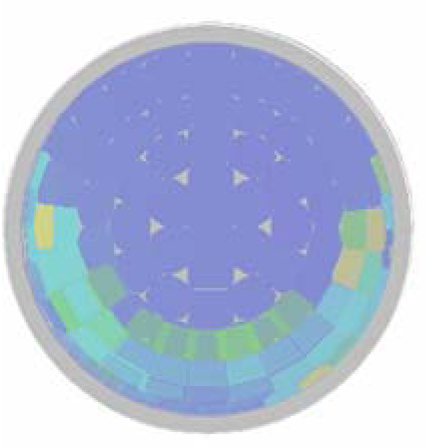
Robinson, Stone "Irradiation modelling made simple: the cumulative sky approach" 2004



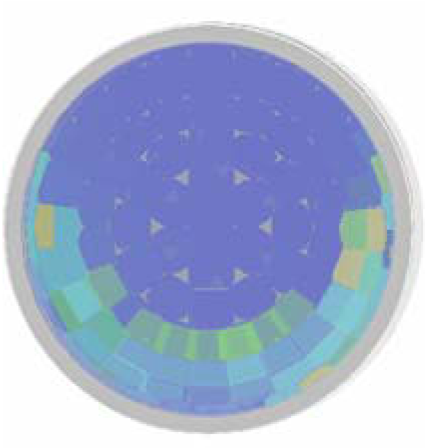
Gencumulative Sky



Simulate Hourly
~4380 simulations



Simulate Daily
~365 simulations



Simulate Monthly
~12 simulations

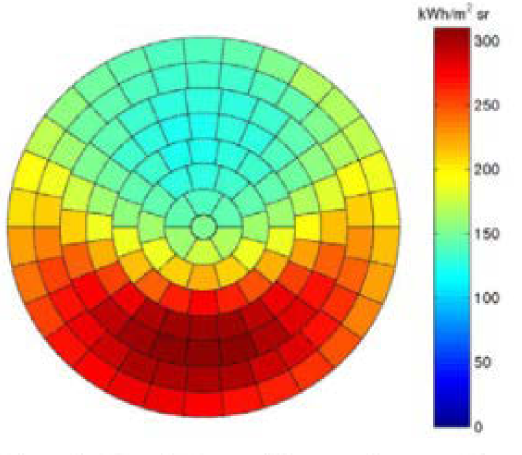
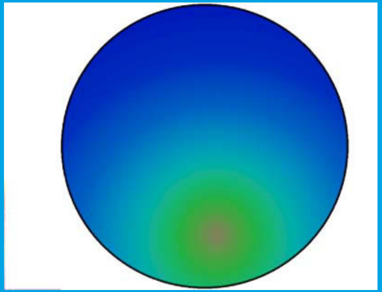


Figure 1 Cumulative diffuse sky radiance distribution for Oslo (based on 10yr mean solar data).

Simulate Yearly
~1 simulations

Gendaylit

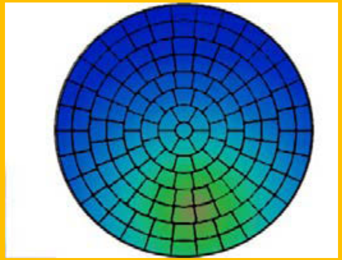


Perez Model

sky2_1axisTrack_3.rad - Notepad

```
# start of sky definition for daylighting studies
# location name: Richmond LAT: 37.5 LON: -77.33 Elev: 54.0
# Sun position calculated w. PVLib
!gendaylit -ang -45.5637656782 -94.442474712 -W 0 0 -g 0.62 -O 1
skyfunc glow sky_mat
0
0
4 1 1 1 0

sky_mat source sky
0
0
4 0 0 1 180

skyfunc glow ground_glow
0
0
4 0.996611520829 0.996611520829 0.996611520829 0
```

GencumSky



Robinson and Stone

cumulative.rad - Notepad

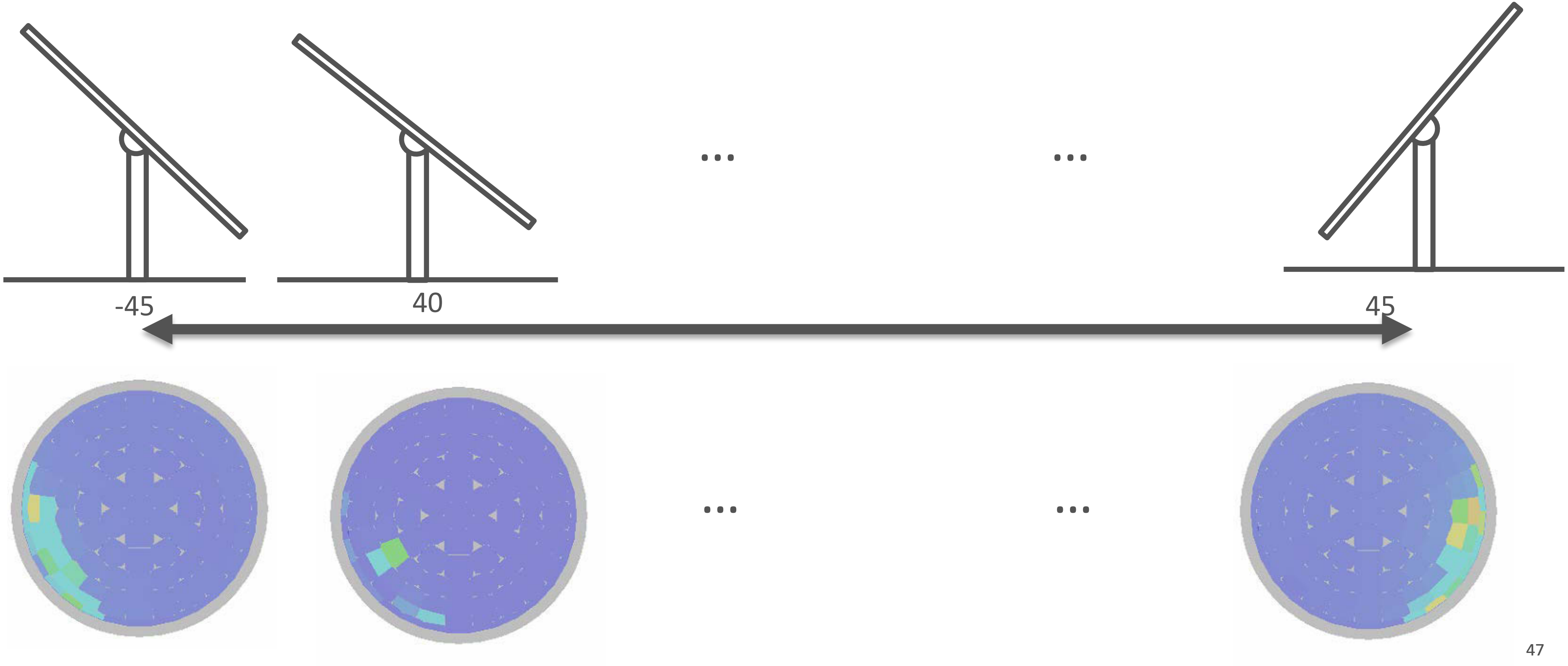
```
#Cumulative Sky Definition
void brightfunc skyfunc
2 skybright cumulative.cal
0
0

skyfunc glow sky_glow
0
0
4 1 1 1 0

sky_glow source sky
0
0
4 0 0 1 180

skyfunc glow ground_glow
0
```

cumulative.cal - Notepad

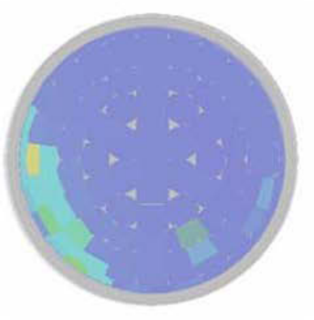
```
{ This .cal file was generated automatically by GenCumulativeSky }
{ gencumulativesky +s1 -h 0 -a 37.5 -o -77.33 -m -75.0 -E -time 0 24 -date 1 1 1 31 EPWs\USA_VA_Ric

skybright=row0+row1+row2+row3+row4+row5+row6+row7;

row0=if(and(alt-0, 12-alt),select(floor(0.5+az/12.00)+1,
        8958.339326,
        9046.086505,
        9096.713150,
        9121.150082,
        9152.248589,
        9249.219459,
        9505.784432,
        10066.865817,
        11157.326638,
        13114.966034,
        16200.616329,
```

Cummulative Sky by Tracker Angle

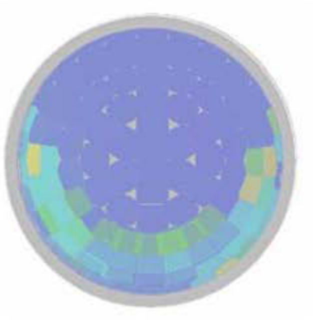
S. Ayala Pelaez, C. Deline, P. Greenberg, J. S. Stein, and R. K. Kostuk, “Model and Validation of Single-Axis Tracking with Bifacial PV - Preprint,” IEEE Journal of Photovoltaics, 2019, vol 9 no. 3, pp. 715-721.



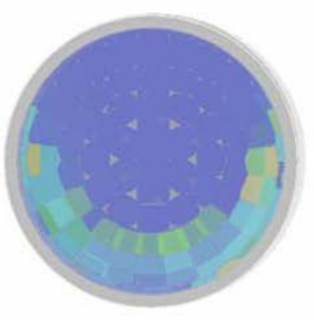
Cummulative Skies



Hourly
~4380 simulations



Accumulate Daily
~365 simulations



Accumulate Monthly
~12 simulations

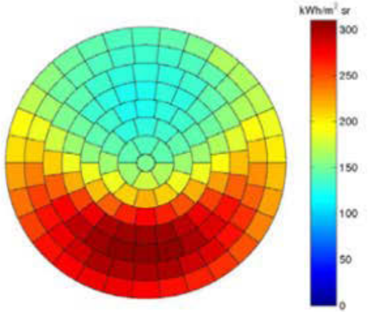
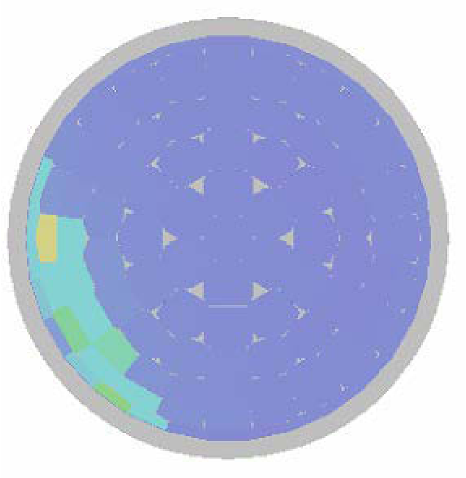


Figure 1 Cumulative diffuse sky radiance distribution for Oslo (based on 10yr mean solar data).

Accumulate Yearly
~1 simulations



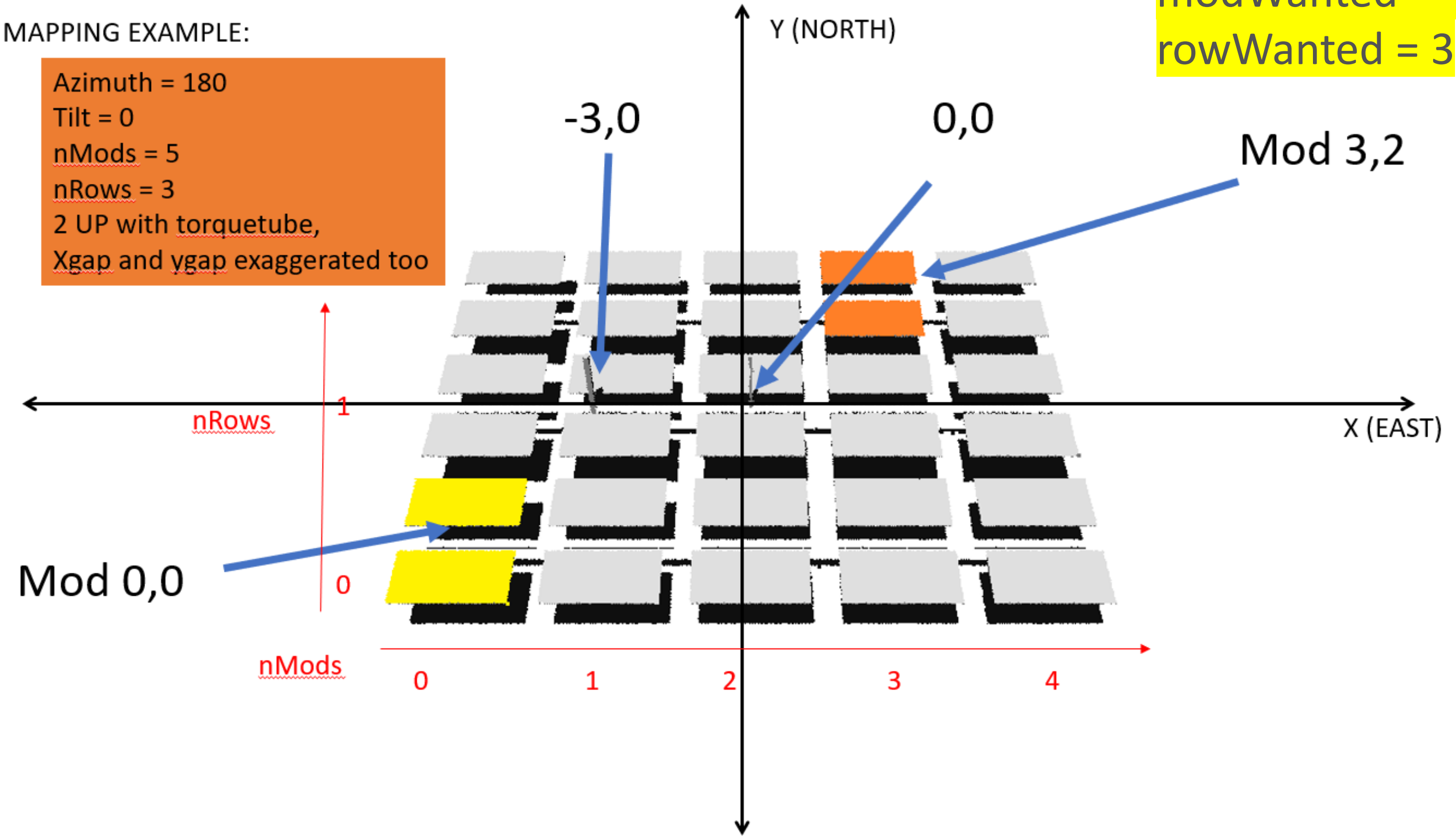
Tracking
Acccumulate by tracker angle
-45 to 45: ~19 simulations

Analysis Objects

MAPPING EXAMPLE:

Azimuth = 180
Tilt = 0
nMods = 5
nRows = 3
2 UP with torquetube,
Xgap and ygap exaggerated too

modWanted = 4
rowWanted = 3

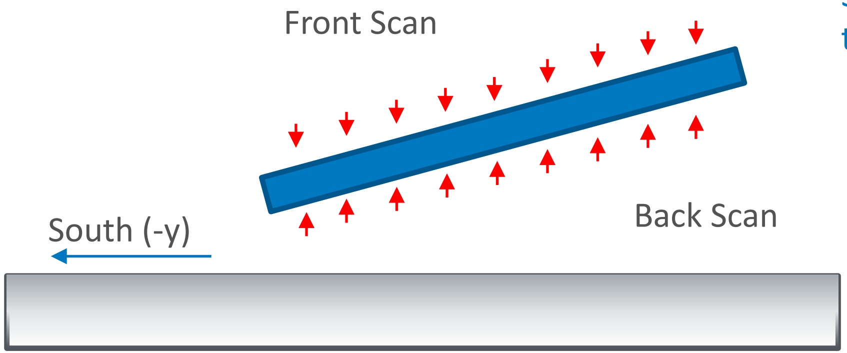


sensorsy

Sensors are spaced equidistant to each other
No ‘safety distance’ from edges.

The detectors measure irradiance at the first surface they ‘see’.

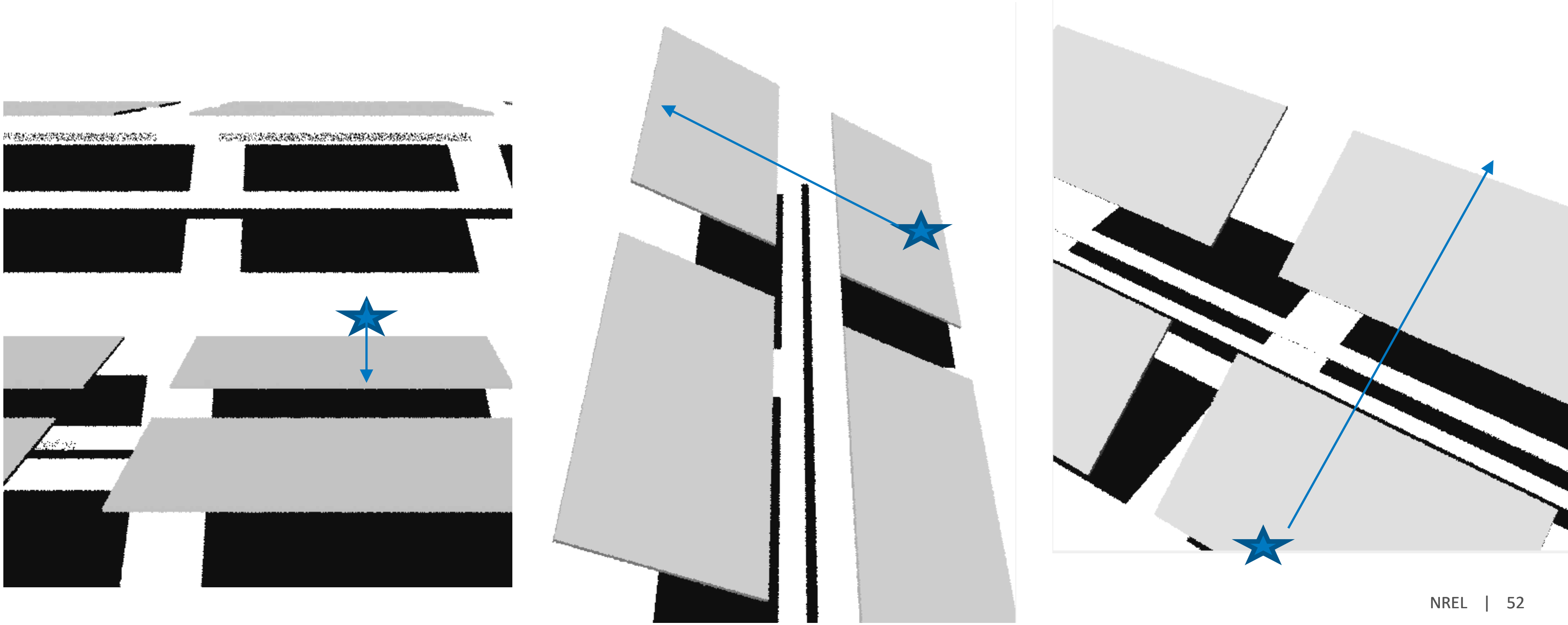
Sensors are placed close to the surfaces to avoid hitting torque tubes.



Complex geometries might require “Cleaning Up” the results

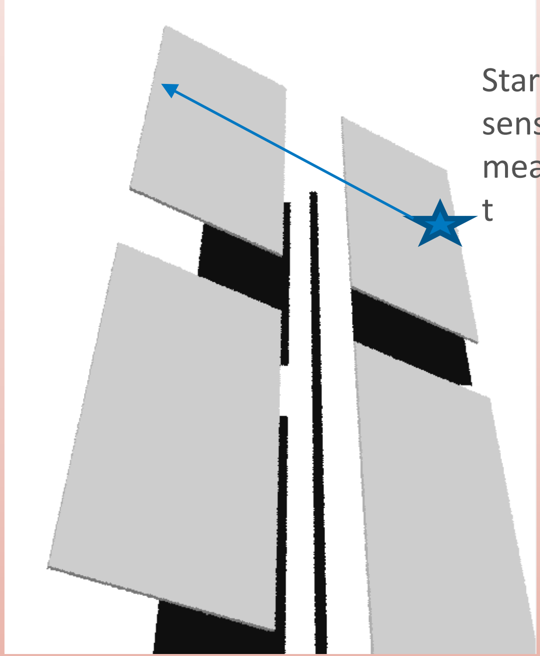
Azimuth

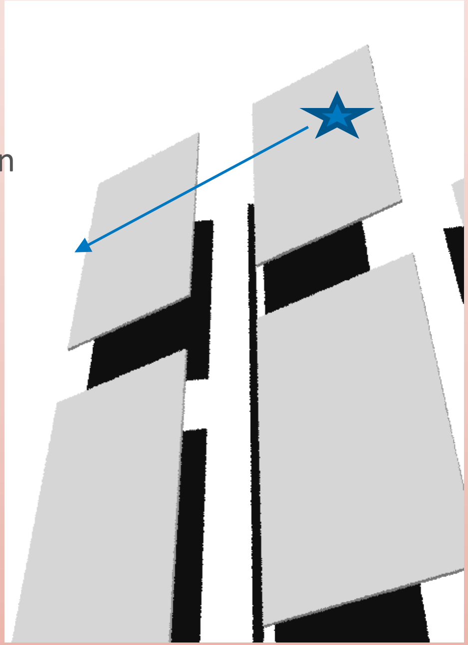
0 | 90 | 210



Tracking Sensor Position

Tracking: same azimuth_axis, tilts are negative for the afternoon.
Sensor locations fall on the same place in the module!





Bifacial_radiance Training Part 1

Github
Tutorials
Jupyter Notebooks

Website to store and manage code, track and control changes and new versions.

Open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text.

https://github.com/NREL/bifacial_radiance/

Example starting-guide: https://realpython.com/jupyter-notebook-introduction/

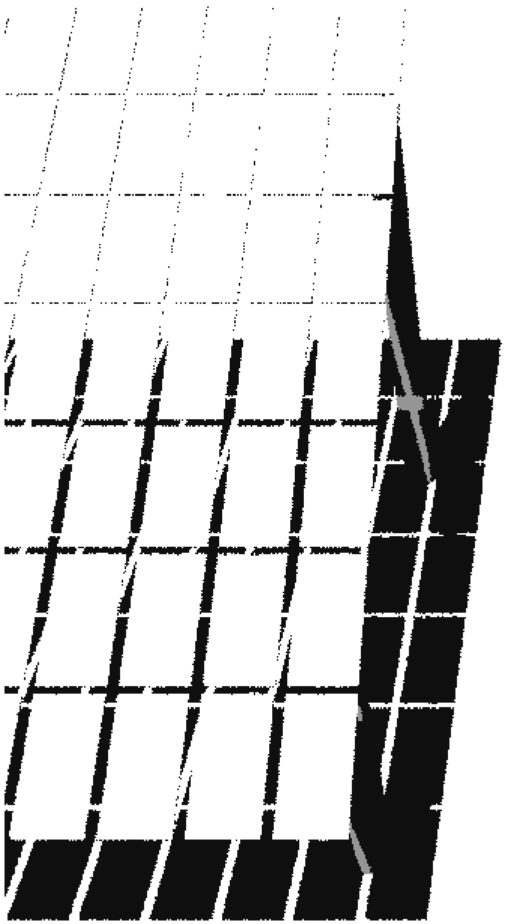
Journals

- 1 - Introductory Example - Fixed Tilt simple setup.ipynb
- 2 - Introductory Example - Single Axis Tracking with cumulative Sky.ipynb
- 3 - Medium Level Example - Single Axis Tracking - hourly.ipynb
- 4 - Medium Level Example - Debugging your Scene with Custom Objects (Fixed Tilt 2-up with Torque Tube + CLEAN Routine + CustomObject).ipynb
- 5 - Medium Level Example - Bifacial Carports and Canopies + sampling across a module!.ipynb
- 6 - Advanced topics - Understanding trackerdict structure.ipynb
- 7 - Advanced topics - Multiple SceneObjects Example.ipynb
- 8 - Advanced topics - Calculating Power Output and Electrical Mismatch.ipynb
- 9 - Advanced topics - 1 axis torque tube Shading for 1 day (Research documentation).ipynb

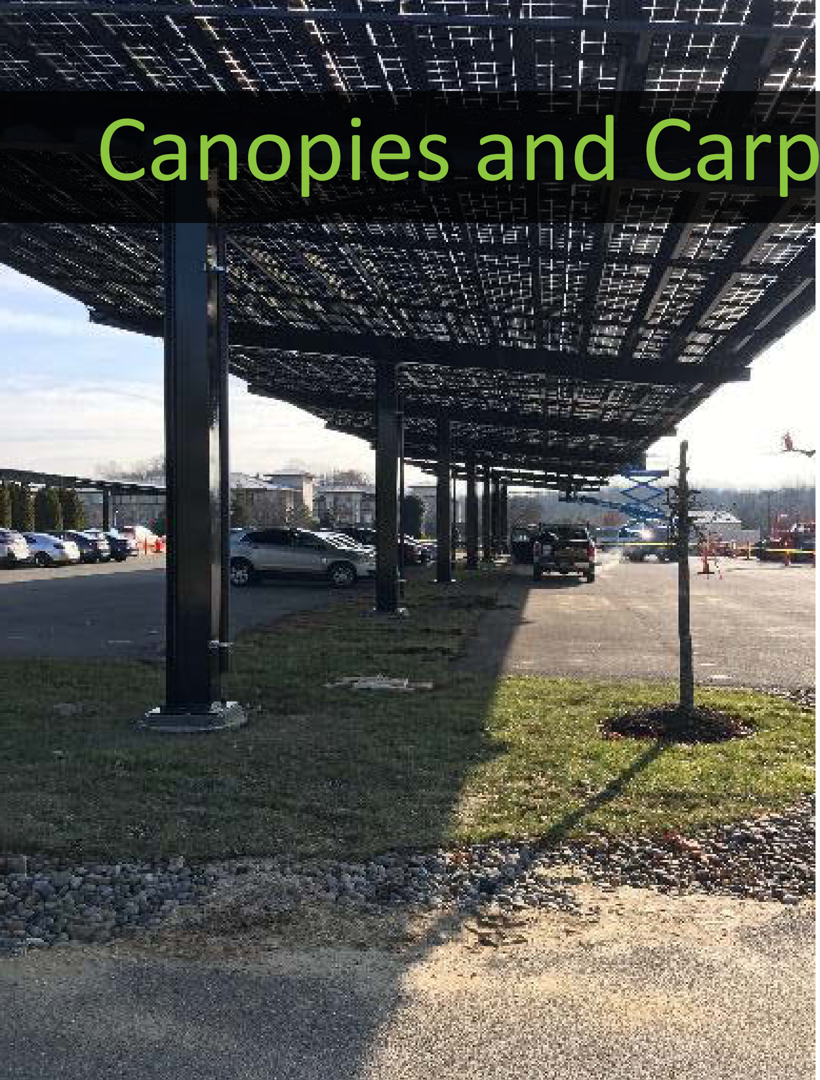
https://github.com/NREL/bifacial_radiance/tree/master/docs/tutorials

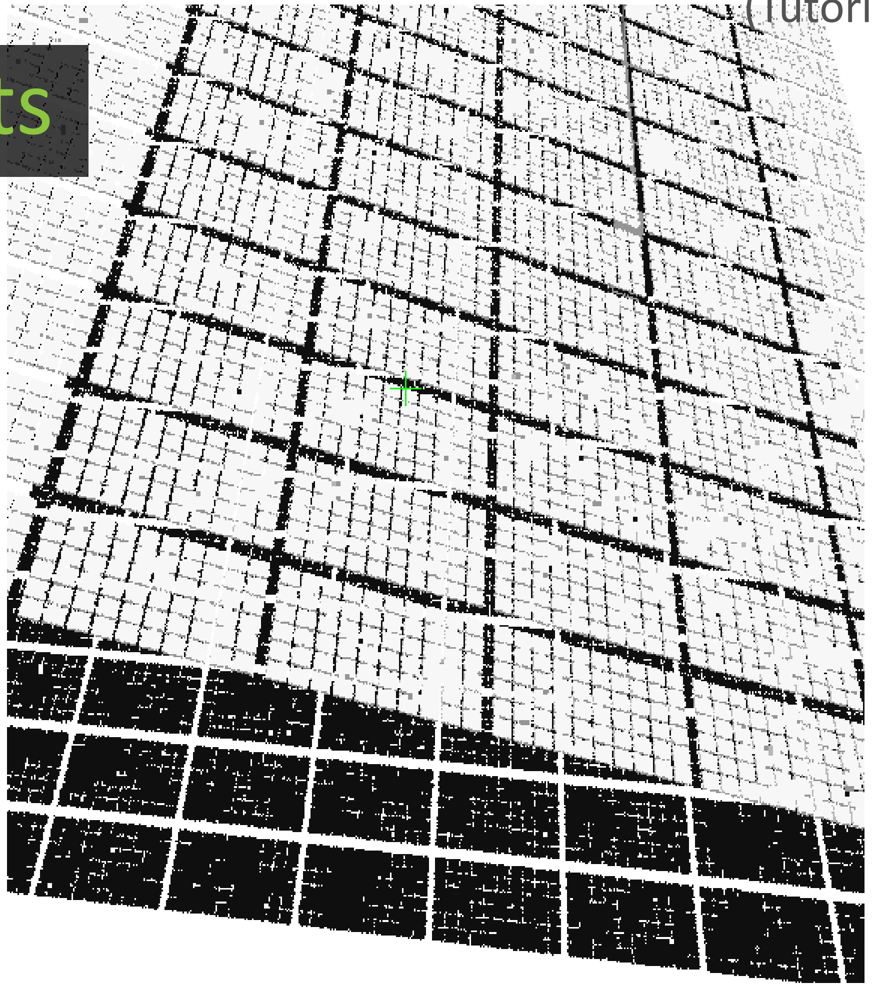
Canopies and Carports

(Tutorial 5)



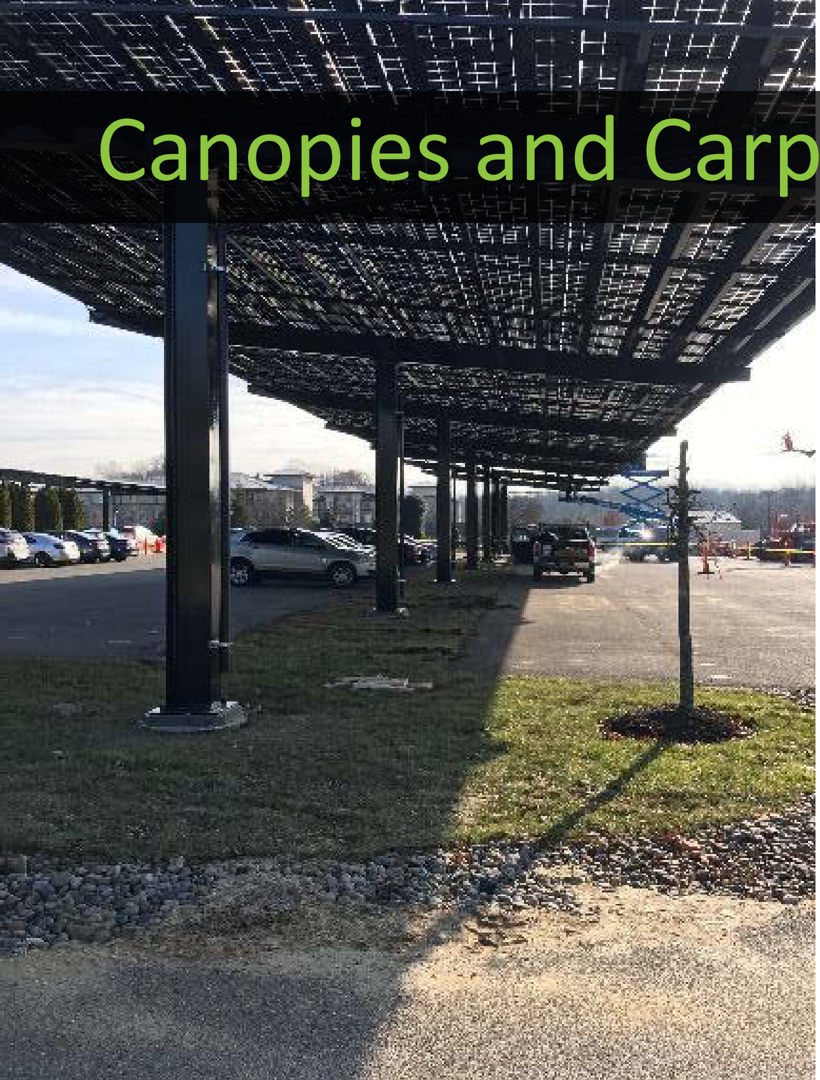
Canopies and Carports





Canopies and Carports

(Tutorial 5)

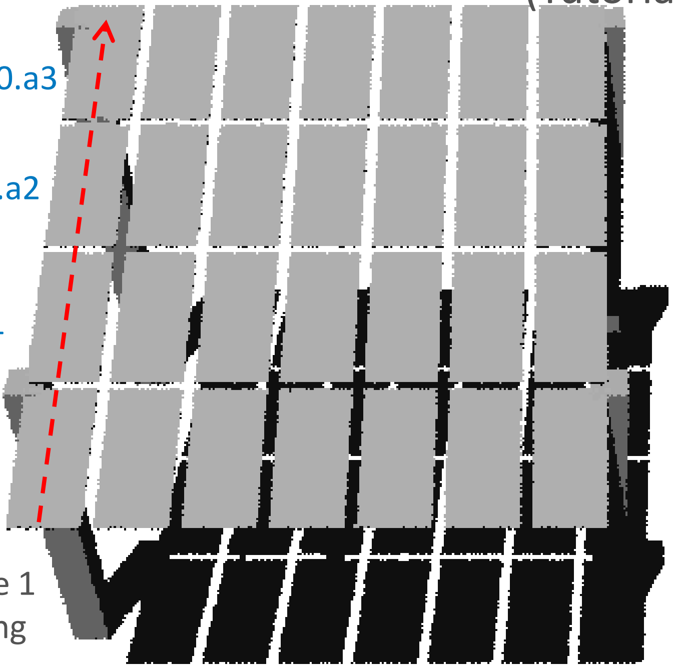


a0.0.a3

a0.0.a2

a0.0.a1

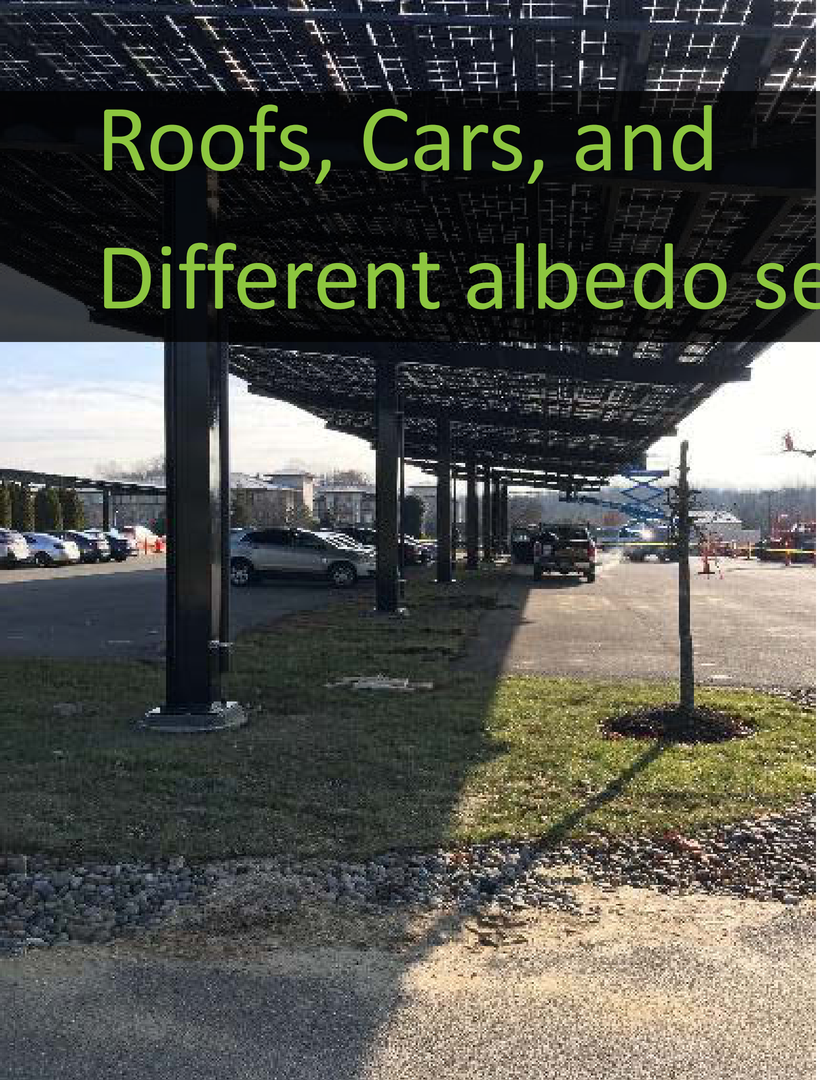
a0.0.a0

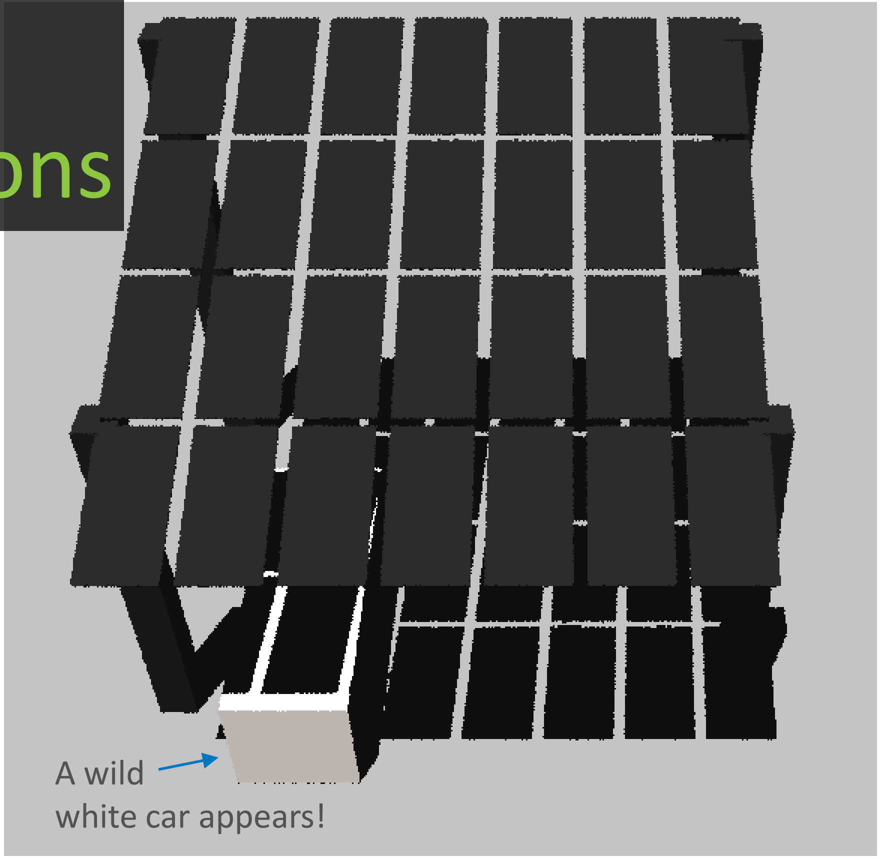


Module 1 Scanning

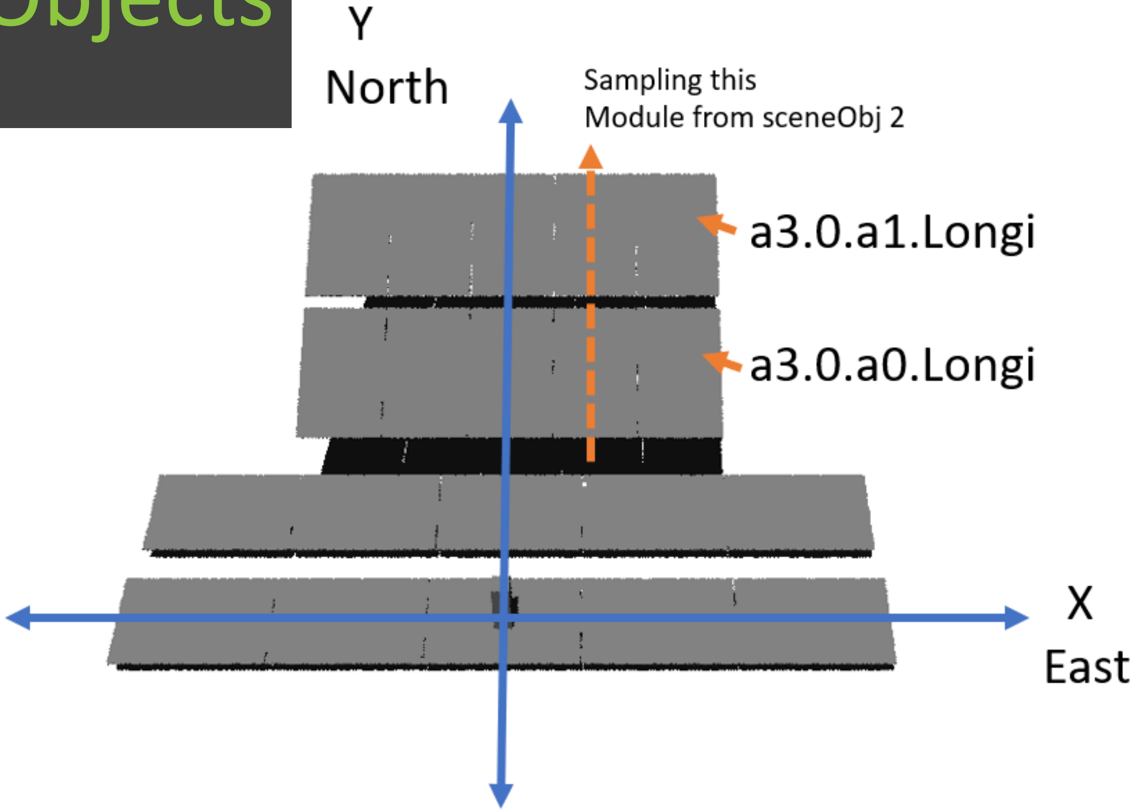
1	irr_HotelCaprortMod1.csv							
2	x	y	z	rearZ	mattype	rearMat	Wm2Fron	Wm2Back
3	-3.3	-3.62011	4.42757	4.33757	a0.0.a0.PrismSolar.6457	a0.0.a0.PrismSolar.2310	787.6552	161.1096
4	-3.3	-3.43446	4.49514	4.40514	a0.0.a0.PrismSolar.6457	a0.0.a0.PrismSolar.2310	787.6781	158.4337
39	...	...	...	...	...	...	...	...
40	...	...	...	...	...	...	...	...
41	-3.3	3.063169	6.860084	6.770084	a0.0.a3.PrismSolar.6457	a0.0.a3.PrismSolar.2310	787.4609	139.0144
42	-3.3	3.248815	6.927654	6.837654	a0.0.a3.PrismSolar.6457	a0.0.a3.PrismSolar.2310	787.4696	135.2156
43	-3.3	3.434462	6.995223	6.905223	a0.0.a3.PrismSolar.6457	a0.0.a3.PrismSolar.2310	787.4783	132.7424
44	-3.3	3.620109	7.062793	6.972793	a0.0.a3.PrismSolar.6457	a0.0.a3.PrismSolar.2310	787.4871	129.142

Roofs, Cars, and Different albedo sections





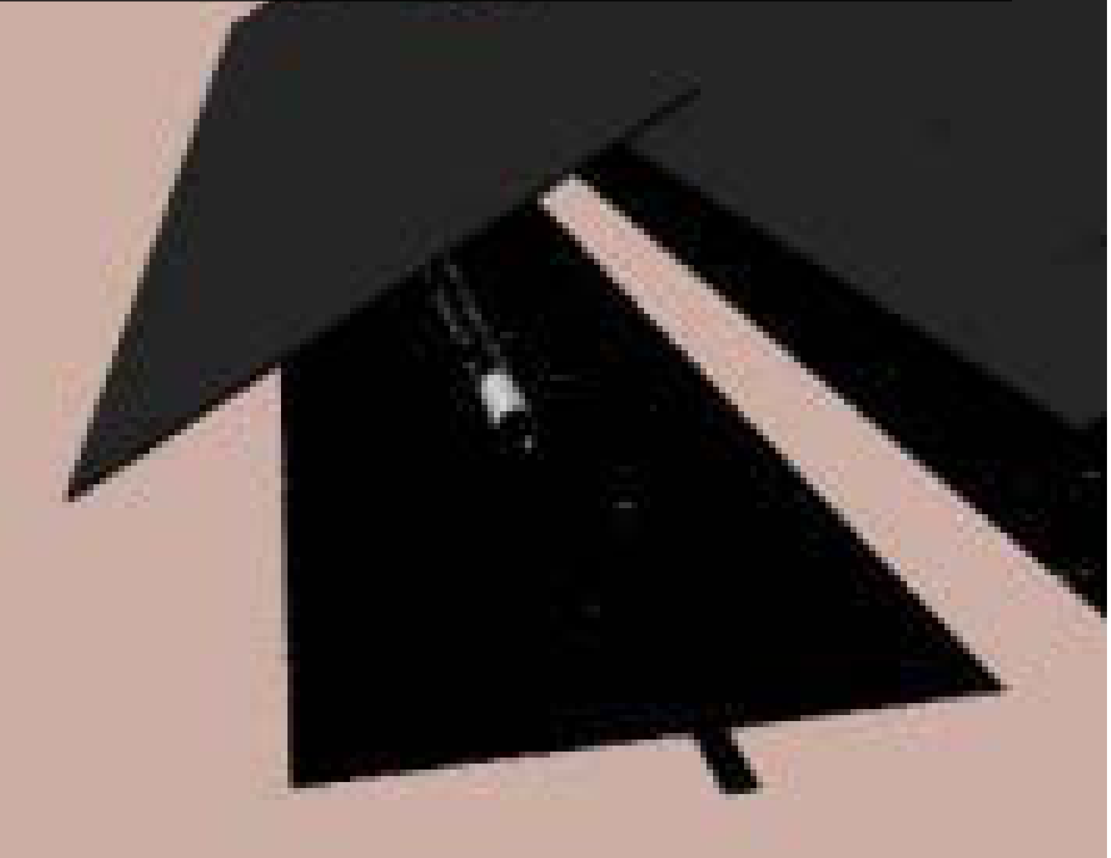
Multiple Scene Objects



https://medium.com/@MySunIndia/limited-time-opportunity-to-get-huge-subsidy-for-rooftop-solar-in-india-for-residential-users-b99f19b2c592

75 kW HSAT | 5 bifacial technologies

Shading Factors



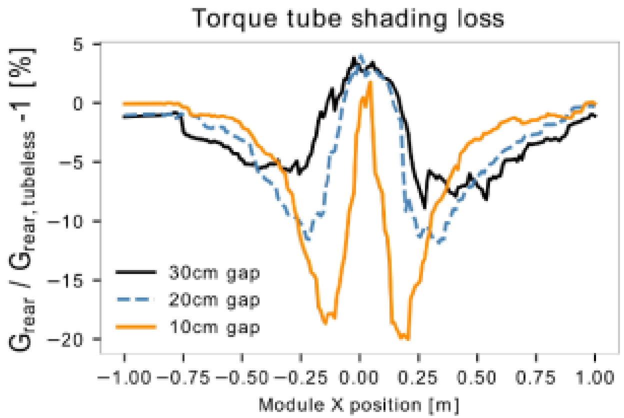
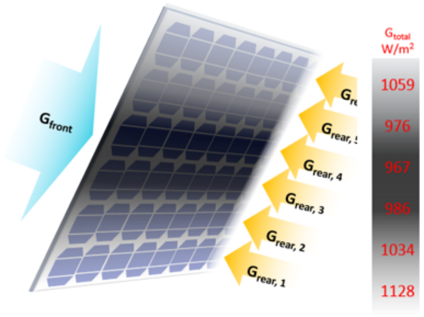


Fig. 13. a) RADIANCE image showing torque tube behind a modules row and b) G_{rear} across the module averaged over a sunny day.

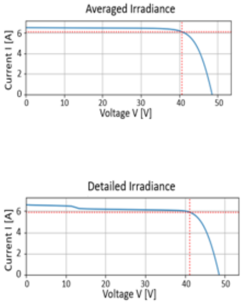
S. Ayala Pelaez, C. Deline, S. M. MacAlpine, B. Marion, J. S. Stein and R. K. Kostuk, "Comparison of Bifacial Solar Irradiance Model Predictions With Field Validation," in *IEEE Journal of Photovoltaics*, vol. 9, no. 1, pp. 82-88, Jan. 2019. doi: 10.1109/JPHOTOV.2018.2877000
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8534404&isnumber=8585410

Electrical Mismatch Losses



Averaging Irradiance for the module ~1031 W/m²

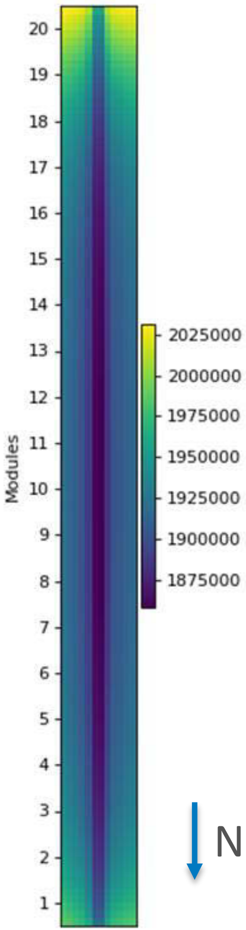
Detailed Irradiance value



P_0

P_1

$$L_{DC} = M = 1 - \frac{P_1}{P_0}$$

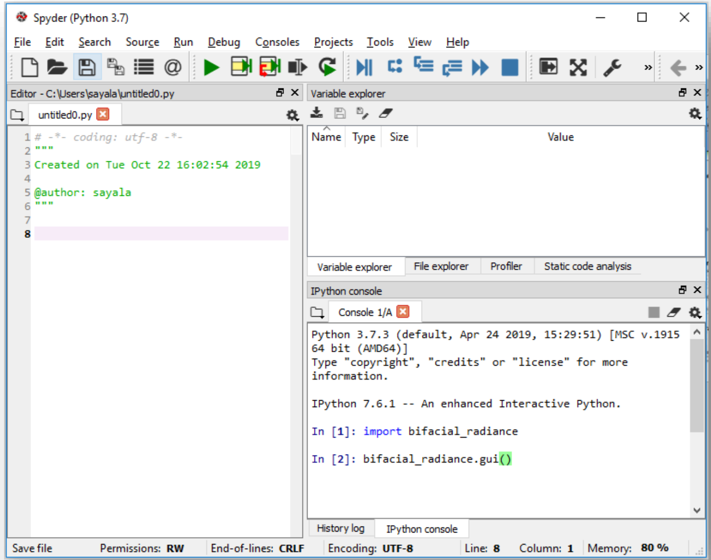


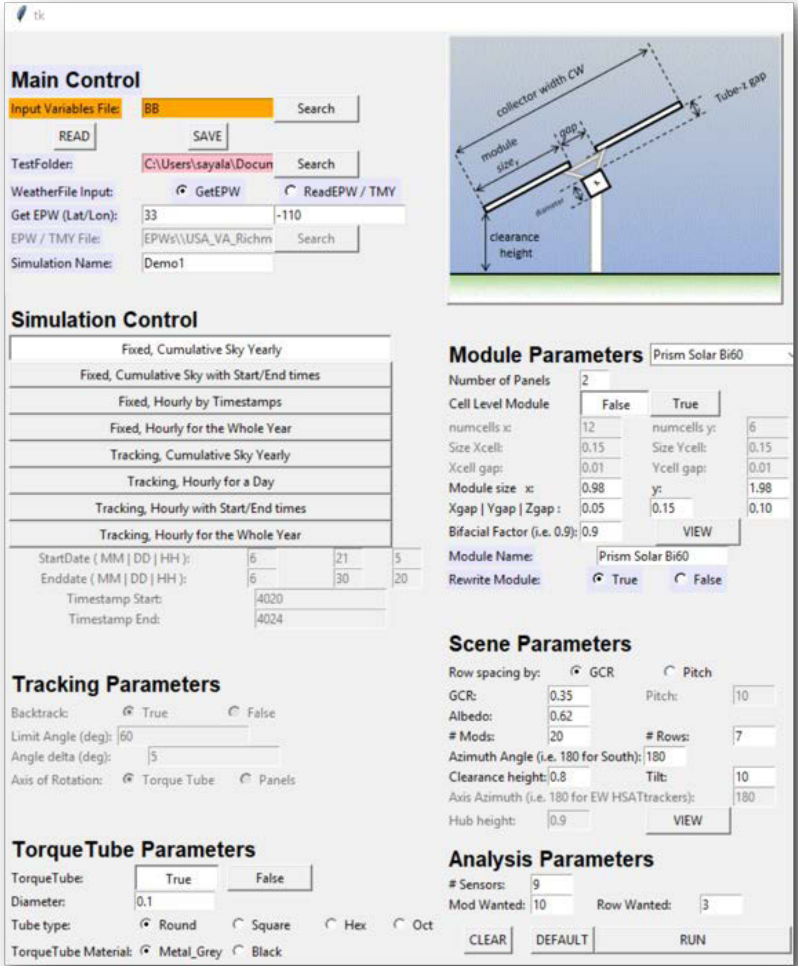
GUI (Visual Interface)

In Spyder Console:

import bifacial_radiance

bifacial_radiance.gui()





Q&A

https://github.com/NREL/bifacial_radiance/issues

silvana.ayala@nrel.gov

www.nrel.gov

NREL/PR-5K00-75218

A portion of the research was performed using computational resources sponsored by the Department of Energy's Office of Energy Efficiency and Renewable Energy and located at the National Renewable Energy Laboratory.

This work was authored in part by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under Solar Energy Technologies Office (SETO) Agreement Number 34910, and Award Number DE-EE0008564. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

NREL Transforming ENERGY