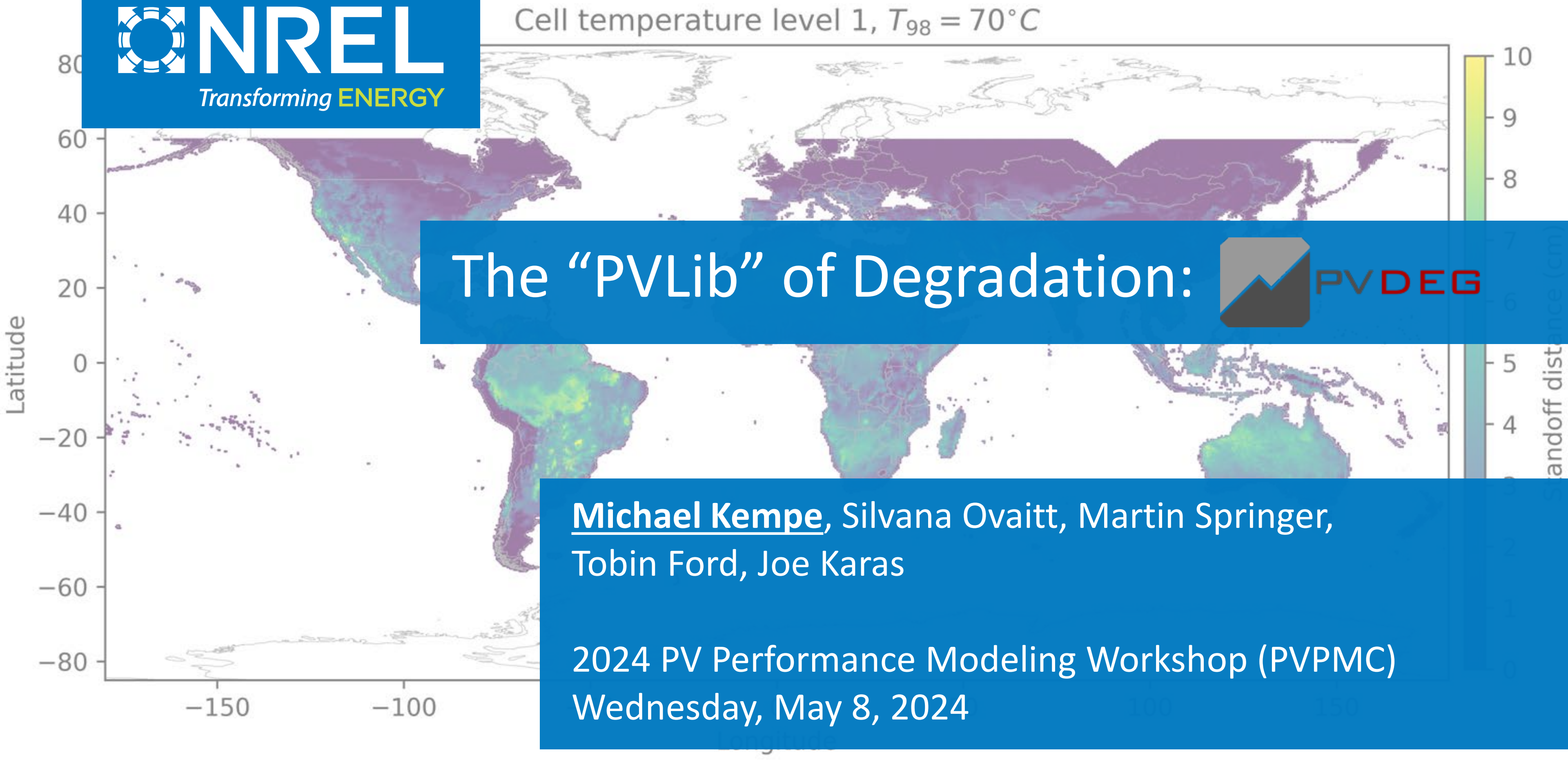
NREL
Transforming ENERGY



The “PVLib” of Degradation: PVDEG

Michael Kempe, Silvana Ovaitt, Martin Springer,
Tobin Ford, Joe Karas

2024 PV Performance Modeling Workshop (PVPMC)
Wednesday, May 8, 2024

Photo from iStock-627281636

https://github.com/NREL/pvdegradationtools

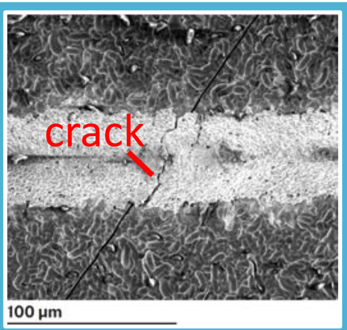
Outline

1. PVDeg Goals
2. Structure & details
3. Example: Standards for PV Standoff
4. Example: LETID
5. Geo-spatial modeling implementation
6. Roadmap

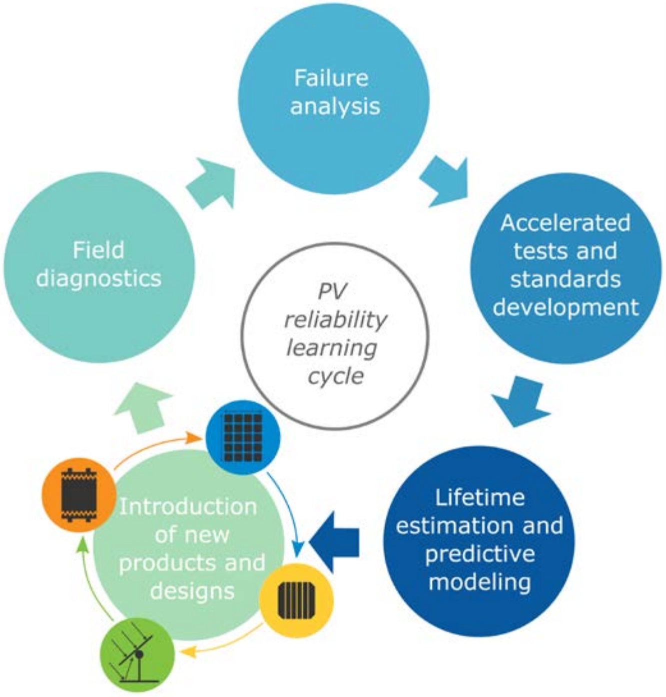
https://github.com/NREL/pvdegradationtools

DESIGN-FOR-RELIABILITY

***Challenge**: Design-for-reliability needs to keep with rapid change and scaling of industry.*



[Image by Tim Silverman / NREL]



Failure analysis

Accelerated tests and standards development

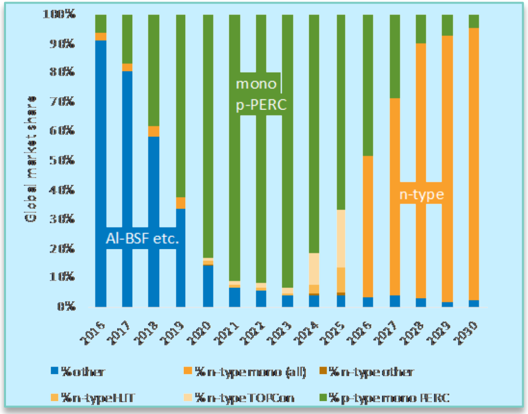
Lifetime estimation and predictive modeling

Introduction of new products and designs

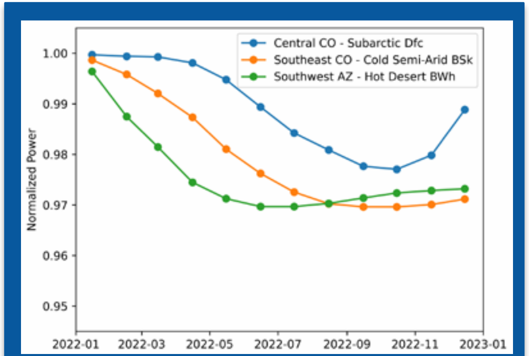
Field diagnostics

PV reliability learning cycle

[P. Hacke, et al., (2019) In Advanced Micro-and Nanomaterials for Photovoltaics]



Jarett Zuboy. DuraMAT Tech Scouting 2022



[M. Springer, **et. al.**, *Prog Photovolt Res Appl.* 2022;1–8., doi: DOI: 10.1002/pip.3645]

Laboratory to field extrapolation

- There is no single equation describing the degradation of a PV module as a whole. We are only able to describe degradation as one mode or mechanism at a time.

Ester Hydrolysis*

$$\log\left(\frac{C}{C-x}\right) = A\cdot t\cdot RH^2\cdot e^{\left(\frac{-Ea}{kT}\right)}$$

CIGS Degradation*** BET model

$$R = k_o\cdot\left[\frac{RH}{1-RH+\epsilon}\right]\cdot e^{\left(\frac{-Ea}{kT}\right)}$$

Si Cell Metallic Corrosion**

$$\mathrm{TF} = F1\cdot e^{-b\cdot RH}\cdot e^{\left(\frac{Ea}{kT}\right)}$$

Arrhenius

$$R=R_o\cdot e^{\left(\frac{-Ea}{kT}\right)}$$

UV Degradation****

$$R_D = R_0\cdot I^X\cdot e^{\left(\frac{-Ea}{kT}\right)}$$

Individual equations are difficult to determine and to parameterize, but they are usually simple.

- To extrapolate to the field you need to:

Get meteorological Data → Estimate module temperature → Determine other parameters such as humidity, stressor ramp rates, voltage, UV spectrum, UV dose & etc. → Run extrapolation to the field

These first three steps are highly repetitive and similar for most researchers.

---

*J. E. Pickett and D. J. Coyle, "Hydrolysis kinetics of condensation polymers under humidity aging conditions," Polymer Degradation and Stability, vol. 98, no. 7, pp. 1311-1320, 2013,

**Whitfield, Salomon, Yang Suez, "Damp Heat versus Field Reliability for Crystalline Silicon", 38th IEEE PVSC (2012).

***Coyle, Blaydes, Northey, Pickett, Nagarkar, Zhao, Gardner, "Life Prediction for CIGS solar modules part 1: modeling moisture ingress and degradation", Prog. Photovolt: Res. Appl. (2011).

**** M. D. Kempe et al., "Highly Accelerated UV Stress Testing for Transparent Flexible Frontsheets," in 2020 47th IEEE Photovoltaic Specialists Conference (PVSC), 2020, pp. 1823-1823.

Goals for this project

Python code library to simplify repetitive tasks

- access meteorological data, perform geospatial analysis or monte-carlo simulations…
- Allow for easy extensibility to add new degradation related functions
- Standardize variable names and code communication.

Create living databases of information on degradation and material properties

- pre-defined set of material and degradation properties
- allow users to add their own

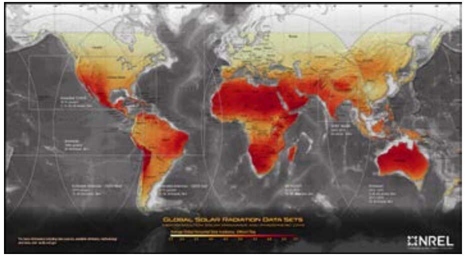
Focus user experience

- Tutorials – based on Jupyter notebooks
- Create simple interfaces such that one does not need to be a Python expert to use the code for common degradation models.
- Scalability – from laptop to HPC for production of maps.

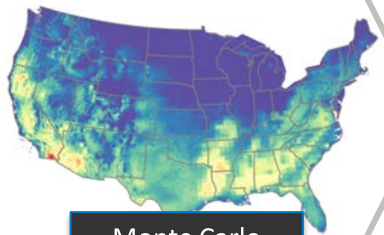
PVDEG PV Degradation Tools

The open-source integration pipeline for PV degradation analysis!

Stressors – NSRDB **Material Libraries** **Degradation models** **Geospatial Analysis**



$$R_D = R_o G^p e^{\left(\frac{-E_a}{RT}\right)}$$



Monte Carlo uncertainty

Geographical Mapping

Optimization

powered by

rev &gaps pvlib python™ jupyter The HDF Group aws binder

https://github.com/NREL/pvdegradationtools

The “PVLib” of Degradation: PVDEG

Key differences

- ☐ Reliability/Durability focus
- ☐ Parallelization support structure
- ☐ Mapping, Monte Carlo, and analysis support functions
- ☐ DATASETS

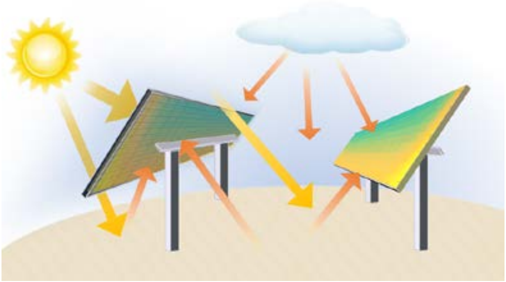
Advantages of creating a “PVLib” for degradation tools

- ✓ Bigger centralized location for all things PV degradation focused on a bottoms up approach for degradation modes and mechanisms
- ✓ Open source practices included (easier to install, use, etc)
- ✓ Longer-term maintenance of the repo (more possibility of code not becoming orphan, or outdated)
- ✓ Bigger team helping maintenance, documentation, and implementation into these geospatial or parallelization features
- ✓ Still get the first author attribution or contributions DOI for your resume and professional metrics from the Zenodo Releases

Degradation Requires a Different Focus

- For degradation, only UV flux is important, and temperature is typically very important.
- Humidity may also be important and is different outside, vs the inside of a module package.

$$D = D_o \int_0^t RH(t)^n \cdot e^{\frac{-E_a}{RT(t)}} \int_\lambda \left[e^{-C_2\lambda} \cdot G(\lambda,t)\right]^x d\lambda dt$$



To model degradation on the back side, you need to know the spectral sensitivity of materials and other parameters. This is just one composite example of common forms:

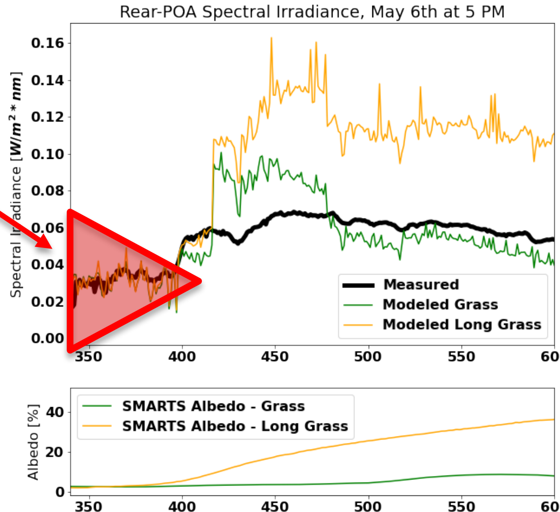
- ***C_2*** - empirical value describing spectral sensitivity to damage
- ***X*** – reciprocity factor describing effect of higher intensity. Typically, between 0.4 and 0.8.
- ***Ea*** – Arrhenius activation energy
- ***n*** – Dependence of degradation on humidity. Can be positive negative or zero and have many other forms other than this.

Most degradation occurs here.

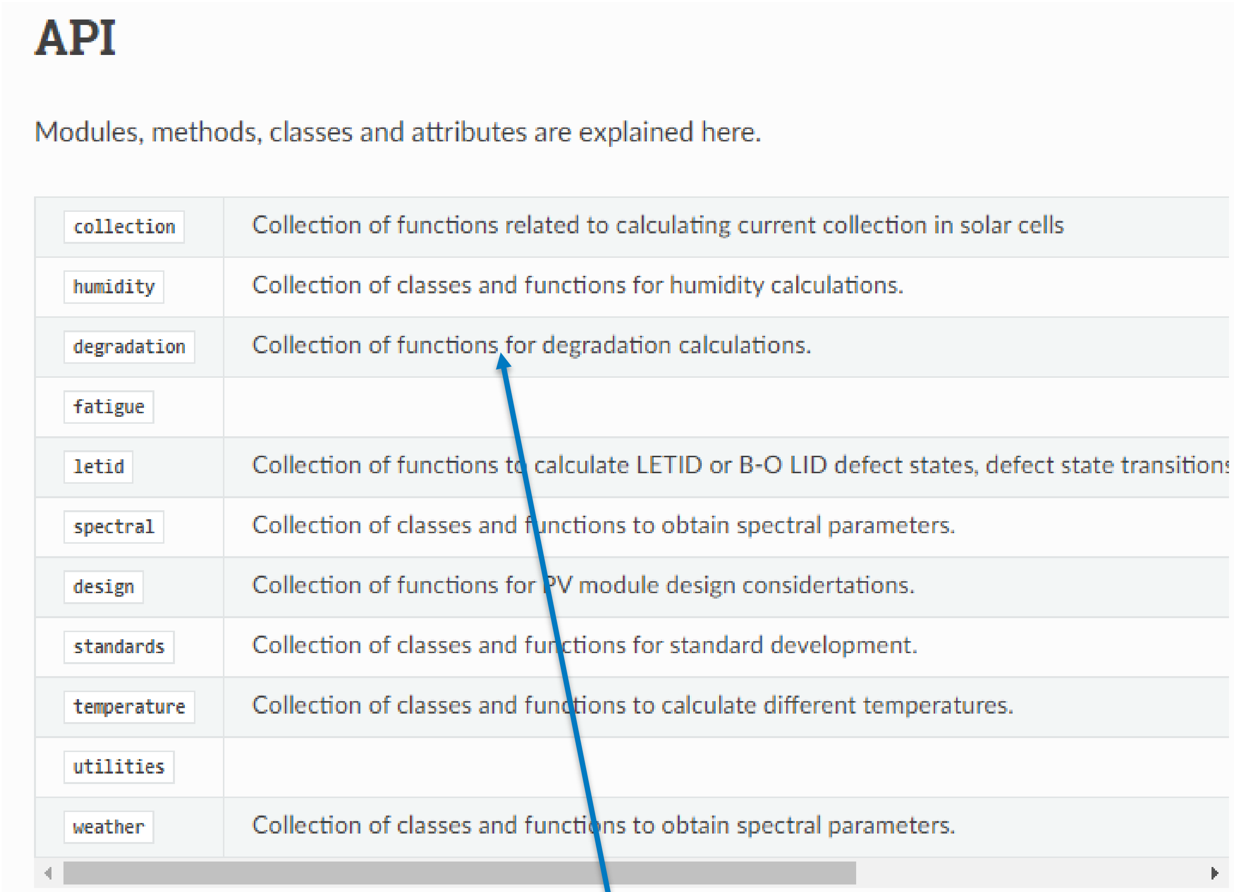
Action spectra are typically highly non-linear.

Photon Damage

$$\sim e^{-C_2\lambda}$$



Code library



API

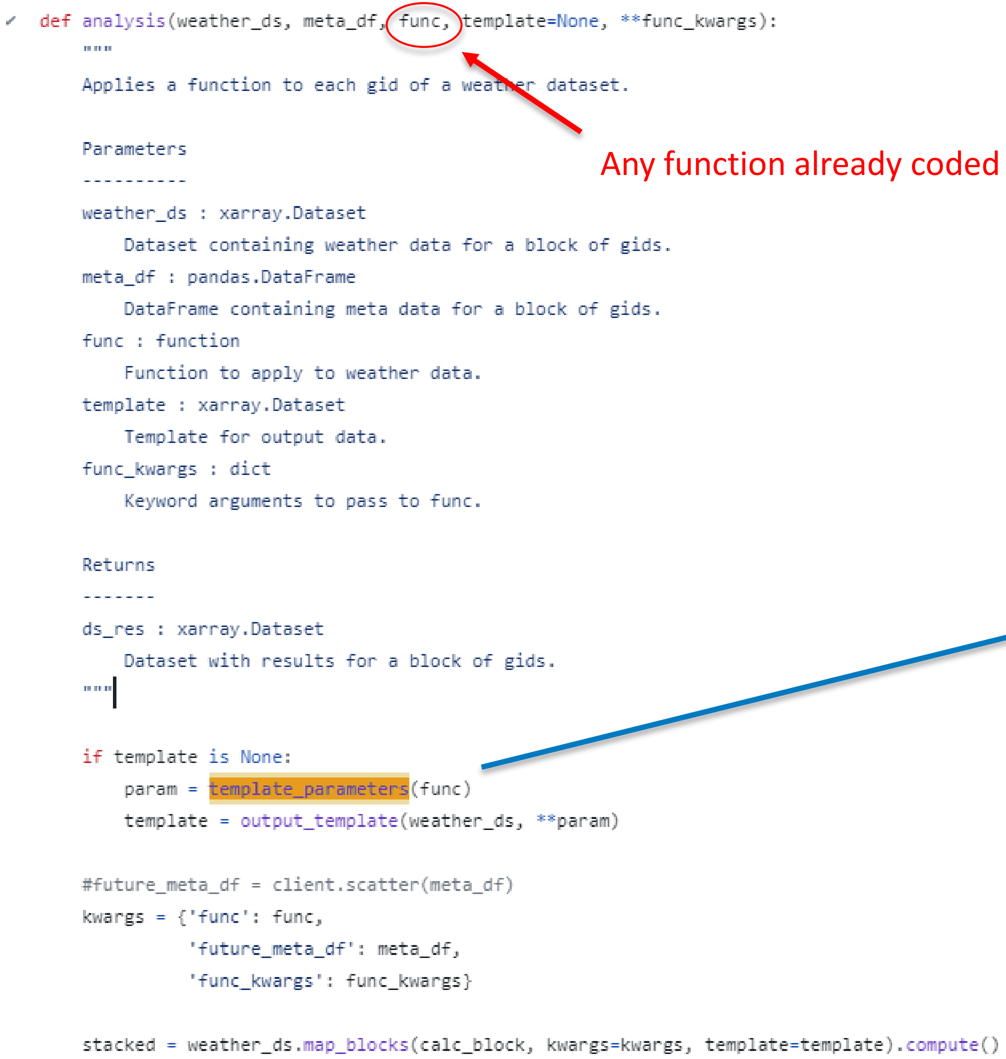
Modules, methods, classes and attributes are explained here.

`collection`	Collection of functions related to calculating current collection in solar cells
`humidity`	Collection of classes and functions for humidity calculations.
`degradation`	Collection of functions for degradation calculations.
`fatigue`	
`letid`	Collection of functions to calculate LETID or B-O LID defect states, defect state transition
`spectral`	Collection of classes and functions to obtain spectral parameters.
`design`	Collection of functions for PV module design considertations.
`standards`	Collection of classes and functions for standard development.
`temperature`	Collection of classes and functions to calculate different temperatures.
`utilities`	
`weather`	Collection of classes and functions to obtain spectral parameters.

E.g. $$R_D = R_o G^p e^{\left(\frac{-E_a}{RT}\right)}$$

- Peer-reviewed functions
- Auxiliary data handling and calculations functions

Geospatial Parallel Analysis Approach

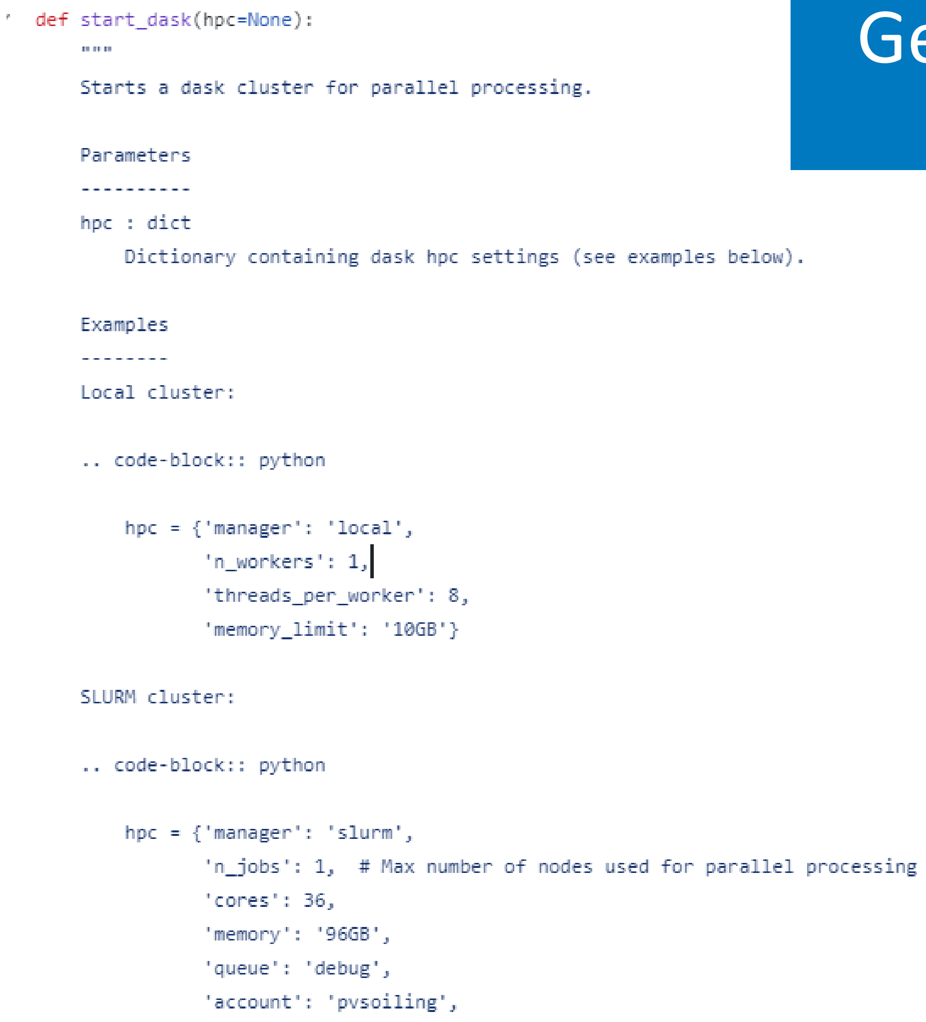


Any function already coded

```python
def analysis(weather_ds, meta_df, func, template=None, **func_kwargs):
    """
    Applies a function to each gid of a weather dataset.

    Parameters
    ----------
    weather_ds : xarray.Dataset
        Dataset containing weather data for a block of gids.
    meta_df : pandas.DataFrame
        DataFrame containing meta data for a block of gids.
    func : function
        Function to apply to weather data.
    template : xarray.Dataset
        Template for output data.
    func_kwargs : dict
        Keyword arguments to pass to func.

    Returns
    -------
    ds_res : xarray.Dataset
        Dataset with results for a block of gids.
    """

    if template is None:
        param = template_parameters(func)
        template = output_template(weather_ds, **param)

    #future_meta_df = client.scatter(meta_df)
    kwargs = {'func': func,
              'future_meta_df': meta_df,
              'func_kwargs': func_kwargs}

    stacked = weather_ds.map_blocks(calc_block, kwargs=kwargs, template=template).compute()
```

The magic: specifying known format of outputs

```python
if func == standards.standoff:

    shapes = {'x':        ('gid',),
              'T98_inf':  ('gid',),
              'T98_0':    ('gid',),
              }

    attrs = {'x' :      {'units': 'cm'},
             'T98_0' :   {'units': 'Celsius'},
             'T98_inf' : {'units': 'Celsius'}}

    add_dims = {}
```



```
def start_dask(hpc=None):
    """
    Starts a dask cluster for parallel processing.

    Parameters
    ----------
    hpc : dict
        Dictionary containing dask hpc settings (see examples below).

    Examples
    --------
    Local cluster:

    .. code-block:: python

        hpc = {'manager': 'local',
               'n_workers': 1,
               'threads_per_worker': 8,
               'memory_limit': '10GB'}

    SLURM cluster:

    .. code-block:: python

        hpc = {'manager': 'slurm',
               'n_jobs': 1,  # Max number of nodes used for parallel processing
               'cores': 36,
               'memory': '96GB',
               'queue': 'debug',
               'account': 'pvsoiling',
```

Geospatial Parallel Analysis Approach

Easy call for local or remote parallelization.

Will send a list of sites with calculation information to a computer which will later pull in all the meteorological data to run the calculations with parallel processing

Future Goal:
Sep. 2024: AWS parameters with this approach

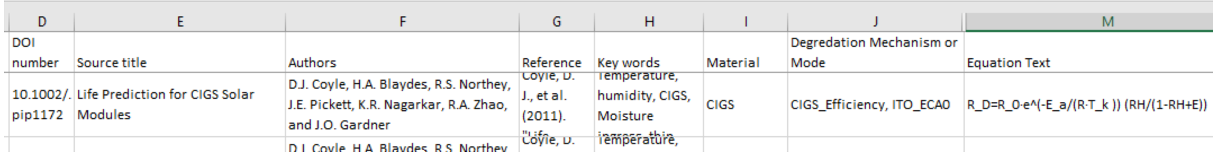
Material and Degradaiton Library

Includes:

- material properties
- parameters for degradation calculations
- Known constants and other empirical factors
- Degradation equations

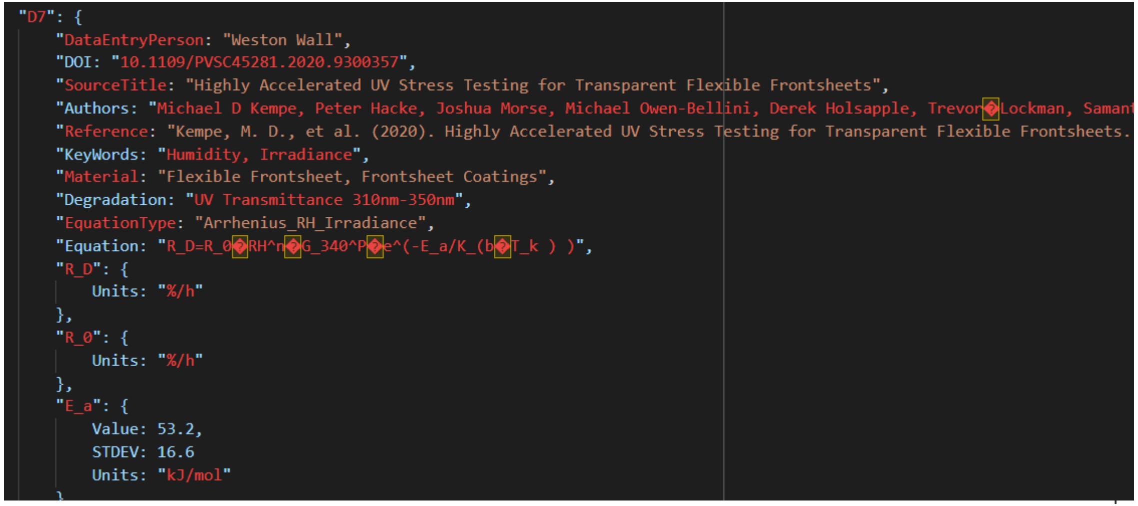
- Searchable database of PV related degradation parameters.
- Comprehensive literature search for most common values already included
- Proposed taxonomy using JSONs

Data Gathering:



D	E	F	G	H	I	J	M
DOI number	Source title	Authors	Reference	Key words	Material	Degradation Mechanism or Mode	Equation Text
10.1002/.pip1172	Life Prediction for CIGS Solar Modules	D.J. Coyle, H.A. Blaydes, R.S. Northey, J.E. Pickett, K.R. Nagarkar, R.A. Zhao, and J.O. Gardner	Coyle, D. J., et al. (2011). "Life	Temperature, humidity, CIGS, Moisture ingress, thin	CIGS	CIGS_Efficiency, ITO_ECA0	R_D=R_0·e^(-E_a/(R·T_k)) (RH/(1-RH+E))
		D.J. Coyle, H.A. Blaydes, R.S. Northey	Coyle, D.	Temperature,			

Structured proposed in JSON format (taxonomy still in development):



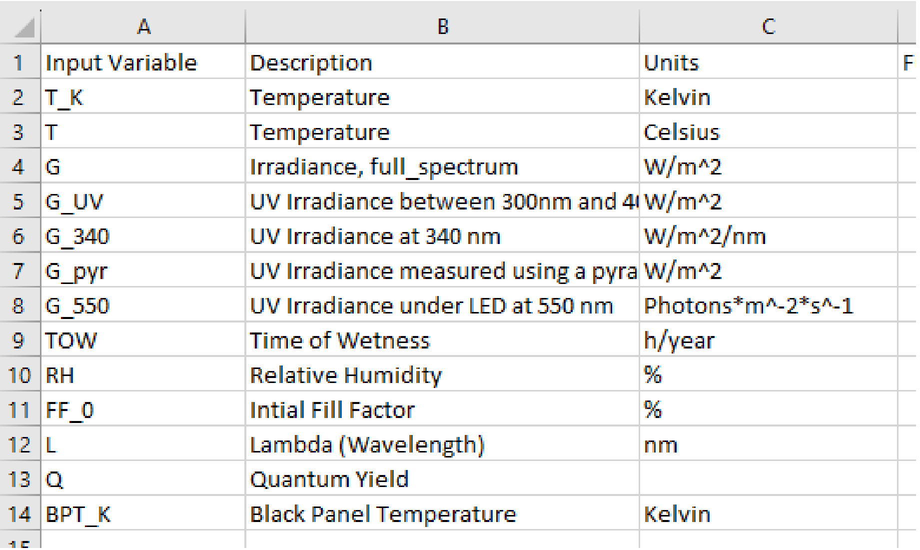
```
"D7": {
    "DataEntryPerson: "Weston Wall",
    "DOI: "10.1109/PVSC45281.2020.9300357",
    "SourceTitle: "Highly Accelerated UV Stress Testing for Transparent Flexible Frontsheets",
    "Authors: "Michael D Kempe, Peter Hacke, Joshua Morse, Michael Owen-Bellini, Derek Holsapple, Trevor�Lockman, Samant
    "Reference: "Kempe, M. D., et al. (2020). Highly Accelerated UV Stress Testing for Transparent Flexible Frontsheets.
    "KeyWords: "Humidity, Irradiance",
    "Material: "Flexible Frontsheet, Frontsheet Coatings",
    "Degradation: "UV Transmittance 310nm-350nm",
    "EquationType: "Arrhenius_RH_Irradiance",
    "Equation: "R_D=R_0�RH^n�G_340^P�e^(-E_a/K_(b�T_k ) )",
    "R_D": {
        Units: "%/h"
    },
    "R_0": {
        Units: "%/h"
    },
    "E_a": {
        Value: 53.2,
        STDEV: 16.6
        Units: "kJ/mol"
    }
```

Material library

Using standardized variable names form PV-terms and developing new ones as needed.
Requires some harmonization of data pulled from various sources.

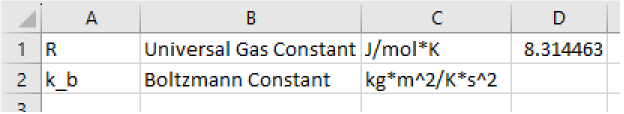
Reliability PVTerms

Stressor Parameters



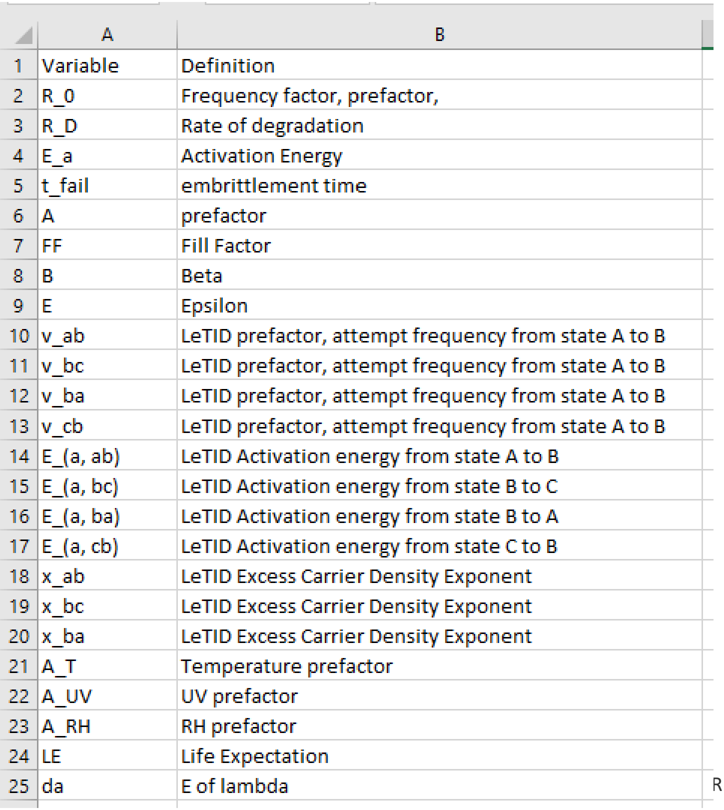
	A	B	C	
1	Input Variable	Description	Units	F
2	T_K	Temperature	Kelvin	
3	T	Temperature	Celsius	
4	G	Irradiance, full_spectrum	W/m^2	
5	G_UV	UV Irradiance between 300nm and 4	W/m^2	
6	G_340	UV Irradiance at 340 nm	W/m^2/nm	
7	G_pyr	UV Irradiance measured using a pyra	W/m^2	
8	G_550	UV Irradiance under LED at 550 nm	Photons*m^-2*s^-1	
9	TOW	Time of Wetness	h/year	
10	RH	Relative Humidity	%	
11	FF_0	Intial Fill Factor	%	
12	L	Lambda (Wavelength)	nm	
13	Q	Quantum Yield		
14	BPT_K	Black Panel Temperature	Kelvin	

Universal Constants



	A	B	C	D
1	R	Universal Gas Constant	J/mol*K	8.314463
2	k_b	Boltzmann Constant	kg*m^2/K*s^2	

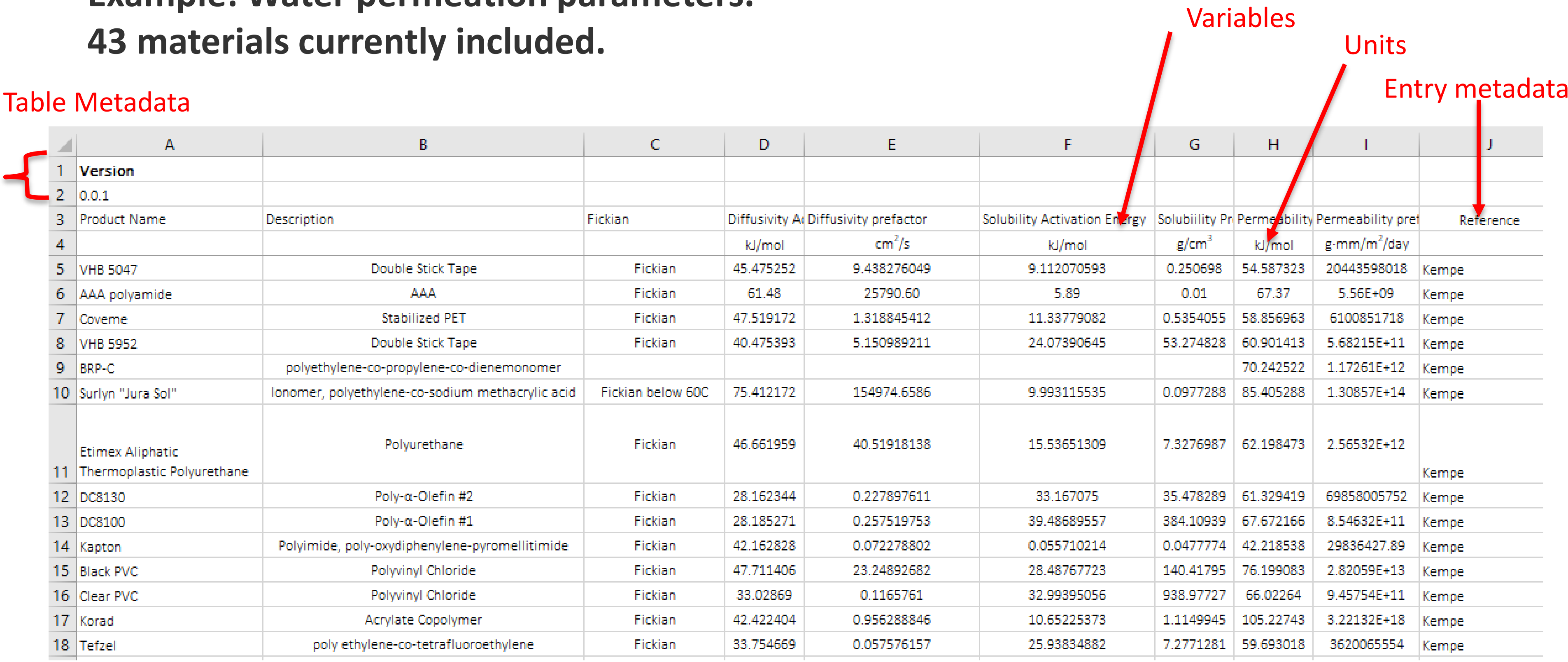
Modeling constants



	A	B
1	Variable	Definition
2	R_0	Frequency factor, prefactor,
3	R_D	Rate of degradation
4	E_a	Activation Energy
5	t_fail	embrittlement time
6	A	prefactor
7	FF	Fill Factor
8	B	Beta
9	E	Epsilon
10	v_ab	LeTID prefactor, attempt frequency from state A to B
11	v_bc	LeTID prefactor, attempt frequency from state A to B
12	v_ba	LeTID prefactor, attempt frequency from state A to B
13	v_cb	LeTID prefactor, attempt frequency from state A to B
14	E_(a, ab)	LeTID Activation energy from state A to B
15	E_(a, bc)	LeTID Activation energy from state B to C
16	E_(a, ba)	LeTID Activation energy from state B to A
17	E_(a, cb)	LeTID Activation energy from state C to B
18	x_ab	LeTID Excess Carrier Density Exponent
19	x_bc	LeTID Excess Carrier Density Exponent
20	x_ba	LeTID Excess Carrier Density Exponent
21	A_T	Temperature prefactor
22	A_UV	UV prefactor
23	A_RH	RH prefactor
24	LE	Life Expectation
25	da	E of lambda

Material library

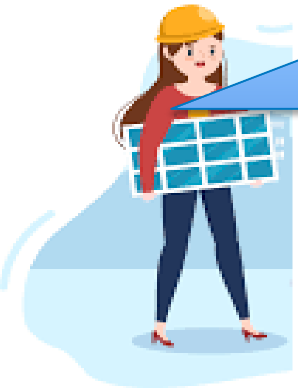
Example: Water permeation parameters.
43 materials currently included.



	A	B	C	D	E	F	G	H	I	J
1	**Version**									
2	0.0.1									
3	Product Name	Description	Fickian	Diffusivity A	Diffusivity prefactor	Solubility Activation Energy	Solubiility Pr	Permeability	Permeability pref	Reference
4				kJ/mol	cm^2/s	kJ/mol	g/cm^3	kJ/mol	$g{\cdot}mm/m^2/day$	
5	VHB 5047	Double Stick Tape	Fickian	45.475252	9.438276049	9.112070593	0.250698	54.587323	20443598018	Kempe
6	AAA polyamide	AAA	Fickian	61.48	25790.60	5.89	0.01	67.37	5.56E+09	Kempe
7	Coveme	Stabilized PET	Fickian	47.519172	1.318845412	11.33779082	0.5354055	58.856963	6100851718	Kempe
8	VHB 5952	Double Stick Tape	Fickian	40.475393	5.150989211	24.07390645	53.274828	60.901413	5.68215E+11	Kempe
9	BRP-C	polyethylene-co-propylene-co-dienemonomer						70.242522	1.17261E+12	Kempe
10	Surlyn "Jura Sol"	Ionomer, polyethylene-co-sodium methacrylic acid	Fickian below 60C	75.412172	154974.6586	9.993115535	0.0977288	85.405288	1.30857E+14	Kempe
11	Etimex Aliphatic Thermoplastic Polyurethane	Polyurethane	Fickian	46.661959	40.51918138	15.53651309	7.3276987	62.198473	2.56532E+12	Kempe
12	DC8130	Poly-α-Olefin #2	Fickian	28.162344	0.227897611	33.167075	35.478289	61.329419	69858005752	Kempe
13	DC8100	Poly-α-Olefin #1	Fickian	28.185271	0.257519753	39.48689557	384.10939	67.672166	8.54632E+11	Kempe
14	Kapton	Polyimide, poly-oxydiphenylene-pyromellitimide	Fickian	42.162828	0.072278802	0.055710214	0.0477774	42.218538	29836427.89	Kempe
15	Black PVC	Polyvinyl Chloride	Fickian	47.711406	23.24892682	28.48767723	140.41795	76.199083	2.82059E+13	Kempe
16	Clear PVC	Polyvinyl Chloride	Fickian	33.02869	0.1165761	32.99395056	938.97727	66.02264	9.45754E+11	Kempe
17	Korad	Acrylate Copolymer	Fickian	42.422404	0.956288846	10.65225373	1.1149945	105.22743	3.22132E+18	Kempe
18	Tefzel	poly ethylene-co-tetrafluoroethylene	Fickian	33.754669	0.057576157	25.93834882	7.2771281	59.693018	3620065554	Kempe

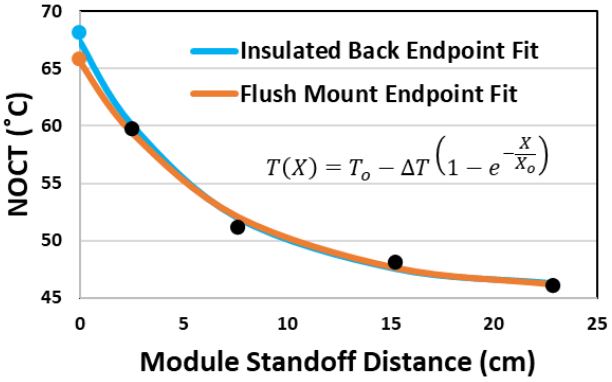
This is a visualization / input tool. Taxonomy and data implemented through a JSON file

Example Use Case:
IEC 63126



I want the panels I install to be safe,
but I don’t want to spend more money than necessary on racking.
I know hot panels are no-bueno,
and that the closer they are to the roof the hotter they’ll be.
How do I know the right distance for my city, i.e. Phoenix?

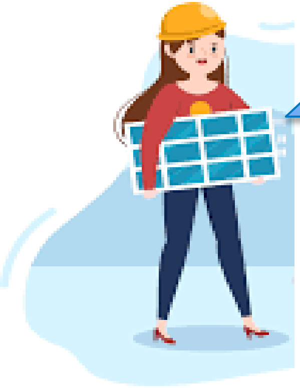
Standoff images for distances of (A) flush mount (B) 2.5 cm, and (C) 10 cm.



M. K. Fuentes, "A simplified thermal model for Flat-Plate photovoltaic arrays," United States, 1987-05-01 1987. [Online]. Available: https://www.osti.gov/biblio/6802914

Example Use Case: IEC 63126

IEC 63126 specifies more rigorous testing for modules deployed in combinations of locations and racking that result in **high temperatures** defined as the 98th percentile temperature of 70°C, 80°C or 90°C



My Module is rated for the 70°C high temperature

$$X_{eff} = -X_o \ln\left(1 - \frac{T_o - T_{98}}{\Delta T}\right)$$

X_0 = 6.5 cm
T_{98} = 70°C
T_0 = Insulated back module temperature
ΔT = Difference between insulated back and open rack modules

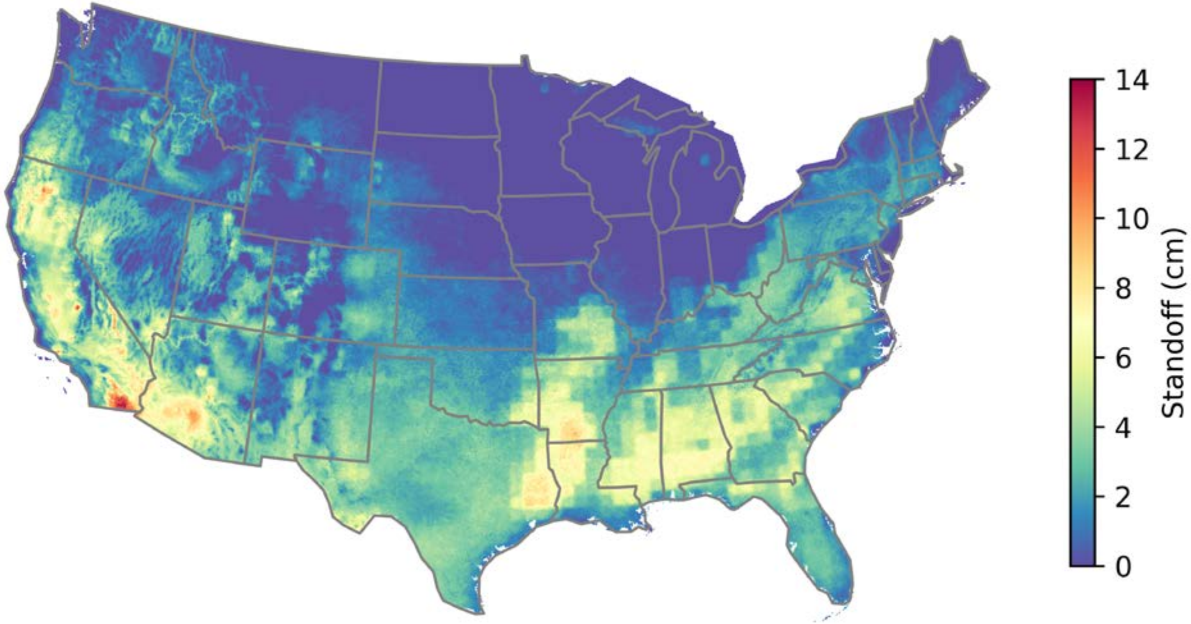
M. K. Fuentes, "A simplified thermal model for Flat-Plate photovoltaic arrays," United States, 1987-05-01 1987. [Online]. Available: https://www.osti.gov/biblio/6802914

Example Use Case: IEC 63126

IEC 63126 specifies more rigorous testing for modules deployed in combinations of locations and racking that result in **high temperatures** defined as the 98th percentile temperature of 70°C, 80°C or 90°C

$$X_{eff} = -X_o \ln\left(1 - \frac{T_o - T_{98}}{\Delta T}\right)$$

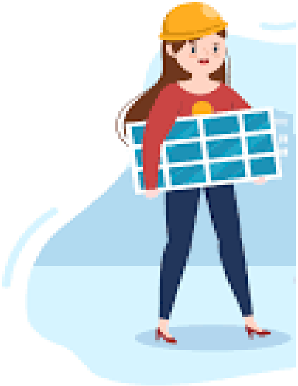
$X_0 = 6.5$ cm
$T_{98} = 70$°C
T_0 = Insulated back module temperature
ΔT = Difference between insulated back and open rack modules

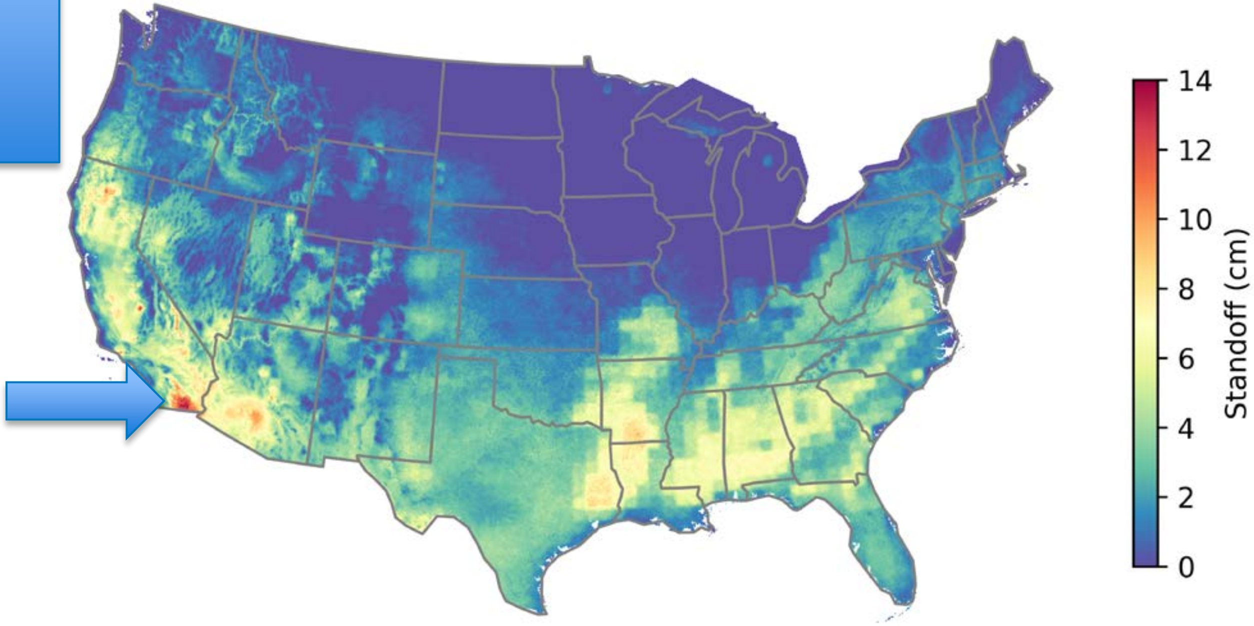


M. K. Fuentes, "A simplified thermal model for Flat-Plate photovoltaic arrays," United States, 1987-05-01 1987. [Online]. Available: https://www.osti.gov/biblio/6802914

Example Use Case: IEC 63126

I should install with at leat 14 cm of gap.

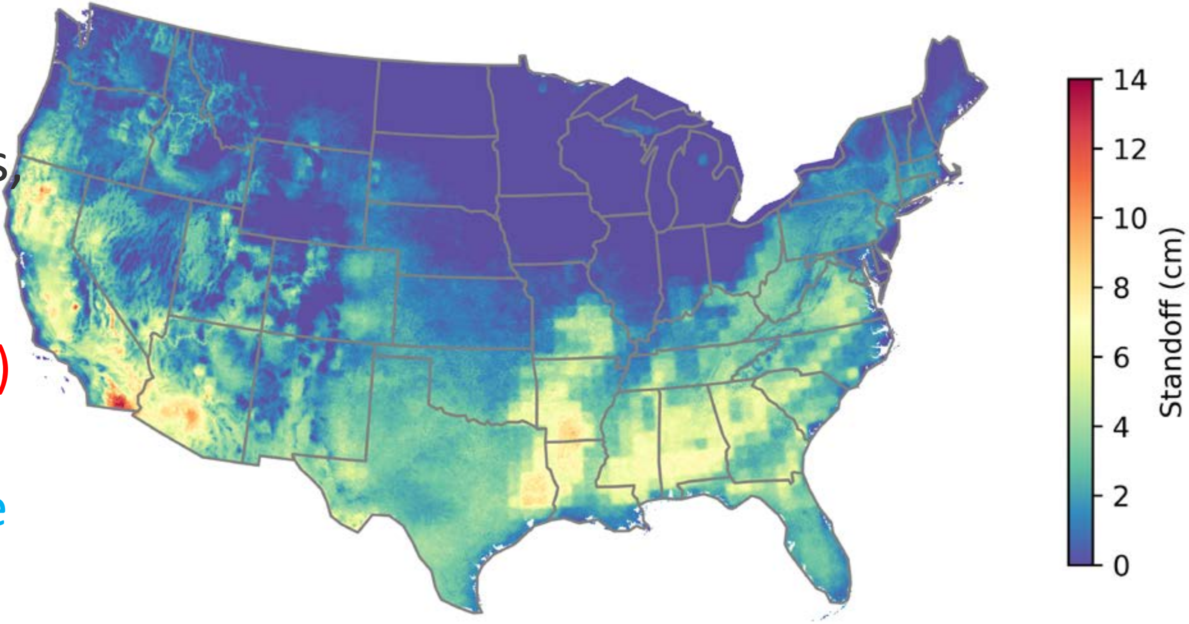




M. K. Fuentes, "A simplified thermal model for Flat-Plate photovoltaic arrays," United States, 1987-05-01 1987. [Online]. Available: https://www.osti.gov/biblio/6802914

Example Use Case: IEC 63126

- Large amounts of calculation with minimal effort
- Integrates the NSRDB data, module temperature models, and the PVDeg functions in the GitHub repository.
- <1 hour (with parallelization)
- Single-location calculations easily accessible through the journals
- Similar maps will be used in the new version of IEC 63126.



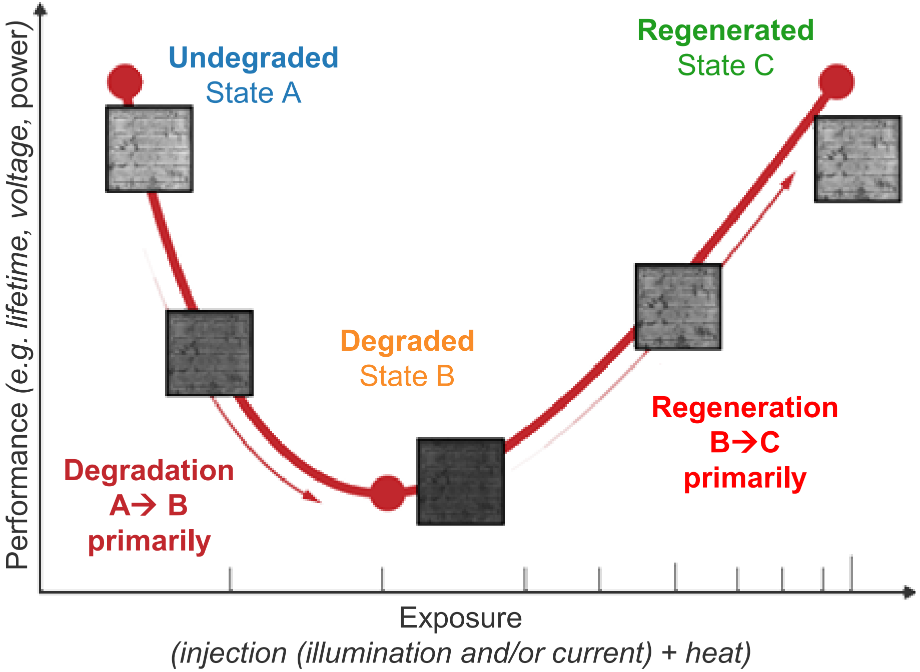
M. K. Fuentes, "A simplified thermal model for Flat-Plate photovoltaic arrays," United States, 1987-05-01 1987. [Online]. Available: https://www.osti.gov/biblio/6802914

Colab Utilized for Simple Access

- Enforcement of the module installation compliance to IEC TS 63126 must be accomplished through the local building codes enforcement process.
- Most of these high temperature systems will be small rooftop systems designed by people with very little PV system performance skills.
- Using PVDeg, we have created a Jupyter notebook that can be accessed through Colab enabling local code enforcement entities or small system designers to run standardized calculations to estimate the 98th percentile temperature.
- https://tinurl.com/IEC-63126-Standoff

LETID implementation

Review of LETID and B-O LID



J. Karas *et al.*, *Progress in Photovoltaics: Research and Applications*, 2022, doi: 10.1002/pip.3573.
I.L. Repins *et al*, *MRS Bulletin*, 2023, doi: 10.1557/s43577-022-00438-8

- **Light-and elevated temperature-induced degradation (LETID)**
  - Relatively recently-discovered degradation mode in silicon
  - Some early cases shows ~10% degradation; more typically 0-3%
  - Losses eventually “regenerate”, but this can take decades depending on climate and technology
- **Boron-oxygen light-induced degradation (B-O LID)**
  - More well-known and better understood defect in mono c-Si
  - Motivated the industry transition to Ga-doped wafers
  - Compared to LETID: faster and less severe. Often accounted for by “First Year” losses in warranties and financial models.
- **Both LETID and B-O LID can be described by a 3-state model**
  - Degradation (**A** → **B**) followed by regeneration (**B** → **C**)
  - Kinetics and time constants are different in LETID and B-O LID, but they can be modeled similarly.

- Progression between states depends on time, carrier injection (either illumination or electrical current), and temperature

LETID implementation

LETID and B-O LID Modeling

Performance loss is a function of the number of defects in state B.

$$Degradation \propto N_B$$

Defect state transitions depend on simultaneous, competing reaction rates

$$\frac{dN_A}{dt} = k_{AB} \cdot N_A + k_{BA} \cdot N_B$$

$$\frac{dN_B}{dt} = k_{AB} \cdot N_A + k_{CB} \cdot N_C - (k_{BA} + k_{BC}) \cdot N_B$$

$$\frac{dN_C}{dt} = k_{BC} \cdot N_B - k_{CB} \cdot N_C$$

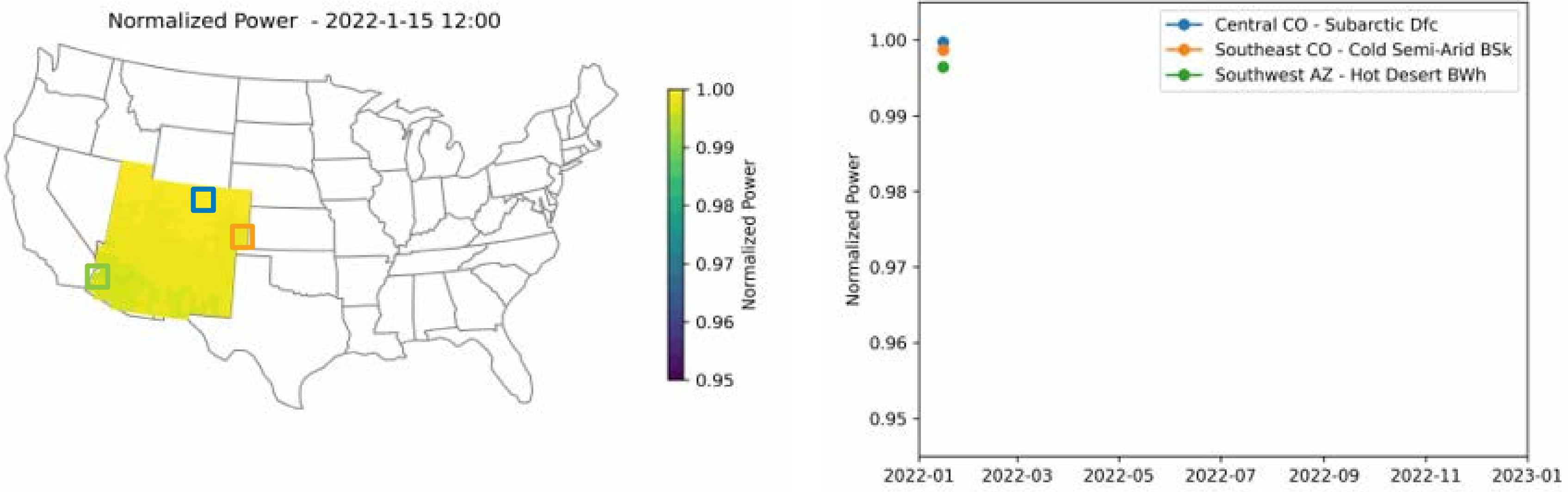
Reaction rates (k_{ij}) have Arrhenius behavior, with modification for injection (excess electronic carrier density in the device)

Kinetic parameters compiled from literature:
$E_{a,ij} \mid v'_{ij} \mid x_{ij}$

$$k_{ij} = v_{ij} \cdot \exp\left(\frac{E_{a,ij}}{kT}\right)$$

$$v_{ij} = v'_{ij} \cdot \Delta n^{x_{ij}}$$

LETID implementation



Credit: Joseph F. Karas

I. L. Repins *et al.*, “Long-term impact of light- and elevated temperature-induced degradation on photovoltaic arrays,” *MRS Bull.*, 2023, doi: 10.1557/s43577-022-00438-8.

Summary

https://github.com/NREL/PVDegradationTools

- PV deployment is growing and evolving exponentially, and we can’t wait many years to know if things are durable and might last 50 years; therefore, a very robust understanding of the modes and mechanisms for failure is needed.
- Open source and flexible Python code is being developed to simplify this long-term extrapolation to the field.
- Extrapolation to the field involves a lot of repetitive process which we are automating enabling users to focus on the unique and fundamental aspects of a given degradation.
- We are also creating living libraries of data to facilitate understanding of the complex and multi-faceted degradation of PV modules.

github.com/NREL/PVDegradationTools

This work was authored 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). The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government.

Funding provided as part of the Durable Modules Materials Consortium (DuraMAT), an Energy Materials Network Consortium funded by the U S Department of Energy, Office of Energy Efficiency and Renewable Energy, Solar Energy Technologies Office Agreement Number 32509.

NREL
Transforming ENERGY

Backup Slides

www.nrel.gov

NREL/PR-5K00-89537

Abstract

The “PVLib” of Degradation: PVDeg

Michael Kempe, Silvana Ovaitt, Tobin Ford, and Martin Springer

Keywords: Durability, PVDeg, Reliability, Photovoltaic, Modeling, Python

The Photovoltaic (PV) industry constantly aims for lower costs through higher-efficiency cells, improved module designs, and improvements in durability. This leads to the use of new materials, designs, and manufacturing processes, and not always with a sufficient amount of durability testing. To help drive down costs there is a desire to create modules that will last for up to 50 years of service life. To accomplish this, every degradation mode and mechanism must be identified and either eliminated or otherwise mitigated. This involves the extrapolation of laboratory results to the field conditions. There is a need to organize the existing degradation data into an accessible format and to provide industry relevant tools for extrapolation from laboratory to field conditions. While the basic equations used to model degradation are sometimes very simple, the full analysis involves calculations are cumbersome but ubiquitous for many degradation processes. A simplified, modeling framework to accomplish these repetitive processes will facilitate the analysis to help researchers keep up with the rapid pace of technological changes.

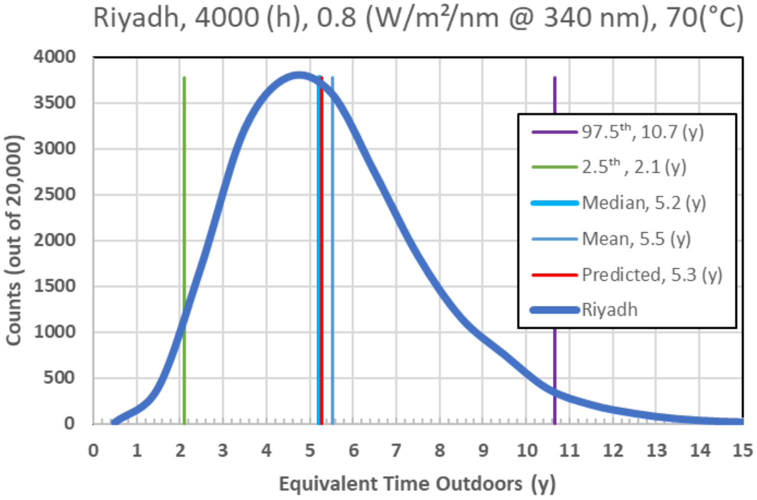
In this talk, we will describe our progress creating the open-source tool PVDeg. This tool can be used to search for and analyze degradation information and extrapolate PV module performance and durability to field exposure. PVDeg simplifies many of the common foundational computational operations for obtaining meteorological data and using it to generate a model of the PV deployment. This prediction tool repository also contains various degradation models as well as a library of material parameters suitable for estimating the durability assessment of materials and components. We use an integration pipeline approach that allows us to leverage weather data from the National Solar Radiation Database, and other weather sources, to perform geospatial degradation analysis in the US and worldwide. We hope to become a repository that can be used for weathering and degradation analysis for various applications beyond the PV industry. During the talk, we will provide the PVPMC attendees the opportunity to interact with the tool via a Google Collab tutorial they can run on their phones or laptops.

Monte Carlo functionality

$$R_D = R_0 \cdot I^X \cdot e^{\left(\frac{-Ea}{kT}\right)}$$

Data for Single axis tracking.

- There is a lot of uncertainty in the extrapolation to the field.
- To capture the uncertainty you need the model parameters, their uncertainties, and the correlation coefficients.
- We are developing code to create large numbers of parameter sets of which each of them can be run through the extrapolation calculation to get a distribution of the degradation estimation.

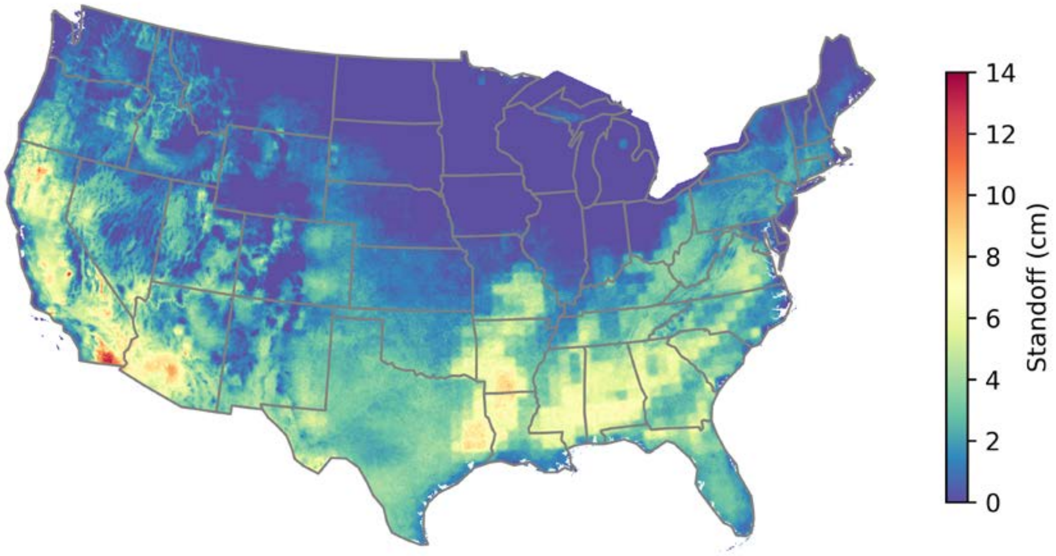


The condition of A3 from IEC 62788-7-2 is only equivalent to a few years of frontside exposure in Riyadh on a single axis tracker.

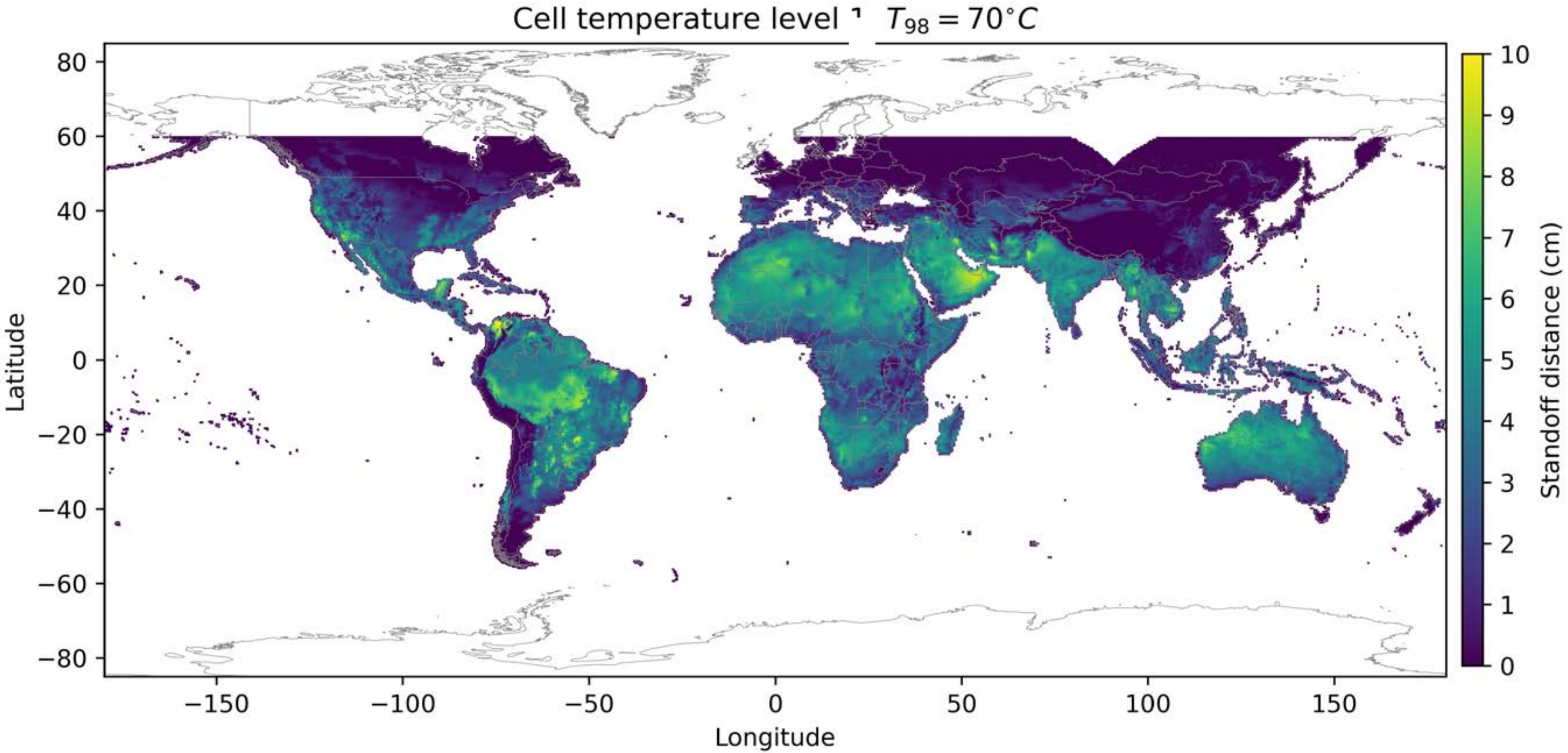
*M. D. Kempe et al., "Highly Accelerated UV Stress Testing for Transparent Flexible Frontsheets," in 2020 47th IEEE Photovoltaic Specialists Conference (PVSC), 2020, pp. 1823-1823.

One step further: Geospatial analysis

```
# Import package
import pvdeg

# Start compute cluster
pvdeg.geospatial.start_dask()

# Get weather data
weather_db = 'NSRDB'
weather_arg = {'satellite': 'Americas',
               'names': 2022,
               'NREL_HPC': True,}
weather_ds, meta_df = pvdeg.weather.get(
    weather_db, geospatial=True, **weather_arg)

# Specify function and input parameters
geo = {'func': pvdeg.standards.standoff,
       'weather_ds': weather_NM_sub,
       'meta_df': meta_NM_sub}

# Perform calculation
standoff_res = pvdeg.geospatial.analysis(**geo)

# Post process results
fig, ax = pvdeg.geospatial.plot_USA(standoff_res['x'],
                    cmap='viridis', vmin=0, vmax=None,
                    title='Minimum estimated air standoff',
                    cb_title='Standoff (cm)')
```



Extension to world map



Map to be included in IEC TS 63126, edition 2, 2024.