PV Degradation Modeling: Applying Geospatial Workflows with “PVDeg”

Tobin Ford, Silvana Ovaitt, Martin Springer, Michael Kempe (NREL)

Awarded FY22
Core Modelling Call

Period of Performance: FY23-26
Funding: $750k

Contribution to DuraMAT Consortium Goals – Project Overview

The goal of this work is to create an online tool that can be used to search for degradation information and extrapolate PV module performance and durability to field exposure. A graphical user interface will aid in the understanding of the results. The prediction tool will be built as a module and published as open-source, allowing users to expand on the existing framework.

Project Highlights

- PVDeg: open-source tool for degradation analysis.
- Searchable database of degradation parameters.
- Geospatial analysis via high-performance computing (HPC) and in AWS.
- Multivariate Monte Carlo methods to quantify error in calculations.

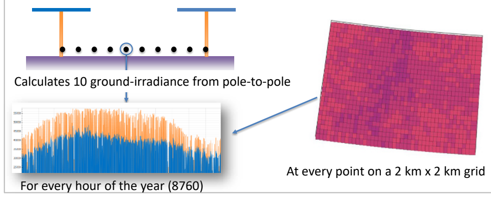
Impact

- Degradation occurs due to many different modes and mechanisms, each with different dependencies on stress factors. To develop a 50-year module, one must individually address every failure mode and eliminate or minimize it.
- The most difficult task in performance prediction is determining a degradation model and obtaining relevant degradation parameters. Next, extrapolation is preformed using time-consuming, repetitive and standardized methods.

SAM Integration

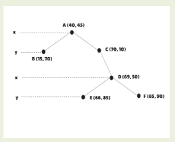
PVDeg seamlessly integrates the latest SAM versions and modules into geospatial workflows using PySAM. Its coding structure closely follows PySAM’s, minimizing the learning curve and eliminating the need to learn a new platform or package. Additionally, it is designed to easily incorporate future updates, ensuring continuous compatibility. PVDeg supports both **pvsamv1** and **pvwatts8,** enabling detailed geospatial workflows with undress of configurable parameters. Users can define setups in the SAM GUI and seamlessly integrate them into their analyses.

PVDeg+PySAM Agrivoltaics example



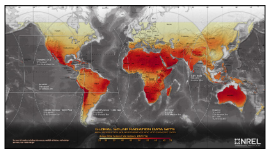
GeoGridFusion: Local Geospatial

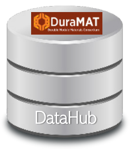
GeoGridFusion enables users outside of NREL, particularly those without access to HPC resources, to utilize and store gridded geospatial data from datasets like NSRDB and PVGIS. The goal of this project is to make our advancements widely accessible.



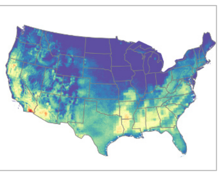
PV Degradation Tools – The PV-focused, open-source, integration pipeline for PV degradation analysis!

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





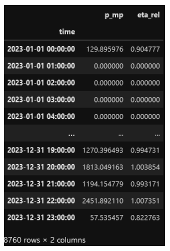
$$R_D = R_o G^p e^{\left(\frac{-E_a}{RT}\right)}$$



PVDeg Geospatial Workflow

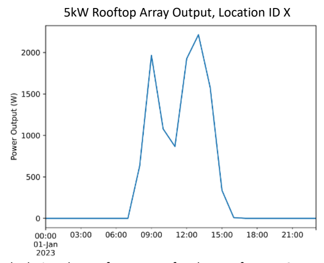
As a simple example for our geospatial framework, we calculate PV system DC output using the ADR module efficiency model with PVLIB[1]. Our goal is to demonstrate how existing tools like the functions in PVLIB can be integrated with PVDeg for geospatial calculations.

This framework requires a function that accepts weather data (as a Pandas Dataframe) and metadata (as a dictionary), including latitude, longitude and other relevant details, and produces a result with uniform structure.

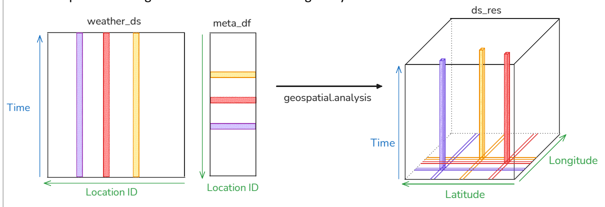


time	p_mp	eta_rel
2023-01-01 00:00:00	129.895976	0.904777
2023-01-01 01:00:00	0.000000	0.000000
2023-01-01 02:00:00	0.000000	0.000000
2023-01-01 03:00:00	0.000000	0.000000
2023-01-01 04:00:00	0.000000	0.000000
...	...	...
2023-12-31 19:00:00	1270.396493	0.994731
2023-12-31 20:00:00	1813.049163	1.003854
2023-12-31 21:00:00	1194.154779	0.993171
2023-12-31 22:00:00	2451.892110	1.007351
2023-12-31 23:00:00	57.535457	0.822763

8760 rows × 2 columns

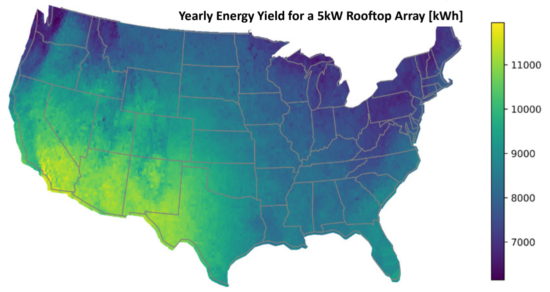


Example output from a function calculating the performance of a 5kW rooftop PV System. The output is in DataFrame format, but PVDeg also supports single or multiple variables in numeric or time-series form. The plot on the right shows the data for a single day at one location



The figure above illustrates the input data structure and transformation for geospatial calculations. The weather dataset has two axes: Location ID and Time, with each Location ID containing all weather data for that location over time. Bars represent TMY data for the three locations. Metadata is stored in a DataFrame, where each row corresponds to a location’s metadata.

The mapping process transforms these inputs into a structured output with latitude and longitude dimensions. Depending on the model, the dataset may include additional dimensions such as time.



Example: 5kW Rooftop Performance

```python
def dc_output(weather_df, meta):
    solpos = pvdeg.spectral.solar_position(
        weather_df = weather_df,
        meta = meta
    )

    # Define panel orientation
    tilt, orient = meta['latitude'], 180

    # Compute plane-of-array (POA) irradiance
    total_irrad = get_total_irradiance(
        tilt, orient, solpos.apparent_zenith, solpos.azimuth,
        weather_df.dni, weather_df.ghi, weather_df.dhi
    )
    weather_df['poa_global'] = total_irrad.poa_global

    # Estimate PV module temperature
    weather_df['temp_pv'] = pvlib.temperature.faiman(
        weather_df.poa_global, weather_df.temp_air, weather_df.wind_speed
    )

    # ADR model parameters
    adr_params = {
        'k_a': 0.99924, 'k_d': -5.49097, 'tc_d': 0.01918,
        'k_rs': 0.06999, 'k_rsh': 0.26144
    }

    # Compute efficiency
    weather_df['eta_rel'] = pvefficiency_adr(
        weather_df.poa_global, weather_df.temp_pv, **adr_params
    )

    # Compute power output
    P_STC, G_STC = 5000.0, 1000.0  # Standard conditions (W, W/m²)
    weather_df['p_mp'] = P_STC * weather_df.eta_rel *
   (weather_df.poa_global / G_STC)

    return weather_df[['p_mp', 'eta_rel']]
```

Geospatial Calculation

To adapt the function above for geospatial calculations on Terabyte-scale datasets, we must describe a mapping of the input data to the output in the form of a template.

```python
from pvdeg.decorators import geospatial_quick_shape

@geospatial_quick_shape('timeseries', ['p_mp', 'eta_rel'])
def dc_output(weather_df, meta):
    ...
```

```python
template=pvdeg.geospatial.output_template(
    ds_gids=geo_weather,
    shapes = {
        'p_mp': ('gid','time'),
        'eta_rel': ('gid','time'),
    }
)
```

Finally, we provide an Xarray Dataset of weather data and a Pandas DataFrame of metadata containing meteorological data for all locations and perform calculations in parallel.

Takeaways

- Simplify geospatial calculations with PVDeg.
- Apply single-site functions across geospatial data.
- Run geospatial PySAM models natively.
- Store and merge multiple geospatial datasets locally, using GeoGridFusion.

[1]A. Driesse, “Simulating PV system DC output using the ADR module efficiency model — pvlib python 0.11.2 documentation,” *Readthedocs.io*, 2020. https://pvlib-python.readthedocs.io/en/stable/gallery/adr-pvarray/plot_simulate_system.html#sphx-glr-gallery-adr-pvarray-plot-simulate-system-py

This work was authored by NREL for the U.S. Department of Energy (DOE), operated under Contract No. DE-AC36-08GO28308. 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. (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. 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.

Photovoltaic Reliability Workshop (PVRW)
March 4-6, 2025
Golden, Colorado
NREL/PO-5K00-93424