ERADIATE

A radiative transfer model for the Earth observation community

Vincent Leroy, Yves Govaerts, Yvan Nollet, Sebastian Schunke, Nicolas Misk // Rayference

CEOS WGCV IVOS 35 // 27th September 2023 // DLR, Oberpfaffenhofen, Germany

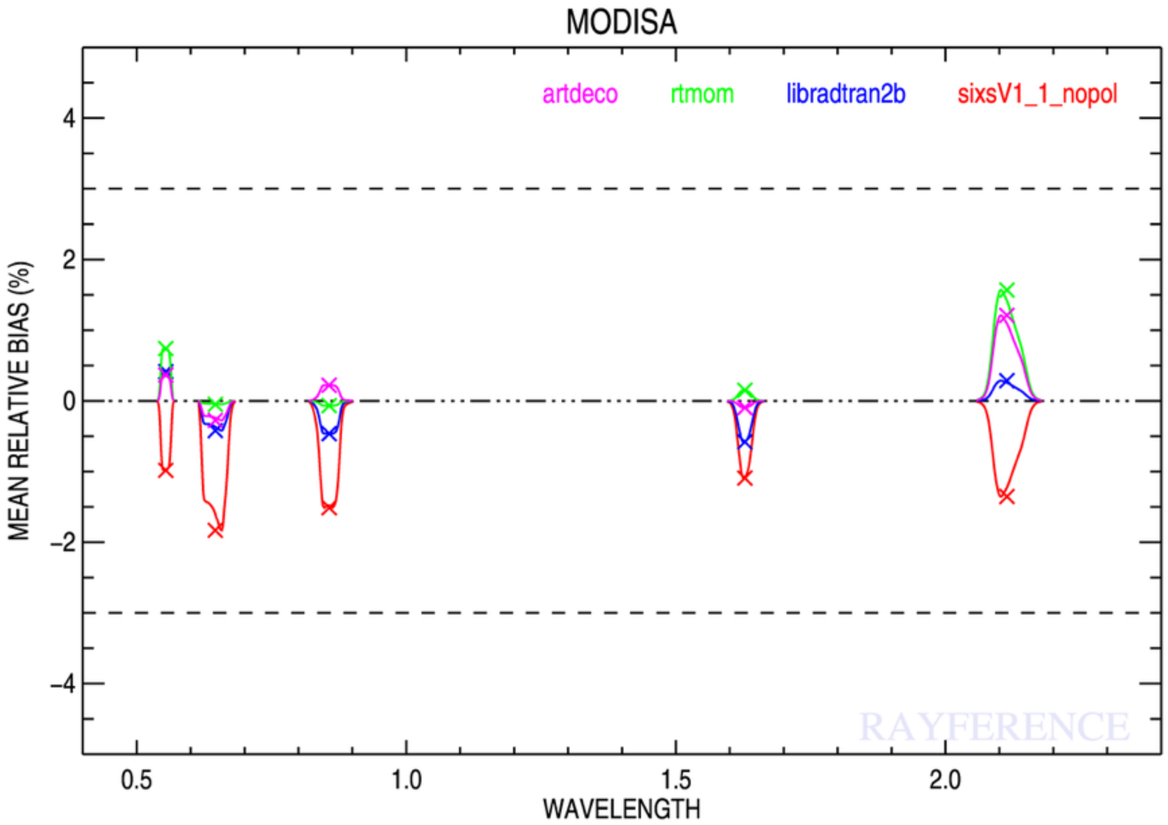
esa

Copernicus
Europe's eyes on Earth

RAYFERENCE

The root cause: cal/val requires more accurate RTMs

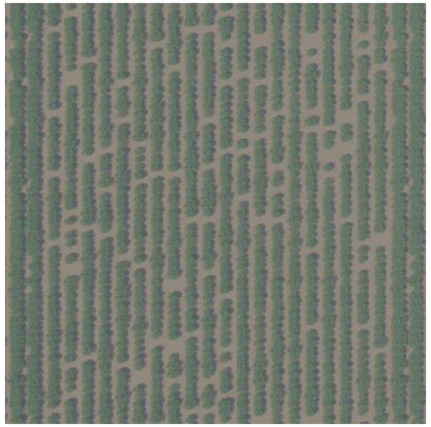
- Modern instruments can reach an accuracy of 2-3% and better
- We use RTMs as part of the calibration process in many cases
- Depending on the one you pick, you won’t get the same result
  - Even if you set up the same simulation
  - Uncertainty can go up to 3-4%
  - Attributable to many different factors; an important driver is the atmospheric molecular absorption modelling (usage of standard profiles, without accounting for molecular concentration changes)

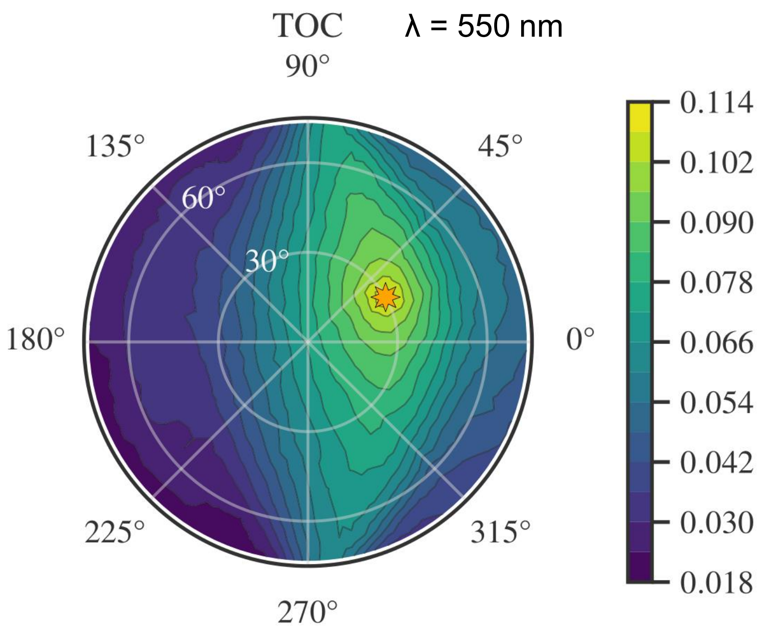


Govaerts (2019), work order for Eradiate Development Phase 1

The root cause: cal/val requires more accurate RTMs

- Most RTMs currently used for cal/val applications assume a plane-parallel geometry with a flat surface
  - Earth is not flat!
  - It has relief and objects at its surface

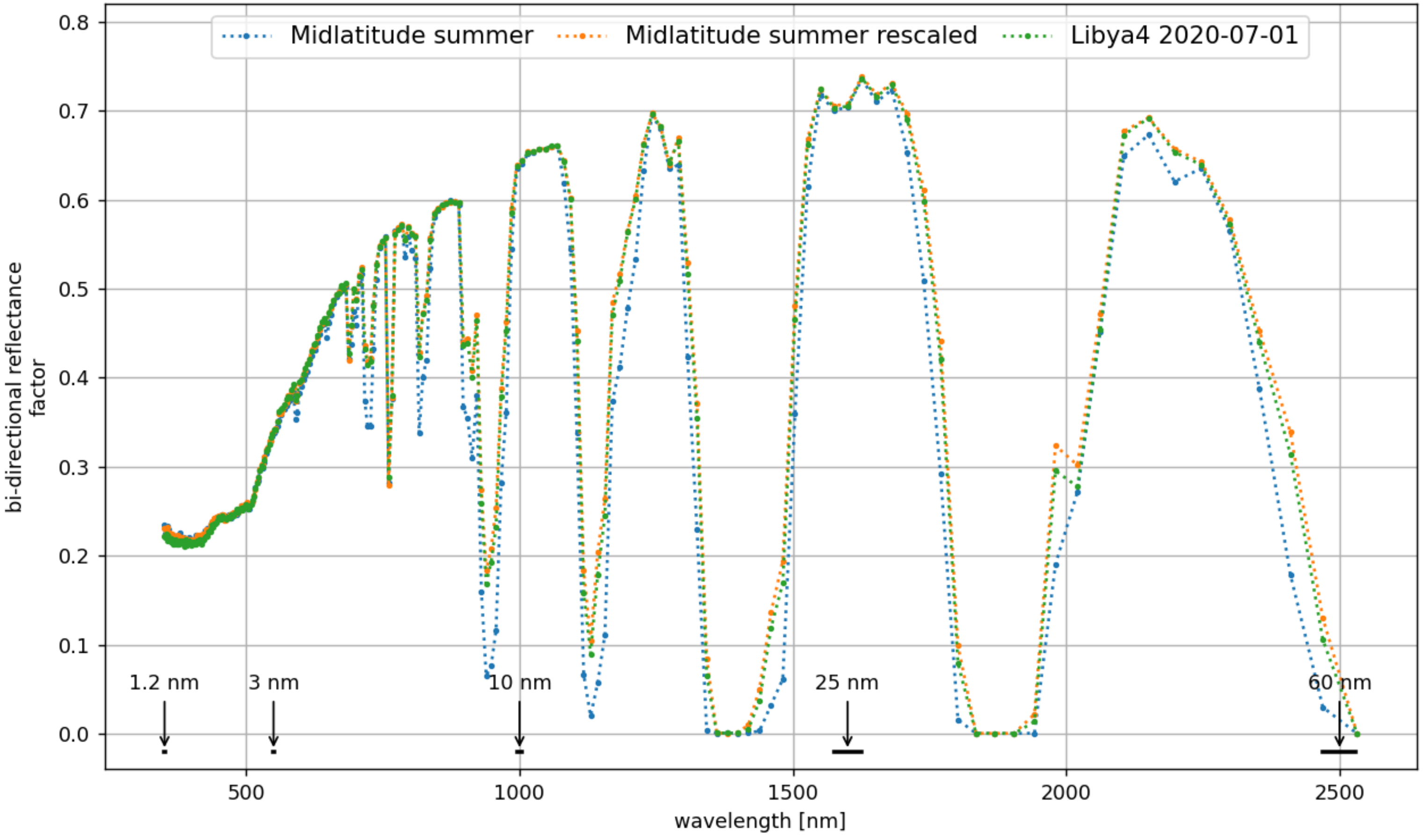




This is why we started building Eradiate
But how do we build such a model?

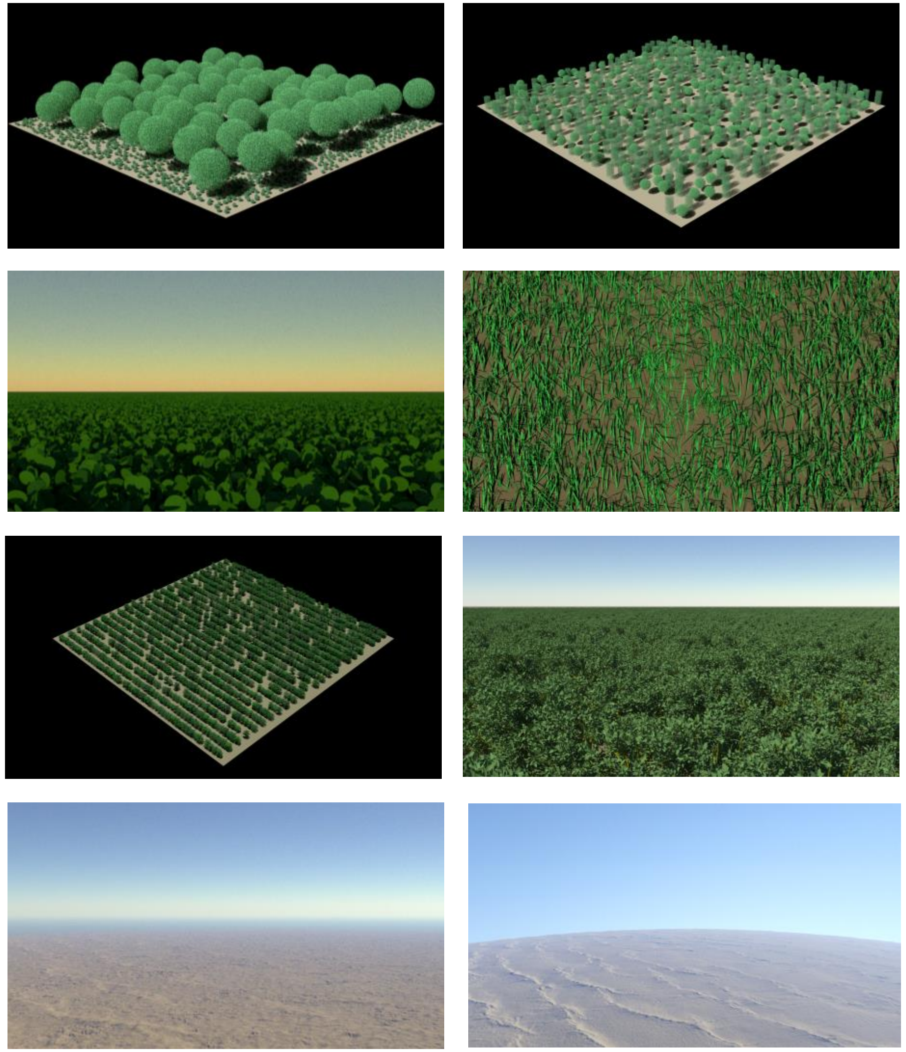
Atmospheric modelling

- Correlated k-distribution method
  - Very common (flexible, simple, reasonably accurate) ⇒ easy to compare to other RTMs
- Support of detailed thermophysical profiles
  - Standard profiles (AFGL 1986)
  - Molecular concentration rescaling
  - Fully customizable profiles (arbitrary input, *e.g*. CAMS data)



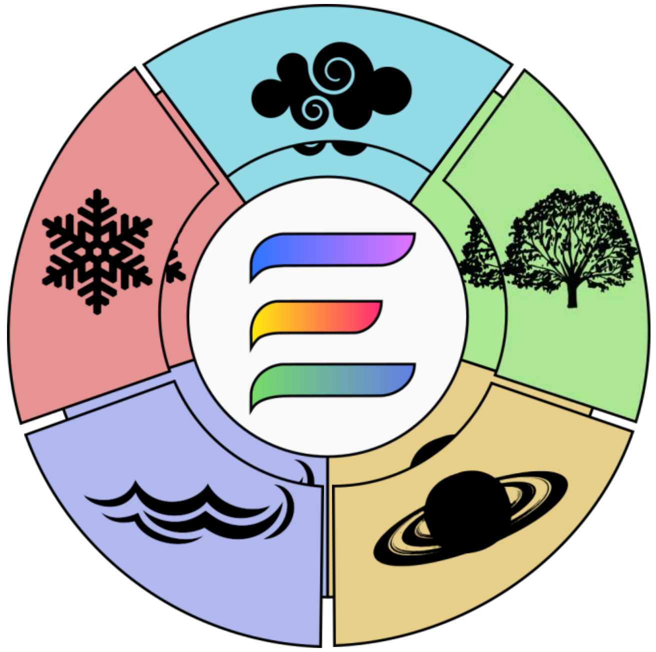
3D surface modelling

- Vegetation: detailed plant modelling
- Relief: support for digital elevation models
- Planetary curvature: spherical-shell geometry
- Arbitrary complexity
  ⇒ **Monte Carlo ray tracing is the way**



Breaking walls between communities

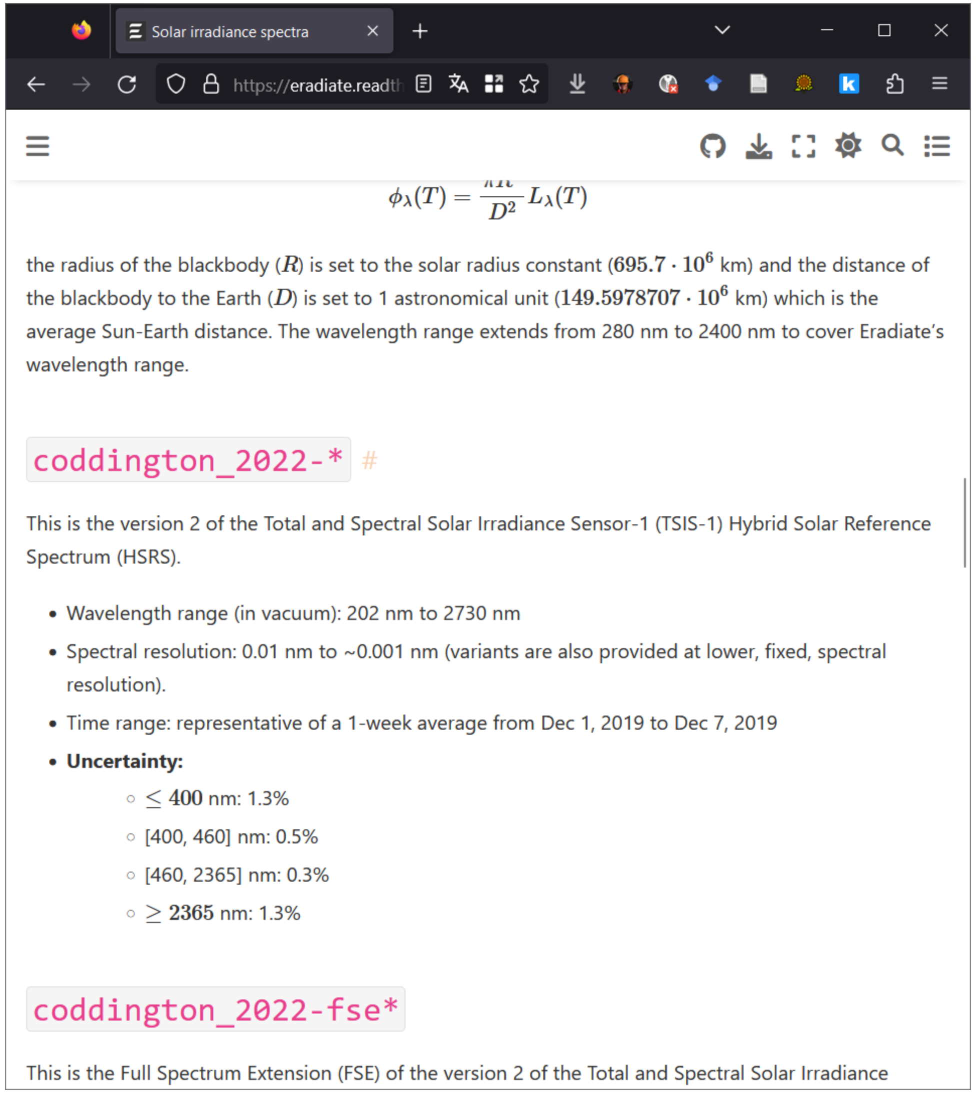
- RT modelling is done by compartmentalized subcommunities
  ⇒ many models, generally with a specific focus
  ⇒ highly accurate modelling requires convergence
- Very impressive work on software infrastructure by computer graphics community
  ⇒ import technology from there
  ⇒ Eradiate's radiometric kernel is rendering software
- Modular architecture: easy to add new components without knowing/breaking everything



Beyond physics: Software engineering

It’s cal/val: traceability matters

- Data traceability: explain where shipped data comes from
- Algorithm transparency: it’s fully open-source



Solar irradiance spectra

https://eradiate.readth

$$\phi_\lambda(T) = \frac{\pi R^2}{D^2} L_\lambda(T)$$

the radius of the blackbody (R) is set to the solar radius constant ($695.7 \cdot 10^6$ km) and the distance of the blackbody to the Earth (D) is set to 1 astronomical unit ($149.5978707 \cdot 10^6$ km) which is the average Sun-Earth distance. The wavelength range extends from 280 nm to 2400 nm to cover Eradiate’s wavelength range.

`coddington_2022-*`

This is the version 2 of the Total and Spectral Solar Irradiance Sensor-1 (TSIS-1) Hybrid Solar Reference Spectrum (HSRS).

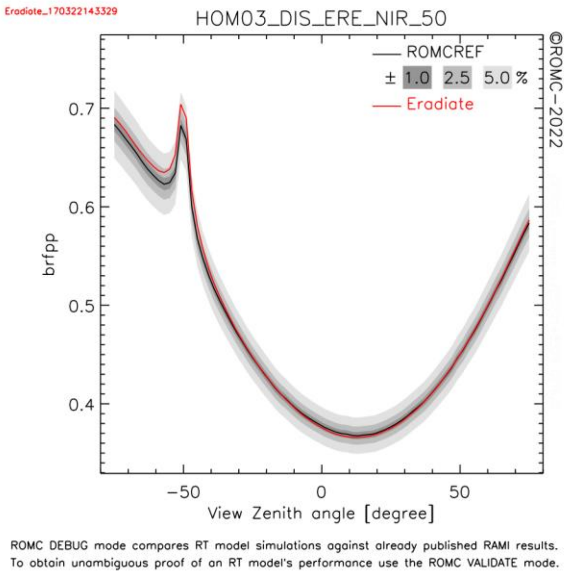
- Wavelength range (in vacuum): 202 nm to 2730 nm
- Spectral resolution: 0.01 nm to ~0.001 nm (variants are also provided at lower, fixed, spectral resolution).
- Time range: representative of a 1-week average from Dec 1, 2019 to Dec 7, 2019
- **Uncertainty:**
  - ≤ 400 nm: 1.3%
  - [400, 460] nm: 0.5%
  - [460, 2365] nm: 0.3%
  - ≥ 2365 nm: 1.3%

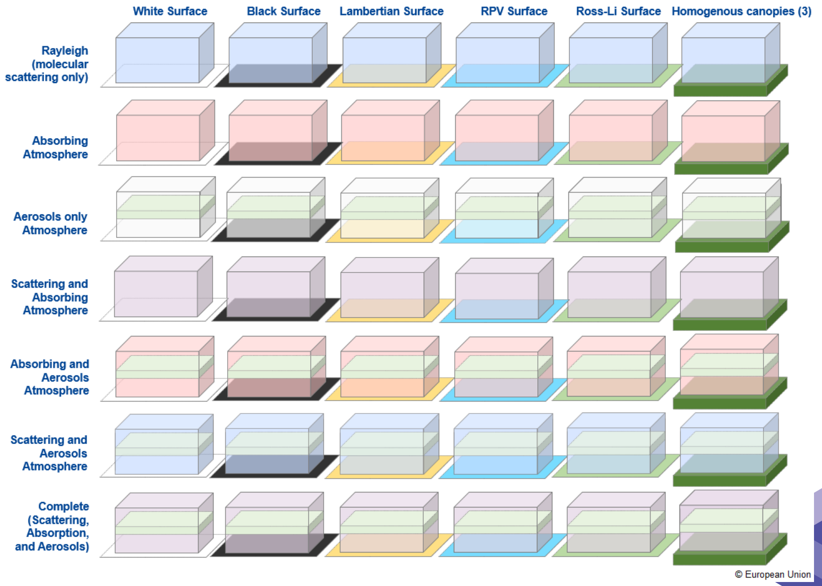
`coddington_2022-fse*`

This is the Full Spectrum Extension (FSE) of the version 2 of the Total and Spectral Solar Irradiance

We want to trust it

- Regular and systematic testing
  - Unit tests for individual components
  - Larger use case-based system tests
- Benchmarking
  - RAMI series (ROMC / RAMI-V / RAMI4ATM)
  - Extension to more relevant benchmarks (*e.g.* IPRT)





Usability in mind

Highly customizable input

- NetCDF-centric workflow
- Easy to bring your own data

Python API

- Interactive usage
  ⇒ Jupyter notebooks
- Scriptable
  ⇒ write you own app

Smooth learning curve

- Thorough documentation
- Progressive tutorials

```python
import eradiate.scenes as ertsc
import eradiate.experiments as ertxp

exp = ertxp.OneDimExperiment(
    surface=ertsc.bsdfs.RPVBSDF(),
    atmosphere=ertsc.atmosphere.MolecularAtmosphere.afgl_1986(),
    illumination=ertsc.illumination.DirectionalIllumination(
        zenith=15.0,
        azimuth=0.0,
    ),
    measures=ertsc.measure.MultiDistantMeasure.from_viewing_angles(
        id="toa_brf",
        zeniths=np.arange(-75, 76, 5),  # Cover the [-75°, 75°] range with 5°
        azimuths=0,  # Same value as SAA to cover the principal plane
        spectral_cfg={
            "srf": "sentinel_2a-msi-5",
            "bin_set": "10nm",
        },
        spp=1000,
    ),
)
```

```
[7]: results = eradiate.run(exp)
     results
```

Spectral loop [710:15]: 48/48 00:05, ETA=00:00

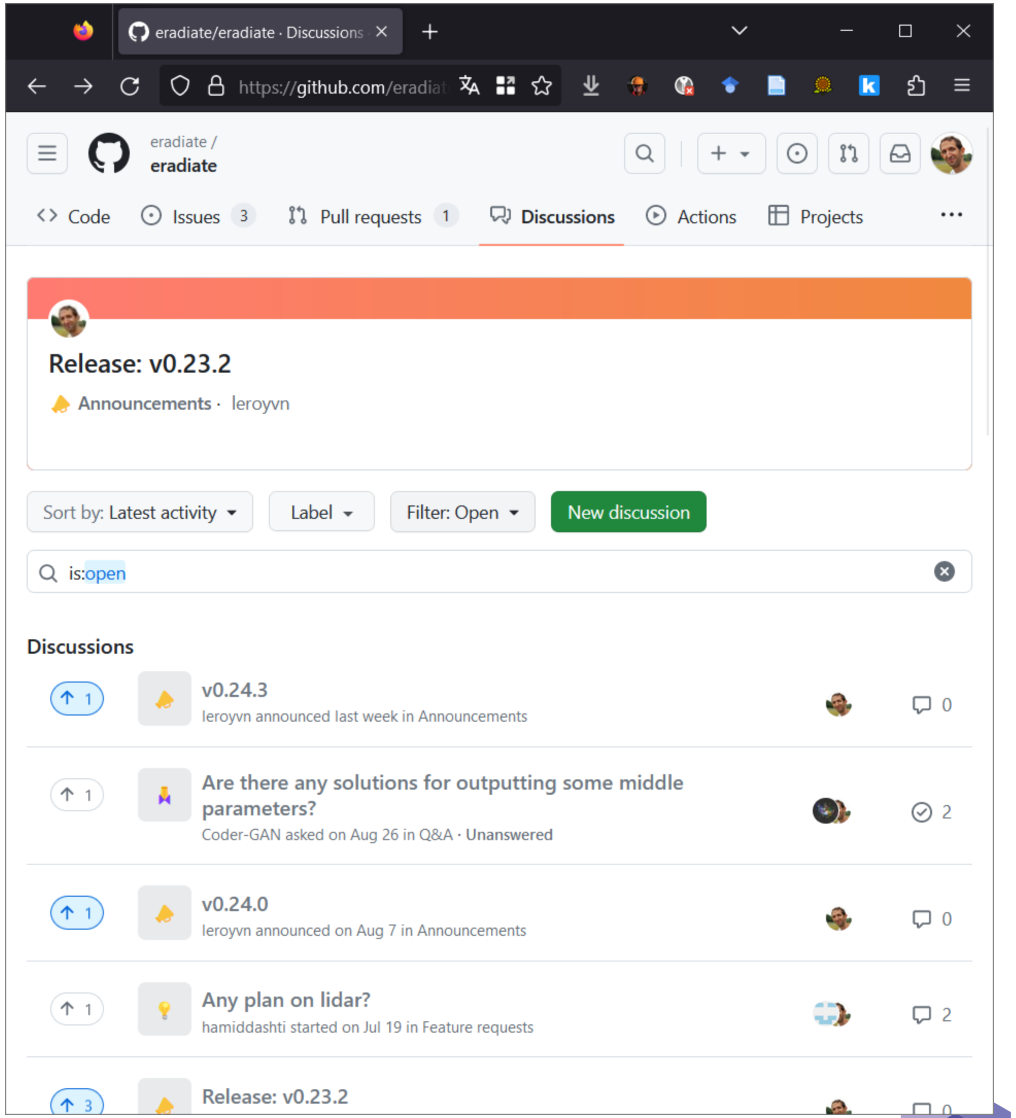
[7]: xarray.Dataset

- Dimensions: (**sza**: 1, **saa**: 1, **w**: 3, **y_index**: 1, **x_index**: 31, **srf_w**: 22)
- Coordinates: (13)
- Data variables:

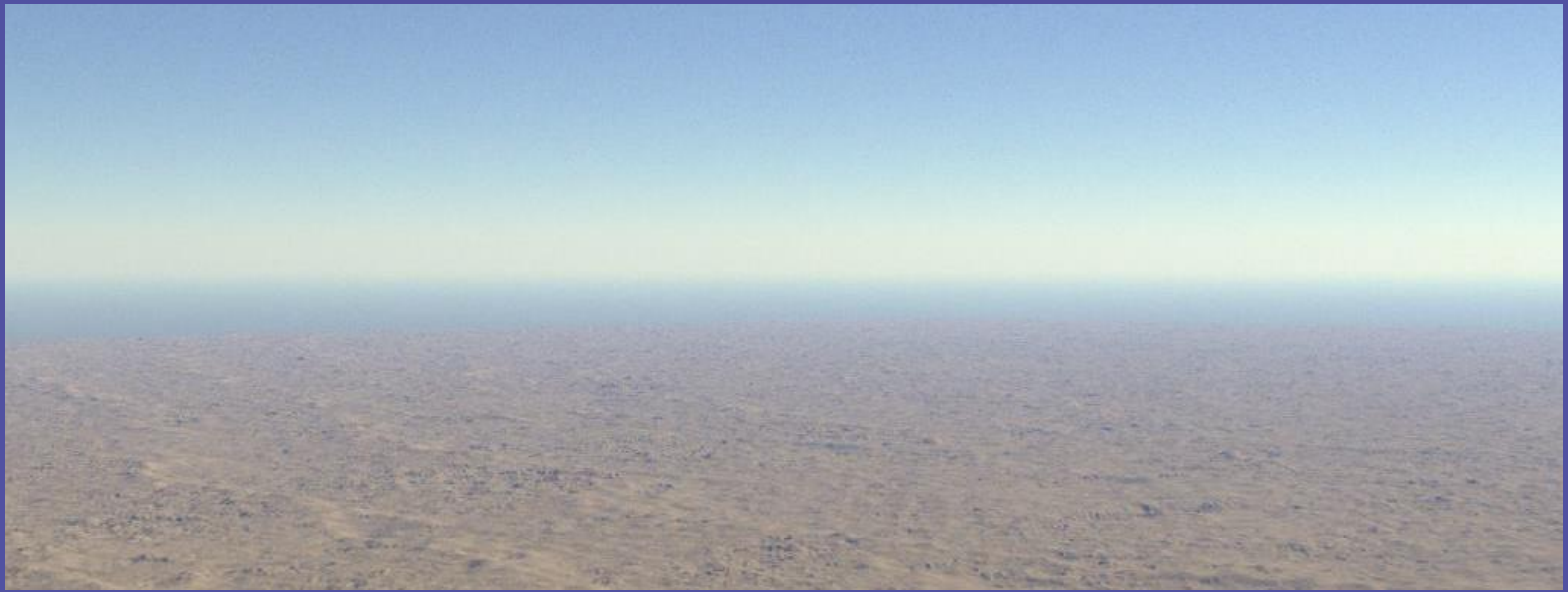
radiance	(sza, saa, w, y_index, x_index)	float64	0.09475 0.0949 ......
spp	(sza, saa, w)	float64	1e+03 1e+03 1e+03
irradiance	(sza, saa, w)	float64	1.412 1.39 1.349
srf	(srf_w)	float64	0.0 0.02836 0.123...
brdf	(sza, saa, w, y_index, x_index)	float64	0.06711 0.06722

It’s built for the community

- Open-source (LGPL) hosted on GitHub
- Contributions welcome
  - Request features or datasets
  - Issue/bug reports
  - Improvements and fixes to documentation / code
  - Additional data
- A problem? Come and open a discussion!



Visit eradiate.eu
Follow us on GitHub github.com/eradiate/eradiate
Subscribe to our newsletter



Libya-4 DTM with nonabsorbing atmosphere

Built with support from

Copernicus
Europe's eyes on Earth

esa