skysim Documentation

Release 0.1.dev54

David Kirkby

Sep 11, 2019

CONTENTS

This is the documentation for skysim: a package to simulate the optical spectrum of sky brightness and extinction.

This package implements models described in:

- Noll 2012, “An atmospheric radiation model for Cerro Paranal. I. The optical spectral range”, 10.1051/0004-6361/201219040
- Leinert 1998, “The 1997 reference of diffuse night sky brightness”, 10.1051/aas:1998105

Part I

Links

- Source code
- Docs
- Issues

Part II

General Documentation

CHAPTER

ONE

INSTALLATION

1.1 Requirements

skysim requires python 3.6 or greater, as well as the following packages:

- numpy
- scipy
- pyyaml
- healpy
- astropy

CHAPTER

TWO

REFERENCE/API

2.1 skysim.airglow Module

Compute optical airglow emission and transmission.

Refer to Section 4 of Nolll 2012 for details.

2.1.1 Functions

airglow_scattering(z)	Calculate net Rayleigh and Mie scattering of airglow emission.
airmass_ag(z)	Calculate the effective airmass of airglow emission from ~90km.
get_airglow(lam, z[, p, H, Rayleigh, Mie, ...])	Calculate airglow flux.

airglow_scattering

skysim.airglow.**airglow_scattering**(*z*)

Calculate net Rayleigh and Mie scattering of airglow emission.

Use equations (24) and (25) of Noll 2012 to approximate the net effect of scattering as an optical depth multiplier. Negative values are possible and indicate that scattering of indirect airglow into the line of sight exceeds scattering of direct airglow out of the line of sight.

Parameters

z
[float or array] Zenith angle(s) in degrees.

Returns

tuple
Tuple (fR, fM) giving the net optical depth multipliers for Rayleigh and Mie scattering, respectively. The components fR, fM will be floats or arrays matching the input z shape.

airmass_ag

skysim.airglow.**airmass_ag**(*z*)

Calculate the effective airmass of airglow emission from ~90km.

Use equation (23) of Noll 2012.

Parameters

z
[float or array] Zenith angle(s) in degrees.

Returns

float or array
Airmass(es) corresponding to each input zenith angle.

get_airglow

`skysim.airglow.`**`get_airglow`**(*lam*, *z*, *p=744.0*, *H=2.64*, *Rayleigh=True*, *Mie=True*, *absorption=True*)

Calculate airglow flux.

Automatically broadcasts over input arrays, but the wavelength input `lam` must be 1D and appear in the last axis.

The CPU time scales with the number of unique values in `z` so calculating on a grid with many duplicates of the same value is relatively efficient (e.g., `skysim.utils.AltAzGrid`).

Parameters

lam
[float or 1D array] Wavelength in nanometers.

z
[float or array] Zenith angle(s) in degrees.

p
[float or array] Pressure at the observation elevation in hPa, used for Rayleigh scattering.

H
[float or array] Elevation of the observation in km, used for Rayleigh scattering.

Rayleigh
[bool] Apply Rayleigh scattering effects.

Mie
[bool] Apply aerosol Mie scattering effects.

absorption
[bool] Apply molecular (but not ozone) absorption effects.

Returns

tuple
Tuple (cont, line) of arrays of airglow continuum and line fluxes in ph / (s cm2 nm).

2.2 skysim.transmission Module

Compute scattering and absorption effects on transmission.

Refer to Section 2 of Noll 2012 for details.

2.2.1 Functions

`tau0M`(lam[, lam0, k0, alpha])	Calculate zenith optical depth due to Mie scattering off aerosols.
`tau0R`(lam[, p, H])	Calculate zenith optical depth due to Rayleigh scattering.
`tau0ma`(lam)	Calculate zenith optical depth of molecular absorption.
`tau0o3`(lam)	Calculate zenith optical depth of ozone absorption.

tau0M

`skysim.transmission.`**`tau0M`**(*lam*, *lam0=400.0*, *k0=0.013*, *alpha=-1.38*)

Calculate zenith optical depth due to Mie scattering off aerosols.

Use equation (4) of Noll 2012:

$$k^M(\lambda) = k_0 \lambda^\alpha$$

Parameters

lam
[float or array] Wavelength in nanometers.

lam0
[float] Optical depth is constant below this wavelength in nanometers.

k0
[float] Extinction at 1000nm in mag / airmass.

alpha
[float] Extinction wavelength power.

tau0R

`skysim.transmission.`**`tau0R`**(*lam*, *p=744.0*, *H=2.64*)

Calculate zenith optical depth due to Rayleigh scattering.

Use equation (3) of Noll 2012:

$$\tau_0^R(\lambda) = \frac{p}{1013.25\,\mathrm{hPa}} \left(0.00864 + 6.5 \times 10^{-6} \frac{H}{1\,\mathrm{km}}\right) \lambda^{-(0.3916 + 0.074\lambda + 0.050/\lambda)}$$

Automatically broadcasts over any input arrays.

Parameters

lam
[float or array] Wavelength in nanometers.

p
[float or array] Pressure at the observation elevation in hPa.

H
[float or array] Elevation of the observation in km.

Returns

float or array
Optical depth(s) calculated for the input parameters.

tau0ma

`skysim.transmission.`**`tau0ma`**(*lam*)

Calculate zenith optical depth of molecular absorption.

The main absorbers are the molecular oxygen bands (A ~762nm, B ~688nm, gamma ~628nm) and water vapor bands (~720nm, 820nm, 940nm). See Section 2 and Fig.2 of Noll 2012 for details.

Narrow asorption features are resampled to the requested wavelength grid using a flux-conserving algorithm.

Parameters

lam
[float or array] Wavelength in nanometers.

Returns

float or array
Zenith optical depth of molecular absorption.

tau0o3

`skysim.transmission.`**`tau0o3`**(*lam*)

Calculate zenith optical depth of ozone absorption.

The main features are the Huggins band in the near-UV and the broad Chappuis bands around 600nm. See Section 2 and Fig.2 of Noll 2012 for details.

Narrow asorption features are resampled to the requested wavelength grid using a flux-conserving algorithm.

Parameters

lam
[float or array] Wavelength in nanometers.

Returns

float or array
Zenith optical depth of molecular absorption.

2.3 skysim.zodiacal Module

Compute optical zodiacal scattering and extinction.

See Section 3.3 of Noll 2012 for details.

2.3.1 Functions

airmass_zodi(zenith_angle)	Calculate the airmass used for zodiacal light calculations.
ecl_elong(ecl_lon, ecl_lat)	Calculate the elongation for a specified ecliptic longitude and latitude.
get_zodiacal(lam, ecl_lon, ecl_lat, z[, p, …])	Calculate zodiacal flux.
get_zodiacal_flux500(ecl_lon, ecl_lat)	Return zodical flux at 500nm incident above the atmosphere.
zodiacal_color_factor(lam, elong)	Calculate the redenning of the solar spectrum.
zodiacal_scattering(I0)	Calculate net Rayleigh and Mie scattering of zodiacal light.

airmass_zodi

`skysim.zodiacal.airmass_zodi(zenith_angle)`

Calculate the airmass used for zodiacal light calculations.

Use equation (2) of Noll 2012, which is originally from Rozenberg 1966.

Parameters

z
[float or array] Zenith angle(s) in degrees.

Returns

float or array
Airmass(es) corresponding to each input zenith angle.

ecl_elong

`skysim.zodiacal.ecl_elong(ecl_lon, ecl_lat)`

Calculate the elongation for a specified ecliptic longitude and latitude.

Use equation (11) of Leinert 1998:

$$\cos\epsilon = \cos(\lambda - \lambda_\odot)\cos\beta$$

Parameters

ecl_lon
[float or array] Heliocentric ecliptic longitude in degrees.

ecl_lat
[float or array] Ecliptic latitude in degrees.

Returns

float or array
Ecliptic elongation in degrees.

get_zodiacal

`skysim.zodiacal.`**`get_zodiacal`**(*lam*, *ecl_lon*, *ecl_lat*, *z*, *p=744.0*, *H=2.64*, *redden=True*, *Rayleigh=True*, *Mie=True*, *absorption=True*)

Calculate zodiacal flux.

Automatically broadcasts over input arrays, but the wavelength input `lam` must be 1D and appear in the last axis.

Parameters

lam
[float or 1D array] Wavelength in nanometers.

ecl_lon
[float or array] Heliocentric ecliptic longitude in degrees.

ecl_lat
[float or array] Ecliptic latitude in degrees.

z
[float or array] Zenith angle in degrees, used to calculate airmass.

p
[float] Pressure at the observation elevation in hPa, used for Rayleigh scattering.

H
[float] Elevation of the observation in km, used for Rayleigh scattering.

redden
[bool] Apply redenning of solar spectrum.

Rayleigh
[bool] Apply Rayleigh scattering effects.

Mie
[bool] Apply aerosol Mie scattering effects.

absorption
[bool] Apply molecular (but not ozone) absorption effects.

Returns

float or array
Array of radiances corresponding to each input wavelength, in units of ph / (arcsec2 m2 s nm).

get_zodiacal_flux500

`skysim.zodiacal.`**`get_zodiacal_flux500`**(*ecl_lon*, *ecl_lat*)

Return zodical flux at 500nm incident above the atmosphere.

Automatically broadcasts over ecl_lon and ecl_lat. Return value is scalar when both inputs are scalar.

Parameters

ecl_lon
[float or array] Heliocentric ecliptic longitude in degrees.

ecl_lat
[float or array] Ecliptic latitude in degrees.

Returns

float or array
Zodiacal flux at 500nm incident above the atmosphere, in units of 1e-8 W / (m2 sr um).

zodiacal_color_factor

`skysim.zodiacal.`**`zodiacal_color_factor`**(*lam*, *elong*)

Calculate the redenning of the solar spectrum.

Interpolate in ecliptic longitude between the color factors calculated with equation (22) of Leinert 1998, as described in Section 8.4.2.

The output is automatically broadcast over its inputs.

Parameters

lam
[float or array] Observed wavelength in nm.

elong
[float or array] Ecliptic elongation angle in degrees, which can be obtained using `ecl_elong()`.

Returns

float or array
The color correction factor(s) to apply to the solar spectrum.

zodiacal_scattering

`skysim.zodiacal.`**`zodiacal_scattering`**(*I0*)

Calculate net Rayleigh and Mie scattering of zodiacal light.

Use equations (19) and (20) of Noll 2012 to approximate the net effect of scattering as an optical depth multiplier. Negative values are possible and indicate that scattering of indirect zodiacal light into the line of sight exceeds scattering of direct zodiacal light out of the line of sight.

Parameters

I0
[float or array] Zodiacal flux in 1e-8 W / (m2 sr um). All values must be >= 0.

Returns

tuple
Tuple (fR, fM) giving the net optical depth multipliers for Rayleigh and Mie scattering, respectively. The components fR, fM will be floats or arrays matching the input z shape. Returns zero when I0 is zero.

PYTHON MODULE INDEX

S

INDEX

A

E

G

S

T

Z