flatstar

Release v0.2.1a

Leonardo A. dos Santos

May 19, 2022

CONTENTS:

1 flatstar API **3**

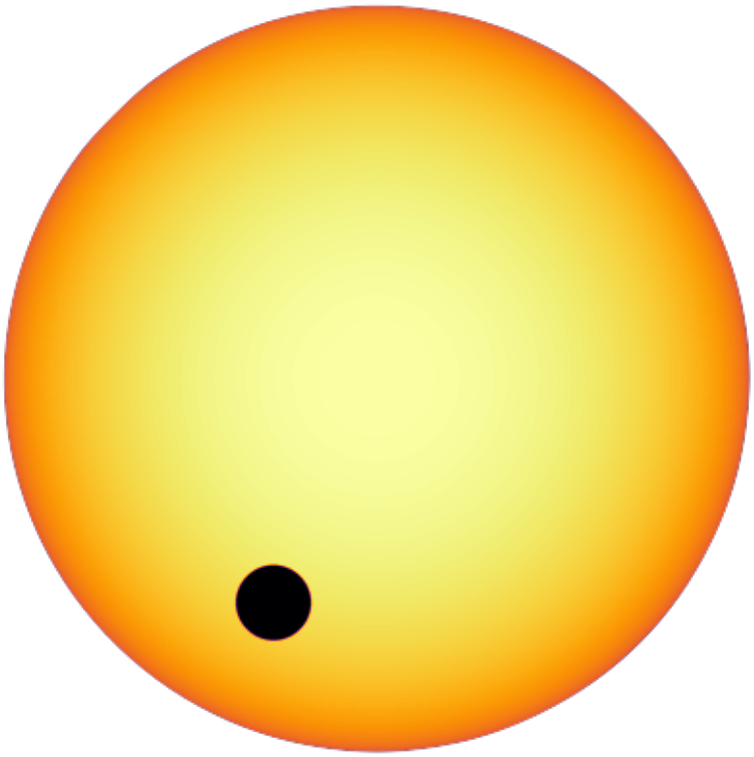
1.1 *draw* 3

1.2 *limb_darkening* 4

Python Module Index **7**

Index **9**

`flatstar` is an open-source Python tool for drawing stellar disks as `numpy.ndarray` objects with scientifically-rigorous limb darkening. Each pixel has an accurate fractional intensity in relation to the total stellar intensity of `1.0`. It is ideal for ray-tracing simulations of stars and planetary transits.



You can contribute to the project using GitHub.

CHAPTER

ONE

FLATSTAR API

1.1 *draw*

`draw.star`(*grid_size*, *radius=0.5*, *limb_darkening_law=None*, *ld_coefficient=None*, *custom_limb_darkening=None*, *supersampling=None*, *upscaling=None*, *resample_method=None*)

Make a normalized drawing of a star with a corresponding limb-darkening law in a square grid. The normalization is made in such a way that the flattened sum of the values inside the two-dimensional array is equal to 1.0. The normalization factor is calculated before the resampling, so more complex resampling algorithms may produce more inaccurate normalizations (by a factor of a few to hundreds of ppm) depending on the requested grid size and supersampling factor. If very precise normalized maps are required, then it is better to not use supersampling or use a `"box"` resampling algorithm.

Parameters

grid_size (``int``): Size of the square grid in number pixels.

radius (``int`` or ``float``, optional): Stellar radius in units of `grid_size`. Default is 0.5.

limb_darkening_law (``None`` or ``str``, optional): String with the name of the limb-darkening law. The options currently implemented are: `'linear'`, `'quadratic'`, `'square-root'`, `'logarithmic'` (or `'log'`), `'exponential'` (or `'exp'`), `'sing-three'` (or `'sing'`, or `'s3'`), `'claret-four'` (or `'claret'`, or `'c4'`), `None` (no limb-darkening), or `'custom'`. In case you choose the latter, you need to provide a callable function that defines your custom law using the parameter `custom_limb_darkening`. Default is `None`.

ld_coefficient (``float`` or ``array-like``): In case of a linear limb-darkening law, the value of the coefficient should be a float. In all other options it should be array-like. Default is `None`.

custom_limb_darkening (``callable`` or ``None``, optional) In case you want to use a custom limb-darkening law, you need provide a function that defines it. The first parameter of this function must be `mu` (cosine of the angle between a line normal to the stellar surface and the line of sight), and the second must be the coefficient (in case it uses multiple coefficients, it must accept them as an array-like object). Default is `None`.

supersampling (``int``, ``float``, or ``None``, optional): For low-resolution grid sizes, in order to avoid intensity maps with hard edges, you can supersample the array by a certain factor defined by this parameter, and then the map is downscaled to your requested grid size using the algorithm defined in `resample_method`. Default is `None` (no supersampling).

upscaling (``int``, ``float``, or ``None``, optional): For fast output of high-resolution grids, you may want to upscale them from a low-resolution setup to save about one order of magnitude in computation time. This parameter is the factor by which to upscale the grids to match the requested grid size. The resizing algorithm is defined in `resample_method`. Default is `None` (no upscaling).

resample_method (``str`` or ``None``, optional): Resampling algorithm. The options currently available are: `"nearest"`, `"box"`, `"bilinear"`, `"hamming"`, `"bicubic"`, and `"lanczos"`. If `None`, then fallback to `"box"`. Default is `None`.

Returns

grid (`flatstar.utils.StarGrid` object): Intensity map of the star.

`draw.`**`planet_transit`**(*star_grid*, *planet_to_star_ratio*, *impact_parameter=0.0*, *phase=0.0*, *rescaling_factor=None*, *resample_method=None*)

Draw a transit in the `StarGrid` object.

Parameters

star_grid (``flatstar.utils.StarGrid`` object):

planet_to_star_ratio (``float``): Ratio between the radii of the planet and the star.

impact_parameter (``float``, optional): Impact parameter of the transit in units of stellar radii. Default is 0.

phase (``float``, optional): Phase of the transit. -0.5, 0.0, and +0.5 correspond respectively to the time of first contact, transit mid-center, and time of fourth contact. Default is 0.

rescaling_factor (``float`` or ``None``, optional) Resize the grid by a factor defined by this parameter. If `None`, no resizing is performed. Default is `None`.

resample_method (``str`` or ``None``, optional): Resampling algorithm. The options currently available are: `"nearest"`, `"box"`, `"bilinear"`, `"hamming"`, `"bicubic"`, and `"lanczos"`. If `None`, then fallback to `"box"`. Default is `None`.

Returns

star_grid (`flatstar.utils.StarGrid` object): Updated `StarGrid` object containing the transit.

1.2 *limb_darkening*

`limb_darkening.`**`linear`**(*mu*, *c*, *i0=1.0*)

Calculates the intensity of a given cell in the stellar surface using a linear limb-darkening law.

Parameters

mu (``float`` or ``numpy.ndarray``): Cosine of the angle between a line normal to the stellar surface and the line of sight.

c (``float``): Limb-darkening coefficient.

i0 (``float``, optional): Intensity without limb-darkening. Default is 1.0.

Returns

i_mu (`float` or `numpy.ndarray`): Intensity with limb-darkening. The format is the same as the input `mu`.

`limb_darkening.`**`quadratic`**(*mu*, *c*, *i0=1.0*)

Calculates the intensity of a given cell in the stellar surface using a quadratic limb-darkening law.

Parameters

mu (``float`` or ``numpy.ndarray``): Cosine of the angle between a line normal to the stellar surface and the line of sight.

c (``array-like``): Limb-darkening coefficients in the order c1, c2.

i0 (``float``, optional): Intensity without limb-darkening. Default is 1.0.

Returns

i_mu (float or numpy.ndarray): Intensity with limb-darkening. The format is the same as the input mu.

limb_darkening.**square_root**(*mu*, *c*, *i0=1.0*)

Calculates the intensity of a given cell in the stellar surface using a square-root limb-darkening law.

Parameters

mu (``float`` or ``numpy.ndarray``): Cosine of the angle between a line normal to the stellar surface and the line of sight.

c (``array-like``): Limb-darkening coefficients in the order c1, c2.

i0 (``float``, optional): Intensity without limb-darkening. Default is 1.0.

Returns

i_mu (float or numpy.ndarray): Intensity with limb-darkening. The format is the same as the input mu.

limb_darkening.**logarithmic**(*mu*, *c*, *i0=1.0*)

Calculates the intensity of a given cell in the stellar surface using a logarithmic limb-darkening law.

Parameters

mu (``float`` or ``numpy.ndarray``): Cosine of the angle between a line normal to the stellar surface and the line of sight.

c (``array-like``): Limb-darkening coefficients in the order c1, c2.

i0 (``float``, optional): Intensity without limb-darkening. Default is 1.0.

Returns

i_mu (float or numpy.ndarray): Intensity with limb-darkening. The format is the same as the input mu.

limb_darkening.**exponential**(*mu*, *c*, *i0=1.0*)

Calculates the intensity of a given cell in the stellar surface using an exponential limb-darkening law.

Parameters

mu (``float`` or ``numpy.ndarray``): Cosine of the angle between a line normal to the stellar surface and the line of sight.

c (``array-like``): Limb-darkening coefficients in the order c1, c2.

i0 (``float``, optional): Intensity without limb-darkening. Default is 1.0.

Returns

i_mu (float or numpy.ndarray): Intensity with limb-darkening. The format is the same as the input mu.

limb_darkening.**sing_three**(*mu*, *c*, *i0=1.0*)

Calculates the intensity of a given cell in the stellar surface using the Sing et al (2009) limb-darkening law.

Parameters

mu (``float`` or ``numpy.ndarray``): Cosine of the angle between a line normal to the stellar surface and the line of sight.

c (``array-like``): Limb-darkening coefficients in the order c1, c2, c3.

i0 (``float``, optional): Intensity without limb-darkening. Default is 1.0.

Returns

i_mu (float or numpy.ndarray): Intensity with limb-darkening. The format is the same as the input mu.

limb_darkening.**claret_four**(*mu*, *c*, *i0=1.0*)

Calculates the intensity of a given cell in the stellar surface using the Claret et al. (2000) limb-darkening law.

Parameters

mu (``float`` or ``numpy.ndarray``): Cosine of the angle between a line normal to the stellar surface and the line of sight.

c (``array-like``): Limb-darkening coefficients in the order c1, c2, c3, c4.

i0 (``float``, optional): Intensity without limb-darkening. Default is 1.0.

Returns

i_mu (float or numpy.ndarray): Intensity with limb-darkening. The format is the same as the input mu.

PYTHON MODULE INDEX

d

draw, 3

l

limb_darkening, 4

INDEX

C

claret_four() (*in module limb_darkening*), 6

D

draw
    module, 3

E

exponential() (*in module limb_darkening*), 5

L

limb_darkening
    module, 4
linear() (*in module limb_darkening*), 4
logarithmic() (*in module limb_darkening*), 5

M

module
    draw, 3
    limb_darkening, 4

P

planet_transit() (*in module draw*), 4

Q

quadratic() (*in module limb_darkening*), 4

S

sing_three() (*in module limb_darkening*), 5
square_root() (*in module limb_darkening*), 5
star() (*in module draw*), 3