
SysSimPyMMEN

Release 0.1.0

Matthias Yang He

Mar 21, 2023

CONTENTS

1	Publication	3
2	Contents	5
2.1	Installation	5
2.2	Basic usage	6
2.3	Fitting individual systems	9
2.4	Detailed API	14
	Python Module Index	29
	Index	31

SysSimPyMMEN is a framework for inferring the MMEN from the SysSim models.

Tip: What's behind the name?

SysSim – a comprehensive forward modeling framework for studying planetary systems based on the *Kepler* mission. In particular, [SysSimExClusters](#) provides clustered planetary system models that characterize the underlying occurrence and intra-system correlations of multi-planet systems.

Py – this package is written almost entirely in Python 3. This is unlike the SysSim codebase which is written in [Julia](#).

MMEN – short for “minimum-mass extrasolar nebula”. This is the idea of a power-law profile for the minimum mass in disk solids required to form the existing exoplanets, if they formed in their present locations (*in situ* formation).

PUBLICATION

Our paper for this project has been accepted to the Astronomical Journal! You can view an initial preprint on the [arXiv](#).

CONTENTS

2.1 Installation

2.1.1 Using pip

You can install the most recent stable version of SysSimPyMMEN using `pip`:

```
python -m pip install syssimpymmen
```

2.1.2 From source

All of the code is publicly available on [Github](#). Thus an alternative to `pip` is to download or clone the repository and install it from there:

```
git clone https://github.com/hematthi/SysSimPyMMEN.git
python -m pip install .
```

There are multiple ways of cloning; if you have trouble with the above or prefer another, see [this Github guide](#). You should also fork the repository first if you want to make your own changes to the source code later.

2.1.3 Installing SysSimPyPlots

In order to use SysSimPyMMEN, you will also need to install [SysSimPyPlots](#) – see the [installation page](#) for that package.

That page will also include instructions for downloading some simulated catalogs. These are not required for using all of the functions in SysSimPyMMEN, but would be required for applying the functions to those simulated systems and running some of the examples in the following tutorials.

You are now ready to use SysSimPyMMEN!

2.1.4 Dependencies

SysSimPyMMEN has been tested on Python >3.7 and uses:

- `numpy` (for almost everything)
- `matplotlib` (for making plots)
- `scipy` (for some miscellaneous functions)
- `sysimpypplots` (for some core functions and loading catalogs, as described above)

2.2 Basic usage

2.2.1 Computing solid surface densities

You can compute the solid surface densities of the annular disks required to form each planet from their masses and semi-major axes:

```
sigmas = solid_surface_density_CL2013(planet_masses, planet_semi_a) # g/cm^2
```

where `planet_masses` is an array of the planet masses (in Earth masses) and `planet_semi_a` is an array of the planet semi-major axes (in AU).

Here, we've used the Chiang & Laughlin (2013) prescription for the feeding zone of each planet, which simply sets them equal to their semi-major axes; the output `sigmas` is an array of solid surface densities computed for each planet.

Tip: Several other prescriptions are available; see [solid_surface_density_S2014](#), [solid_surface_density_nHill](#), and [solid_surface_density_system_RC2014](#). Some of these may require additional parameters, such as the planet radii (for “S2014”) and the stellar mass (for “S2014” and “nHill”).

You can also define and use your own custom prescription for the feeding zone width (`delta_a`) by using the core function that all of the above functions call, [solid_surface_density](#).

2.2.2 Fitting a power-law

It is easy to fit a power-law to the solid surface densities as a function of semi-major axis:

```
a0 = 0.3 # AU; separation for the normalization constant
sigma0, beta = fit_power_law_MMEN(planet_semi_a, sigmas, a0=a0)
```

Here, `sigma0` is the normalization constant for the solid surface density (g/cm^2) at the chosen separation `a0`, and `beta` is the slope or power-law index.

2.2.3 An example using a SysSim catalog

A full working example using a simulated catalog is shown below (make sure to replace `load_dir` with the path to where you downloaded the simulated catalog):

```
from syssimpyplots.general import *
from syssimpyplots.load_sims import *

from syssimpymmen.mmen import *

# Load a simulated catalog:
load_dir = '/path/to/a/simulated/catalog/' # replace with your path where you downloaded
↳ a simulated catalog!
sss_per_sys, sss = compute_summary_stats_from_cat_obs(file_name_path=load_dir)

# Compute solid surface densities for all planets in the catalog:
sigma_obs_CL2013, mass_obs, a_obs = solid_surface_density_CL2013_given_observed_
↳ catalog(sss_per_sys, max_core_mass=np.inf)

# Fit a power-law profile for the MMEN:
a0 = 0.3 # AU; choice for the normalization point
sigma0_obs_CL2013, beta_obs_CL2013 = fit_power_law_MMEN(a_obs, sigma_obs_CL2013, a0=a0)

# Plot solid surface density vs semi-major axis:
fig = plt.figure(figsize=(8,8))
plot = GridSpec(1,1,left=0.15,bottom=0.15,right=0.95,top=0.95,wspace=0,hspace=0)
ax = plt.subplot(plot[0,0])
plt.scatter(a_obs, np.log10(sigma_obs_CL2013), marker='o', s=10, alpha=0.2, color='b')
a_array = np.linspace(1e-3,2,1001)
plt.plot(a_array, np.log10(MMEN_power_law(a_array, sigma0_obs_CL2013, beta_obs_CL2013,
↳ a0=a0)), lw=2, ls='--', color='k', label=r'Power-law fit ($\Sigma_0 = \{0.0f\}$, $\beta_0 = \{0.2f\}$'.format(sigma0_obs_CL2013, beta_obs_CL2013))
↳ = \{0.2f\}$')
ax.tick_params(axis='both', labelsize=20)
plt.gca().set_xscale("log")
ax.get_xaxis().set_major_formatter(ticker.ScalarFormatter())
plt.xticks([0.05, 0.1, 0.2, 0.4, 0.8])
plt.xlim([0.04,0.9])
plt.ylim([0.5,5.5])
plt.xlabel(r'Semimajor axis, $a$ (AU)', fontsize=20)
plt.ylabel(r'Surface density, $\log_{10}(\Sigma/\rm g\ cm^{-2})$', fontsize=20)
plt.legend(loc='upper right', bbox_to_anchor=(1.,1.), ncol=1, frameon=False, fontsize=16)
plt.show()
```

Note: In this example, the function `solid_surface_density_CL2013_given_observed_catalog` takes the observed catalog, uses a mass-radius relation to draw a set of planet masses from the planet radii, and enforces a limit on the maximum mass via `max_core_mass` before calling the function `solid_surface_density_CL2013` we started with at the top of this page.

We've removed the maximum core mass limit (by setting `max_core_mass=np.inf`) to show you the broad range of

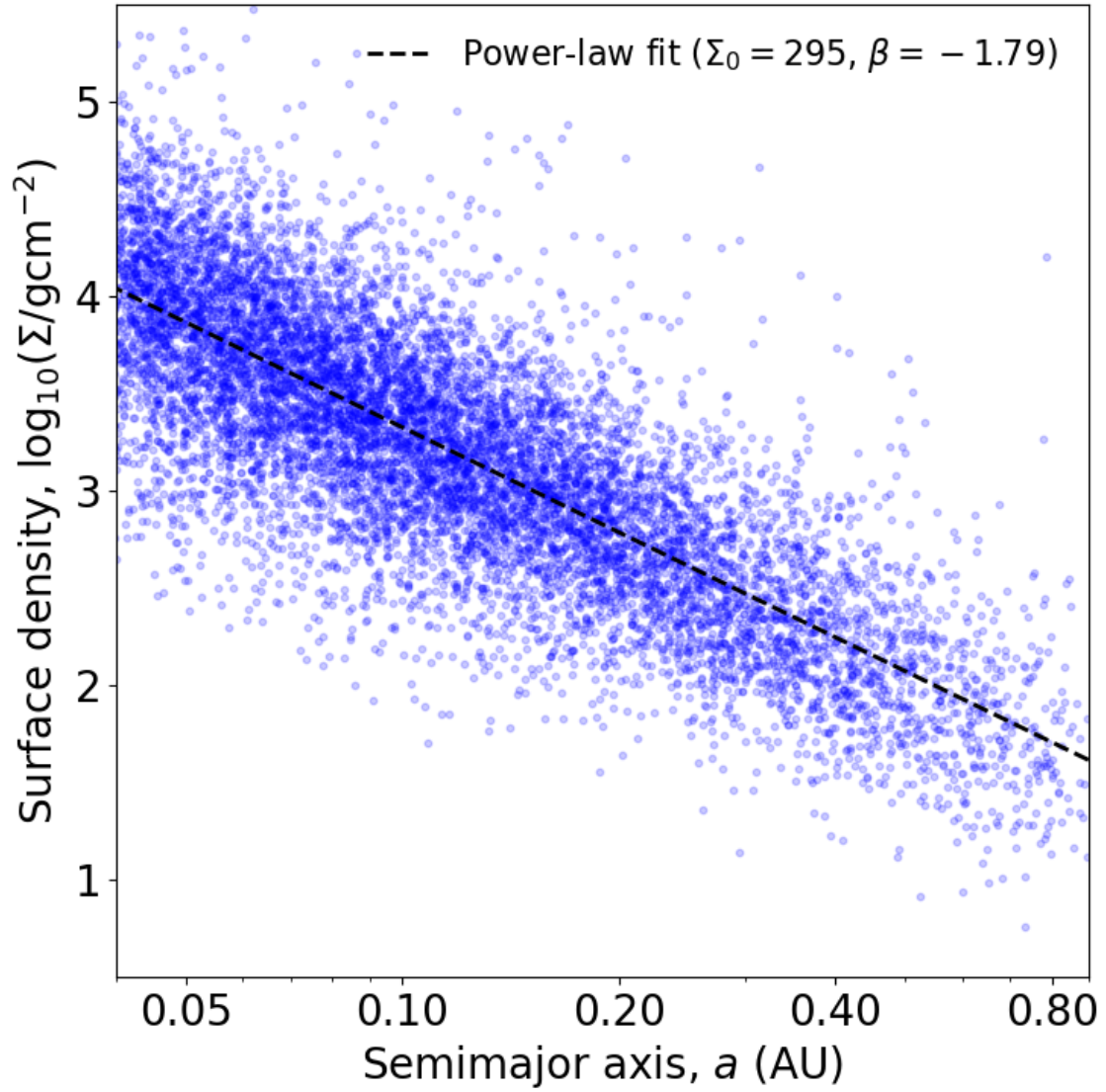


Fig. 1: The solid surface densities and MMEN fit to a simulated observed catalog.

surface densities arising from the broad range of planet masses. By default, it is set to 10 Earth masses.

2.3 Fitting individual systems

The usage for fitting individual planetary systems can be exactly the same as for fitting a catalog! On the *previous page*, the inputs `planet_masses` and `planet_semi_a` could have been either arrays for many planets in an entire catalog, or arrays of just a few planets in a single planetary system. The following shows an example of fitting three hypothetical planets in the same system:

```
planet_masses = np.array([1.7, 4.5, 2.2]) # arbitrary planet masses (Earth masses)
planet_semi_a = np.array([0.1, 0.3, 0.8]) # arbitrary semi-major axes (AU)

sigmas = solid_surface_density_CL2013(planet_masses, planet_semi_a) # g/cm^2
a0 = 0.3 # AU; separation for the normalization constant
sigma0, beta = fit_power_law_MMEN(planet_semi_a, sigmas, a0=a0)
```

2.3.1 Scaling up solid surface densities

You may want to “scale up” the power-law fit such that it is at least above solid surface densities of each planet in the system. After-all, the concept of a MMEN is to estimate the minimum surface density profile containing enough mass to form every planet in their present locations, and thus the surface density must be at least equal to that local to each planet!

This is easily done via the following function:

```
sigma0_scaled, beta, scale_factor = fit_power_law_and_scale_up_MMEN(planet_semi_a,
    ↪ sigmas, a0=a0)
```

Here, `scale_factor` is a value greater than or equal to one, and the power-law normalization `sigma0` has already been multiplied by the scale factor. The power-law slope (`beta`) should be unaffected. You can visually check the effect of the scaling by plotting the scaled and un-scaled power-law profiles:

```
# Plot solid surface density vs semi-major axis:
fig = plt.figure(figsize=(8,8))
plot = GridSpec(1,1,left=0.15,bottom=0.15,right=0.95,top=0.95,wspace=0,hspace=0)
ax = plt.subplot(plot[0,0])
plt.scatter(planet_semi_a, np.log10(sigmas), marker='o', s=50*planet_masses, color='b')
a_array = np.linspace(1e-3,2,1001)
plt.plot(a_array, np.log10(MMEN_power_law(a_array, sigma0, beta, a0=a0)), lw=2, ls='--',
    ↪ color='k', label=r'Power-law fit ($\Sigma_0 = \{:\0.0f\}$, $\beta = \{:\0.2f\}$)'.
    ↪ format(sigma0, beta))
plt.plot(a_array, np.log10(MMEN_power_law(a_array, sigma0_scaled, beta, a0=a0)), lw=2,
    ↪ ls='--', color='r', label=r'Scaled up by $\alpha = \{:\0.2f\}$ ($\Sigma_0^{*} = \{:\0.0f\}$)'.
    ↪ format(scale_factor, sigma0_scaled))
ax.tick_params(axis='both', labels=20)
plt.gca().set_xscale("log")
ax.get_xaxis().set_major_formatter(ticker.ScalarFormatter())
plt.xticks([0.05, 0.1, 0.2, 0.4, 0.8])
plt.xlim([0.05,1.])
plt.ylim([1.,4.])
```

(continues on next page)

(continued from previous page)

```
plt.xlabel(r'Semimajor axis, $a$ (AU)', fontsize=20)
plt.ylabel(r'Surface density, $\log_{10}(\Sigma/\rm g\ cm^{-2})$', fontsize=20)
plt.legend(loc='upper right', bbox_to_anchor=(1.,1.), ncol=1, frameon=False, fontsize=16)
plt.show()
```

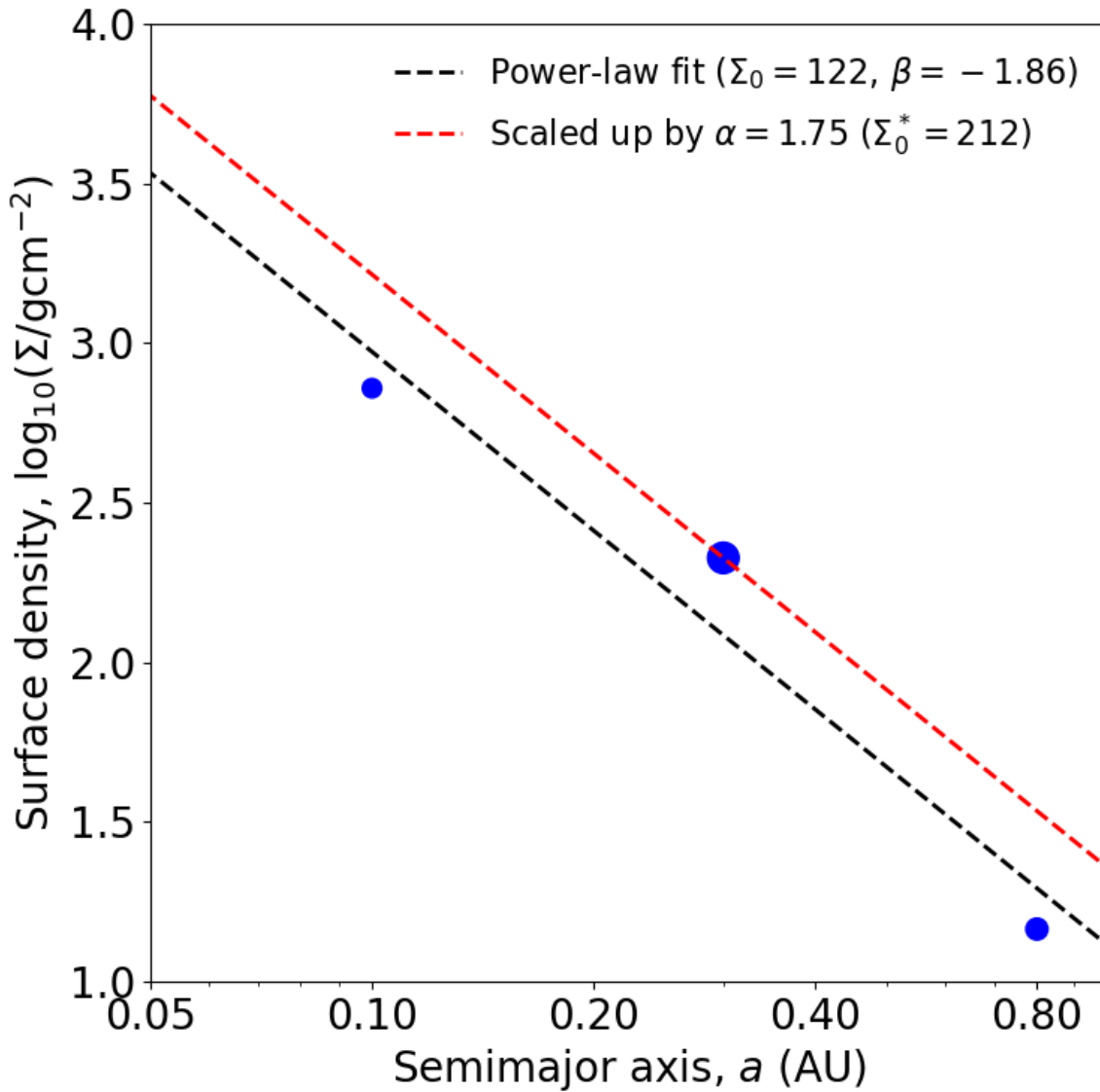


Fig. 2: The MMEN fit to an example system with three planets, with and without scaling up.

2.3.2 Visualizing individual systems

We provide a function for fitting and plotting individual systems from a simulated physical catalog. For each system, it will fit an MMEN power-law to the observed (transiting) planets in the system, and then to all the planets in the system, to illustrate how undetected planets can bias the fit. Try it!

```
from syssimpyplots.general import *
from syssimpyplots.load_sims import *

from syssimpymmen.mmen import *

# Load a simulated catalog:
load_dir = '/path/to/a/simulated/catalog/' # replace with your path where you downloaded
↳ a simulated catalog!

# To load and process the simulated physical catalog of stars and planets:
sssp_per_sys, sssp = compute_summary_stats_from_cat_phys(file_name_path=load_dir, load_
↳ full_tables=True)

# To loop through and plot individual physical systems with at least 3 observed planets:
a0 = 0.3
prescription = 'RC2014' # choose the prescription for the feeding zones
plot_feeding_zones_and_power_law_fit_MMEN_per_system_observed_and_physical(sssp_per_sys,
↳ sssp, n_mult_min=3, prescription=prescription, a0=a0, scale_up=True, N_sys=10)
```

2.3.3 Plotting a diversity of MMEN

You can fit MMEN to a large number of systems, and then plot the distribution of power-law parameters (i.e. σ_0 and β) using the `plot_2d_points_and_contours_with_histograms` function from `SysSimPyPlots`:

```
# To import the "plot_2d_points_and_contours_with_histograms()" function:
from syssimpyplots.plot_params import *

scale_up = True
y_sym = '\Sigma_0^*' if scale_up else '\Sigma_0'

# Fit a power-law to each system in the physical catalog:
fit_per_sys_dict = fit_power_law_MMEN_per_system_physical(sssp_per_sys, sssp,
↳ prescription=prescription, a0=a0, scale_up=True, N_sys=100000) # may take a minute

# To plot the distribution of fitted power-law parameters (sigma_0 vs. beta):
plot_2d_points_and_contours_with_histograms(fit_per_sys_dict['beta'], fit_per_sys_dict[
↳ 'sigma_0'], x_min=-8., x_max=4., y_min=1e-1, y_max=1e6, log_y=True, xlabel_text=r'\beta
↳ ', ylabel_text=r'\log_{10}(\frac{\sigma_0}{\rm g, cm^{-2}})' % y_sym, y_str_format='{:.0f}', x_
↳ symbol=r'\beta$', y_symbol=r'%s$' % y_sym)
plt.show()
```

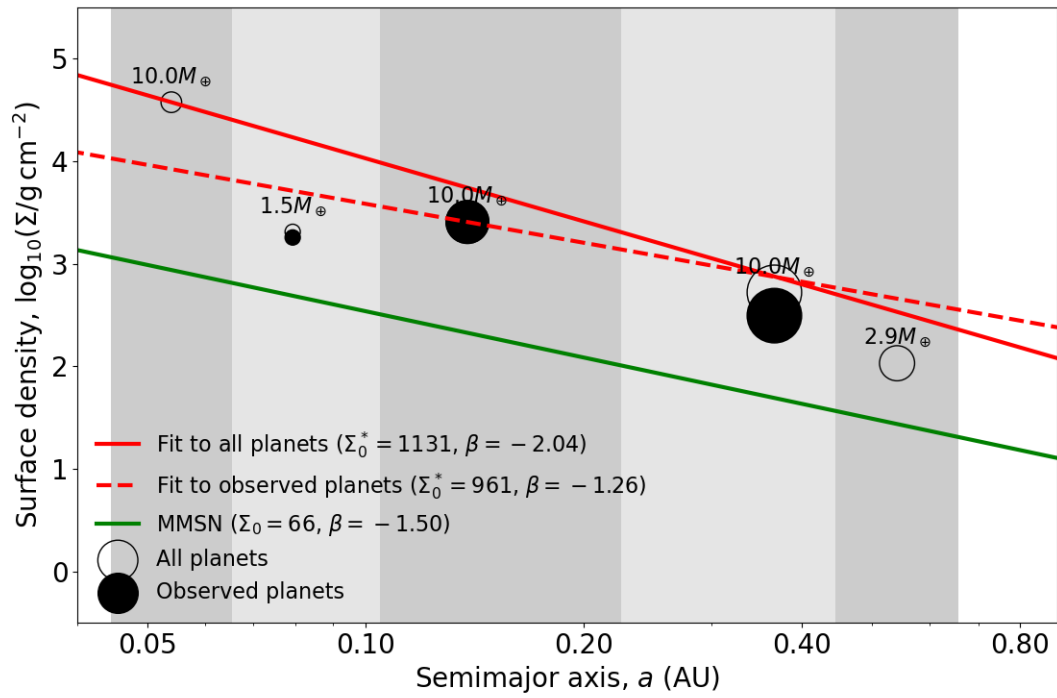


Fig. 3: An example of how missing planets can alter the inferred MMEN for a given system. The function will output many other similar figures, not shown here.

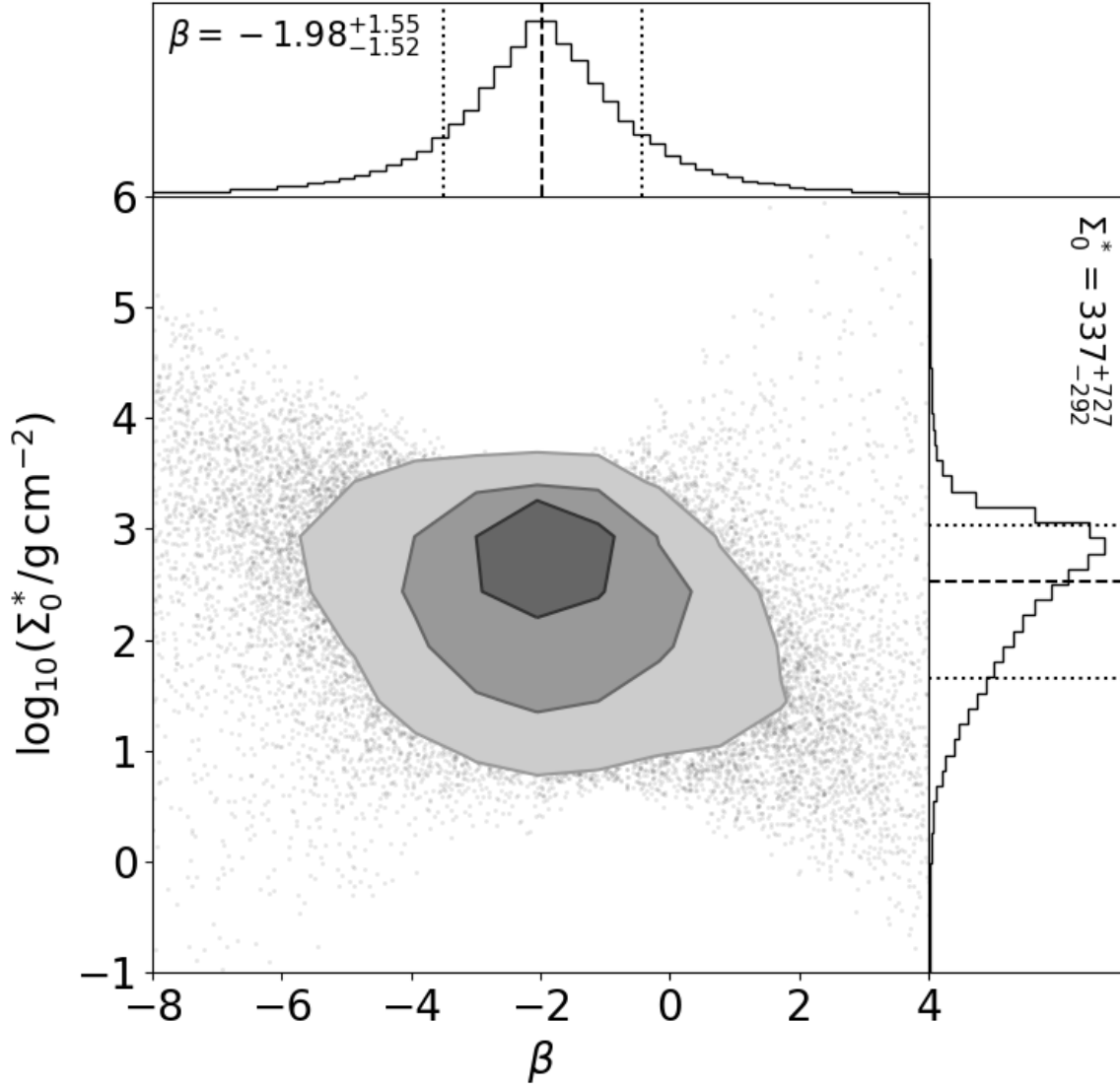


Fig. 4: The distribution of power-law fit parameters from a physical catalog, showing the diversity of MMEN.

2.4 Detailed API

Here, we provide detailed documentation for all of the functions in SysSimPyMMEN.

generate_planet_mass_from_radius_Ning2018_table(*R*)

Draw a planet mass from the Ning et al. (2018) mass-radius model interpolated on a precomputed table.

Parameters

R (*float*) – The planet radius (Earth radii).

Return type

The planet mass (Earth masses).

Note: Requires a globally defined interpolation object, `logMR_table_interp` (defined in the same module).

_logM_linear_given_radius(*R*, *_logM_r1*=0.04, *_logM_r2*=0.3, *r1*=0.5, *r2*=1.472)

Compute the standard deviation in log planet mass at a given planet radius based on a linear relation.

Parameters

- **R** (*float*) – The planet radius (Earth radii).
- **_logM_r1** (*float*, *default*=`_logM_at_radius_min`) – The standard deviation in log(mass) (Earth masses) at radius *r1*.
- **_logM_r2** (*float*, *default*=`_logM_at_radius_switch`) – The standard deviation in log(mass) (Earth masses) at radius *r2*.
- **r1** (*float*, *default*=0.5) – The planet radius (Earth radii) corresponding to `_logM_r1`.
- **r2** (*float*, *default*=`radius_switch`) – The planet radius (Earth radii) corresponding to `_logM_r2`.

Return type

The standard deviation in log(mass) (Earth masses) at the given radius.

generate_planet_mass_from_radius_lognormal_mass_around_earthlike_rocky(*R*, *_min*=1.0, *_max*=100.0)

Draw a planet mass from a lognormal distribution centered around the Earth-like rocky model from Li Zeng (<https://www.cfa.harvard.edu/~lzeng/tables/massradiusEarthlikeRocky.txt>).

Parameters

- **R** (*float*) – The planet radius (Earth radii).
- **_min** (*float*, *default*=1.) – The minimum planet density (g/cm³) allowed.
- **_max** (*float*, *default*=100.) – The maximum planet density (g/cm³) allowed.

Return type

The planet mass (Earth masses).

Note: Requires a globally defined interpolation object, `M_earthlike_rocky_interp` (defined in the same module).

generate_planet_mass_from_radius_Ning2018_table_above_lognormal_mass_earthlike_rocky_below(*R*, *R_switch*=1.472, *_min*=1.0, *_max*=100.0)

Draw a planet mass from a combined model (the Ning et al. 2018 model for large radii, and the lognormal distribution centered around the Earth-like rocky model for small radii).

Parameters

- **R** (*float*) – The planet radius (Earth radii).
- **R_switch** (*float*, *default=radius_switch*) – The transition radius (Earth radii) defining whether to draw from the Ning et al. 2018 model (above this radius) or the lognormal-around Earth-like rocky model (below this radius).
- **_min** (*float*, *default=1.*) – The minimum planet density (g/cm³) allowed.
- **_max** (*float*, *default=100.*) – The maximum planet density (g/cm³) allowed.

Return type

The planet mass (Earth masses).

```
generate_planet_mass_from_radius_Ning2018_table_above_lognormal_mass_earthlike_rocky_below_vec(R,
                                                                                               R_switch=
                                                                                               _min=1.0,
                                                                                               _max=100
```

Draw a planet mass from a combined model (the Ning et al. 2018 model for large radii, and the lognormal distribution centered around the Earth-like rocky model for small radii).

Parameters

- **R** (*float*) – The planet radius (Earth radii).
- **R_switch** (*float*, *default=radius_switch*) – The transition radius (Earth radii) defining whether to draw from the Ning et al. 2018 model (above this radius) or the lognormal-around Earth-like rocky model (below this radius).
- **_min** (*float*, *default=1.*) – The minimum planet density (g/cm³) allowed.
- **_max** (*float*, *default=100.*) – The maximum planet density (g/cm³) allowed.

Return type

The planet mass (Earth masses).

```
feeding_zone_S2014(M, R, a, Mstar=1.0)
```

Compute the feeding zone width of a planet using the Schlichting (2014) prescription:

$$\text{delta_a} = 2^{(3/2)} * a * ((a * M) / (R * Mstar))^{(1/2)}.$$

Parameters

- **M** (*float or array[float]*) – The planet mass (Earth masses).
- **R** (*float or array[float]*) – The planet radius (Earth radii).
- **a** (*float or array[float]*) – The semi-major axis (AU).
- **Mstar** (*float or array[float]*, *default=1.*) – The stellar mass (solar masses).

Returns

delta_a – The feeding zone width (AU) of the planet.

Return type

float or array[float]

feeding_zone_nHill(*M*, *a*, *Mstar*=1.0, *n*=10.0)

Compute the feeding zone width of a planet using a number of Hill radii:

$$\text{delta_a} = n * R_{\text{Hill}} = n * a * (M / (3 * M_{\text{star}}))^{1/3}.$$

Parameters

- **M** (*float* or *array[float]*) – The planet mass (Earth masses).
- **a** (*float* or *array[float]*) – The semi-major axis (AU).
- **Mstar** (*float* or *array[float]*, *default*=1.) – The stellar mass (solar masses).
- **n** (*float* or *array[float]*, *default*=10.) – The number of Hill radii to use as the feeding zone.

Returns

delta_a – The feeding zone width (AU) of the planet.

Return type

float or array[float]

feeding_zone_RC2014(*a_sys*)

Compute the feeding zone widths of all the planets in a multi-planet system using the Raymond & Cossou (2014) prescription.

Uses the geometric means of the neighboring planets' semi-major axes as the boundaries of their feeding zones.

Note: Assumes the same ratio for the inner edge of the innermost planet as its outer edge, and the same ratio for the outer edge of the outermost planet as its inner edge.

Parameters

a_sys (*array[float]*) – The semi-major axes (AU) of all the planets.

Returns

- **delta_a_sys** (*array[float]*) – The feeding zone widths (AU) of all the planets.
- **a_bounds_sys** (*array[float]*) – The boundaries (AU) of the feeding zones for the planets (length = n+1 where n is the number of planets). For example, *a_bounds_sys[i]* and *a_bounds_sys[i+1]* will be the inner and outer boundaries of the *i*th planet.

solid_surface_density(*M*, *a*, *delta_a*)

Compute the solid surface density associated with a planet.

This divides the mass of the planet by the surface area of an annulus of a given width, centered at its semi-major axis.

Parameters

- **M** (*float* or *array[float]*) – The planet mass (Earth masses).
- **a** (*float* or *array[float]*) – The semi-major axis (AU).
- **delta_a** (*float* or *array[float]*) – The feeding zone width (AU).

Returns

sigma_solid – The solid surface density (g/cm²) local to the planet.

Return type

float or array[float]

solid_surface_density_CL2013(*M*, *a*)

Compute the solid surface density of a planet using the Chiang & Laughlin (2013) prescription for the feeding zone width (set equal to the semi-major axis; $\delta a = a$).

Parameters

- **M** (*float* or *array[float]*) – The planet mass (Earth masses).
- **a** (*float* or *array[float]*) – The semi-major axis (AU).

Return typeThe solid surface density (g/cm^2) local to the planet.**solid_surface_density_S2014**(*M*, *R*, *a*, *Mstar=1.0*)

Compute the solid surface density of a planet using the Schlichting (2014) prescription for the feeding zone width (see [syssimpymmen.mmen.feeding_zone_S2014\(\)](#)).

Parameters

- **M** (*float* or *array[float]*) – The planet mass (Earth masses).
- **R** (*float* or *array[float]*) – The planet radius (Earth radii).
- **a** (*float* or *array[float]*) – The semi-major axis (AU).
- **Mstar** (*float* or *array[float]*, *default=1.*) – The stellar mass (solar masses).

Return typeThe solid surface density (g/cm^2) local to the planet.**solid_surface_density_nHill**(*M*, *a*, *Mstar=1.0*, *n=10.0*)

Compute the solid surface density of a planet using a number of Hill radii for the feeding zone width (see [syssimpymmen.mmen.feeding_zone_nHill\(\)](#)).

Parameters

- **M** (*float* or *array[float]*) – The planet mass (Earth masses).
- **a** (*float* or *array[float]*) – The semi-major axis (AU).
- **Mstar** (*float* or *array[float]*, *default=1.*) – The stellar mass (solar masses).
- **n** (*float* or *array[float]*, *default=10.*) – The number of Hill radii to use as the feeding zone.

Return typeThe solid surface density (g/cm^2) local to the planet.**solid_surface_density_system_RC2014**(*M_sys*, *a_sys*)

Compute the solid surface densities of all planets in a multi-planet system using the Raymond & Cossou (2014) prescription for their feeding zone widths (see [syssimpymmen.mmen.feeding_zone_RC2014\(\)](#)).

Parameters

- **M_sys** (*array[float]*) – The planet masses (Earth masses).
- **a_sys** (*array[float]*) – The semi-major axes (AU) of the planets.

Return typeThe solid surface densities (g/cm^2) local to the planets.

solid_surface_density_prescription(*M*, *R*, *a*, *Mstar*=1.0, *n*=10.0, *prescription*='CL2013')

Compute the solid surface density of a planet or planetary system using a given prescription for the feeding zone width.

Wrapper function that calls the appropriate function based on the *prescription* string.

Parameters

- **M** (*float or array[float]*) – The planet masses (Earth masses).
- **R** (*float or array[float]*) – The planet radii (Earth radii). Only used for the 'S2014' prescription.
- **a** (*float or array[float]*) – The semi-major axes (AU) of the planets.
- **Mstar** (*float or array[float]*, *default*=1.) – The stellar mass or masses (solar masses). Only used for the 'S2014' or 'nHill' prescriptions.
- **n** (*float*, *default*=10.) – The number of Hill radii to use as the feeding zone. Only used for the 'nHill' prescription.
- **prescription** (*{'CL2013', 'S2014', 'nHill', 'RC2014'}*, *default*='CL2013') – The string indicating the prescription to use for computing the feeding zone widths.

Return type

The solid surface densities (g/cm²) local to the planets.

Note: If the prescription is 'RC2014', the input arrays (i.e. *M* and *a*) must all correspond to planets in the same system!

solid_surface_density_CL2013_given_physical_catalog(*sssp_per_sys*, *max_core_mass*=10.0)

Compute the solid surface densities for all planets in a physical catalog, using the Chiang & Laughlin (2013) prescription for the feeding zone widths.

Parameters

- **sssp_per_sys** (*dict*) – The dictionary containing the planetary and stellar properties for each system in a physical catalog (2-d and 1-d arrays).
- **max_core_mass** (*float*, *default*=10.) – The maximum allowed (solid) core mass (Earth masses).

Returns

- **sigma_all** (*array[float]*) – The solid surface densities (g/cm²) of all the planets.
- **a_all** (*array[float]*) – The semi-major axes (AU) of all the planets.

solid_surface_density_S2014_given_physical_catalog(*sssp_per_sys*, *sssp*, *max_core_mass*=10.0)

Compute the solid surface densities for all planets in a physical catalog, using the Schlichting (2014) prescription for the feeding zone widths.

Parameters

- **sssp_per_sys** (*dict*) – The dictionary containing the planetary and stellar properties for each system in a physical catalog (2-d and 1-d arrays).
- **sssp** (*dict*) – The dictionary containing the planetary and stellar properties of all planets in a physical catalog (1-d arrays).
- **max_core_mass** (*float*, *default*=10.) – The maximum allowed (solid) core mass (Earth masses).

Returns

- **sigma_all** (*array[float]*) – The solid surface densities (g/cm^2) of all the planets.
- **a_all** (*array[float]*) – The semi-major axes (AU) of all the planets.

solid_surface_density_nHill_given_physical_catalog(*sssp_per_sys, sssp, max_core_mass=10.0, n=10.0*)

Compute the solid surface densities for all planets in a physical catalog, using a number of Hill radii for the feeding zone widths.

Parameters

- **sssp_per_sys** (*dict*) – The dictionary containing the planetary and stellar properties for each system in a physical catalog (2-d and 1-d arrays).
- **sssp** (*dict*) – The dictionary containing the planetary and stellar properties of all planets in a physical catalog (1-d arrays).
- **max_core_mass** (*float, default=10.*) – The maximum allowed (solid) core mass (Earth masses).
- **n** (*float, default=10.*) – The number of Hill radii to use as the feeding zone.

Returns

- **sigma_all** (*array[float]*) – The solid surface densities (g/cm^2) of all the planets.
- **a_all** (*array[float]*) – The semi-major axes (AU) of all the planets.

solid_surface_density_RC2014_given_physical_catalog(*sssp_per_sys, max_core_mass=10.0*)

Compute the solid surface densities for all planets in multi-planet systems in a physical catalog, using the Raymond & Cossou (2014) prescription for the feeding zone widths.

Parameters

- **sssp_per_sys** (*dict*) – The dictionary containing the planetary and stellar properties for each system in a physical catalog (2-d and 1-d arrays).
- **max_core_mass** (*float, default=10.*) – The maximum allowed (solid) core mass (Earth masses).

Returns

- **sigma_all_2p** (*array[float]*) – The solid surface densities (g/cm^2) of the planets in multi-planet systems.
- **a_all_2p** (*array[float]*) – The semi-major axes (AU) of the planets in multi-planet systems.
- **mult_all_2p** (*array[float]*) – The multiplicities of the multi-planet systems each planet belongs to.

solid_surface_density_CL2013_given_observed_catalog(*sssp_per_sys, max_core_mass=10.0*)

Compute the solid surface densities for all planets in an observed catalog, using the Chiang & Laughlin (2013) prescription for the feeding zone widths.

Parameters

- **sssp_per_sys** (*dict*) – The dictionary containing the planetary and stellar properties for each system in a physical catalog (2-d and 1-d arrays).
- **max_core_mass** (*float, default=10.*) – The maximum allowed (solid) core mass (Earth masses).

Returns

- **sigma_obs** (*array[float]*) – The solid surface densities (g/cm^2) of all the observed planets.
- **core_mass_obs** (*array[float]*) – The core masses (Earth masses) of all the observed planets. These are the total masses of the planets computed from their observed radii using a mass-radius relation, capped at *max_core_mass*.
- **a_obs** (*array[float]*) – The semi-major axes (AU) of all the observed planets.

solid_surface_density_S2014_given_observed_catalog(*sss_per_sys*, *max_core_mass=10.0*)

Compute the solid surface densities for all planets in an observed catalog, using the Schlichting (2014) prescription for the feeding zone widths.

Parameters

- **sss_per_sys** (*dict*) – The dictionary containing the planetary and stellar properties for each system in a physical catalog (2-d and 1-d arrays).
- **max_core_mass** (*float*, *default=10.*) – The maximum allowed (solid) core mass (Earth masses).

Returns

- **sigma_obs** (*array[float]*) – The solid surface densities (g/cm^2) of all the observed planets.
- **core_mass_obs** (*array[float]*) – The core masses (Earth masses) of all the observed planets. These are the total masses of the planets computed from their observed radii using a mass-radius relation, capped at *max_core_mass*.
- **a_obs** (*array[float]*) – The semi-major axes (AU) of all the observed planets.

solid_surface_density_nHill_given_observed_catalog(*sss_per_sys*, *max_core_mass=10.0*, *n=10.0*)

Compute the solid surface densities for all planets in an observed catalog, using a number of Hill radii for the feeding zone widths.

Parameters

- **sss_per_sys** (*dict*) – The dictionary containing the planetary and stellar properties for each system in a physical catalog (2-d and 1-d arrays).
- **max_core_mass** (*float*, *default=10.*) – The maximum allowed (solid) core mass (Earth masses).
- **n** (*float*, *default=10.*) – The number of Hill radii to use as the feeding zone.

Returns

- **sigma_obs** (*array[float]*) – The solid surface densities (g/cm^2) of all the observed planets.
- **core_mass_obs** (*array[float]*) – The core masses (Earth masses) of all the observed planets. These are the total masses of the planets computed from their observed radii using a mass-radius relation, capped at *max_core_mass*.
- **a_obs** (*array[float]*) – The semi-major axes (AU) of all the observed planets.

solid_surface_density_RC2014_given_observed_catalog(*sss_per_sys*, *max_core_mass=10.0*)

Compute the solid surface densities for all planets in observed multi-planet systems in an observed catalog, using the Raymond & Cossou (2014) prescription for the feeding zone widths.

Parameters

- **sss_per_sys** (*dict*) – The dictionary containing the planetary and stellar properties for each system in a physical catalog (2-d and 1-d arrays).
- **max_core_mass** (*float*, *default=10.*) – The maximum allowed (solid) core mass (Earth masses).

Returns

- **sigma_obs_2p** (*array[float]*) – The solid surface densities (g/cm^2) of the observed planets in multi-planet systems.
- **core_mass_obs_2p** (*array[float]*) – The core masses (Earth masses) of the observed planets in multi-planet systems. These are the total masses of the planets computed from their observed radii using a mass-radius relation, capped at *max_core_mass*.
- **a_obs_2p** (*array[float]*) – The semi-major axes (AU) of the observed planets in multi-planet systems.
- **mult_obs_2p** (*array[float]*) – The multiplicities of the observed multi-planet systems each planet belongs to.

MMSN(*a*, *F=1.0*, *Zrel=0.33*)

Compute the solid surface density of the minimum mass solar nebula (MMSN) at a given separation, as defined by Eq. 2 in Chiang & Youdin (2010) (<https://arxiv.org/pdf/0909.2652.pdf>).

Note: The normalization is such that the default values of *F=1* and *Zrel=0.33* give 1 Earth mass of solids in an annulus centered on Earth's semi-major axis.

Parameters

- **a** (*float or array[float]*) – The semi-major axis (AU).
- **F** (*float or array[float]*, *default=1.*) – A fudge factor.
- **Zrel** (*float or array[float]*, *default=0.33*) – The relative enrichment in metallicity.

Returns

sigma_p – The solid surface density (g/cm^2) of the MMSN at the given separation *a*.

Return type

float or array[float]

MMEN_power_law(*a*, *sigma0*, *beta*, *a0=1.0*)

Evaluate a power-law profile for the solid surface density of the minimum mass extrasolar nebula (MMEN) at a given separation.

Given by $\sigma(a) = \sigma_0 (a/a_0)^\beta$, where σ_0 (g/cm^2) is the normalization at semi-major axis *a0* (AU), and β is the power-law index.

Parameters

- **a** (*float or array[float]*) – The semi-major axis (AU) at which to evaluate the solid surface density.
- **sigma0** (*float*) – The solid surface density normalization (g/cm^2) at separation *a0*.
- **beta** (*float*) – The power-law index.
- **a0** (*float*, *default=1.*) – The normalization point for the separation (AU).

Returns

sigma_a – The solid surface density (g/cm^2) at the given separation *a*.

Return type

float or array[float]

fit_power_law_MMEN(*a_array*, *sigma_array*, *a0*=1.0, *p0*=1.0, *p1*=-1.5)

Fit a power-law representing the minimum mass extrasolar nebula (MMEN) of a planetary system given the planets' semi-major axes and solid surface densities.

Parameters

- **a_array** (*array[float]*) – The semi-major axes (AU) of the planets.
- **sigma_array** (*array[float]*) – The solid surface densities (g/cm^2) of the planets.
- **a0** (*float*, *default*=1.) – The normalization point for the separation (AU).
- **p0** (*float*, *default*=1.) – The initial guess for the normalization parameter, 'log(sigma0)'.
- **p1** (*float*, *default*=-1.5) – The initial guess for the power-law index parameter, 'beta'.

Returns

- **sigma0** (*float*) – The best-fit value for the solid surface density normalization (g/cm^2). Unlike the initial guess (*p0*), this value is unlogged.
- **beta** (*float*) – The best-fit value for the power-law index.

fit_power_law_and_scale_up_MMEN(*a_sys*, *sigma_sys*, *a0*=1.0, *p0*=1.0, *p1*=-1.5)

Call `syssimpymmen.mmen.fit_power_law_MMEN()`, and then scale up the resulting power-law fit to be at/above all of the input solid surface densities given by *sigma_sys*.

Note: Should only be used for fitting MMEN to individual planetary systems; should NOT be used for fitting MMEN to an entire catalog or a collection of planets from different planetary systems.

Parameters

- **a_sys** (*array[float]*) – The semi-major axes (AU) of the planets in the system.
- **sigma_sys** (*array[float]*) – The solid surface densities (g/cm^2) of the planets in the system.
- **a0** (*float*, *default*=1.) – The normalization point for the separation (AU).
- **p0** (*float*, *default*=1.) – The initial guess for the normalization parameter, 'log(sigma0)'.
- **p1** (*float*, *default*=-1.5) – The initial guess for the power-law index parameter, 'beta'.

Returns

- **sigma0** (*float*) – The best-fit value for the solid surface density normalization (g/cm^2), multiplied by a scale factor *scale_factor*. Unlike the initial guess (*p0*), this value is unlogged.
- **beta** (*float*) – The best-fit value for the power-law index.
- **scale_factor** (*float*) – The scale factor required to shift the power-law to be at/above all of the values in *sigma_sys*. This is always unity for systems with two planets, and greater than or equal to one in general.

fit_power_law_MMEN_all_planets_observed(*sss_per_sys*, *max_core_mass*=10.0, *prescription*='CL2013', *n*=10.0, *a0*=1.0, *p0*=1.0, *p1*=-1.5)

Compute the solid surface densities and fit a power-law to them for all planets in an observed catalog.

Parameters

- **sss_per_sys** (*dict*) – The dictionary containing the planetary and stellar properties for each system in a physical catalog (2-d and 1-d arrays).
- **max_core_mass** (*float*, *default=10.*) – The maximum allowed (solid) core mass (Earth masses).
- **prescription** (*{'CL2013', 'S2014', 'nHill', 'RC2014'}*, *default='CL2013'*) – The string indicating the prescription to use for computing the feeding zone widths.
- **n** (*float*, *default=10.*) – The number of Hill radii to use for the feeding zones. Only used for the 'nHill' prescription.
- **a0** (*float*, *default=1.*) – The normalization point for the separation (AU).
- **p0** (*float*, *default=1.*) – The initial guess for the normalization parameter, 'log(sigma0)'.
- **p1** (*float*, *default=-1.5*) – The initial guess for the power-law index parameter, 'beta'.

Returns

outputs_dict – A dictionary containing the solid surface densities, power-law fit parameters, and other properties computed from the observed catalog.

Return type

dict

The output dictionary contains the following fields:

- *sigma_obs*: The solid surface densities (g/cm^2) of all the observed planets (1-d array).
- *core_mass_obs*: The solid core masses (Earth masses) of all the observed planets (1-d array). These are the total masses of the planets computed from their observed radii using a mass-radius relation, capped at *max_core_mass*.
- *a_obs*: The semi-major axes (AU) of all the observed planets (1-d array).
- *sigma0*: The best-fit value for the solid surface density normalization (g/cm^2).
- *beta*: The best-fit value for the power-law index.

fit_power_law_MMEN_all_planets_physical(*sssp_per_sys*, *sssp*, *max_core_mass=10.0*,
prescription='CL2013', *n=10.0*, *a0=1.0*, *p0=1.0*, *p1=-1.5*)

Compute the solid surface densities and fit a power-law to them for all planets in a physical catalog.

Parameters

- **sssp_per_sys** (*dict*) – The dictionary containing the planetary and stellar properties for each system in a physical catalog (2-d and 1-d arrays).
- **sssp** (*dict*) – The dictionary containing the planetary and stellar properties of all planets in a physical catalog (1-d arrays).
- **max_core_mass** (*float*, *default=10.*) – The maximum allowed (solid) core mass (Earth masses).
- **prescription** (*{'CL2013', 'S2014', 'nHill', 'RC2014'}*, *default='CL2013'*) – The string indicating the prescription to use for computing the feeding zone widths.
- **n** (*float*, *default=10.*) – The number of Hill radii to use for the feeding zones. Only used for the 'nHill' prescription.
- **a0** (*float*, *default=1.*) – The normalization point for the separation (AU).
- **p0** (*float*, *default=1.*) – The initial guess for the normalization parameter, 'log(sigma0)'.

- **p1** (*float*, *default=-1.5*) – The initial guess for the power-law index parameter, ‘beta’.

Returns

outputs_dict – A dictionary containing the solid surface densities, power-law fit parameters, and other properties computed from the observed catalog.

Return type

dict

The output dictionary contains the following fields:

- *sigma_all*: The solid surface densities (g/cm^2) of all the planets (1-d array).
- *a_all*: The semi-major axes (AU) of all the planets (1-d array).
- *sigma0*: The best-fit value for the solid surface density normalization (g/cm^2).
- *beta*: The best-fit value for the power-law index.

fit_power_law_MMEN_per_system_observed(*sss_per_sys*, *n_mult_min=2*, *max_core_mass=10.0*,
prescription='CL2013', *n=10.0*, *a0=1.0*, *p0=1.0*, *p1=-1.5*,
scale_up=False)

Fit power-laws to the solid surface densities of the planets *in each multi-planet system* in an observed catalog.

Parameters

- **sss_per_sys** (*dict*) – The dictionary containing the planetary and stellar properties for each system in a physical catalog (2-d and 1-d arrays).
- **n_mult_min** (*int*, *default=2*) – The minimum multiplicity to include.
- **max_core_mass** (*float*, *default=10.*) – The maximum allowed (solid) core mass (Earth masses).
- **prescription** (*{'CL2013', 'S2014', 'nHill', 'RC2014'}*, *default='CL2013'*) – The string indicating the prescription to use for computing the feeding zone widths.
- **n** (*float*, *default=10.*) – The number of Hill radii to use for the feeding zones. Only used for the ‘nHill’ prescription.
- **a0** (*float*, *default=1.*) – The normalization point for the separation (AU).
- **p0** (*float*, *default=1.*) – The initial guess for the normalization parameter, ‘log(sigma0)’.
- **p1** (*float*, *default=-1.5*) – The initial guess for the power-law index parameter, ‘beta’.
- **scale_up** (*bool*, *default=False*) – Whether to scale up the normalization such that the power-law fit for each system is at/above the solid surface densities of every planet in the system.

Returns

fit_per_sys_dict – A dictionary containing the power-law fit parameters and other properties for each included system in the observed catalog.

Return type

dict

The output dictionary contains the following fields:

- *m_obs*: The observed multiplicities of the included systems (greater than or equal to *n_mult_min*; 1-d array).
- *Mstar_obs*: The stellar masses of the included system (1-d array).

- *sigma0*: The best-fit values for the solid surface density normalizations (g/cm^2) of the included systems (1-d array). If *scale_factor=True*, these normalization values have already been multiplied by the scale factor of each system.
- *scale_factor*: The scale factors required to increase the normalizations of each system such that the power-law fits are at/above the solid surface densities of every planet in the system (1-d array).
- *beta*: The best-fit values for the power-law indices of the included systems (1-d array).

fit_power_law_MMEN_per_system_physical(*sssp_per_sys*, *sssp*, *n_mult_min*=2, *max_core_mass*=10.0, *prescription*='CL2013', *n*=10.0, *a0*=1.0, *p0*=1.0, *p1*=-1.5, *scale_up*=False, *N_sys*=10000)

Fit power-laws to the solid surface densities of the planets in *each multi-planet system* in a physical catalog.

Parameters

- **sssp_per_sys** (*dict*) – The dictionary containing the planetary and stellar properties for each system in a physical catalog (2-d and 1-d arrays).
- **sssp** (*dict*) – The dictionary containing the planetary and stellar properties of all planets in a physical catalog (1-d arrays).
- **n_mult_min** (*int*, *default*=2) – The minimum multiplicity to include.
- **max_core_mass** (*float*, *default*=10.) – The maximum allowed (solid) core mass (Earth masses).
- **prescription** ({'CL2013', 'S2014', 'nHill', 'RC2014'}, *default*='CL2013') – The string indicating the prescription to use for computing the feeding zone widths.
- **n** (*float*, *default*=10.) – The number of Hill radii to use for the feeding zones. Only used for the 'nHill' prescription.
- **a0** (*float*, *default*=1.) – The normalization point for the separation (AU).
- **p0** (*float*, *default*=1.) – The initial guess for the normalization parameter, 'log(sigma0)'.
- **p1** (*float*, *default*=-1.5) – The initial guess for the power-law index parameter, 'beta'.
- **scale_up** (*bool*, *default*=False) – Whether to scale up the normalization such that the power-law fit for each system is at/above the solid surface densities of every planet in the system.
- **N_sys** (*int*, *default*=10000) – The maximum number of systems to be included (to save time if there are too many systems in the physical catalog).

Returns

fit_per_sys_dict – A dictionary containing the power-law fit parameters and other properties for each included system in the physical catalog.

Return type

dict

The output dictionary contains the following fields:

- *n_pl*: The multiplicities of the included systems (greater than or equal to *n_mult_min*; 1-d array).
- *sigma0*: The best-fit values for the solid surface density normalizations (g/cm^2) of the included systems (1-d array). If *scale_factor=True*, these normalization values have already been multiplied by the scale factor of each system.
- *scale_factor*: The scale factors required to increase the normalizations of each system such that the power-law fits are at/above the solid surface densities of every planet in the system (1-d array).

- *beta*: The best-fit values for the power-law indices of the included systems (1-d array).

fit_power_law_MMEN_per_system_observed_and_physical(*sssp_per_sys*, *sssp*, *n_mult_min*=2, *max_core_mass*=10.0, *prescription*='CL2013', *n*=10.0, *a0*=1.0, *p0*=1.0, *p1*=-1.5, *scale_up*=False)

Fit power-laws to the solid surface densities of the observed and physical planets *in each multi-planet system* in a physical catalog.

Computes the solid surface densities and fit a power-law to each multi-planet system in the physical catalog, for the observed planets only and then for all the planets in those systems (using the physical planet properties for both).

Parameters

- **sssp_per_sys** (*dict*) – The dictionary containing the planetary and stellar properties for each system in a physical catalog (2-d and 1-d arrays).
- **sssp** (*dict*) – The dictionary containing the planetary and stellar properties of all planets in a physical catalog (1-d arrays).
- **n_mult_min** (*int*, *default*=2) – The minimum multiplicity to include.
- **max_core_mass** (*float*, *default*=10.) – The maximum allowed (solid) core mass (Earth masses).
- **prescription** ({'CL2013', 'S2014', 'nHill', 'RC2014'}, *default*='CL2013') – The string indicating the prescription to use for computing the feeding zone widths.
- **n** (*float*, *default*=10.) – The number of Hill radii to use for the feeding zones. Only used for the 'nHill' prescription.
- **a0** (*float*, *default*=1.) – The normalization point for the separation (AU).
- **p0** (*float*, *default*=1.) – The initial guess for the normalization parameter, 'log(sigma0)'.
- **p1** (*float*, *default*=-1.5) – The initial guess for the power-law index parameter, 'beta'.
- **scale_up** (*bool*, *default*=False) – Whether to scale up the normalization such that the power-law fit for each system is at/above the solid surface densities of every planet in the system.

Returns

fit_per_sys_dict – A dictionary containing the power-law fit parameters and other properties for each multi-planet system in the observed catalog.

Return type

dict

The output dictionary contains the following fields:

- *n_pl_true*: The true multiplicities of the included systems (greater than or equal to *n_mult_min*) (1-d array).
- *n_pl_obs*: The observed multiplicities of the included systems (1-d array).
- *Mp_tot_true*: The total solid core mass (Earth masses) of all the planets in each included system (1-d array).
- *Mp_tot_obs*: The total solid core mass (Earth masses) of only the observed planets in each included system (1-d array).
- *sigma0_true*: The best-fit values for the solid surface density normalizations (g/cm²) from fitting all the planets of the included systems (1-d array). If *scale_factor*=True, these normalization values have already been multiplied by the scale factor of each system.

- *sigma0_obs*: The best-fit values for the solid surface density normalizations (g/cm^2) from fitting only the observed planets of the included systems (1-d array). If *scale_factor=True*, these normalization values have already been multiplied by the scale factor of each system.
- *scale_factor_true*: The scale factors required to increase the normalizations of each system such that the power-law fits are at/above the solid surface densities of every planet in the system (1-d array).
- *scale_factor_obs*: The scale factors required to increase the normalizations of each system such that the power-law fits are at/above the solid surface densities of every observed planet in the system (1-d array).
- *beta_true*: The best-fit values for the power-law indices from fitting all the planets of the included systems (1-d array).
- *beta_obs*: The best-fit values for the power-law indices from fitting only the observed planets of the included systems (1-d array).

plot_feeding_zones_and_power_law_fit_MMEN_per_system_observed_and_physical(*sssp_per_sys*, *sssp*,
n_mult_min=2,
n_mult_max=10,
max_core_mass=10.0,
prescription='CL2013',
n=10.0, *a0*=1.0,
p0=1.0, *p1*=-1.5,
scale_up=False,
N_sys=10)

Make plots of solid surface density versus semi-major axis, including the feeding zones of each planet and a power-law fit to all the physical and observed planets, for each multi-planet system in a physical catalog.

Parameters

- **sssp_per_sys** (*dict*) – The dictionary containing the planetary and stellar properties for each system in a physical catalog (2-d and 1-d arrays).
- **sssp** (*dict*) – The dictionary containing the planetary and stellar properties of all planets in a physical catalog (1-d arrays).
- **n_mult_min** (*int*, *default*=2) – The minimum multiplicity to include.
- **n_mult_max** (*int*, *default*=10) – The maximum multiplicity to include.
- **max_core_mass** (*float*, *default*=10.) – The maximum allowed (solid) core mass (Earth masses).
- **prescription** (*{'CL2013', 'S2014', 'nHill', 'RC2014'}*, *default*='CL2013') – The string indicating the prescription to use for computing the feeding zone widths.
- **n** (*float*, *default*=10.) – The number of Hill radii to use for the feeding zones. Only used for the 'nHill' prescription.
- **a0** (*float*, *default*=1.) – The normalization point for the separation (AU).
- **p0** (*float*, *default*=1.) – The initial guess for the normalization parameter, 'log(sigma0)'.
- **p1** (*float*, *default*=-1.5) – The initial guess for the power-law index parameter, 'beta'.
- **scale_up** (*bool*, *default*=False) – Whether to scale up the normalization such that the power-law fit for each system is at/above the solid surface densities of every planet in the system.
- **N_sys** (*int*, *default*=10) – The maximum number of systems to plot. A separate figure will be plotted for each system.

solid_mass_integrated_r0_to_r_given_power_law_profile(*r*, *r0*, *sigma0*, *beta*, *a0*=1.0)

Compute the total solid mass from a power-law profile for the solid surface density, within a given separation.

Integrates the power-law over an annular area from radius *r0* to *r*.

Parameters

- **r** (*float*) – The separation (AU) of the outer boundary.
- **r0** (*float*) – The separation (AU) of the inner boundary.
- **sigma0** (*float*) – The normalization for the solid surface density (g/cm^2) at separation *a0*.
- **beta** (*float*) – The power-law index.
- **a0** (*float*, *default*=1.) – The normalization point for the separation (AU).

Returns

M_r – The total integrated solid mass (Earth masses).

Return type

float

Warning: The inner boundary *r0* must be non-zero to avoid infinite mass for *beta* <= -2.

PYTHON MODULE INDEX

S

`syssimpymmen.mmen`, [14](#)

INDEX

F

`feeding_zone_nHill()` (in module `syssimpymmen.mmen`), 15
`feeding_zone_RC2014()` (in module `syssimpymmen.mmen`), 16
`feeding_zone_S2014()` (in module `syssimpymmen.mmen`), 15
`fit_power_law_and_scale_up_MMEN()` (in module `syssimpymmen.mmen`), 22
`fit_power_law_MMEN()` (in module `syssimpymmen.mmen`), 21
`fit_power_law_MMEN_all_planets_observed()` (in module `syssimpymmen.mmen`), 22
`fit_power_law_MMEN_all_planets_physical()` (in module `syssimpymmen.mmen`), 23
`fit_power_law_MMEN_per_system_observed()` (in module `syssimpymmen.mmen`), 24
`fit_power_law_MMEN_per_system_observed_and_physical()` (in module `syssimpymmen.mmen`), 26
`fit_power_law_MMEN_per_system_physical()` (in module `syssimpymmen.mmen`), 25

G

`generate_planet_mass_from_radius_lognormal_mass_around_earth_like_rocky()` (in module `syssimpymmen.mmen`), 14
`generate_planet_mass_from_radius_Ning2018_table()` (in module `syssimpymmen.mmen`), 14
`generate_planet_mass_from_radius_Ning2018_table_above_lognormal_mass_earth_like_rocky_below()` (in module `syssimpymmen.mmen`), 14
`generate_planet_mass_from_radius_Ning2018_table_above_lognormal_mass_earth_like_rocky_below()` (in module `syssimpymmen.mmen`), 15

M

`MMEN_power_law()` (in module `syssimpymmen.mmen`), 21
`MMSN()` (in module `syssimpymmen.mmen`), 21
module
`syssimpymmen.mmen`, 14

P

`plot_feeding_zones_and_power_law_fit_MMEN_per_system_observed_and_physical()` (in module `syssimpymmen.mmen`), 27

S

`solid_mass_integrated_r0_to_r_given_power_law_profile()` (in module `syssimpymmen.mmen`), 28
`solid_surface_density()` (in module `syssimpymmen.mmen`), 16
`solid_surface_density_CL2013()` (in module `syssimpymmen.mmen`), 17
`solid_surface_density_CL2013_given_observed_catalog()` (in module `syssimpymmen.mmen`), 19
`solid_surface_density_CL2013_given_physical_catalog()` (in module `syssimpymmen.mmen`), 18
`solid_surface_density_nHill()` (in module `syssimpymmen.mmen`), 17
`solid_surface_density_nHill_given_observed_catalog()` (in module `syssimpymmen.mmen`), 20
`solid_surface_density_nHill_given_physical_catalog()` (in module `syssimpymmen.mmen`), 19
`solid_surface_density_prescription()` (in module `syssimpymmen.mmen`), 17
`solid_surface_density_RC2014_given_observed_catalog()` (in module `syssimpymmen.mmen`), 20
`solid_surface_density_RC2014_given_physical_catalog()` (in module `syssimpymmen.mmen`), 19
`solid_surface_density_S2014()` (in module `syssimpymmen.mmen`), 17
`solid_surface_density_S2014_given_observed_catalog()` (in module `syssimpymmen.mmen`), 20
`solid_surface_density_S2014_given_physical_catalog()` (in module `syssimpymmen.mmen`), 18
`solid_surface_density_system_RC2014()` (in module `syssimpymmen.mmen`), 17
`syssimpymmen.mmen`
module, 14
`_logM_linear_given_radius()` (in module `syssimpymmen.mmen`), 14