FuncComp Documentation

Daniel Garrett
Sibley School of Mechanical and Aerospace Engineering
Cornell University
Ithaca, NY 14850

ABSTRACT

This document describes the open source software package FuncComp. FuncComp contains two modules which calculate single-visit completeness for an assumed population of planets. The module Functional contains the class object Functional which generates the single-visit completeness using an analytical function approach while MonteCarlo does the same using a Monte Carlo trial approach.

CONTENTS

1 Single-Visit Completeness

Obscurational completeness was introduced by Brown as a necessary, but not sufficient, condition for detection of an exoplanet [1]. Assuming distributions for semi-major axis and eccentricity of planetary orbits, Brown defined obscurational completeness as the probability of a planet falling outside a telescope's central obscuration, thus becoming potentially observable. This concept was expanded to include selection effects due to photometric restrictions on exoplanet observability introduced by telescope optics [2]. Single-visit completeness is determined by the assumption that an exoplanet is observable if its angular separation from its star is greater than the observatory's inner working angle (IWA) and less than the observatory's outer working angle (OWA) while also being illuminated such that the difference in brightness between the star and planet (Δmag) is below a limiting value (Δmag_0). The IWA and OWA represent the minimum and maximum angular separation of the field of view. Δmag_0 is where unresolvable confusion between the planet signal and noise occur.

FuncComp provides two modules which calculate single-visit completeness along with other support modules and scripts. The module `MonteCarlo` calculates single-visit completeness using the procedure of Brown [2]. The module `Functional` calculates single-visit completeness using a functional approach (to be published).

2 Python Packages

The following Python packages are used for either `Functional` or `MonteCarlo`:

```
astropy
    astropy.units
copy_reg
matplotlib (optional)
numpy
pp (optional)
scipy
    scipy.integrate
    scipy.interpolate
sys
time (optional)
types
```

The packages `matplotlib` and `time` are used in the demonstration script `Plotting.py` included with FuncComp. The Parallel Python (`pp`) package, if installed, will utilize available cores of multi-core processors to compute single-visit completeness.

3 Supporting Modules

FuncComp includes the following supporting modules: `Population`, `statsFun`, and `util`. The `Population` module contains a class object which when instantiated contains probability density functions of the planetary quantities needed for computing single-visit completeness. The `statsFun` module contains a sampling function. The `util` module contains functions calculating the minimum and maximum Δmag for a given separation value.

3.1 Population

The `Population` module contains a class object which when instantiated contains as attributes minimum and maximum values of needed quantities and methods which calculate the probability density of these quantities. These values and methods may be changed by the user to give the desired planetary population.

3.1.1 Population Class Attributes

The `Population` class object contains the following attributes giving the minimum and maximum values of the planetary population parameters:

arange
: Semi-major axis range defined as [a_min, a_max] (astropy Quantity set in *AU*)

erange
: Eccentricity range defined as [e_min, e_max] (ndarray)

Rrange
: Planetary radius range defined as [R_min, R_max] (astropy Quantity set in *km*)

prange
: Planetary geometric albedo range defined as [p_min, p_max] (ndarray)

3.1.2 Population Class Methods

The `Population` class object contains the following methods which calculate the probability density value for each quantity:

semi_axis
: Semi-major axis probability density function (in AU^{-1})

eccentricity
: Eccentricity probability density function

Radius
: Planetary radius probability density function (in km^{-1})

albedo
: Geometric albedo probability density function

3.2 statsFun

The `statsFun` module contains functions which utilize a rejection sampling method to generate samples from a given probability density function. The function used is `simpSample`.

3.2.1 simpSample Input/Output Description

Inputs

f
: Probability density function (callable)

numTest
: Number of samples desired (integer)

xMin
: Minimum value of population (float)

xMax
: Maximum value of population (float)

Outputs

X
: Array of samples from the population (ndarray)

3.3 util

The `util` module contains two functions `maxdmag` and `mindmag` which calculate the maximum and minimum Δmag for a given separation value and maximum and minimum planet population values.

3.3.1 maxdmag Input/Output Description

Inputs

s

Array of apparent separation in *AU* (ndarray)

ranges

Tuple containing:

pmin : minimum geometric albedo (float)
Rmin : minimum planetary radius in *km* (float)
rmax : maximum planetary distance from star in *AU* (float)

x

Conversion factor for *AU* to *km* (float)

Output

maxdmag

Array of maximum Δmag values (ndarray)

3.3.2 mindmag Input/Output Description

Inputs

s

Array of apparent separation in *AU* (ndarray)

ranges

Tuple containing:

pmax : maximum geometric albedo (float)
Rmax : maximum planetary Radius in *km* (float)
rmin : minimum planetary distance from star in *AU* (float)
rmax : maximum planetary distance from star in *AU* (float)

x

Conversion factor for *AU* to *km* (float)

Output

mindmag

Array of minimum Δmag values (ndarray)

4 Functional Module

The `Functional` module contains the `Functional` class object which when instantiated contains the information for single-visit completeness. The module also contains functions `Jac`, `onef_zeta`, `onef_R2`, `onef_r`, `onef_dmags`, and `onef_dmagsz` which aid in the determination of the single-visit completeness joint probability density function $f_{\bar{s},\overline{\Delta\mathrm{mag}}}(s,\Delta\mathrm{mag})$.

4.1 Functional Class Object

4.1.1 Initialization Input/Output Description

Inputs

smin
: Minimum apparent separation value in AU, default is zero (float)

smax
: Maximum apparent separation value in AU, default is None and the maximum apparent separation is determined based on the assumed population values from the instantiated `Population` object (float)

ns
: Number of points in separation to calculate, default is 400 (integer)

dmagmin
: Minimum Δmag, default is None and minimum Δmag is determined based on the assumed population values from the instantiated `Population` object (float)

dmagmax
: Maximum Δmag, default is None and maximum Δmag is determined based on the assumed population values from the instantiated `Population` object (float)

ndmag
: Number of points in Δmag to calculate, default is 400 (integer)

Attributes

s
: Points sampled in apparent separation in AU (ndarray)

dmag
: Points sampled in Δmag (ndarray)

pc
: Single-visit completeness joint probability density function evaluated at s and dmag in $[AU \cdot \text{mag}]^{-1}$ (ndarray)

grid
: Interpolant of single-visit completeness joint probability density function $f_{\bar{s},\overline{\Delta\text{mag}}}(s, \Delta\text{mag})$ in $[AU \cdot \text{mag}]^{-1}$ (callable(s,Δmag))

pdf
: Vectorized version of **grid** (callable(s,Δmag))

4.1.2 comp Input/Output Description

The method `comp` belonging to the `Functional` class object calculates the single-visit completeness.

Inputs

smin
: Array of minimum apparent separation in AU (ndarray)

smax
: Array of maximum apparent separation in AU (ndarray)

dmagmin
: Array of minimum Δmag values (ndarray)

dmagmax
: Array of maximum Δmag values (ndarray)

Output

f

Array of completeness values (ndarray)

4.2 f_zeta Class Object

The `f_zeta` class object generates the probability density function for $\zeta = pR^2$ used in calculating the single-visit completeness joint probability density function.

4.2.1 Initialization Input/Output Description

Inputs

ranges

Tuple containing:

pmin : minimum value of geometric albedo (float)
pmax : maximum value of geometric albedo (float)
Rmin : minimum value of planetary radius in *km* (float)
Rmax : maximum value of planetary radius in *km* (float)

pdfs

Tuple containing:

f_p : probability density function for geometric albedo (callable(p))
f_R : probability density function for planetary radius (callable(R))

n

Number of sample points in ζ

Attributes

pmin

Minimum value of geometric albedo (float)

pmax

Maximum value of geometric albedo (float)

Rmin

Minimum value of planetary radius in *km* (float)

Rmax

Maximum value of planetary radius in *km* (float)

f_p

Probability density function for geometric albedo

f_R

Probability density function for planetary radius

f_z

Probability density function for $\zeta = pR^2$ if p and R are both not constant

4.2.2 pRconst Input/Output Description

The method `pRconst` belonging to the `f_zeta` class object calculates the probability density function of $\zeta = pR^2$ when p and R take constant values.

Input

zi

Value of ζ in km^2 (ndarray)

Output

f

This method is a dummy method, 1. is returned

4.2.3 pconst Input/Output Description

The method `pconst` belonging to the `f_zeta` class object calculates the probability density function of $\zeta = pR^2$ when p is constant and R is a random variable.

Input

zi

Value of ζ in km^2 (ndarray)

Output

f

Probability density of $\zeta = pR^2$

4.2.4 Rconst Input/Output Description

The method `Rconst` belonging to the `f_zeta` class object calculates the probability density function of $\zeta = pR^2$ when R is constant and p is a random variable.

Input

zi

Value of ζ in km^2 (ndarray)

Output

f

Probability density of $\zeta = pR^2$

4.3 Jac Input/Output Description

The function `Jac` calculates the determinant of the Jacobian matrix of the change of variables necessary for calculating the joint probability density function.

Input

b

Phase angle in *rad* (ndarray)

Output

J

Determinant of Jacobian matrix (ndarray)

4.4 onef_zeta Input/Output Description

The function `onef_zeta` calculates the probability density function $f_{\zeta}(\zeta)$ for $\zeta = pR^2$ for values of p and R which are random variables.

Inputs

zetai

Value of $\zeta = pR^2$ in km^2 (float)

pdfs

Tuple containing:

f_p : Probability density function for albedo (callable(p))
f_R : Probability density function for planetary radius in (callable(R))

ranges

Tuple containing:

pmin : minimum value of geometric albedo (float)
pmax : maximum value of geometric albedo (float)
Rmin : minimum value of planetary radius in *km* (float)
Rmax : maximum value of planetary radius in *km* (float)

Output

f

Probability density for given zetai in km^{-2} (float)

4.5 onef_R2 Input/Output Description

The function `onef_R2` calculates the probability density of planetary radius squared.

Inputs

R2

Value of planetary radius squared (float)

pdf

Probability density function of planetary radius (callable(R))

Output

f

Value of probability density function of planetary radius squared in km^{-2} (float)

4.6 onef_r Input/Output Description

The function `onef_r` calculates the probability density of orbital radius or distance from the star to the planet, r, when eccentricity, e, and semi-major axis, a, are random variables.

Inputs

ri

Value of orbital radius in *AU* (float)

pdfs

Tuple containing:

f_e : probability density function for eccentricity (callable(e))
f_a : probability density function for semi-major axis (callable(a))

ranges

Tuple containing:

amin : minimum semi-major axis in *AU* (float)
amax : maximum semi-major axis in *AU* (float)

emin : minimum eccentricity (float)
emax : maximum eccentricity (float)

Output

f

Value of probability density for orbital radius in AU^{-1} (float)

4.7 onef_r_aeconst Input/Output Description

The function `onef_r_aeconst` calculates the probability density of r when a and e are constant.

Inputs

r

Value of r in AU (float)

a

Constant value of a in AU (float)

e

Constant value of e (float)

Output

f

Probability density of r

4.8 onef_r_aconst Input/Output Description

The function `onef_r_aconst` calculates the probability density of r when a is constant and e is random.

Inputs

r

Value of r in AU (float)

a

Constant value of a in AU (float)

e

Array containing minimum and maximum values of e (ndarray)

f_e

Probability density function of eccentricity (callable(e))

Output

f

Probability density of r

4.9 onef_r_econst Input/Output Description

The function `onef_r_econst` calculates the probability density of r when e is constant and a is random.

Inputs

r
Value of r in AU (float)

e
Constant value of e (float)

a
Array containing minimum and maximum values of a in AU (ndarray)

f_a
Probability density function of semi-major axis (callable(a))

Output

f
Probability density of r

4.10 onef_dmags Input/Output Description

The function `onef_dmags` calculates the single-visit completeness joint probability density of separation and Δmag, $f_{\bar{s},\overline{\Delta\text{mag}}}(s,\Delta\text{mag})$.

Inputs

dmag
Value of difference in magnitude, Δmag (float)

s
Value of apparent separation in AU (float)

ranges
Tuple containing:

pmin : minimum value of geometric albedo (float)
pmax : maximum value of geometric albedo (float)
Rmin : minimum value of planetary radius in km (float)
Rmax : maximum value of planetary radius in km (float)
rmin : minimum value of orbital radius in AU (float)
rmax : maximum value of orbital radius in AU (float)
zmin : minimum value of $\zeta = pR^2$ in km^2 (float)
zmax : maximum value of $\zeta = pR^2$ in km^2 (float)

val
Value of $\sin^2(\beta^*)\Phi(\beta^*)$ (float)

pdfs
Tuple containing:

f_z : probability density function for $\zeta = pR^2$ (callable(ζ))
f_r : probability density function for orbital radius (callable(r))

funcs
Tuple containing:

binv1 : inverse function of $\sin^2(\beta)\Phi(\beta)$ for $0 < \beta < \beta^*$ (callable(P))
binv2 : inverse function of $\sin^2(\beta)\Phi(\beta)$ for $\beta^* < \beta < \pi$ (callable(P))

x
Conversion factor between AU and km (float)

Output

f

Joint probability density for given s and dmag in $[AU \cdot \text{mag}]^{-1}$ (float)

4.11 onef_dmagsz Input/Output Description

The function `onef_dmagsz` calculates the joint probability density function $f_{\bar{s},\overline{\Delta\text{mag}},\bar{\zeta}}(s, \Delta\text{mag}, \zeta)$.

Inputs

z

Value of $\zeta = pR^2$ in km^2 (float)

dmag

Value of difference in brightness magnitude, Δmag (float)

s

Value of apparent separation in AU (float)

val

Value of $\sin^2(\beta^*)\Phi(\beta^*)$ (float)

pdfs

Tuple containing:

f_z : probability density function for $\zeta = pR^2$ (callable(ζ))

f_r : probability density function for orbital radius (callable(r))

funcs

Tuple containing:

binv1 : inverse function of $\sin^2(\beta)\Phi(\beta)$ for $0 < \beta < \beta^*$ (callable(P))

binv2 : inverse function of $\sin^2(\beta)\Phi(\beta)$ for $\beta^* < \beta < \pi$ (callable(P))

x

Conversion factor between AU and km (float)

Output

f

Joint probability density $f_{\bar{s},\overline{\Delta\text{mag}},\bar{\zeta}}(s, \Delta\text{mag}, \zeta)$ in $\left[AU \cdot \text{mag} \cdot km^2\right]^{-1}$ (float)

5 MonteCarlo Module

The `MonteCarlo` module contains the `MonteCarlo` class object which when instantiated contains the information for single-visit completeness using a Monte Carlo trial approach. The module also contains the function `oneMCsim` which aids in the determination of the single-visit completeness joint probability density function $f_{\bar{s},\overline{\Delta\text{mag}}}(s, \Delta\text{mag})$.

5.1 MonteCarlo Class Object

5.1.1 Initialization Input/Output Description

Inputs

Nplanets

Number of planetary samples, default is 1e6 (float)

bins

Number of bins in apparent separation for 2-D histogram, default is 400 (integer)

bindmag

Number of bins in difference in magnitude for 2-D histogram, default is 400 (integer)

smin
Minimum value of apparent separation in AU, default is 0 (float)

smax
Maximum value of apparent separation in AU, default is None and maximum value is determined based on assumed planetary population from the instantiated `Population` object (float)

dmagmin
Minimum value of difference in magnitude, default is None and minimum value is determined based on assumed planetary population from the instantiated `Population` object (float)

dmagmax
Maximum value of difference in magnitude, default is None and maximum is determined based on assumed planetary population from the instantiated `Population` object (float)

Attributes

s
Points sampled in apparent separation in AU (ndarray)

dmag
Points sampled in Δmag (ndarray)

Hc
Normalized 2-D histogram representing single-visit completeness joint probability density function evaluated at s and dmag in $[AU \cdot \Delta\text{mag}]^{-1}$ (ndarray)

grid
Interpolant of single-visit completeness joint probability density function $f_{\bar{s},\overline{\Delta\text{mag}}}(s, \Delta\text{mag})$ in $[AU \cdot \Delta\text{mag}]^{-1}$ (callable(s,Δmag))

pdf
Vectorized version of **grid** (callable(s,Δmag))

5.1.2 comp Input/Output Description

The method `comp` belonging to the `MonteCarlo` class object calculates the single-visit completeness.

Inputs

smin
Array of minimum apparent separation in AU (ndarray)

smax
Array of maximum apparent separation in AU (ndarray)

dmagmin
Array of minimum Δmag values (ndarray)

dmagmax
Array of maximum Δmag values (ndarray)

Output

f
Array of single-visit completeness values (ndarray)

5.2 oneMCsim Input/Output Description

The function `oneMCsim` performs the Monte Carlo trials to help calculate the single-visit completeness joint probability density function $f_{\bar{s},\overline{\Delta\mathrm{mag}}}(s,\Delta\mathrm{mag})$.

Inputs

nplan

Number of sample planets (integer)

pdfs

Tuple containing:

f_e : probability density function for eccentricity (callable(e))
f_a : probability density function for semi-major axis (callable(a))
f_R : probability density function for planetary radius (callable(R))
f_p : probability density function for geometric albedo (callable(p))

ranges

Tuple containing:

emin : minimum eccentricity (float)
emax : maximum eccentricity (float)
amin : minimum semi-major axis in *AU* (float)
amax : maximum semi-major axis in *AU* (float)
Rmin : minimum planetary radius in *km* (float)
Rmax : maximum planetary radius in *km* (float)
pmin : minimum geometric albedo (float)
pmax : maximum geometric albedo (float)
smin : minimum apparent separation in *AU* (float)
smax : maximum apparent separation in *AU* (float)
dmagmin : minimum Δmag (float)
dmagmax : maximum Δmag (float)

binh

Tuple containing:

bins : number of bins in apparent separation for 2-D histogram (integer)
bindmag : number of bins in Δmag for 2-D histogram (integer)

x

Conversion factor of *AU* to *km* (float)

Outputs

hc

2-D histogram values for apparent separation and Δmag in $[AU \cdot \mathrm{mag}]^{-1}$ (ndarray)

sedges

Edge values for 2-D histogram bins in apparent separation in *AU* (ndarray)

dmagedges

Edge values for 2-D histogram bins in Δmag (ndarray)

References

[1] Brown, R. A., 2004. "Obscurational completeness". *The Astrophysical Journal,* **607**(2), p. 1003.

[2] Brown, R. A., 2005. "Single-visit photometric and obscurational completeness". *The Astrophysical Journal,* **624**(2), p. 1010.