ExoSpec

Generated by Doxygen 1.8.13

Contents

Chapter 1

ExoSpec

1.1 Introduction

Exospec is a python tool for fitting your multi-wavelength transit light-curves It can accept an arbitrary number of wavelength channels and an arbitrary number of auxiliary measurements. Currently the fitting has two Gaussian Process kernel options: the kernel outlined in Gibson 2011 which incorporates auxiliary measurements made over the course of transit observation and a more general squared exponential.

1.2 Installation

ExoSpec uses numpy and matplotlib, as well as the Python packages batman, emcee, corner, pandas and george.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 5

Class Documentation

5.1 DifferentFileSize Class Reference

Raise when one the light curve file does not have the same size with light curve file for the lowest wavelength (this file serves as reference)

5.1.1 Detailed Description

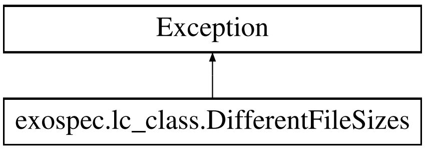
Raise when one the light curve file does not have the same size with light curve file for the lowest wavelength (this file serves as reference)

The documentation for this class was generated from the following file:

- /Users/heatherp/Documents/Courses/Computational/Project/ExoplanetSpectra/exospec/lc_class.py

5.2 exospec.lc_class.DifferentFileSizes Class Reference

Inheritance diagram for exospec.lc_class.DifferentFileSizes:



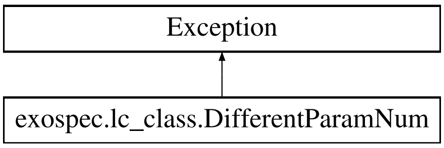
The documentation for this class was generated from the following file:

- /Users/heatherp/Documents/Courses/Computational/Project/ExoplanetSpectra/exospec/lc_class.py

5.3 exospec.lc_class.DifferentParamNum Class Reference

Raise when one the light curve file does not have the same number of parameters with light curve file for the lowest wavelength (this file serves as reference)

Inheritance diagram for exospec.lc_class.DifferentParamNum:



5.3.1 Detailed Description

Raise when one the light curve file does not have the same number of parameters with light curve file for the lowest wavelength (this file serves as reference)

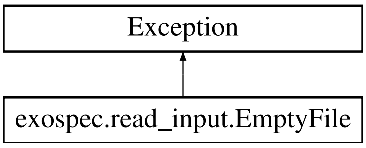
The documentation for this class was generated from the following file:

- /Users/heatherp/Documents/Courses/Computational/Project/ExoplanetSpectra/exospec/lc_class.py

5.4 exospec.read_input.EmptyFile Class Reference

Raise exception when the input file is empty or only has comments.

Inheritance diagram for exospec.read_input.EmptyFile:



5.4.1 Detailed Description

Raise exception when the input file is empty or only has comments.

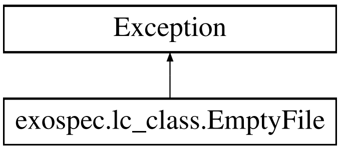
The documentation for this class was generated from the following file:

- /Users/heatherp/Documents/Courses/Computational/Project/ExoplanetSpectra/exospec/read_input.py

5.5 exospec.lc_class.EmptyFile Class Reference

Raise when the light curve file is empty.

Inheritance diagram for exospec.lc_class.EmptyFile:



5.5.1 Detailed Description

Raise when the light curve file is empty.

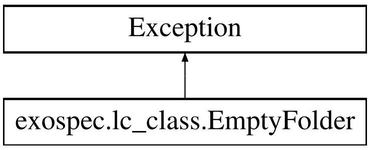
The documentation for this class was generated from the following file:

- /Users/heatherp/Documents/Courses/Computational/Project/ExoplanetSpectra/exospec/lc_class.py

5.6 exospec.lc_class.EmptyFolder Class Reference

Raise exception when the light curve folder is empty.

Inheritance diagram for exospec.lc_class.EmptyFolder:



5.6.1 Detailed Description

Raise exception when the light curve folder is empty.

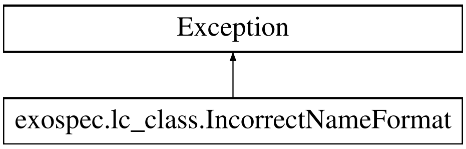
The documentation for this class was generated from the following file:

- /Users/heatherp/Documents/Courses/Computational/Project/ExoplanetSpectra/exospec/lc_class.py

5.7 exospec.lc_class.IncorrectNameFormat Class Reference

Raise when the light curve file name is not under the expected format sample_lc_<wavelength>.txt.

Inheritance diagram for exospec.lc_class.IncorrectNameFormat:



5.7.1 Detailed Description

Raise when the light curve file name is not under the expected format sample_lc_<wavelength>.txt.

The documentation for this class was generated from the following file:

- /Users/heatherp/Documents/Courses/Computational/Project/ExoplanetSpectra/exospec/lc_class.py

5.8 exospec.lc_class.LightCurve Class Reference

List all the light curve files in the indicated folder if the user decided to use another wavelength resolution than the one provided with the files, the code compute the resulting number of new wavelength to be used loop over the light curve files to get their file names and original wavelengths if the user decided to use a new wavelength resolution, the code computes the new wavelength to be used instantiate the new lc objects (with new wave_bin_size) Warning: The code is not able yet to handle situation where files_num is not proportional to wave_bin_size.

Public Member Functions

- def **__init__** (self, PathToLC, wave_bin_size)
- def LC_dic (self)

  Return LC_dic.
- def wave_length (self)

  Return wave_length.
- def new_wave_length (self)

  Return new_wave_length.
- def store_transit_model (self, transit_model)

  Stores transit_model.

Public Attributes

- **files_list**
- **files_num**
- **wave_length**
- **new_wave_length**
- **LC_dic**
- **obj_mcmc**
- **obj_chain**
- **obj_mcmcGP**
- **obj_chainGP**
- **transit_model**

5.8.1 Detailed Description

List all the light curve files in the indicated folder if the user decided to use another wavelength resolution than the one provided with the files, the code compute the resulting number of new wavelength to be used loop over the light curve files to get their file names and original wavelengths if the user decided to use a new wavelength resolution, the code computes the new wavelength to be used instantiate the new lc objects (with new wave_bin_size) Warning: The code is not able yet to handle situation where files_num is not proportional to wave_bin_size.

5.8.2 Member Function Documentation

5.8.2.1 LC_dic()

```
def exospec.lc_class.LightCurve.LC_dic (
              self )
```

Return LC_dic.

Returns

LC_dic the dictionary that contains the light curve objects and which are references by their wavelength as keys

5.8.2.2 new_wave_length()

```
def exospec.lc_class.LightCurve.new_wave_length (
              self )
```

Return new_wave_length.

Returns

new_wave_length a list containing the new user defined wavelength of the ligth curve objects

5.8.2.3 store_transit_model()

```
def exospec.lc_class.LightCurve.store_transit_model (
              self,
              transit_model )
```

Stores transit_model.

Parameters

transit_model	

5.8.2.4 wave_length()

```
def exospec.lc_class.LightCurve.wave_length (
self )
```

Return wave_length.

Returns

wave_length a list containing the original wavelength of the ligth curve files

The documentation for this class was generated from the following file:

- /Users/heatherp/Documents/Courses/Computational/Project/ExoplanetSpectra/exospec/lc_class.py

5.9 exospec.lc_class.LightCurveData Class Reference

this class is used by the class LightCurve to extract, store and process the data from the light curve files

Public Member Functions

- def **__init__** (self, Path_to_files)
- def len_file (self)

  Returns len_file.
- def time (self)

  Return time.
- def flux (self)

  Returns flux.
- def ferr (self)

  Returns ferr.
- def param_num (self)

  Returns param_num.
- def param_name (self)

  Returns param_name.
- def param_list (self)

  Returns param_list.
- def new_time_bin (self, bin_size)

  Change the time resolution of a light curve object enables the user to use a new time resolution.
- def plot_flux_time (self, bin_size)

  Plot the flux against the time plot the flux with a new time resolution using function new_time_bin.
- def plot_flux_param (self, param_index)

  Plot the flux against a parameter plot the flux againt the parameter indicated by the user.

Public Attributes

- **len_file**
- **time**
- **flux**
- **ferr**
- **param_num**
- **param_name**
- **param_list**

5.9.1 Detailed Description

this class is used by the class LightCurve to extract, store and process the data from the light curve files

Parameters

Path_to_files	A list that contains the path(s) to the file(s) that will be used to create a light curve object (multiple files can be lumped together to create a ligh curve object)

5.9.2 Member Function Documentation

5.9.2.1 ferr()

```
def exospec.lc_class.LightCurveData.ferr (
              self )
```

Returns ferr.

Returns

ferr the error on the flux

5.9.2.2 flux()

```
def exospec.lc_class.LightCurveData.flux (
              self )
```

Returns flux.

Returns

flux the flux vector

5.9.2.3 len_file()

```
def exospec.lc_class.LightCurveData.len_file (
              self )
```

Returns len_file.

Returns

len_file the length of the light curve file

5.9.2.4 new_time_bin()

```
def exospec.lc_class.LightCurveData.new_time_bin (
              self,
              bin_size )
```

Change the time resolution of a light curve object enables the user to use a new time resolution.

Parameters

bin_size	number of time points to lump together

5.9.2.5 param_list()

```
def exospec.lc_class.LightCurveData.param_list (
              self )
```

Returns param_list.

Returns

param_list a list containing the parameters

5.9.2.6 param_name()

```
def exospec.lc_class.LightCurveData.param_name (
              self )
```

Returns param_name.

Returns

param_name a list containing the names of the parameters

5.9.2.7 param_num()

```
def exospec.lc_class.LightCurveData.param_num (
              self )
```

Returns param_num.

Returns

param_num the number of parameters defined in the light curve file

5.9.2.8 plot_flux_param()

```
def exospec.lc_class.LightCurveData.plot_flux_param (
              self,
              param_index )
```

Plot the flux against a parameter plot the flux againt the parameter indicated by the user.

Parameters

param_index	the index of the param selected by the user in param_list

5.9.2.9 plot_flux_time()

```
def exospec.lc_class.LightCurveData.plot_flux_time (
              self,
              bin_size )
```

Plot the flux against the time plot the flux with a new time resolution using function new_time_bin.

Parameters

bin_size	number of time points to lump together

5.9.2.10 time()

```
def exospec.lc_class.LightCurveData.time (
              self )
```

Return time.

Returns

time the time vector

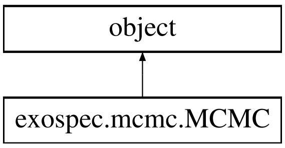
The documentation for this class was generated from the following file:

- /Users/heatherp/Documents/Courses/Computational/Project/ExoplanetSpectra/exospec/lc_class.py

5.10 exospec.mcmc.MCMC Class Reference

Class to run MCMC to fit curve and produce basic diagnostic plots and statistics Uses emcee (to run MCMC) and corner (to produce triangle plots)

Inheritance diagram for exospec.mcmc.MCMC:



Public Member Functions

- def __init__ (self, t, val, err, ln_prob_fn, transit_params, hyper_params, num_walkers, num_threads)

  The constructor.
- def run (self, pos, burnin_steps, production_run_steps)

  Runs the MCMC should run emcee given a log probability function result is the MCMC chains which are saved as an object attribute.
- def save_chain (self, filename)

  Saves the chain as a numpy array.
- def get_mean_acceptance_fraction (self)

  Allows the user to access the mean acceptance fraction, which should be around 1/2.
- def get_median_and_errors (self)

  Best fit parameters and 1 sigma errors.
- def triangle_plot (self, extra_burnin_steps=0, theta_true=None, plot_transit_params=True, plot_hyper_↩ params=True, save_as_dir=".", save_as_name="triangle.png")

  Makes a triangle plot If an error is encountered the function returns 1 but does not raise an exception.
- def walker_plot (self, extra_burnin_steps=0, theta_true=None, plot_transit_params=True, plot_hyper_↩ params=True, save_as_dir=".", save_as_name="walkers.png")

  Plots the chains of each walker and a histogram showing how each parameter was sampled If an error is encountered the function returns 1 but does not raise an exception.
- def light_curve_plot (self, model, extra_burnin_steps=0, theta_true=None, plot_transit_params=True, plot_↩ hyper_params=True, save_as_dir=".", save_as_name="light_curve")

  Plots the chains of each walker and a histogram showing how each parameter was sampled If an error is encountered the function returns 1 but does not raise an exception.

5.10.1 Detailed Description

Class to run MCMC to fit curve and produce basic diagnostic plots and statistics Uses emcee (to run MCMC) and corner (to produce triangle plots)

5.10.2 Constructor & Destructor Documentation

5.10.2.1 __init__()

```
def exospec.mcmc.MCMC.__init__ (
              self,
              t,
              val,
              err,
              ln_prob_fn,
              transit_params,
              hyper_params,
              num_walkers,
              num_threads )
```

The constructor.

Parameters

self	The object pointer
t	A numpy array of the independent variable for the data to be fitted
val	A numpy array of the dependent variable for the data to be fitted
err	A numpy array of the errors on the dependent variable
ln_prob_fn	The log probability function to be sampled by the MCMC chain
transit_params	A list of strings giving the names of the curve's parameter's
hyper_params	A list of strings giving the names of noise parameters
num_walkers	Integer giving the number of walkers for the MCMC run
num_threads	An integer giving the number of threads to use on each core

5.10.3 Member Function Documentation

5.10.3.1 get_mean_acceptance_fraction()

```
def exospec.mcmc.MCMC.get_mean_acceptance_fraction (
              self )
```

Allows the user to access the mean acceptance fraction, which should be around 1/2.

Parameters

self	The object pointer

Returns

Mean acceptance fraction

5.10.3.2 get_median_and_errors()

```
def exospec.mcmc.MCMC.get_median_and_errors (
              self )
```

Best fit parameters and 1 sigma errors.

Parameters

self	The object pointer

Returns

Three numpy arrays giving the median and one sigma errors for each parameter

5.10.3.3 light_curve_plot()

```
def exospec.mcmc.MCMC.light_curve_plot (
              self,
              model,
              extra_burnin_steps = 0,
              theta_true = None,
              plot_transit_params = True,
              plot_hyper_params = True,
              save_as_dir = ".",
              save_as_name = "light_curve" )
```

Plots the chains of each walker and a histogram showing how each parameter was sampled If an error is encountered the function returns 1 but does not raise an exception.

These plots are useful for visualization but should not cause the code to crash, as the main purpose is to create and save the MCMC chains

Parameters

self	The object pointer
model	A function that returns the lightcurve shape as a function of the light curve parameters and time
extra_burnin_steps	Number of steps (in addition to burnin_steps from run) at the start of each chain to neglect
theta_true	Numpy array of true parameter values if known (used for test data)
plot_transit_params	Boolean value specifying whether or not to plot the transit parameters
plot_hyper_params	Boolean value specifying whether or not to plot the hyper parameters
save_as_dir	Directory where plot should be saved. Default is current working Directory
save_as_name	Name under which plot should be saved

Return values

0	if successful
1	on failure

5.10.3.4 run()

```
def exospec.mcmc.MCMC.run (
              self,
              pos,
              burnin_steps,
              production_run_steps )
```

Runs the MCMC should run emcee given a log probability function result is the MCMC chains which are saved as an object attribute.

Parameters

self	The object pointer
pos	A 2D numpy array giving the initial positions of the walkers in parameter space
burnin_steps	An integer giving the number of initial steps to take to start exploring the parameter space before starting to save the chains
production_run_steps	The number of steps to take for each walker after the burnin phase

Returns

A 2D numpy array with all the samples for each of the transit and hyper parameters

5.10.3.5 save_chain()

```
def exospec.mcmc.MCMC.save_chain (
              self,
              filename )
```

Saves the chain as a numpy array.

Parameters

self	The object pointer
filename	The filename including path where the chains should be saved

Return values

0	if successful
1	if an IO error occurs

5.10.3.6 triangle_plot()

```
def exospec.mcmc.MCMC.triangle_plot (
              self,
              extra_burnin_steps = 0,
              theta_true = None,
              plot_transit_params = True,
              plot_hyper_params = True,
              save_as_dir = ".",
              save_as_name = "triangle.png" )
```

Makes a triangle plot If an error is encountered the function returns 1 but does not raise an exception.

These plots are useful for visualization but should not cause the code to crash, as the main purpose is to create and save the MCMC chains

Parameters

self	The object pointer
extra_burnin_steps	Number of steps (in addition to burnin_steps from run) at the start of each chain to neglect
theta_true	Numpy array of true parameter values if known (used for test data)
plot_transit_params	Boolean value specifying whether or not to plot the transit parameters
plot_hyper_params	Boolean value specifying whether or not to plot the hyper parameters
save_as_dir	Directory where plot should be saved. Default is current working Directory
save_as_name	Name under which plot should be saved

Return values

0	if successful
1	on failure

5.10.3.7 walker_plot()

```
def exospec.mcmc.MCMC.walker_plot (
              self,
              extra_burnin_steps = 0,
              theta_true = None,
              plot_transit_params = True,
              plot_hyper_params = True,
              save_as_dir = ".",
              save_as_name = "walkers.png" )
```

Plots the chains of each walker and a histogram showing how each parameter was sampled If an error is encountered the function returns 1 but does not raise an exception.

These plots are useful for visualization but should not cause the code to crash, as the main purpose is to create and save the MCMC chains

Parameters

self	The object pointer
extra_burnin_steps	Number of steps (in addition to burnin_steps from run) at the start of each chain to neglect
theta_true	Numpy array of true parameter values if known (used for test data)
plot_transit_params	Boolean value specifying whether or not to plot the transit parameters
plot_hyper_params	Boolean value specifying whether or not to plot the hyper parameters
save_as_dir	Directory where plot should be saved. Default is current working Directory
save_as_name	Name under which plot should be saved

Return values

0	if successful
1	on failure

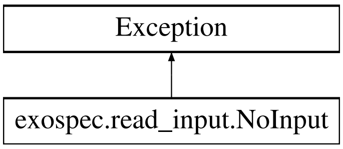
The documentation for this class was generated from the following file:

- /Users/heatherp/Documents/Courses/Computational/Project/ExoplanetSpectra/exospec/mcmc.py

5.11 exospec.read_input.NoInput Class Reference

Raise exception when the input entry is not properly set.

Inheritance diagram for exospec.read_input.NoInput:



5.11.1 Detailed Description

Raise exception when the input entry is not properly set.

The documentation for this class was generated from the following file:

- /Users/heatherp/Documents/Courses/Computational/Project/ExoplanetSpectra/exospec/read_input.py

5.12 exospec.read_input.read_input Class Reference

reads the input file and stores the input entries

Public Member Functions

- def **__init__** (self, input_file)
- def param_dic (self)

  Returns the parameters dictionary.
- def is_float (self, string)

  Check if the string contains a float.

Public Attributes

- **param_dic**

5.12.1 Detailed Description

reads the input file and stores the input entries

5.12.2 Member Function Documentation

5.12.2.1 param_dic()

```
def exospec.read_input.read_input.param_dic (
              self )
```

Returns the parameters dictionary.

Returns

param_dic the parameters dictionary

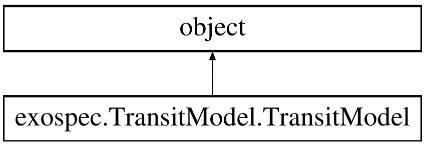
The documentation for this class was generated from the following file:

- /Users/heatherp/Documents/Courses/Computational/Project/ExoplanetSpectra/exospec/read_input.py

5.13 exospec.TransitModel.TransitModel Class Reference

Class to estimate the Transit Model with the customized kernel.

Inheritance diagram for exospec.TransitModel.TransitModel:



Public Member Functions

- def __init__ (self, kwargs)

  The constructor.
- def set_values (self, dict_of_values, kwargs)

  Set the parameters of the model based on the values provided.
- def read_limb_dark_params (self, kwargs)

  Forms a list of the limb darkening parameters from the user input In case some values are missed, the default ones will be taken.
- def read_errors_data (self, kwargs)

  Collects the data about errors that was passed In case some values are missed, the default ones will be taken.
- def update_data (self, time=None, obs=None, kwargs)

  Updates the data for the given parameters.
- def update_transit_params (self, rp_new, u_new)

  Updates the parameters of the model.
- def update_kernel_params (self, a_new=None, gamma_new=None, variance_new=None)

  Updates the hyperparameters of the kernel function.

- def updateTransitMode (self)

  Updates the transit model parameters.
- def model (self)

  Returns the flux values array.
- def **model** (self, params)
- def meanfnc (self, t)

  Mean function for the kernel meaan estimation.
- def kernelfnc (self, x1, x2, p=None)

  Computes the kernel function for the arbitrary sources of errors in the observations.
- def lnlike_gp (self)

  Computes the log likelihood from gaussian process.
- def lnprior_base (self)

  Checks if the batman parameters are within the predefined prior ranges.
- def lnprior_gp (self)

  Checks if the kernel parameters are within the predefined prior ranges.
- def lnprob_gp (self)

  Computes the log probability of the parameters for the given data.
- def sample_conditional (self, p, t, y, yerr)

  For a given set of parameters get predicted y values at t, and separate this into the transit signal component and the noise component.
- def lnprob_mcmc (self, p, t, y, yerr)

  MCMC API for the Transit Model object.

Public Attributes

- **batman_default_params**
- **transit_default_priors**
- **kernel_default_params**
- **kernel_default_priors**
- **data_defaults**
- **n_errors**
- **err_names**
- **errors_list**
- **params**
- **batman_model**
- **model**
- **data_dict**
- **model_initialized**
- **kernel_type**
- **lnprob**

5.13.1 Detailed Description

Class to estimate the Transit Model with the customized kernel.

More details.

5.13.2 Constructor & Destructor Documentation

5.13.2.1 __init__()

```
def exospec.TransitModel.TransitModel.__init__ (
              self,
              kwargs )
```

The constructor.

Takes the disctionary of the parameters and data to customize the object. In case some values are missed, the default one are used.

Parameters

self	The object pointer
∗∗*kwargs*	Accepts the dictionary of data, transit and kernel parameters

5.13.3 Member Function Documentation

5.13.3.1 kernelfnc()

```
def exospec.TransitModel.TransitModel.kernelfnc (
              self,
              x1,
              x2,
              p = None )
```

Computes the kernel function for the arbitrary sources of errors in the observations.

Parameters

self	The object pointer
x1	First time coordinate
x2	Second time coordinate
p(=None)	Kernel auxiliary parameters

Returns

Covariance between two points in time

5.13.3.2 lnlike_gp()

```
def exospec.TransitModel.TransitModel.lnlike_gp (
              self )
```

Computes the log likelihood from gaussian process.

Parameters

self	The object pointer

Returns

Log likelihood of a set of observations under the Gaussian process model.

5.13.3.3 lnprior_base()

```
def exospec.TransitModel.TransitModel.lnprior_base (
              self )
```

Checks if the batman parameters are within the predefined prior ranges.

Parameters

self	The object pointer

Returns

Returns 0 in case transit parameters whithin the prior range and -inf otherwise

5.13.3.4 lnprior_gp()

```
def exospec.TransitModel.TransitModel.lnprior_gp (
              self )
```

Checks if the kernel parameters are within the predefined prior ranges.

Parameters

self	The object pointer

Returns

Returns -inf in case parameters out of the range and 0.0 if within the prior range

5.13.3.5 lnprob_gp()

```
def exospec.TransitModel.TransitModel.lnprob_gp (
              self )
```

Computes the log probability of the parameters for the given data.

Parameters

self	The object pointer

Returns

Log probability of the parameters

5.13.3.6 lnprob_mcmc()

```
def exospec.TransitModel.TransitModel.lnprob_mcmc (
              self,
              p,
              t,
              y,
              yerr )
```

MCMC API for the Transit Model object.

Parameters

self	The object pointer
p	Parameters of the transit
t	Time data
y	Observations data
yerr	Errors data

Returns

Log probability of the chosen parameters

5.13.3.7 meanfnc()

```
def exospec.TransitModel.TransitModel.meanfnc (
              self,
              t )
```

Mean function for the kernel meaan estimation.

Parameters

self	The object pointer
t	The time data

5.13.3.8 model()

```
def exospec.TransitModel.TransitModel.model (
              self )
```

Returns the flux values array.

Parameters

self	The object pointer

Returns

Model-generated observation for the given transit parameters

5.13.3.9 read_errors_data()

```
def exospec.TransitModel.TransitModel.read_errors_data (
              self,
              kwargs )
```

Collects the data about errors that was passed In case some values are missed, the default ones will be taken.

Parameters

self	The object pointer
∗∗*kwargs*	Dictionary of the parameters to pass

5.13.3.10 read_limb_dark_params()

```
def exospec.TransitModel.TransitModel.read_limb_dark_params (
              self,
              kwargs )
```

Forms a list of the limb darkening parameters from the user input In case some values are missed, the default ones will be taken.

Parameters

self	The object pointer
∗∗*kwargs*	Dictionary of the parameters to pass

5.13.3.11 sample_conditional()

```
def exospec.TransitModel.TransitModel.sample_conditional (
              self,
              p,
              t,
              y,
              yerr )
```

For a given set of parameters get predicted y values at t, and separate this into the transit signal component and the noise component.

Parameters

self	The object pointer
p	Parameters of the transit
t	Time data
y	Observations data
yerr	Errors data

Returns

Predicted observations

5.13.3.12 set_values()

```
def exospec.TransitModel.TransitModel.set_values (
              self,
              dict_of_values,
              kwargs )
```

Set the parameters of the model based on the values provided.

In case some values are missed, the default ones will be taken.

Parameters

self	The object pointer
dict_of_values	Dictionary of the parameters to pass

5.13.3.13 update_data()

```
def exospec.TransitModel.TransitModel.update_data (
              self,
              time = None,
              obs = None,
              kwargs )
```

Updates the data for the given parameters.

Parameters

self	The object pointer
time(=None)	Time data
obs(=None)	Observations data
∗∗*kwargs*	Handles arbitrary number of the errors that was passed

5.13.3.14 update_kernel_params()

```
def exospec.TransitModel.TransitModel.update_kernel_params (
              self,
              a_new = None,
              gamma_new = None,
              variance_new = None )
```

Updates the hyperparameters of the kernel function.

Parameters

self	The object pointer
a_new	New value of the kernel_a
gamma_new	New value of the kernel_gamma

5.13.3.15 update_transit_params()

```
def exospec.TransitModel.TransitModel.update_transit_params (
              self,
              rp_new,
              u_new )
```

Updates the parameters of the model.

Parameters

self	The object pointer
rp_new	New value of the rp parameter
u_new	New list of values for the limb darkening

5.13.3.16 updateTransitMode()

```
def exospec.TransitModel.TransitModel.updateTransitMode (
              self )
```

Updates the transit model parameters.

Parameters

self	The object pointer

The documentation for this class was generated from the following file:

- /Users/heatherp/Documents/Courses/Computational/Project/ExoplanetSpectra/exospec/TransitModel.py

Chapter 6

File Documentation

6.1 /Users/heatherp/Documents/Courses/Computational/Project/ExoplanetSpectra/bin/exospec↩ _main.py File Reference

Fits the transits for multiple wavelengths to produce the transmission spectrum.

Variables

- exospec_main.input_file = read_input(sys.argv[1])

  main Reads from the user's input file.
- **exospec_main.input_param_dic** = input_file.param_dic
- **exospec_main.mpi_flag** = input_param_dic['mpi_flag']
- **exospec_main.lc_path** = input_param_dic['lc_path']
- **exospec_main.wave_bin_size** = input_param_dic['wave_bin_size']
- **exospec_main.LC** = exospec.lc_class.LightCurve(lc_path, wave_bin_size)
- **exospec_main.LC_dic** = LC.LC_dic
- **exospec_main.comm** = MPI.COMM_WORLD
- **exospec_main.rank** = comm.Get_rank()
- int **exospec_main.j** = 0
- **exospec_main.jmax** = len(LC_dic) / comm.Get_size()
- **exospec_main.dest**
- **exospec_main.tag**
- **exospec_main.chain**
- **exospec_main.source**
- **exospec_main.i**
- **exospec_main.obj_chain**

6.1.1 Detailed Description

Fits the transits for multiple wavelengths to produce the transmission spectrum.

6.1.2 Variable Documentation

6.1.2.1 chain

```
exospec_main.chain
```

Initial value:

```
=  np.empty(
                             np.shape(LC_dic[LC_dic.keys()[0]].obj_chain))
```

6.1.2.2 input_file

```
exospec_main.input_file = read_input(sys.argv[1])
```

main Reads from the user's input file.

Performs the MCMC fit for each wavelength, saving the chains. Plots and saves the transmission spectrum.

6.2 /Users/heatherp/Documents/Courses/Computational/Project/ExoplanetSpectra/exospec/deliverable File Reference

Deliverables module to parse MCMC data into latex code and plots.

Functions

- def exospec.deliverables.latex_table (LC_dic, transit_params, hyper_params, separate_flag, confidence, filename)

  Writes the best fit transit & hyper parameters from the MCMC fit as a latex table.
- def exospec.deliverables.simple_table (LC_dic, filename)

  Writes the best fit transit parameters from MCMC fit to an output file.
- def exospec.deliverables.plot_single_wavelength (wl_id, mcmc_obj, model, extra_burnin_steps=0, theta_true=None, plot_transit_params=True, plot_hyper_params=True, save_as_dir="")

  Produces triangle, walker and lightcurve plots for the MCMC results for a single wavelength.
- def exospec.deliverables.plot_transmission_spec (LC_dic, save_as_dir="")

  Produces a plot of the best-fit radius as a function of wavelengths.
- def exospec.deliverables.get_median_and_errors (flatchain)

  Obtains the median parameter values and 1 sigma errors from the MCMC flatchain.
- def exospec.deliverables.post_processing_all_wl (input_param_dic, LC_dic)

  Runs the post processing once all wavelength mcmc chains are finished This includes saving a simple table, latex table, and transmission spectrum.

6.2.1 Detailed Description

Deliverables module to parse MCMC data into latex code and plots.

6.2.2 Function Documentation

6.2.2.1 get_median_and_errors()

```
def exospec.deliverables.get_median_and_errors (
              flatchain )
```

Obtains the median parameter values and 1 sigma errors from the MCMC flatchain.

Parameters

flatchain	A 2D numpy array with all the samples for each of the transit and hyper parameters

Returns

median Array of median value for each parameter
err_plus Array of upper error for each parameter
err_minus Array of lower error for each parameter

6.2.2.2 latex_table()

```
def exospec.deliverables.latex_table (
              LC_dic,
              transit_params,
              hyper_params,
              separate_flag,
              confidence,
              filename )
```

Writes the best fit transit & hyper parameters from the MCMC fit as a latex table.

Rows correspond to the light curve with a specified wavelength, and each column corresponds to a given parameter. The use has an option of outputting a single table with transit and hyper parameters, or outputting two separate tables.

Parameters

LC_dic	A light curve dictionary with finished chains stored.
transit_params	List of transit parameter names.
hyper	List of hyper parameter names.
separate_flag	Flag to toggle if combine parameters in a single table or separate them into two tables
confidence	The confidence level desired for the parameter bounds.
filename	The name of the file to write the table in.

6.2.2.3 plot_single_wavelength()

```
def exospec.deliverables.plot_single_wavelength (
              wl_id,
              mcmc_obj,
              model,
              extra_burnin_steps = 0,
              theta_true = None,
              plot_transit_params = True,
              plot_hyper_params = True,
              save_as_dir = "" )
```

Produces triangle, walker and lightcurve plots for the MCMC results for a single wavelength.

Parameters

wl_id	The wavelength being processed
mcmc_obj	An object of the MCMC class that has been run for the wavelength wl_id
model	A function that takes a set of parameters and produces plots of the transit's lightcurve
extra_burnin_steps	Number of steps (in addition to burnin_steps from run) at the start of each chain to neglect
theta_true	Numpy array of true parameter values if known (used for test data)
plot_transit_params	Boolean value specifying whether or not to plot the transit parameters
plot_hyper_params	Boolean value specifying whether or not to plot the hyper parameters
save_as_dir	Directory where plot should be saved. Default is current working Directory

6.2.2.4 plot_transmission_spec()

```
def exospec.deliverables.plot_transmission_spec (
              LC_dic,
              save_as_dir = "" )
```

Produces a plot of the best-fit radius as a function of wavelengths.

Parameters

LC_dic	A lightcurve dictionary with a lightcurve object for each wavelengths
save_as_dir	Directory where plot should be saved. Default is current working Directory

6.2.2.5 post_processing_all_wl()

```
def exospec.deliverables.post_processing_all_wl (
              input_param_dic,
              LC_dic )
```

Runs the post processing once all wavelength mcmc chains are finished This includes saving a simple table, latex table, and transmission spectrum.

Parameters

input_param_dic	
LC_dic	

6.2.2.6 simple_table()

```
def exospec.deliverables.simple_table (
              LC_dic,
              filename )
```

Writes the best fit transit parameters from MCMC fit to an output file.

Output file columns are: wavelength, radius of planet, first limb darkening parameter, second limb darkening parameter, followed by lower bound of corresponding confidence intervals then upper bound of corresponding confidence intervals. Note that this currently only works for the quadratic limb darkening model.

Parameters

LC_dic	A light curve dictionary with finished chains stored.
filename	The name of the file to write the table in.

6.3 /Users/heatherp/Documents/Courses/Computational/Project/ExoplanetSpectra/exospec/fitting↩ _single_lc.py File Reference

Fits a transit lightcurve for a single wavelength.

Functions

- def exospec.fitting_single_lc.run_mcmc_single_wl (input_param_dic, LC_dic, wl_id)

  Definition which carries out the procedure for fitting a single wavelength.

6.3.1 Detailed Description

Fits a transit lightcurve for a single wavelength.

6.3.2 Function Documentation

6.3.2.1 run_mcmc_single_wl()

```
def exospec.fitting_single_lc.run_mcmc_single_wl (
              input_param_dic,
              LC_dic,
              wl_id )
```

Definition which carries out the procedure for fitting a single wavelength.

This follows the steps of: initializing a TransitModel object with correct data for that wavelength, initializing an mcmc object and running it, saving the results of mcmc run, and making walker and triangle plots for that mcmc run. input_param_dic A dictionary of input parameter values LC_dic A light curve dictionary containing the wavelength data wl_id The key for the wavelength to be fit

Return values

0	if successful

Index