



astrocalc Documentation

Release v0.4.2

Dave Young

2023

TABLE OF CONTENTS

1	Features	3
2	How to cite astrocalc	5
2.1	Installation	5
2.1.1	Development	5
2.2	Initialisation	6
2.2.1	Modifying the Settings	6
2.2.2	Basic Python Setup	6
2.3	Todo List	6
2.4	Release Notes	12
3	API Reference	13
3.1	Modules	13
3.1.1	commonutils (<i>module</i>)	13
3.1.2	coords (<i>module</i>)	13
3.1.3	distances (<i>module</i>)	14
3.1.4	times (<i>module</i>)	14
3.1.5	getpackagepath (<i>module</i>)	15
3.2	Classes	15
3.2.1	separations (<i>class</i>)	16
3.2.2	translate (<i>class</i>)	17
3.2.3	unit_conversion (<i>class</i>)	18
3.2.4	converter (<i>class</i>)	22
3.2.5	conversions (<i>class</i>)	25
3.2.6	now (<i>class</i>)	28
3.3	Functions	28
3.3.1	getpackagepath (<i>function</i>)	29
3.3.2	coordinates_to_array (<i>function</i>)	29
3.3.3	luminosity_to_flux (<i>function</i>)	29
3.4	A-Z Index	30
4	Release Notes	31
	Python Module Index	33
	Index	35

DOI 10.5281/zenodo.8014736

downloads 163/month

downloads 163/month

downloads 163/month

downloads 163/month

coverage 78%

docs passing

perform common astronomy-related calculations, conversions and measurements within python scripts or from the command-line.

Documentation for astrocalc is hosted by [Read the Docs](#) ([development version](#) and [master version](#)). The code lives on [github](#). Please report any issues you find [here](#).

FEATURES

-

HOW TO CITE ASTROCALC

If you use `astrocalc` in your work, please cite using the following BibTeX entry:

```
@software{Young_astrocalc,  
author = {Young, David R.},  
doi = {10.5281/zenodo.8014736},  
license = {GPL-3.0-only},  
title = ,  
url = {https://github.com/thespacedoctor/astrocalc}  
}
```

2.1 Installation

The easiest way to install `astrocalc` is to use `pip` (here we show the install inside of a conda environment):

```
conda create -n astrocalc python=3.7 pip  
conda activate astrocalc  
pip install astrocalc
```

Or you can clone the [github repo](#) and install from a local version of the code:

```
git clone git@github.com:thespacedoctor/astrocalc.git  
cd astrocalc  
python setup.py install
```

To upgrade to the latest version of `astrocalc` use the command:

```
pip install astrocalc --upgrade
```

To check installation was successful run `astrocalc -v`. This should return the version number of the install.

2.1.1 Development

If you want to tinker with the code, then install in development mode. This means you can modify the code from your cloned repo:

```
git clone git@github.com:thespacedoctor/astrocalc.git  
cd astrocalc  
python setup.py develop
```

[Pull requests](#) are welcomed!

2.2 Initialisation

Before using astrocalc you need to use the `init` command to generate a user settings file. Running the following creates a `yaml` settings file in your home folder under `~/.config/astrocalc/astrocalc.yaml`:

```
astrocalc init
```

The file is initially populated with astrocalc's default settings which can be adjusted to your preference.

If at any point the user settings file becomes corrupted or you just want to start afresh, simply trash the `astrocalc.yaml` file and rerun `astrocalc init`.

2.2.1 Modifying the Settings

Once created, open the settings file in any text editor and make any modifications needed.

2.2.2 Basic Python Setup

If you plan to use `astrocalc` in your own scripts you will first need to parse your settings file and set up logging etc. One quick way to do this is to use the `fundamentals` package to give you a logger, a settings dictionary and a database connection (if connection details given in settings file):

```
## SOME BASIC SETUP FOR LOGGING, SETTINGS ETC
from fundamentals import tools
from os.path import expanduser
home = expanduser("~")
settingsFile = home + "/.config/astrocalc/astrocalc.yaml"
su = tools(
    arguments={"settingsFile": settingsFile},
    docString=__doc__,
)
arguments, settings, log, dbConn = su.setup()
```

2.3 Todo List

Todo: add usage info create a sublime snippet for usage

(The *original entry* is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/coords/coordinates_to_array.py:docstring` of `astrocalc.coords.coordinates_to_array.coordinates_to_array`, line 19.)

Todo:

- replace `get_angular_separation` throughout all code using `dryxPython`
 - replace `getAngularSeparation` throughout all code using `dryxPython`
-

(The *original entry* is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/coords/separations.py:docstring` of `astrocalc.coords.separations.separations`, line 16.)

Todo:

- replace `shift_coordinates` class in all other code
-

(The *original entry* is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/coords/translate.py:docstring` of `astrocalc.coords.translate.translate`, line 14.)

Todo:

- add usage info
 - create a sublime snippet for usage
 - add `ra_sexagesimal_to_decimal`
-

(The *original entry* is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/coords/unit_conversion.py:docstring` of `astrocalc.coords.unit_conversion.unit_conversion`, line 12.)

Todo:

- replace `dryxPython declination_sexagesimal_to_decimal` with this version in all my code
 - replace `coords_sex_to_dec` in all code
-

(The *original entry* is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/coords/unit_conversion.py:docstring` of `astrocalc.coords.unit_conversion.unit_conversion.dec_sexagesimal_to_decimal`, line 26.)

Todo:

- replace `ra_to_sex` from `dryxPython` in all code
-

(The *original entry* is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/coords/unit_conversion.py:docstring` of `astrocalc.coords.unit_conversion.unit_conversion.ra_decimal_to_sexagesimal`, line 19.)

Todo:

- replace `dec_to_sex` in `dryxPython` in all code
-

(The *original entry* is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/coords/unit_conversion.py:docstring` of `astrocalc.coords.unit_conversion.unit_conversion.dec_decimal_to_sexagesimal`, line 19.)

Todo:

- replace `calculate_cartesians` in all code
-

(The *original entry* is located in /home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/coords/unit_conversion.py:docstring of astrocalc.coords.unit_conversion.unit_conversion.ra_dec_to_cartesian, line 15.)

Todo:

- replace convert_mpc_to_redshift in all code

(The *original entry* is located in /home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/distances/converter.py:docstring of astrocalc.distances.converter.converter.distance_to_redshift, line 16.)

Todo:

- replace convert_redshift_to_distance in all other code

(The *original entry* is located in /home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/distances/converter.py:docstring of astrocalc.distances.converter.converter.redshift_to_distance, line 22.)

Todo:

- update key arguments values and definitions with defaults
- update return values and definitions
- update usage examples and text
- update docstring text
- check sublime snippet exists
- clip any useful text to docs mindmap
- regenerate the docs and check rendering of this docstring

(The *original entry* is located in /home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/docstring_test.py:docstring of astrocalc.docstring_test.docstring_test, line 38.)

Todo:

- nice!

(The *original entry* is located in /home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/docstring_test.py:docstring of astrocalc.docstring_test.docstring_test, line 52.)

Todo:

- add usage info
- create a sublime snippet for usage
- add mjd_to_date

- add `decimal_day_to_day_hour_min_sec`
 - add `date_to_mjd`
 - convert all functions in `__init__` to modules
-

(The *original entry* is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/times/conversions.py:docstring` of `astrocalc.times.conversions.conversions`, line 12.)

Todo:

- replace `getMJDFromSqlDate` in all code
-

(The *original entry* is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/times/conversions.py:docstring` of `astrocalc.times.conversions.conversions.ut_datetime_to_mjd`, line 18.)

Todo:

- replace `getDateFromMJD` in all code
 - replace `getSqlDateFromMJD` in all code
-

(The *original entry* is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/times/conversions.py:docstring` of `astrocalc.times.conversions.conversions.mjd_to_ut_datetime`, line 13.)

Todo:

- replace `decimal_day_to_day_hour_min_sec` in all other code
-

(The *original entry* is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/times/conversions.py:docstring` of `astrocalc.times.conversions.conversions.decimal_day_to_day_hour_min_sec`, line 21.)

Todo:

- add `clutil`
 - remove `getCurrentMJD` from all other code
-

(The *original entry* is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/times/now.py:docstring` of `astrocalc.times.now.now.get_mjd`, line 11.)

Todo: add usage info create a sublime snippet for usage

(The *original entry* is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/coords/coordinates_to_array.py:docstring` of `astrocalc.coords.coordinates_to_array.coordinates_to_array`, line 19.)

Todo:

- replace `get_angular_separation` throughout all code using `dryxPython`
 - replace `getAngularSeparation` throughout all code using `dryxPython`
-

(The [original entry](#) is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/coords/separations.py:docstring` of `astrocalc.coords.separations.separations`, line 16.)

Todo:

- replace `shift_coordinates` class in all other code
-

(The [original entry](#) is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/coords/translate.py:docstring` of `astrocalc.coords.translate.translate`, line 14.)

Todo:

- add usage info
 - create a sublime snippet for usage
 - add `ra_sexagesimal_to_decimal`
-

(The [original entry](#) is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/coords/unit_conversion.py:docstring` of `astrocalc.coords.unit_conversion.unit_conversion`, line 12.)

Todo:

- replace `dec_to_sex` in `dryxPython` in all code
-

(The [original entry](#) is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/coords/unit_conversion.py:docstring` of `astrocalc.coords.unit_conversion.unit_conversion.dec_decimal_to_sexagesimal`, line 19.)

Todo:

- replace `dryxPython` `declination_sexagesimal_to_decimal` with this version in all my code
 - replace `coords_sex_to_dec` in all code
-

(The [original entry](#) is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/coords/unit_conversion.py:docstring` of `astrocalc.coords.unit_conversion.unit_conversion.dec_sexagesimal_to_decimal`, line 26.)

Todo:

- replace `calculate_cartesians` in all code
-

(The [original entry](#) is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/coords/unit_conversion.py:docstring` of `astrocalc.coords.unit_conversion.unit_conversion.ra_dec_to_cartesian`, line 15.)

Todo:

- replace `ra_to_sex` from `dryxPython` in all code
-

(The [original entry](#) is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/coords/unit_conversion.py:docstring` of `astrocalc.coords.unit_conversion.unit_conversion.ra_decimal_to_sexagesimal`, line 19.)

Todo:

- replace `convert_mpc_to_redshift` in all code
-

(The [original entry](#) is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/distances/converter.py:docstring` of `astrocalc.distances.converter.converter.distance_to_redshift`, line 16.)

Todo:

- replace `convert_redshift_to_distance` in all other code
-

(The [original entry](#) is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/distances/converter.py:docstring` of `astrocalc.distances.converter.converter.redshift_to_distance`, line 22.)

Todo:

- add usage info
 - create a sublime snippet for usage
 - add `mjd_to_date`
 - add `decimal_day_to_day_hour_min_sec`
 - add `date_to_mjd`
 - convert all functions in `__init__` to modules
-

(The [original entry](#) is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/times/conversions.py:docstring` of `astrocalc.times.conversions.conversions`, line 12.)

Todo:

- replace `decimal_day_to_day_hour_min_sec` in all other code
-

(The [original entry](#) is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/times/conversions.py:docstring` of `astrocalc.times.conversions.conversions.decimal_day_to_day_hour_min_sec`, line 21.)

Todo:

- replace `getDateFromMJD` in all code
-

- replace `getSQLDateFromMJD` in all code
-

(The [original entry](#) is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/times/conversions.py:docstring` of `astrocalc.times.conversions.conversions.mjd_to_ut_datetime`, line 13.)

Todo:

- replace `getMJDFromSqlDate` in all code
-

(The [original entry](#) is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/times/conversions.py:docstring` of `astrocalc.times.conversions.conversions.ut_datetime_to_mjd`, line 18.)

Todo:

- add `clutil`
 - remove `getCurrentMJD` from all other code
-

(The [original entry](#) is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/envs/develop/lib/python3.7/site-packages/astrocalc-0.4.2-py3.7.egg/astrocalc/times/now.py:docstring` of `astrocalc.times.now.now.get_mjd`, line 11.)

Todo:

- nice!
-

(The *original entry* is located in `/home/docs/checkouts/readthedocs.org/user_builds/astrocalc/checkouts/develop/docs/source/_template_` line 1.)

2.4 Release Notes

v0.4.1 - May 2, 2020

- Code now compliant with Python 3.7

Release Notes

API REFERENCE

3.1 Modules

<code>astrocalc.commonutils</code>	<i>common tools used throughout package</i>
<code>astrocalc.coords</code>	<i>Calculations and conversions based on sky-coordinates</i>
<code>astrocalc.distances</code>	<i>Tools to work with and manipulate astronomical distances</i>
<code>astrocalc.times</code>	<i>Calculations and conversions based on astronomical times</i>
<code>astrocalc.commonutils.getpackagepath</code>	<i>Get common file and folder paths for the host package</i>

3.1.1 commonutils (module)

common tools used throughout package

Sub-modules

<code>getpackagepath</code>	<i>Get common file and folder paths for the host package</i>
-----------------------------	--

3.1.2 coords (module)

Calculations and conversions based on sky-coordinates

Classes

<code>separations(log, ra1, dec1, ra2, dec2[, ...])</code>	<i>The worker class for the separations module</i>
<code>translate(log, ra, dec, northArcsec, eastArcsec)</code>	<i>Translate a set of coordinates north and east by distances given in arcsecs</i>
<code>unit_conversion(log[, settings])</code>	<i>The worker class for the unit_conversion module</i>

Functions

<code>coordinates_to_array(log, ra, dec)</code>	<i>Convert a single value RA, DEC or list of RA and DEC to numpy arrays</i>
---	---

Sub-modules

<code>coordinates_to_array(log, ra, dec)</code>	<i>Convert a single value RA, DEC or list of RA and DEC to numpy arrays</i>
<code>separations(log, ra1, dec1, ra2, dec2[, ...])</code>	<i>The worker class for the separations module</i>
<code>translate(log, ra, dec, northArcsec, eastArcsec)</code>	<i>Translate a set of coordinates north and east by distances given in arcsecs</i>
<code>unit_conversion(log[, settings])</code>	<i>The worker class for the unit_conversion module</i>

3.1.3 distances (module)

Tools to work with and manipulate astronomical distances

Classes

<code>converter(log[, settings])</code>	<i>A converter to switch distance between various units of measurement</i>
---	--

Sub-modules

<code>converter(log[, settings])</code>	<i>A converter to switch distance between various units of measurement</i>
---	--

3.1.4 times (module)

Calculations and conversions based on astronomical times

Classes

<code>conversions(log[, settings])</code>	<i>The worker class for the conversions module</i>
<code>now(log[, settings])</code>	<i>Report the current time into various formats</i>

Sub-modules

<code>conversions(log[, settings])</code>	<i>The worker class for the conversions module</i>
<code>now(log[, settings])</code>	<i>Report the current time into various formats</i>

3.1.5 getpackagepath (module)

Get common file and folder paths for the host package

Author David Young

Functions

<code>getpackagepath()</code>	<i>Get the root path for this python package</i>
-------------------------------	--

Sub-modules

<code>getpackagepath()</code>	<i>Get the root path for this python package</i>
<code>os</code>	<i>OS routines for NT or Posix depending on what system we're on.</i>

`getpackagepath()`

Get the root path for this python package

Used in unit testing code

3.2 Classes

<code>astrocalc.coords.separations</code>	<i>The worker class for the separations module</i>
<code>astrocalc.coords.translate</code>	<i>Translate a set of coordinates north and east by distances given in arcsecs</i>
<code>astrocalc.coords.unit_conversion</code>	<i>The worker class for the unit_conversion module</i>
<code>astrocalc.distances.converter</code>	<i>A converter to switch distance between various units of measurement</i>
<code>astrocalc.times.conversions</code>	<i>The worker class for the conversions module</i>
<code>astrocalc.times.now</code>	<i>Report the current time into various formats</i>

3.2.1 separations (class)

class `separations` (*log, ra1, dec1, ra2, dec2, settings=False*)

Bases: `object`

The worker class for the separations module

Key Arguments

- `log` – logger
- `settings` – the settings dictionary
- `ra1` – the right-ascension of the first location. Decimal degrees or sexagesimal.
- `dec1` – the declination of the first location. Decimal degrees or sexagesimal.
- `ra2` – the right-ascension of the second location. Decimal degrees or sexagesimal.
- `dec2` – the declination of the second location. Decimal degrees or sexagesimal.

Usage

Todo:

- replace `get_angular_separation` throughout all code using `dryxPython`
 - replace `getAngularSeparation` throughout all code using `dryxPython`
-

You can input sexagesimal coordinates,

```
from astrocalc.coords import separations
calculator = separations(
    log=log,
    ra1="23:32:23.2324",
    dec1="-13:32:45.43553",
    ra2="23:32:34.642",
    dec2="-12:12:34.9334",
)
angularSeparation, north, east = calculator.get()
print(angularSeparation, north, east)

# OUT: '4813.39431', '4810.50214', '166.83941'
```

or decimal degrees,

```
from astrocalc.coords import separations
calculator = separations(
    log=log,
    ra1=2.3342343,
    dec1=89.23244233,
    ra2=45.343545345,
    dec2=87.3435435
)
angularSeparation, north, east = calculator.get()
print(angularSeparation, north, east)

# OUT: '7774.4375', '-6800.0358', '4625.7620'
```

or even a mix of both

```

from astrocalc.coords import separations
calculator = separations(
    log=log,
    ra1=352.5342343,
    dec1=89.23,
    ra2="23:32:34.642",
    dec2="89:12:34.9334"
)
angularSeparation, north, east = calculator.get()
print(angularSeparation, north, east)

# OUT: '78.9', '-73.1', '29.9')

```

Methods

get()	<i>Calculate the angular separation between two locations on the sky</i>
-------	--

get ()
Calculate the angular separation between two locations on the sky
 Input precision should be respected.

Key Arguments

Return

- angularSeparation – total angular separation between coordinates (arcsec)
- north – north-south separation between coordinates (arcsec)
- east – east-west separation between coordinates (arcsec)

See main class usage for details.

3.2.2 translate (class)

class translate (log, ra, dec, northArcsec, eastArcsec, settings=False)

Bases: object

Translate a set of coordinates north and east by distances given in arcsecs

Key Arguments

- log – logger
- settings – the settings dictionary. Default *False*
- ra – ra (decimal or sexagesimal)
- dec – dec (decimal or sexagesimal)
- northArcsec – number of arcsecs to move location to the north
- eastArcsec – number of arcsecs to move location to the east

Todo:

- replace shift_coordinates class in all other code

Usage

To shift a set of coordinates north and east by given distances:

```
# TRANSLATE COORDINATES ACROSS SKY
from astrocalc.coords import translate
ra, dec = translate(
    log=log,
    settings=settings,
    ra="14.546438",
    dec="-45.34232334",
    northArcsec=4560,
    eastArcsec=+967800
).get()
```

Methods

get()	<i>translate the coordinates</i>
-------	----------------------------------

get()
translate the coordinates

Return

- ra – the right-ascension of the translated coordinate
- dec – the declination of the translated coordinate

3.2.3 unit_conversion (class)

class unit_conversion (log, settings=False)

Bases: object

The worker class for the unit_conversion module

Key Arguments

- log – logger
- settings – the settings dictionary (prob not required)

Usage

Todo:

- add usage info
 - create a sublime snippet for usage
 - add ra_sexagesimal_to_decimal
-

usage code

- @review: when complete, clean unit_conversion class
- @review: when complete add logging
- @review: when complete, decide whether to abstract class to another module

Methods

<code>dec_decimal_to_sexagesimal(dec[, delimiter])</code>	<i>Convert a declination between decimal degrees and sexagesimal.</i>
<code>dec_sexagesimal_to_decimal(dec)</code>	<i>Convert a declination from sexagesimal format to decimal degrees.</i>
<code>get()</code>	<i>get the unit_conversion object</i>
<code>ra_dec_to_cartesian(ra, dec)</code>	<i>Convert an RA, DEC coordinate set to x, y, z cartesian coordinates</i>
<code>ra_decimal_to_sexagesimal(ra[, delimiter])</code>	<i>Convert a right-ascension between decimal degrees and sexagesimal.</i>
<code>ra_sexagesimal_to_decimal(ra)</code>	<i>Convert a right-ascension from sexagesimal format to decimal degrees.</i>

`dec_decimal_to_sexagesimal(dec, delimiter=':')`

Convert a declination between decimal degrees and sexagesimal.

Precision should be respected.

Key Arguments

- `dec` – DEC in decimal degrees. Will try and convert to float before performing calculation.
- `delimiter` – how to delimit the RA units. Default :

Return

- `sexagesimal` – ra in sexagesimal units

Usage

Todo:

- replace `dec_to_sex` in `dryxPython` in all code

```
from astrocalc.coords import unit_conversion
converter = unit_conversion(
    log=log
)
dec = converter.dec_decimal_to_sexagesimal(
    dec="-3.454676456",
    delimiter=":"
)
print(dec)

# OUT: -03:27:16.8
```

`dec_sexagesimal_to_decimal(dec)`

Convert a declination from sexagesimal format to decimal degrees.

Precision should be respected. If a float is passed to this method, the same float will be returned (useful if unclear which format coordinates are in).

The code will attempt to read the sexagesimal value in whatever form it is passed. Any of the following should be handled correctly:

- +1:58:05.45341
- 01:5:05
- +1 58 05.45341
- -23h53m05s

Key Arguments

- dec - DEC in sexagesimal format.

Return

- decDeg – declination converted to decimal degrees

Usage

Todo:

- replace dryxPython declination_sexagesimal_to_decimal with this version in all my code
 - replace coords_sex_to_dec in all code
-

```
from astrocalc.coords import unit_conversion
converter = unit_conversion(
    log=log
)
dec = converter.dec_sexagesimal_to_decimal(
    dec="-23:45:21.23232"
)
print(dec)

# OUTPUT: -23.7558978667
```

get()

get the unit_conversion object

Return

- unit_conversion

```
- @review: when complete, clean get method
- @review: when complete add logging
```

ra_dec_to_cartesian(ra, dec)

Convert an RA, DEC coordinate set to x, y, z cartesian coordinates

Key Arguments

- ra – right ascension in sexagesimal or decimal degrees.
- dec – declination in sexagesimal or decimal degrees.

Return

- cartesians – tuple of (x, y, z) coordinates

Todo:

- replace calculate_cartesians in all code
-

Usage

```
from astrocalc.coords import unit_conversion
converter = unit_conversion(
    log=log
)
x, y, z = converter.ra_dec_to_cartesian(
    ra="23 45 21.23232",
    dec="+01:58:5.45341"
)
print(x, y, z)

# OUTPUT: 0.9973699780687104, -0.06382462462791459, 0.034344492110465606
```

ra_decimal_to_sexagesimal (ra, delimiter=':')

Convert a right-ascension between decimal degrees and sexagesimal.

Precision should be respected.

Key Arguments

- ra – RA in decimal degrees. Will try and convert to float before performing calculation.
- delimiter – how to delimit the RA units. Default :

Return

- sexagesimal – ra in sexagesimal units

Usage

Todo:

- replace ra_to_sex from dryxPython in all code
-

```
from astrocalc.coords import unit_conversion
converter = unit_conversion(
    log=log
)
ra = converter.ra_decimal_to_sexagesimal(
    ra="-23.454676456",
    delimiter=":"
)
print(ra)

# OUT: 22:26:10.87
```

ra_sexagesimal_to_decimal (ra)

Convert a right-ascension from sexagesimal format to decimal degrees.

Precision should be respected. If a float is passed to this method, the same float will be returned (useful if unclear which format coordinates are in).

The code will attempt to read the sexagesimal value in whatever form it is passed. Any of the following should be handled correctly

- 23:45:21.23232
- 23h45m21.23232s
- 23 45 21.23232
- 2 04 21.23232
- 04:45 21

Key Arguments

- ra – ra in sexagesimal units

Return

- decimalDegrees

Usage

```
- replace dryxPython ra_sexagesimal_to_decimal with this version in all my_
↪code

from astrocalc.coords import unit_conversion
converter = unit_conversion(
    log=log
)
ra = converter.ra_sexagesimal_to_decimal(
    ra="04:45 21"
)
print(ra)

# OUTPUT: 71.3375
```

3.2.4 converter (class)

class converter (log, settings=False)

Bases: object

A converter to switch distance between various units of measurement

Key Arguments

- log – logger
- settings – the settings dictionary

Usage

To instantiate a converter object:

```
from astrocalc.distances import converter
c = converter(log=log)
```

Methods

<code>distance_to_redshift(mpc)</code>	<i>Convert a distance from MPC to redshift</i>
<code>redshift_to_distance(z[, WM, WV, H0])</code>	<i>convert redshift to various distance measurements</i>

distance_to_redshift (*mpc*)

Convert a distance from MPC to redshift

The code works by iteratively converting a redshift to a distance, correcting itself and honing in on the true answer (within a certain precision)

Key Arguments

- `mpc` – distance in MPC (assumes a luminosity distance).

Return

- `redshift`

Todo:

- replace `convert_mpc_to_redshift` in all code
-

Usage

```
from astrocalc.distances import converter
c = converter(log=log)
z = c.distance_to_redshift(
    mpc=500
)

print(z)

# OUTPUT: 0.108
```

redshift_to_distance (*z, WM=0.3, WV=0.7, H0=70.0*)

convert redshift to various distance measurements

Key Arguments

- `z` – redshift measurement.
- `WM` – Omega_matter. Default *0.3*
- `WV` – Omega_vacuum. Default *0.7*
- `H0` – Hubble constant. (km s⁻¹ Mpc⁻¹) Default *70.0*

Return

- **results** – result dictionary including
 - `dcmr_mpc` – co-moving radius distance
 - `da_mpc` – angular distance
 - `da_scale` – angular distance scale
 - `dl_mpc` – luminosity distance (usually use this one)
 - `dmod` – distance modulus (determined from luminosity distance)

Todo:

- replace convert_redshift_to_distance in all other code
-

Usage

```
from astrocalc.distances import converter
c = converter(log=log)
dists = c.redshift_to_distance(
    z=0.343
)

print("Distance Modulus: " + str(dists["dmod"]) + " mag")
print("Luminosity Distance: " + str(dists["dl_mpc"]) + " Mpc")
print("Angular Size Scale: " + str(dists["da_scale"]) + " kpc/arcsec")
print("Angular Size Distance: " + str(dists["da_mpc"]) + " Mpc")
print("Comoving Radial Distance: " + str(dists["dcmr_mpc"]) + " Mpc")

# OUTPUT :
# Distance Modulus: 41.27 mag
# Luminosity Distance: 1795.16 Mpc
# Angular Size Scale: 4.85 kpc/arcsec
# Angular Size Distance: 999.76 Mpc
# Comoving Radial Distance: 1339.68 Mpc

from astrocalc.distances import converter
c = converter(log=log)
dists = c.redshift_to_distance(
    z=0.343,
    WM=0.286,
    WV=0.714,
    H0=69.6
)

print("Distance Modulus: " + str(dists["dmod"]) + " mag")
print("Luminosity Distance: " + str(dists["dl_mpc"]) + " Mpc")
print("Angular Size Scale: " + str(dists["da_scale"]) + " kpc/arcsec")
print("Angular Size Distance: " + str(dists["da_mpc"]) + " Mpc")
print("Comoving Radial Distance: " + str(dists["dcmr_mpc"]) + " Mpc")

# OUTPUT :
# Distance Modulus: 41.29 mag
# Luminosity Distance: 1811.71 Mpc
# Angular Size Scale: 4.89 kpc/arcsec
# Angular Size Distance: 1008.97 Mpc
# Comoving Radial Distance: 1352.03 Mpc
```

3.2.5 conversions (class)

class conversions (*log*, *settings=False*)

Bases: object

The worker class for the conversions module

Key Arguments

- *log* – logger
- *settings* – the settings dictionary

Usage

Todo:

- add usage info
 - create a sublime snippet for usage
 - add `mjd_to_date`
 - add `decimal_day_to_day_hour_min_sec`
 - add `date_to_mjd`
 - convert all functions in `__init__` to modules
-

usage code

Methods

<code>decimal_day_to_day_hour_min_sec(daysFloat)</code>	<i>Convert a day from decimal format to hours mins and sec</i>
<code>get()</code>	<i>get the conversions object</i>
<code>mjd_to_ut_datetime(mjd[, sqlDate, ...])</code>	<i>mjd to ut datetime</i>
<code>ut_datetime_to_mjd(utDatetime)</code>	<i>ut datetime to mjd</i>

decimal_day_to_day_hour_min_sec (*daysFloat*)

Convert a day from decimal format to hours mins and sec

Precision should be respected.

Key Arguments

- *daysFloat* – the day as a decimal.

Return

- *daysInt* – day as an integer
- *hoursInt* – hour as an integer (None if input precision too low)
- *minsInt* – mins as an integer (None if input precision too low)
- *secFloat* – secs as a float (None if input precision too low)

Usage

Todo:

- replace *decimal_day_to_day_hour_min_sec* in all other code
-

```
from astrocalc.times import conversions
converter = conversions(
    log=log
)
daysInt, hoursInt, minsInt, secFloat = converter.decimal_day_to_day_hour_min_
↪sec(
    daysFloat=24.2453
)
print(daysInt, hoursInt, minsInt, secFloat)

# OUTPUT: 24, 5, 53, None

daysInt, hoursInt, minsInt, secFloat = converter.decimal_day_to_day_hour_min_
↪sec(
    daysFloat=24.1232435454
)
print("%(daysInt)s days, %(hoursInt)s hours, %(minsInt)s mins, %(secFloat)s_
↪sec" % locals())

# OUTPUT: 24 days, 2 hours, 57 mins, 28.242 sec
```

get()

get the conversions object

Return

- conversions

```
- @review: when complete, clean get method
- @review: when complete add logging
```

mjd_to_ut_datetime (*mjd*, *sqlDate=False*, *datetimeObject=False*)

mjd to ut datetime

Precision should be respected.

Key Arguments

- *mjd* – time in MJD.
- *sqlDate* – add a ‘T’ between date and time instead of space
- *datetimeObject* – return a datetime object instead of a string. Default *False*

Todo:

- replace *getDateFromMJD* in all code
 - replace *getSQLDateFromMJD* in all code
-

Return

- *utDatetime* - the UT datetime in string format

Usage

```

from astrocalc.times import conversions
converter = conversions(
    log=log
)
utDate = converter.mjd_to_ut_datetime(
    mjd=57504.61577585013
)
print(utDate)

# OUT: 2016-04-26 14:46:43.033

utDate = converter.mjd_to_ut_datetime(
    mjd=57504.61577585013,
    sqlDate=True
)
print(utDate)

# OUT: 2016-04-26T14:46:43.033

```

ut_datetime_to_mjd (*utDatetime*)
ut datetime to mjd

If the date given has no time associated with it (e.g. 20160426), then the datetime assumed is 20160426 00:00:00.0.

Precision should be respected.

Key Arguments

- *utDatetime* – UT datetime. Can accept various formats e.g. 201604261444, 20160426, 20160426144444.5452, 2016-04-26 14:44:44.234, 20160426 14h44m44.432s

Return

- *mjd* – the MJD

Todo:

- replace getMJDFromSqlDate in all code
-

Usage

```

from astrocalc.times import conversions
converter = conversions(
    log=log
)
mjd = converter.ut_datetime_to_mjd(utDatetime="20160426t1446")
print(mjd)

# OUT: 57504.6153

mjd = converter.ut_datetime_to_mjd(utDatetime="2016-04-26 14:44:44.234")
print(mjd)

# OUT: 57504.61440

```

3.2.6 now (class)

class `now` (*log*, *settings=False*)

Bases: `object`

Report the current time into various formats

Key Arguments

- `log` – logger
- `settings` – the settings dictionary

Methods

<code>get_mjd()</code>	<i>Get the current time as an MJD</i>
------------------------	---------------------------------------

get_mjd()

Get the current time as an MJD

Return

- `mjd` – the current MJD as a float

Usage

Todo:

- add `clutil`
 - remove `getCurrentMJD` from all other code
-

```
from astrocalc.times import now
mjd = now(
    log=log
).get_mjd()
```

3.3 Functions

<code>astrocalc.commonutils.getpackagepath. getpackagepath</code>	<i>Get the root path for this python package</i>
---	--

<code>astrocalc.coords. coordinates_to_array</code>	<i>Convert a single value RA, DEC or list of RA and DEC to numpy arrays</i>
---	---

<code>astrocalc.luminosity_to_flux</code>	<i>Convert luminosity to a flux</i>
---	-------------------------------------

3.3.1 getpackagepath (function)

getpackagepath()

Get the root path for this python package

Used in unit testing code

3.3.2 coordinates_to_array (function)

coordinates_to_array(log, ra, dec)

Convert a single value RA, DEC or list of RA and DEC to numpy arrays

Key Arguments

- ra – list, numpy array or single ra value
- dec – list, numpy array or single dec value
- log – logger

Return

- raArray – input RAs as a numpy array of decimal degree values
- decArray – input DECs as a numpy array of decimal degree values

Usage

Todo: add usage info create a sublime snippet for usage

```
ra, dec = coordinates_to_array(  
    log=log,  
    ra=ra,  
    dec=dec  
)
```

3.3.3 luminosity_to_flux (function)

luminosity_to_flux(lumErg_S, dist_Mpc)

Convert luminosity to a flux

Key Arguments

- lumErg_S – luminosity in ergs/sec
- dist_Mpc – distance in Mpc

Return

- fluxErg_cm2_S – flux in ergs/cm2/s

3.4 A-Z Index

Modules

<code>astrocalc.commonutils</code>	<i>common tools used throughout package</i>
<code>astrocalc.coords</code>	<i>Calculations and conversions based on sky-coordinates</i>
<code>astrocalc.distances</code>	<i>Tools to work with and manipulate astronomical distances</i>
<code>astrocalc.times</code>	<i>Calculations and conversions based on astronomical times</i>
<code>astrocalc.commonutils.getpackagepath</code>	<i>Get common file and folder paths for the host package</i>

Classes

<code>astrocalc.coords.separations</code>	<i>The worker class for the separations module</i>
<code>astrocalc.coords.translate</code>	<i>Translate a set of coordinates north and east by distances given in arcsecs</i>
<code>astrocalc.coords.unit_conversion</code>	<i>The worker class for the unit_conversion module</i>
<code>astrocalc.distances.converter</code>	<i>A converter to switch distance between various units of measurement</i>
<code>astrocalc.times.conversions</code>	<i>The worker class for the conversions module</i>
<code>astrocalc.times.now</code>	<i>Report the current time into various formats</i>

Functions

<code>astrocalc.commonutils.getpackagepath.getpackagepath</code>	<i>Get the root path for this python package</i>
<code>astrocalc.coords.coordinates_to_array</code>	<i>Convert a single value RA, DEC or list of RA and DEC to numpy arrays</i>
<code>astrocalc.luminosity_to_flux</code>	<i>Convert luminosity to a flux</i>

RELEASE NOTES

v0.4.1 - May 2, 2020

- Code now compliant with Python 3.7

Release Notes

PYTHON MODULE INDEX

c

`astrocalc.commonutils`, [13](#)
`astrocalc.commonutils.getpackagepath`,
 [15](#)
`astrocalc.coords`, [13](#)

d

`astrocalc.distances`, [14](#)

t

`astrocalc.times`, [14](#)

INDEX

A

`astrocalc.commonutils`
 module, 13
`astrocalc.commonutils.getpackagepath`
 module, 15
`astrocalc.coords`
 module, 13
`astrocalc.distances`
 module, 14
`astrocalc.times`
 module, 14

C

`conversions` (*class in astrocalc.times*), 25
`converter` (*class in astrocalc.distances*), 22
`coordinates_to_array()` (*in module astrocalc.coords*), 29

D

`dec_decimal_to_sexagesimal()`
 (*unit_conversion method*), 19
`dec_sexagesimal_to_decimal()`
 (*unit_conversion method*), 19
`decimal_day_to_day_hour_min_sec()` (*conversions method*), 25
`distance_to_redshift()` (*converter method*), 23

G

`get()` (*conversions method*), 26
`get()` (*separations method*), 17
`get()` (*translate method*), 18
`get()` (*unit_conversion method*), 20
`get_mjd()` (*now method*), 28
`getpackagepath()` (*in module astrocalc.commonutils.getpackagepath*), 15, 29

L

`luminosity_to_flux()` (*in module astrocalc*), 29

M

`mjd_to_ut_datetime()` (*conversions method*), 26

module

`astrocalc.commonutils`, 13
 `astrocalc.commonutils.getpackagepath`, 15
 `astrocalc.coords`, 13
 `astrocalc.distances`, 14
 `astrocalc.times`, 14

N

`now` (*class in astrocalc.times*), 28

R

`ra_dec_to_cartesian()` (*unit_conversion method*), 20
`ra_decimal_to_sexagesimal()`
 (*unit_conversion method*), 21
`ra_sexagesimal_to_decimal()`
 (*unit_conversion method*), 21
`redshift_to_distance()` (*converter method*), 23

S

`separations` (*class in astrocalc.coords*), 16

T

`translate` (*class in astrocalc.coords*), 17

U

`unit_conversion` (*class in astrocalc.coords*), 18
`ut_datetime_to_mjd()` (*conversions method*), 27