astrotool

Marc van der Sluys

Jul 13, 2021

CONTENTS:

CHAPTER

ONE

ASTROTOOL PACKAGE

1.1 Submodules

1.1.1 astrotool.constants module

Definition of constants for AstroTool.

astrotool.constants.**AU = 149597870.7**
Astronomical unit in km

astrotool.constants.**as2r = 4.84813681109536e-06**
Arcseconds to radians

astrotool.constants.**d2r = 0.017453292519943295**
Degrees to radians

astrotool.constants.**earth_rad = 6378.1366**
Earth radius in km

astrotool.constants.**h2r = 0.2617993877991494**
Hours to radians

astrotool.constants.**jd1820 = 2385801**
JD in 1820 (for Delta-T fit)

astrotool.constants.**jd1900 = 2415021**
JD in 1900

astrotool.constants.**jd2000 = 2451545**
JD in 2000.0

astrotool.constants.**mas2r = 4.84813681109536e-09**
Milliarcseconds to radians

astrotool.constants.**moon_rad = 1737.5**
Moon radius in km

astrotool.constants.**pi = 3.141592653589793**
pi

astrotool.constants.**pi2 = 6.283185307179586**
2 pi

astrotool.constants.**pio2 = 1.5707963267948966**
pi/2

astrotool.constants.**r2d = 57.29577951308232**
Radians to degrees

1.1.2 astrotool.coordinates module

Coordinate transformations and related functions for AstroTool.

astrotool.coordinates.**ecl2eq**(*lon*, *lat*, *eps*)
Convert (geocentric) spherical ecliptical coordinates to spherical equatorial coordinates.

Parameters

- **lon** (*double*) – Ecliptical longitude (rad).
- **lat** (*double*) – Ecliptical latitude (rad).
- **eps** (*double*) – Obliquity of the ecliptic (rad).

Returns

tuple containing (ra, dec):

- ra (double): Right ascension (rad).
- dec (double): Declination (rad).

Return type tuple (double,double)

References

- Explanatory Supplement to the Astronomical Almanac 3rd Ed, Eq.14.43

astrotool.coordinates.**eq2ecl**(*ra*, *dec*, *eps*)
Convert equatorial coordinates to ecliptical.

Parameters

- **ra** (*double*) – Right ascension (rad).
- **dec** (*double*) – Declination (rad).
- **eps** (*double*) – Obliquity of the ecliptic (rad).

Returns

tuple containing (lon, lat):

- lon (double): Ecliptical longitude (rad).
- lat (double): Ecliptical latitude (rad).

Return type tuple (double,double)

astrotool.coordinates.**geoc2topoc_ecl**(*lon_gc*, *lat_gc*, *dist_gc*, *rad_gc*, *eps*, *lst*, *lat_obs*, *ele_obs=0*, *debug=False*)
Convert spherical ecliptical coordinates from the geocentric to the topocentric system.

Parameters

- **lon_gc** (*double*) – Geocentric ecliptic longitude (rad).

- **lat_gc** (*double*) – Geocentric ecliptic latitude (rad).
- **dist_gc** (*double*) – Geocentric distance (AU).
- **rad_gc** (*double*) – Geocentric semi-diameter (rad).
- **eps** (*double*) – Obliquity of the ecliptic (rad).
- **lst** (*double*) – Local sidereal time (rad).
- **lat_obs** (*double*) – Geographical latitude of the observer (rad).
- **ele_obs** (*double*) – Altitude/elevation of the observer above sea level (metres, optional, default value = 0).
- **debug** (*bool*) – Print debug output (True/False, optional, default value = True).

Returns

tuple containing (lon_tc, lat_tc, rad_tc):

- lon_tc (double): Topocentric ecliptic longitude (rad).
- lat_tc (double): Topocentric ecliptic latitude (rad).
- rad_tc (double): Topocentric semi-diameter (rad).

Return type tuple (double,double,double)

`astrotool.coordinates.`**`obliquity`**`(`*`jd`*`)`

Compute the obliquity of the ecliptic in radians from the JD(E).

Parameters **jd** (*double*) – Julian day (days).

Returns eps: Obliquity of the ecliptic (rad).

Return type double

References

- Seidelman 1992, Eq. 3.222-1.

`astrotool.coordinates.`**`par2horiz`**`(`*`ha`*`, `*`dec`*`, `*`phi`*`)`

Convert parallactic coordinates to horizontal.

Parameters

- **ha** (*double*) – Hour angle (rad).
- **dec** (*double*) – Declination (rad).
- **phi** (*double*) – Geographical latitude (rad, N>0).

Returns

tuple containing (az, alt):

- az (double): Azimuth (rad, S=0).
- alt (double): Altitude (rad).

Return type tuple (double,double)

astrotool.coordinates.**precess_from_2000**(*jd*, *ra*, *dec*)
Compute precession in equatorial coordinates from J2000 to that of the specified JD.

J2000 is the equinox of many catalogues, including the Hipparcos one.

Parameters

- **jd** (*double*) – Julian day (days).
- **ra** (*double*) – Right ascension (rad).
- **dec** (*double*) – Declination (rad).

Returns

tuple containing (ra_new, dec_new):

- ra_new (double): Right ascension for the target equinox (rad).
- dec_new (double): Declination for the target equinox (rad).

Return type tuple (double,double)

astrotool.coordinates.**proper_motion**(*jd_start*, *jd_target*, *ra*, *dec*, *pma*, *pmd*)
Compute the proper motion from jd_start to jd_target for the given positions and proper motions.

Parameters

- **jd_start** (*double*) – Julian day of the initial epoch (days).
- **jd_target** (*double*) – Julian day of the target epoch (days).
- **ra** (*double*) – Right ascension (numpy array, rad).
- **dec** (*double*) – Declination (numpy array, rad).
- **pma** (*double*) – Proper motion in right ascension (numpy array, rad/yr).
- **pmd** (*double*) – Proper motion in declination (numpy array, rad/yr).

Returns

tuple containing (ra_target, dec_target):

- ra_target (double): Right ascension for the target epoch (rad).
- dec_target (double): Declination for the target epoch (rad).

Return type tuple (double,double)

1.1.3 astrotool.date_time module

Date and time functions for AstroTool.

astrotool.date_time.**date_time2jd**(*year*, *month*, *day*, *hour*, *minute*, *second*)
Compute the Julian Day for a given year, month, day, hour, minute and second.

Notes

- Date and time are expressed in UT.

Parameters

- **year** (*int*) – Year CE (UT). Note that year=0 = 1 BCE.
- **month** (*int*) – Month number of year (UT; 1-12).
- **day** (*int*) – Day of month with fraction (UT; 1.0-31.999).
- **hour** (*int*) – Hour of time of day (UT).
- **minute** (*int*) – Minute of hour of time (UT).
- **second** (*double*) – Second of minute of time (UT).

Returns jd: Julian day (days).

Return type double

Note:

- uses julian_day().

---

astrotool.date_time.**deltat**(*jd*)

Return the value of DeltaT through interpolation.

For the date range -700 - now, the value for Delta T is obtained by interpolation of known historical values. Outside this range, a lenghtening of the day of 1.8 ms/century is assumed, as well as that the minimum of the parabola is DeltaT=12s in 1820.

Parameters **jd** (*double*) – Julian day (days).

Returns Delta T (s).

Return type double

References

- International Earth Rotation and Reference Systems Service of the U.S. Naval Observatory.
- Robert van Gent's website on Delta T.
- Extrapolation of Delta T.

astrotool.date_time.**deltat_1820**(*jd*)

Return a rough estimate for the value of Delta T.

A lenghtening of the day of 1.8 ms/century is assumed, as well as and that the minimum of the parabola is DeltaT=12s in 1820.

Parameters **jd** (*double*) – Julian day (days).

Returns Delta T (s).

Return type double

References

- Extrapolation of Delta T.

astrotool.date_time.**doy_from_datetime**(*date_time*)
Compute the day of year for a given datetime.

Parameters **date_time** (*datetime*) – Date and time.

Returns Day of year.

Return type (int)

astrotool.date_time.**doy_from_ymd**(*year*, *month*, *day*)
Compute the day of year for a given year, month and day.

Parameters

- **year** (*int*) – Year of date.
- **month** (*int*) – Month of date.
- **day** (*int*) – Day of date.

Returns Day of year.

Return type (int)

astrotool.date_time.**fix_date_time**(*year*, *month*, *day*, *hour*, *minute*, *second*)
Fix a given set of date and time variables (year, month, day, hour, minute and second) to make them consistent.

For example, month=13 will be corrected to month=1 in the next year, day=32 to a date in the next month, hour=24 to hour=0 on the next day, et cetera. This is useful, because some sources list hours between 1 and 24, rather than 0-23, on which Python's datetime crashes. In rare cases of roundoff of 59.9 or a leap second, second=60. More generally, this makes it straightforward to add or subtract dates and times and to take into account timezones, DST, et cetera.

Parameters

- **year** (*int*) – Year CE. Note that year=0 = 1 BCE.
- **month** (*int*) – Month number of year.
- **day** (*int*) – Day of month with fraction.
- **hour** (*int*) – Hour of time of day.
- **minute** (*int*) – Minute of hour of time.
- **second** (*double*) – Second of minute of time.

Returns

tuple containing (year CE, month, day, hour, minute, second):

- year (int): Year CE. Note that year=0 = 1 BCE.
- month (int): Month number of year (UT; 1-12).
- day (int): Day of month with fraction (UT; 1-31).
- hour (int): Hour of time of day (0-23).

- minute (int): Minute of hour of time (0-59).
- second (double): Second of minute of time (0.000-59.999).

Return type tuple (int,int,int, int,int,double)

Note:

- uses date_time2jd() and jd2date_time().

`astrotool.date_time.gmst(jd)`

Calculate Greenwich Mean Sidereal Time for any instant, in radians.

Parameters **jd** (*double*) – Julian day (days).

Returns gmst: Greenwich mean sidereal time (rad).

Return type double

References

- Explanatory Supplement to the Astronomical Almanac, 3rd ed, Eq. 6.66 (2012).

`astrotool.date_time.jd2date_time(jd)`

Compute the date and time from a given Julian Day.

Parameters **jd** (*double*) – Julian day (days).

Returns

tuple containing (year CE, month, day, hour, minute, second):

- year (int): Year CE (UT). Note that year=0 = 1 BCE.
- month (int): Month number of year (UT; 1-12).
- day (int): Day of month with fraction (UT; 1.0-31.999).
- hour (int): Hour of time of day (UT).
- minute (int): Minute of hour of time (UT).
- second (double): Second of minute of time (UT).

Return type tuple (int,int,int, int,int,double)

Note:

- uses jd2ymd().

`astrotool.date_time.jd2tjc(jd)`

Compute the time in Julian centuries since 2000.0.

Parameters **jd** (*double*) – Julian day (days).

Returns tjc: Time in Julian centuries since 2000.0 (UT).

Return type double

astrotool.date_time.**jd2tjm**(*jd*)
Compute the time in Julian millennia since 2000.0.

Parameters **jd** (*double*) – Julian day (days).

Returns tjm: Time in Julian millennia since 2000.0 (UT).

Return type double

astrotool.date_time.**jd2year**(*jd*)
Compute a year with fraction from a given Julian Day.

Parameters **jd** (*double*) – Julian day (days).

Returns Year CE, with decimals. Note that year=0 indicates 1 BCE.

Return type double

astrotool.date_time.**jd2ymd**(*jd*)
Compute the calendar date from a given Julian Day.

Notes

- Date and time are expressed in UT.
- Decimals can be returned in the day to indicate the time of day, e.g. 1.0 for midnight and 1.5 for noon on the first day of the month.

Parameters **jd** (*double*) – Julian day (days).

Returns

Tuple containing (year, month, day):

- year (int): Year CE (UT). Note that year=0 indicates 1 BCE.
- month (int): Month number of year (UT; 1-12).
- day (double): Day of month with fraction (UT; 1.0-31.999).

Return type tuple (int,int,double)

astrotool.date_time.**julian_day**(*year*, *month*, *day*)
Compute the Julian Day for a given year, month and day.

Notes

- Date and time are expressed in UT.
- Decimals can be used in the day to take into account the time of day other than midnight, e.g. 1.5 for noon on the first day of the month.

Parameters

- **year** (*int*) – Year CE (UT). Note that year=0 = 1 BCE.
- **month** (*int*) – Month number of year (UT; 1-12).
- **day** (*double*) – Day of month with fraction (UT; 1.0-31.999).

Returns jd: Julian day (days).

Return type double

astrotool.date_time.**weekday_en_abbr_from_datetime**(*datetime*)
Return an English abbreviation of the weekday for a given datetime.

Parameters **datetime** (*datetime*)

Returns String with the three-character English abbreviation of the weekday (Mon-Sun).

Return type (str)

astrotool.date_time.**ymdhms_us_from_datetime64**(*dt64*)
Convert (array of) datetime64 to a calendar (array of) year, month, day, hour, minute, seconds, microsecond with these quantites indexed on the last axis.

Parameters **dt64** (*datetime64*) – (numpy array of) datetime(s) (of arbitrary shape).

Returns

(..., 7) calendar array with last axis representing year, month, day, hour, minute, second, microsecond.

Return type uint32 array

Note:

- Nicked from https://stackoverflow.com/a/56260054/1386750

1.2 Module contents

1.2.1 AstroTool package

AstroTool is a Python package to do basic astronomical calculations in Python or on the command line. The package can be used under the conditions of the GPLv3 licence. These pages contain the API documentation. For more information on the Python package, licence, source code and data files, see the AstroTool homepage.

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

a

INDEX

Y