
pyorbital Documentation

Release v1.3.1

The Pytroll crew

Mar 23, 2022

Contents

| | | |
|----------|--|-----------|
| 1 | Installation | 3 |
| 2 | TLE files | 5 |
| 3 | Use actual TLEs to increase accuracy | 7 |
| 3.1 | TLE files | 7 |
| 3.2 | Computing satellite position | 8 |
| 3.3 | Computing astronomical parameters | 8 |
| 3.4 | Moon phase and position with Pyorbital | 8 |
| 3.5 | API | 10 |
| 4 | Indices and tables | 13 |
| | Python Module Index | 15 |
| | Index | 17 |

Pyorbital is a python package to compute orbital parameters for satellites from TLE files as well as astronomical parameters of interest for satellite remote sensing. Currently pyorbital only supports low earth orbit satellites.

CHAPTER 1

Installation

Pyorbital comes with a file `platforms.txt` that maps satellite name to NORAD identifier. This file needs to be copied to the appropriate satpy etc directory (`$PPP_CONFIG_DIR`). It is wise to check it contains your satellites of interest. The NORAD identifier can be found as the first number of each line in the Two-Line Elements (eg. from celestrak).

Pyorbital has a module for parsing NORAD TLE-files

```
>>> from pyorbital.orbital import Orbital
>>> from datetime import datetime
>>> # Use current TLEs from the internet:
>>> orb = Orbital("Suomi NPP")
>>> now = datetime.utcnow()
>>> # Get normalized position and velocity of the satellite:
>>> orb.get_position(now)
(array([-0.20015267,  0.09001458,  1.10686756]),
 array([ 0.06148495,  0.03234914,  0.00846805]))
>>> # Get longitude, latitude and altitude of the satellite:
>>> orb.get_lonlatalt(now)
(40.374855865574951, 78.849923885700363, 839.62504115338368)
```

Use actual TLEs to increase accuracy

```
>>> from pyorbital.orbital import Orbital
>>> from datetime import datetime
>>> orb = Orbital("Suomi NPP")
>>> dtobj = datetime(2015,2,7,3,0)
>>> orb.get_lonlatalt(dtobj)
(152.11564698762811, 20.475251739329622, 829.37355785502211)
```

But since we are interesting knowing the position of the Suomi-NPP more than two and half years from now (September 26, 2017) we can not rely on the current TLEs, but rather need a TLE closer to the time of interest:

```
>>> snpp = Orbital('Suomi NPP', tle_file='/data/lang/satellit/polar/orbital_elements/
↳TLE/201502/tle-20150207.txt')
>>> snpp.get_lonlatalt(dtobj)
(105.37373804512762, 79.160752404540133, 838.94605490133154)
```

If we take a TLE from one week earlier we get a slightly different result:

```
>>> snpp = Orbital('Suomi NPP', tle_file='/data/lang/satellit/polar/orbital_elements/
↳TLE/201501/tle-20150131.txt')
>>> snpp.get_lonlatalt(dtobj)
(104.1539184988462, 79.328272480878141, 838.81555967963391)
```

```
>>>>>> master
```

3.1 TLE files

Pyorbital has a module for parsing NORAD TLE-files

```
>>> from pyorbital import tlefile
>>> tle = tlefile.read('noaa 18', '/path/to/my/tle_file.txt')
>>> tle.inclination
99.043499999999995
```

If no path is given pyorbital tries to read the earth observation TLE-files from celestrak.com

3.2 Computing satellite position

The orbital module enables computation of satellite position and velocity at a specific time:

```
>>> from pyorbital.orbital import Orbital
>>> from datetime import datetime
>>> orb = Orbital("noaa 18")
>>> now = datetime.utcnow()
>>> # Get normalized position and velocity of the satellite:
>>> orb.get_position(now)
([0.57529384846822862, 0.77384005228105424, 0.59301408257897559],
 [0.031846489698768146, 0.021287993461926374, -0.05854106186659274])
>>> # Get longitude, latitude and altitude of the satellite:
>>> orb.get_lonlatalt(now)
(-1.1625895579622014, 0.55402132517640568, 847.89381184656702)
```

3.3 Computing astronomical parameters

The astronomy module enables computation of certain parameters of interest for satellite remote sensing for instance the Sun-zenith angle:

```
>>> from pyorbital import astronomy
>>> from datetime import datetime
>>> utc_time = datetime(2012, 5, 15, 15, 45)
>>> lon, lat = 12, 56
>>> astronomy.sun_zenith_angle(utc_time, lon, lat)
62.685986438071602
```

3.4 Moon phase and position with Pyorbital

It is possible to calculate the phase and position of the moon at any given time in the past or future. This is done by an implementation of the astronomical methods described at [Stjaernhimlen](#). The calculations have been compared to the pyephem library, and deviations are small, see below.

3.4.1 Computing the phase of the moon

```
>>> from datetime import datetime
>>> from pyorbital.moon_phase import moon_phase
>>> time_t = datetime(2011, 12, 1, 12)
>>> print moon_phase(time_t)
0.409752921579
```

It works also with datetime sequences:

```
>>> import numpy as np
>>> from pyorbital.moon_phase import moon_phase
>>> time_t = np.arange('2005-02', '2005-03', dtype='datetime64[D]')
```

(continues on next page)

(continued from previous page)

```
>>> print moon_phase(time_t)
[ 6.36537786e-01  5.32201222e-01  4.23413367e-01  3.15141455e-01
 2.13350898e-01  1.24769455e-01  5.62102473e-02  1.34598103e-02
 4.68042562e-05  1.64235364e-02  5.99725484e-02  1.25824054e-01
 2.08064456e-01  3.00824090e-01  3.98939205e-01  4.98161316e-01
 5.95059508e-01  6.86787971e-01  7.70852355e-01  8.44942289e-01
 9.06852663e-01  9.54487201e-01  9.85925226e-01  9.99530703e-01
 9.94086558e-01  9.68941138e-01  9.24152715e-01  8.60612554e-01]
```

3.4.2 Computing the position of the moon

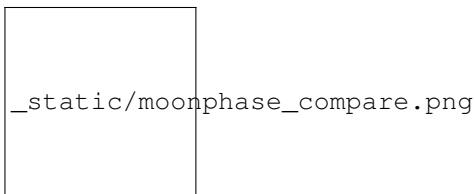
```
>>> from datetime import datetime
>>> from pyorbital import planets
>>> time_t = datetime(2016, 7, 30, 0, 0)
>>> moon = planets.Moon(time_t)
>>> lon = 20.0
>>> lat = 65.0
>>> rasc, decl, alt, azi = moon.topocentric_position(lon, lat)
>>> print alt, azi
6.33389454706 61.6795817556
```

And for an array of longitudes and latitudes:

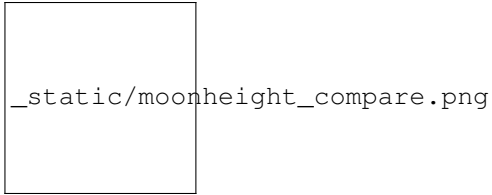
```
>>> import numpy as np
>>> lons = np.arange(100).reshape(10,10)
>>> lats = np.arange(100).reshape(10,10) * 0.9
>>> rasc, decl, alt, azi = moon.topocentric_position(lons, lats)
>>> print alt.shape
(10, 10)
>>> print alt[4,8]
14.9564426346
```

3.4.3 Accuracy

In lack of an absolute truth or reference the moon calculations with Pyorbital have been compared to pyephem. There are indeed deviations, but for the moon phase they are in general rather small. See image below, where we compare the moon phase over 416 days starting from December 1st 2015. As seen from the figure the deviations are within 0.3 %.



For moon height and azimuth differences are larger, as seen from the figures below. We have calculated the position of the moon relative to the City of Norrköping, Sweden, over four months from March 7, 2012. The deviations are within 4 degrees for the height, and within 9 degrees for the azimuth.



3.5 API

3.5.1 Orbital computations

Module for computing the orbital parameters of satellites.

class `pyorbital.orbital.OrbitElements` (*tle*)
Class holding the orbital elements.

class `pyorbital.orbital.Orbital` (*satellite, tle_file=None, line1=None, line2=None*)
Class for orbital computations.

The *satellite* parameter is the name of the satellite to work on and is used to retrieve the right TLE data for internet or from *tle_file* in case it is provided.

find_aol (*utc_time, lon, lat*)

find_aos (*utc_time, lon, lat*)

get_last_an_time (*utc_time*)

Calculate time of last ascending node relative to the specified time

get_lonlatalt (*utc_time*)

Calculate sublon, sublat and altitude of satellite. <http://celestrak.com/columns/v02n03/>

get_next_passes (*utc_time, length, lon, lat, alt, tol=0.001, horizon=0*)

Calculate passes for the next hours for a given start time and a given observer.

Original by Martin.

utc_time: Observation time (datetime object) *length*: Number of hours to find passes (int) *lon*: Longitude of observer position on ground (float) *lat*: Latitude of observer position on ground (float) *alt*: Altitude above sea-level (geoid) of observer position on ground (float) *tol*: precision of the result in seconds *horizon*: the elevation of horizon to compute risetime and falltime.

Return: [(rise-time, fall-time, max-elevation-time), ...]

get_observer_look (*utc_time, lon, lat, alt*)

Calculate observers look angle to a satellite. <http://celestrak.com/columns/v02n02/>

utc_time: Observation time (datetime object) *lon*: Longitude of observer position on ground in degrees east *lat*: Latitude of observer position on ground in degrees north *alt*: Altitude above sea-level (geoid) of observer position on ground in km

Return: (Azimuth, Elevation)

get_orbit_number (*utc_time*, *tbus_style=False*)

Calculate orbit number at specified time. Optionally use TBUS-style orbit numbering (TLE orbit number + 1)

get_position (*utc_time*, *normalize=True*)

Get the cartesian position and velocity from the satellite.

exception `pyorbital.orbital.OrbitalError`

`pyorbital.orbital.get_observer_look` (*sat_lon*, *sat_lat*, *sat_alt*, *utc_time*, *lon*, *lat*, *alt*)

Calculate observers look angle to a satellite. <http://celestrak.com/columns/v02n02/>

utc_time: Observation time (datetime object) *lon*: Longitude of observer position on ground in degrees east
lat: Latitude of observer position on ground in degrees north *alt*: Altitude above sea-level (geoid) of observer position on ground in km

Return: (Azimuth, Elevation)

`pyorbital.orbital.kep2xyz` (*kep*)

3.5.2 TLE handling

exception `pyorbital.tlefile.ChecksumError`

ChecksumError.

`pyorbital.tlefile.SATELLITES = {}`

The platform numbers are given in a file \$PPP_CONFIG/platforms.txt in the following format:

```
# Mappings between satellite catalogue numbers and corresponding # platform names from OSCAR. ALOS-2
39766 CloudSat 29107 CryoSat-2 36508 CSK-1 31598 CSK-2 32376 CSK-3 33412 CSK-4 37216 DMSP-F15
25991 DMSP-F16 28054 DMSP-F17 29522 DMSP-F18 35951 DMSP-F19 39630 EOS-Aqua 27424 EOS-
Aura 28376 EOS-Terra 25994 FY-2D 29640 FY-2E 33463 FY-2F 38049 FY-2G 40367 FY-3A 32958 FY-
3B 37214 FY-3C 39260 GOES-13 29155 GOES-14 35491 GOES-15 36411 Himawari-6 28622 Himawari-7
28937 Himawari-8 40267 INSAT-3A 27714 INSAT-3C 27298 INSAT-3D 39216 JASON-2 33105 Kalpana-1
27525 Landsat-7 25682 Landsat-8 39084 Meteosat-7 24932 Meteosat-8 27509 Meteosat-9 28912 Meteosat-10
38552 Metop-A 29499 Metop-B 38771 NOAA-15 25338 NOAA-16 26536 NOAA-17 27453 NOAA-18 28654
NOAA-19 33591 RadarSat-2 32382 Sentinel-1A 39634 SMOS 36036 SPOT-5 27421 SPOT-6 38755 SPOT-7
40053 Suomi-NPP 37849 TanDEM-X 36605 TerraSAR-X 31698
```

class `pyorbital.tlefile.Tle` (*platform*, *tle_file=None*, *line1=None*, *line2=None*)

Class holding TLE objects.

line1

Return first TLE line.

line2

Return second TLE line.

platform

Return satellite platform name.

`pyorbital.tlefile.fetch` (*destination*)

fetch TLE from internet and save it to *destination*.

`pyorbital.tlefile.main` ()

Main for testing TLE reading.

`pyorbital.tlefile.read` (*platform*, *tle_file=None*, *line1=None*, *line2=None*)

Read TLE for *satellite* from *tle_file*, from *line1* and *line2*, from the newest file provided in the TLES pattern, or from internet if none is provided.

`pyorbital.ttlefile.read_platform_numbers` (*in_upper=False, num_as_int=False*)
Read platform numbers from \$PPP_CONFIG_DIR/platforms.txt if available.

3.5.3 Astronomical computations

Astronomy module. Parts taken from <http://www.geoastro.de/elevaz/basics/index.htm>

`pyorbital.astronomy.cos_zen` (*utc_time, lon, lat*)
Cosine of the sun-zenith angle for *lon, lat* at *utc_time*. *utc_time*: `datetime.datetime` instance of the UTC time
lon and *lat* in degrees.

`pyorbital.astronomy.get_alt_az` (*utc_time, lon, lat*)
Return sun altitude and azimuth from *utc_time, lon, and lat*. *lon, lat* in degrees What is the unit of the returned
angles and heights!? FIXME!

`pyorbital.astronomy.gmst` (*utc_time*)
Greenwich mean sidereal *utc_time*, in radians.

As defined in the AIAA 2006 implementation: <http://www.celestrak.com/publications/AIAA/2006-6753/>

`pyorbital.astronomy.jdays` (*utc_time*)
Get the julian day of *utc_time*.

`pyorbital.astronomy.jdays2000` (*utc_time*)
Get the days since year 2000.

`pyorbital.astronomy.observer_position` (*time, lon, lat, alt*)
Calculate observer ECI position.

<http://celestrak.com/columns/v02n03/>

`pyorbital.astronomy.sun_earth_distance_correction` (*utc_time*)
Calculate the sun earth distance correction, relative to 1 AU.

`pyorbital.astronomy.sun_ecliptic_longitude` (*utc_time*)
Ecliptic longitude of the sun at *utc_time*.

`pyorbital.astronomy.sun_ra_dec` (*utc_time*)
Right ascension and declination of the sun at *utc_time*.

`pyorbital.astronomy.sun_zenith_angle` (*utc_time, lon, lat*)
Sun-zenith angle for *lon, lat* at *utc_time*. *lon, lat* in degrees. The angle returned is given in degrees

CHAPTER 4

Indices and tables

- `genindex`
- `modindex`
- `search`

p

`pyorbital.astronomy`, 12
`pyorbital.orbital`, 10
`pyorbital.tlefile`, 11

C

ChecksumError, 11
cos_zen() (in module *pyorbital.astronomy*), 12

F

fetch() (in module *pyorbital.tlefile*), 11
find_aol() (*pyorbital.orbital.Orbital* method), 10
find_aos() (*pyorbital.orbital.Orbital* method), 10

G

get_alt_az() (in module *pyorbital.astronomy*), 12
get_last_an_time() (*pyorbital.orbital.Orbital* method), 10
get_lonlatalt() (*pyorbital.orbital.Orbital* method), 10
get_next_passes() (*pyorbital.orbital.Orbital* method), 10
get_observer_look() (in module *pyorbital.orbital*), 11
get_observer_look() (*pyorbital.orbital.Orbital* method), 10
get_orbit_number() (*pyorbital.orbital.Orbital* method), 10
get_position() (*pyorbital.orbital.Orbital* method), 11
gmst() (in module *pyorbital.astronomy*), 12

J

jdays() (in module *pyorbital.astronomy*), 12
jdays2000() (in module *pyorbital.astronomy*), 12

K

kep2xyz() (in module *pyorbital.orbital*), 11

L

line1 (*pyorbital.tlefile.Tle* attribute), 11
line2 (*pyorbital.tlefile.Tle* attribute), 11

M

main() (in module *pyorbital.tlefile*), 11

O

observer_position() (in module *pyorbital.astronomy*), 12
Orbital (class in *pyorbital.orbital*), 10
OrbitalError, 11
OrbitElements (class in *pyorbital.orbital*), 10

P

platform (*pyorbital.tlefile.Tle* attribute), 11
pyorbital.astronomy (module), 12
pyorbital.orbital (module), 10
pyorbital.tlefile (module), 11

R

read() (in module *pyorbital.tlefile*), 11
read_platform_numbers() (in module *pyorbital.tlefile*), 11

S

SATELLITES (in module *pyorbital.tlefile*), 11
sun_earth_distance_correction() (in module *pyorbital.astronomy*), 12
sun_ecliptic_longitude() (in module *pyorbital.astronomy*), 12
sun_ra_dec() (in module *pyorbital.astronomy*), 12
sun_zenith_angle() (in module *pyorbital.astronomy*), 12

T

Tle (class in *pyorbital.tlefile*), 11