XYZ Engineer - Technical Task

Muhammad Ahsan

Mar 20, 2021

CONTENTS:

CHAPTER

ONE

XYZ ENGINEER - TECHNICAL TASK

1.1 Task Description

Download the TLEs of a given satellite from the web and propagate it using SGP4 orbit propagator for *x* days. Report the new TLE with the epoch.

1.2 Background Knowledge

A set of six independent elements called "classical Keplerian obital elements" sufficiently describe the size, shape and orientation of an orbit and pinpoint the position of a satellite along the orbit at a particular time [1]. These are:

1. a_o, *semi-major axis* – a constant defining the size of the conic orbit.
2. e_o, *eccentricity* – a constant defining the shape of the conic orbit (0=circular, less than 1=elliptical). In Two Line Element set(TLEs), the value provided is the mean eccentricity. A leading decimal must be applied to this value.
3. i_o, *inclination* – the angle between the equator and the orbit plane. In TLEs, this value is the TEME (True Equator, Mean Equinox) mean inclination.
4. Ω_o, *longitude of the ascending node* – right ascension of the ascending node; the angle between vernal equinox and the point where the orbit crosses the equatorial plane(going north). In TLEs, this is TEME mean right ascension of the ascending node.
5. ω_o, *argument of perigee* – the angle, in the plane of the satellite's orbit, between ascending node and the periapsis point, measured in the direction of the satellite's motion.
6. T_o, *time of periapsis passage* – the time when the satellite was at periapsis.

The TLEs include *eccentricity*, *inclination*, *right ascension of the ascending node* and *argument of perigee*. The subscript '*o*' indicates mean values. Other elements in TLEs include [2]:

- M_o, mean anomaly is the angle (in degrees) measured from perigee, of the satellite location in the orbit referenced to a circular orbit with radius equal to the semi-major axis.

- n_o, mean motion is the mean number of orbits/day the object completes. There are *8* digits after the decimal, leaving no trailing space(s) when the element exceeds *9999*.
- $B^{\star}$, a drag term (also called radiation pressure coefficient) in the SGP4 predictor. The last two characters define an applicable power of 10.
- $\frac{1}{2} \cdot \frac{\mathrm{d}}{\mathrm{dt}} . n_o$, first derivative of mean motion (also called ballistic coefficient) is the daily rate of change in number of revs/day the object completes, divided by 2. This is "catch all" drag term used in the SGP4 predictor.
- $\frac{1}{6} \cdot \frac{\mathrm{d}^2}{\mathrm{dt}^2} . n_o$, second derivative of mean motion is a second order drag term in SGP4 predictor used to model terminal orbit decay. It measures the second time derivative in daily mean motion, divided by *6*. A leading decimal must be applied to this value. The last two characters define an applicable power of *10*.

A sample TLEs of a satellite is shown below in Fig. 1.1.

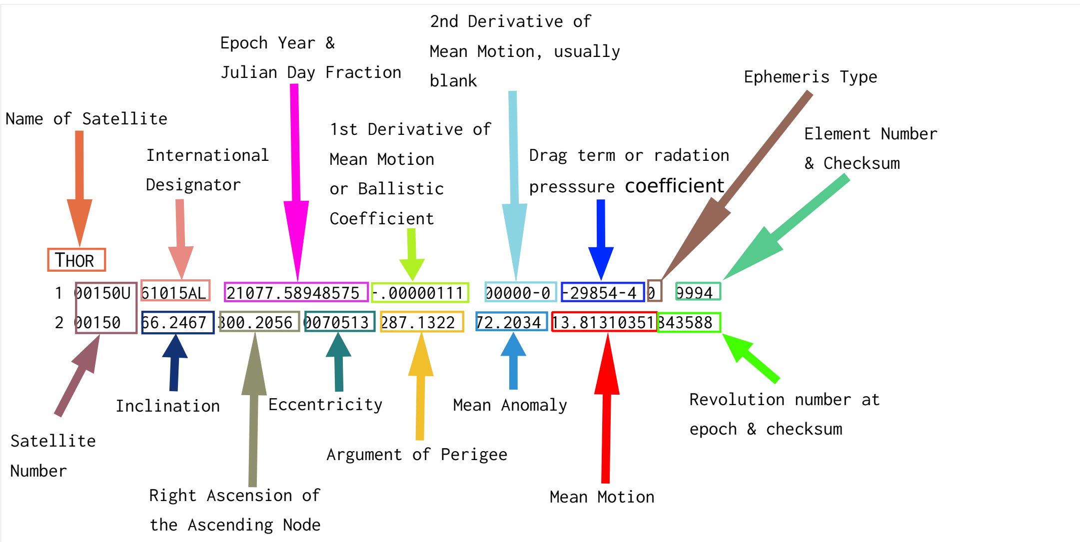

Fig. 1.1: TLE Format Decomposition

TLEs are distributed as two-line element sets in ASCII form with additional first line containing a satellite identifier which is not used by the prediction models. The TLEs also contain the *epoch* time for the data, as well as additional information not used by the propagators.

The elements in the published TLEs are not in the units required by the propagators and therefore must be converted prior to calling the prediction routines. Element units in both the published TLEs and needed in propagators are described in table Table 1.1 [3].

Table 1.1: Units of TLEs

Element	Published Units	Propagator Units
e_o	Unitless	Unitless
i_o	degrees	radians
Ω_o	degrees	radians
ω_o	degrees	radians
M_o	degrees	radians
n_o	revolutions/day	radians/minute
$B^\star$	Earth-radii^{-1}	Earth-radii^{-1}
1st derivative of n_o	revolutions/day^2	radians/minute^2
2nd derivative of n_o	revolutions/day^3	radians/minute^3

Various mathematical satellite prediction models exist for propagting Earth-orbiting satellite with NORAD mean elements. SGP, SGP4, and SGP8 are models for near-Earth satellites (also called LEO satellites) whereas SDP4 and SDP8 are models for deep-space satellites and are beyond the scope of this document. Of all these models, SGP4 is of most interest as we will use it for propagation in this assignment. SGP4 was develped by Crawford in 1970. It is a simplification of the analytical theory of Lane and Crawford. SGP4 uses a power density function for modeling the atmosphere and the solution of Brouwer for its gravitational model [4]. A detailed description of SGP4 is beyond the scope of this document. A curious reader might refer to section 6 of [5] and [6].

In general, a propagation model reads the 2-line element set and calls the appropriate ephemeris package as specified by the user [1]. As per this candidate's understanding, the idea of the assignment is to take the initial TLE, change the epoch in the initial TLE set, and calculate the above classical orbital and additional elements and wrap these elements into a new TLE set. In order to better understand the overall picture, this candidate tried to investigate the problem in Julia as well. For the sake of brevity, the results of this investigation are shared as a small Julia script available in the same repository.

Following the pattern similar to Julia's script we will write a Python code to generate a new TLEs and epoch and wrap them into a new TLEs.

1.3 Proposed Solution

The proposed solution decomposes the task into smaller tasks and attempts them one each as an independent task. A rough decomposition of the solution is as follows:

1. Download TLEs of the satellite
2. Manipulate the TLEs and get current epoch
3. Propagate using SGP4 orbit propagator for x days
4. Get the new epoch and orbital elements with respect to new epoch
5. Repor/wrap the new TLEs with the new epoch

Solution is implemented in *Python*. Other technologies to solve the task include *Docker* and *Sphinx* document generation API.

The required *Python* libraries include *skyfield*, *argparse*, and *dateutil*. There is no need to install or setup any libraries, everything is already done inside the docker container and the solution can be run out of the box.

skyfield is a *Python* library to compute positions of stars, planets, and satellites in orbit around the Earth. Its results should agree with the positions generated by the United States Naval Observatory and their *Astronomical Almanac* to within *0.0005* arcseconds (half a mas or milliarchseconds).

argparse is use to handle the number of days as commandline input argument to python script.

dateutil is used to calculate the new epoch based on the number of days.

The solution consists of a *Python* class named **SimulateX2**. This class has two attributes and about half a dozen accompanying methods. The two attributes are *Satellite Catalog Number(CAN)* and *number of days*. Both of these values are provided by the user as commandline arguments.

Class methods include *getTLEs*, *loadTLEs*, *getEpoch*, *getNewEpoch*, *getPV*, *getLonLatAlt*, *getOrbitalElements printResults*, and *generateTLEs*. Most of these methods are self-explanatory.

1.3.1 Download TLEs

We download the TLEs wth the help of *getTLEs* method and save them locally to a file named *tle-CATNR-<catnr>.txt*. As the “epoch” date of a satellite element set is all-important date and time for which the set of elements is most accurate, and before and after which they go rapidly out of date; the method *getTLEs* makes sure that latest TLEs are fetched from the Internet. This is implemented with *reload=True*.

1.3.2 Manipulate the TLEs to get the Current Epoch

After downloading TLEs from the Internet, we will load them from locally saved file and split and save the satellite name, line 1 and line 2 in different variables. This is achieved with *loadTLEs* method.

We will split TLEs to find the current epoch of the satellite. This is done with *getEpoch* method.

1.3.3 Calculate the New Epoch

New epoch is calculated with the help of *getNewEpoch* method based on the given number of days. We will take the current epoch as obtained via *getEpoch* and add the given number of days to it to find the new TLEs epoch.

1.3.4 Propagate Using SGP4 Orbit Propagator for *x* Days

Skyfield is able to predict the positions of Earth satellites by loading satellite orbital elements from TLEs files and running them through the SGP4 satellite propagation routine.

The elements constituting TLEs can be decomposed into two categories:

- Static elements
- Dynamic elements

Static elements are the ones that remain unchanged across TLEs in general. These include *satellite identifier*, *satellite catalog number*, *international designator*, *second derivative of mean motion (0)*, *ephemeris(0 in TLEs, used internally)*, and *element set number(floats between 1-999)*. Rest of the elements are obtained via certain calculations and categorized as dynamic elements.

We will use the class method *getPV* to obtain the position, and velocity at new epoch. Similarly, we calculate the dynamic elements in *getOrbitalElements* method. We use the method *printResults* to simply print the various fields of new TLEs. In future, these fields will be wrapped into a new TLEs in *generateTLEs* method. Please refer to the code to get more details about the aforementioned methods.

1.3.5 Report New TLEs

Calculation of satellite position is a pre-requisite for deducing the new orbital elements and ultimately generating the new TLEs, hence the two methods *getGeoPosition* and *getLonLatAlt*. The former calculates the geocentric coordinates where x, y, and z positions relative to Earth's center in the Geocentric Celestial Reference System (GCRS) while the latter calculates the point on Earth's globe which is directly beneath the satellite.

The new TLEs will be wrapped into a satellite record object in the method *generateTLEs*.

If we try to estimate the position of a given satellite 10 days from the original epoch, we can see that with slight variations, the results will coincide with the ones we can obtain with the Julia script.

Pasted below results are for *THOR ABLESTAR*'s debris.

```
TLEs are saved in tle-CATNR-150.txt
TLE epoch: A.D. 2021-Mar-18 19:21:38.5422 UTC
New epoch: 2021-03-28 19:21:38.542175+00:00
Longitude: 177deg 46' 41.8"
```

(continues on next page)

(continued from previous page)

```
Latitude: 37deg 20' 01.3"
Altitude (in kilometers): 989.2162075080619
New epoch: 2021-03-28 19:21:38.542175+00:00
Geocentric x,y,z in GCRS: [ 2426.46312711 -5342.64855118  4441.95867708]
Velocity (km/s): [1.82727618 4.99431052 5.07062533]
New epoch: 2021-03-28 19:21:38.542175+00:00

 ========== Line 0 of New TLEs ==========
Satellite name: THOR ABLESTAR DEB

 ========== Line 1 of New TLEs ==========
Satellite number: 150
Satellite classification: U
International designator: 61015AL
Epoch year: 21
Epoch days: 87.80669609
Ballistic drag coefficient: <placeholder>
2nd Derivative of the mean motion (ignored by SGP4): <placeholder>
Drag term: <placeholder>
Ephemeris Type: 0
Element number: 999

 ========== Line 2 of New TLEs ==========
Satellite number: 150
Inclination (in degrees): 66.13235085123695
RAAN: 274deg 51' 24.7"
Eccentricity: 0.005946896805975747
Argument of perigee: 279deg 36' 13.1"
Mean Anomaly: 121deg 06' 56.4"
Mean motion: 4973deg 19' 04.5"
```

BIBLIOGRAPHY

[1] Roger R. Bate, Donald D. Mueller, and Jerry E. White. *Fundamentals of astrodynamics*. Dover Publications, New York, NY, 1971.

[2] T.S. Kelso. NORAD documentation. http://www.celestrak.com/NORAD/documentation/, 2019. Last updated: 2019-12-19.

[3] Dwight E. Andersen. Computing norad mean orbital elements from a state vector. Master's thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, December 1994.

[4] Simon P. Shuster. A survey and performance analysis of orbit propagators for leo, geo, and highly elliptical orbits. Master's thesis, Utah State University, Logan, Utah, May 2017.

[5] Felix R. Hoots and Ronald L. Roehrich. SPACETRACK REPORT NO. 3 models for propagation of NORAD element sets. Technical Report, Defense Documentation Center, Cameron Station, Alexandria VA 22314, December 1980.

[6] David A. Vallado, Paul Crawford, Richard Hujsak, and Ronald T. S. Kelso. Revisiting spacetrack report#3. Technical Report AIAA 2006-6753, Center for Space Standards and Innovation, Colorado Springs, Colorado, 80920, August 2006.