Spacecraft Observation Report

Dakota Richline

Department of Aerospace, Physics, and Space Sciences, Florida Tech

AEE 4263: Space Flight Mechanics

Dr. Paula do Vale Pereira

Spacecraft Observation Report

Spacecraft Selection

I weighed four criteria to choose a spacecraft: low inclination, size, viable passes, and an interesting or historic payload. Starting with viable passes, I used In-The-Sky.org's "Satellite Observing Opportunities" list to find a bright satellite with an evening pass on November 2nd. The winner was "SAKURA (CS) and DELTA 1 R/B" (NORAD #10517): the Delta 2914 second stage that inserted Japan's first prototype communications satellite into orbit in 1977. It has an inclination nearly identical to Melbourne's latitude ($i = 28.7°$), resulting in many high-angle passes that happen to occur at convenient times during the two-week observation period. I chose the spacecraft on October 31st, obtained a TLE from Celestrak, and double-checked my November 2nd pass against N2YO. The three-day-old TLE should be adequate for a dead spacecraft with a semimajor axis of 7478 km (altitude > 1000 km); atmospheric drag is negligible and no thrusters are firing, although J_2 effects result in a nodal rotation rate $\dot{\Omega} \approx -5\,°/\text{day}$. If I'm not mistaken, apsidal rotation should not affect passes for a circular orbit.

Code Overview

I chose to write the code in a Pluto notebook using PyCall.jl to bridge between Julia and the Skyfield Python library, which was more difficult and had no advantage over just using Python. Most of the code is adapted from Skyfield's Earth Satellite documentation with some additions from other Skyfield example code: local date/time conversion using pytz, topographic Sun elevation from JPL ephemeris data, and a tabular output file. The code takes a TLE and local coordinates of the observer as input, and outputs a list of passes with elevation $\geq 15°$ in CSV (comma-separated value) format. The tabular output file 'out.csv' has six columns per pass: local date, local time, azimuth, max elevation, sunlit, and Sun elevation.

```
begin
    using PyCall
    using Conda
    using SatelliteToolbox
    using Pkg
end
```

```
PyObject <module 'datetime' from '/home/dakota/.julia/conda/3/x86_64/lib/python3.10/datet
```

```
begin
    ENV["PYTHON"]=""
    #Pkg.build("PyCall")
    Conda.add("skyfield")
    Conda.add("pytz")
    skyfield = pyimport_conda("skyfield.api","skyfield")
    timezone = pyimport_conda("pytz", "timezone")
    datetime = pyimport("datetime")
end
```

Code adapted from https://rhodesmill.org/skyfield/earth-satellites.html

```
md"""**Code adapted from https://rhodesmill.org/skyfield/earth-satellites.html**"""
```

```
tle = TLE:
                      Name : SAKURA DELTA 1 R/B(1)
          Satellite number : 10517
  International designator : 77118B
        Epoch (Year / Day) : 23 / 304.59269594 (2023-10-31T14:13:28.929)
        Element set number : 999
              Eccentricity :   0.08346580
               Inclination :  28.66040000 deg
                      RAAN : 265.06250000 deg
       Argument of perigee : 182.01920000 deg
              Mean anomaly : 177.68500000 deg
           Mean motion (n) :  13.42567491 revs / day
         Revolution number : 22206
                        B* :   0.00025053 1 / er
                     ṅ / 2 :    1.454e-05 rev / day²
                     n̈ / 6 :            0 rev / day³
```

```
# Validate TLE
    tle = tle"""
SAKURA DELTA 1 R/B(1)
1 10517U 77118B   23304.59269594  .00001454  00000+0  25053-3 0  9990
2 10517  28.6604 265.0625 0834658 182.0192 177.6850 13.42567491222063
"""
```

▸ ("rise above 15°", "culminate", "set below 15°")

```
begin
    ts = skyfield.load.timescale()
    name = "SAKURA DELTA 1 R/B(1)"
    line1 = "1 10517U 77118B   23304.59269594  .00001454  00000+0  25053-3 0  9990"
    line2 = "2 10517  28.6604 265.0625 0834658 182.0192 177.6850 13.42567491222063"
    satellite = skyfield.EarthSatellite(line1, line2, name, ts)

    eph = skyfield.load("de440s.bsp")
    earth = py"""$eph['earth']"""
    sun = py"""$eph['sun']"""
    FIT_topocentric = earth + skyfield.wgs84.latlon(28.065833, -80.624444)

    FIT = skyfield.wgs84.latlon(28.065833, -80.624444)
    eastern = timezone.timezone("US/Eastern")
    t0 = ts.utc(2023, 11, 02, 0, 0, 0)
    t1 = ts.utc(2023, 11, 14, 23, 59, 0)

    t, events = satellite.find_events(FIT, t0, t1, altitude_degrees=15.0)
    event_names = "rise above 15°", "culminate", "set below 15°"
end
```

```
py"""
import csv
f = open('out.csv', 'w')
writer = csv.writer(f)
header = ['Date', 'Local Time', 'Azimuth', 'Max Elevation', 'Sunlit', 'Sun Alt']
writer.writerow(header)

for ti, event in zip($t, $events):
    name = $event_names[event]
    localTime = ti.astimezone($eastern)

    difference = $satellite - $FIT
    topocentric = difference.at(ti)
    alt, az, distance = topocentric.altaz()

    sunlit = $satellite.at(ti).is_sunlit($eph)
    astrometric = $FIT_topocentric.at(ti).observe($sun)
    sunAlt, sunAz, d = astrometric.apparent().altaz()

    if name == "culminate":
        if sunlit == True and sunAlt.degrees < 0:
            visible = True
        else:
            visible = False
        data = [ti.astimezone($eastern).strftime('%m/%d/%Y'), ti.astimezone($eastern).strftime('%H:%M:%S'), '{:.6}'.format(az.degrees), '{:.6}'.format(alt.degrees), sunlit, '{:.4}'.format(sunAlt.degrees)]
        writer.writerow(data)
f.close()
"""
```

Code Details

The program begins by adding necessary packages to the Julia environment, then preparing the root Python environment using Julia's Conda distribution. Prior to the Python/Skyfield code, a macro from SatelliteToolbox.jl parses the TLE and converts it to a data structure. The TLE parser raises an error when either line does not have the correct spacing or number of characters, making it a useful tool to validate TLEs before passing them to Skyfield. All code after this point is written in Python.

The next code section assigns input variables for Skyfield and generates the arrays of pass data. First, TLE variables are assigned and JPL's de440s Planetary Ephemeris file is loaded. The ephemeris is an extremely precise model of planetary motion integrated over an 1100-year span, then encoded as a database of polynomials. "FIT" is defined by its local coordinates both in the geologic and topocentric frames; the latter is required to calculate the Sun's local elevation. The start and end times are defined in UTC, along with a timezone for local time/date conversion. Finally, pass events and their times are calculated and stored in respective arrays.

The last section of Python code calculates the aforementioned six columns for each pass event, but only stores culmination events. A topocentric state vector from the observer to the SC is calculated at culmination time, then converted to azimuth and elevation. Next, the is_sunlit() function is called and the Sun's elevation is calculated from the observer's topocentric position vector. Finally, each culmination event and its supplementary data are stored as a single row in the output CSV file. The CSV file was opened in LibreOffice Calc and formatted as the following section's LaTeX table.

List of Passes

Table 1

Passes between 19:15 — 20:45 ET

Date	Local Time	Azimuth	Max. Elevation	Sunlit	Sun Elevation	Viable
11/01/2023	20:32:13	170.2	86.3	True	-26	No
11/02/2023	19:42:03	167.3	80.9	True	-15	Yes
11/04/2023	19:57:04	355.3	88.3	True	-18	No
11/05/2023	20:02:24	5.3	89.3	True	-33	No
11/07/2023	20:17:24	194.9	78.0	False	-36	No
11/08/2023	19:27:09	191.3	84.7	True	-25	No
11/09/2023	20:32:09	203.2	59.1	False	-40	No
11/10/2023	19:41:58	199.5	68.7	True	-29	No
11/12/2023	19:56:34	206.7	49.7	True	-32	No

Note: Passes are marked viable if they are sunlit and occur during observation days

With the end of daylight savings time, passes earlier in the evening may have been viable but were not considered because I had already completed the observation.

Spacecraft Observation

My viable pass was on November 2nd, which was unfortunately a cloudy evening. It was shortly after another student's pass, so we didn't have enough time to aim the telescope camera before culmination. I might have seen the rocket body; the spot resembled other satellite passes I've seen, it looked too fast to be an airplane, and it was moving in roughly the right direction. It might not have been the rocket body, but I'm fairly sure it was a satellite. Nevertheless, we did see plenty of airplanes, an unscheduled satellite pass, and a very bright Jupiter.