

Development of Open Source tools for SatCom constellation analysis

First deliverable

Author: Juan Luis Cano Rodríguez <hello@juanlu.space>

Publication date: 2020-11-17

Last modification date: 2020-12-14

Introduction

One of the most widely used exchange formats for orbital information are Two-line Element sets (TLEs). TLEs for all known artificial satellites, as well as several large space debris objects, are produced by the United States Armed Forces and disseminated to the public, therefore making them easily available for consumers. These TLEs are meant to be propagated with SGP4, one of the Simplified General Perturbations models published by the United States Department of Defense. The source code for these analytic models, which offer a good tradeoff between running time and accuracy, was made public, again democratizing its usage.

However, “TLEs were designed [...] at a time when both bandwidth for transmission or digital storage were extremely limited”, and as a result they exhibit several limitations, like poor legibility and limited accuracy for the variables. CelesTrak, the main public source of general perturbations orbital data, announced in May 2020 that they would start using the Orbit Mean-Elements Message format (OMM) “that is part of the Orbit Data Messages (ODM) Recommended Standard CCSDS 502.0-B-2 developed by The Consultative Committee for Space Data Systems (CCSDS) in November 2009”. OMMs contains all the information available in TLEs and can include extra data, while being free from the character limits imposed by the previous format.

For SatCom constellations, TLEs have a dual mission: on one hand, they provide an invaluable tool for long term simulation, since SGP4 can capture the most important perturbation effects while retaining good performance. And on the other hand, they are a low cost monitoring tool that can be leveraged without the need to deploy any tracking infrastructure.

Overview

As part of the Development of Open Source tools for SatCom constellation analysis, one of the sub-activities of the OpenSatCom initiative implemented by Libre Space Foundation and Inno3 for the European Space Agency, the first proposed step was the “exploration of ongoing and complete works to read and write CCSDS OMM format”, which is described in this document.

To this end, we selected a list of free/open source libraries for TLE/GP handling and orbit propagation, mostly focused on the Python programming language, and assessed their capabilities. We devoted special attention to **python-sgp4**, since it is the most widely used pure Python library for propagating TLEs with SGP4. For the rest of the libraries we performed a brief assessment and included our conclusions in this report.

Selected libraries

Name	Source code	License	Last commit ¹	Downloads ²
python-sgp4	https://github.com/brandon-rhodes/python-sgp4	MIT	2020-11-14	125 895
Orekit	https://gitlab.orekit.org/orekit/orekit	Apache	2020-11-17	165 254 ³
beyond	https://github.com/galactics/beyond	MIT	2020-11-12	3 906
python-satellitetele	https://gitlab.com/librespacefoundation/python-satellitetele	MIT	2020-07-29	276
oacmpy	https://gitlab.com/jorispio/ccsds2czml	MIT	2020-05-19	12
ccsds-ndm	https://github.com/egemenimre/ccsds-ndm	GPLv3	2020-12-06	166

The Appendix includes other related libraries that were excluded from the analysis.

¹ In master branch or equivalent

² Unless noted, last month downloads as obtained from <https://pypistats.org/packages/>

³ Total downloads from conda as obtained from <https://anaconda.org/conda-forge/orekit> Notice that probably conda is not the canonical way to obtain Orekit, therefore these numbers are under-reported, and also that the scope of the project is much broader and therefore cannot be compared to the other ones.

Analysis of capabilities

python-sgp4 (MIT)

python-sgp4 is the most downloaded Python library for propagating TLEs using SGP4: its implementation of SGP4, directly ported from the public C++ version by the Center for Space Standards and Innovation, has been battle tested since 2012.

In October 2020, the library gained support for *reading* OMM data in XML or CSV form, for the purpose of orbit propagation exclusively. Therefore, it has **partial** read support, since it ignores the covariance matrix and other keys. On the other hand, to date it has **no** write support, although, as explained below, an agreement on the API was reached and stated publicly.

Three changes were proposed to the library during the course of the analysis:

- *“Raise error in export_tle if angles are invalid”* [#72](#) **(Accepted)** Consumers might end up producing orbital element sets that, while correct, might contain out of range values. In particular, TLEs assume that the angles have a value between 0 and 360 degrees (0 to 180 for the inclination), but before this change there was no validation taking place.
- *“Fix export_tle for old satrec objects”* [#73](#) **(Accepted)** The TLE export was invalid in some cases.
- *“Add a export_omm function that produces a mapping containing all the OMM keys”* [#65](#) **(Deferred)** Despite having the ability to load lists of XML and CSV OMMs, python-sgp4 does not have the ability to export OMMs from its internal representation. While working on the implementation, a roadblock was identified that the package maintainer wanted to address personally, and therefore this capability is still missing.

Orekit (Apache 2.0)

Orekit is a broad space dynamics library written in Java, developed under an open governance model. It has a wide range of applications, including (but not limited to) orbit propagation, frame transformation, attitude computation, perturbation analysis, visualization, and more. It can be integrated with other languages, like MATLAB and Python, although this setup is more complex.

It has **full** read support for all ODM formats (OMM, ODM and OEM) in KVN (key-value notation). However, to the best of our knowledge⁴, **no** support for XML read is present, and **no** support for writing OMM files is available either.

⁴ The author of this report does not claim expertise on Java nor Orekit, and the conclusions here might be misguided.



beyond (MIT)

beyond is a simple library for flight dynamics written in pure Python. It implements several algorithms from widely known Astrodynamics textbooks, and it even contains an original implementation of SGP4.

It has **full** read support for all ODM formats in KVN and XML, as well as **full** write support in both formats. Thorough tests are included in the source code that cover this functionality. It is, therefore, the most promising among the analyzed libraries.

python-satellitetele (MIT)

python-satellitetele is a Python package developed by the Libre Space Foundation that includes functions to fetch TLEs from several sources. It contains a handful of predefined sources that serve TLEs for satellites transmitting in radio amateur bands, and it also supports loading arbitrary data from Space-Track.org.

At the moment, python-satellitetele does not have OMM read nor write support, but a complete implementation is available at [125](#) waiting to be accepted in the repository. This contribution adds **full** JSON and CSV read capabilities, as well as **full** XML and JSON write capabilities.

oacmpy (MIT)

oacmpy, also known as Ccsds2CZML, is a Python library focused on converting ODM formats to CZML, a JSON subset used primarily by Cesium.js to describe time-dynamic scenes.

It has **full** support for all ODM formats in XML and KVN, but unfortunately **no** support for writing those, apart from the aforementioned conversion to CZML.

ccsds_ndm (GPLv3)

ccsds_ndm was announced right after the first version of this report was delivered, and happens to be a promising alternative. Its focus is to provide solely I/O capabilities for several Navigation Data Messages (NDM) subformats, including OMM.

At the moment it has full support for XML, and both KVN and JSON are in the works.

Conclusions and future work

Support for OMM and other ODM formats seems to be quite sparse in the open source ecosystem. The fact that Celestrak has started to push for its adoption should serve as a motivation to coalesce towards one canonical open source package to read and write them.

Among all the packages that were analyzed, `beyond` seems to be the most complete, while the most widely used one, `python-sgp4`, probably will not gain full support for these formats any time soon. Interestingly, `beyond` happens to have `python-sgp4` as a dependency: being a seemingly complete solution for reading and writing both TLEs and OMMs, perhaps it would be useful to split that part into a separate, lower-level library that could be reused by other projects in the ecosystem. However, the current implementation has some internal coupling with its other data structures, and some discussion (or a fork) would be needed. We have not reached out to the original author of `beyond` with these ideas yet.

On the other hand, `ccsds_ndm` was created recently with the explicit intent of providing a library solely focused on NDM, including OMM. While at the moment only XML support is present, its author has declared that KVN and JSON are also on the roadmap. Still, its GPLv3 license might somewhat limit adoption in the broader ecosystem.

Summary

Name	OMM read	OMM write	Notes
python-sgp4	XML, CSV	In progress	Only elements, no covariances or metadata. JSON write in progress.
Orekit	KVN	Missing	
beyond	KVN, XML	KVN, XML	
python-satellitetele	In progress	In progress	JSON and CSV read in progress. XML and JSON write in progress.
oacmpy	KVN, XML	Missing	Only export to CZML.
ccsds-ndm	XML	XML	Only non-permissively licensed library. KVN and JSON in the roadmap.

References

- Vallado, D. A., et al. "Revisiting spacetrack report# 3: rev 2. AIAA-2006-6753-Rev2. AIAA Astrodynamics Specialists Conference and Exhibit. Keystone, CO: American Institute of Aeronautics and Astronautics." (2006).
- Orbit Data Messages, Recommended Standard CCSDS 502.0-B-2, Consultative Committee for Space Data Systems.
- "A New Way to Obtain GP Data (aka TLEs)"
<https://celestrak.com/NORAD/documentation/gp-data-formats.php> [Retrieved 2020-11-17].
- Vallado, David A. "Fundamentals of Astrodynamics and Applications, chapter 8." Space Technology Library, (2013).



Appendix: Excluded libraries

- orbitdeterminator <https://github.com/aerospaceresearch/orbitdeterminator> contains a standalone implementation of SGP4 in Python (which is rare) but relies on python-sgp4 for all the I/O.
- odmpy <https://github.com/RazerM/odmpy> contains a high quality implementation of OPM, however the initial idea of adding OMM did not come to reality and the library was last updated in 2014.
- oem <https://github.com/bradsease/oem> is maintained and contains a high quality implementation of OEM, however OMM support does not seem to be within the scope of the project.