CELESTLAB: SPACEFLIGHT DYNAMICS TOOLBOX FOR MISSION ANALYSIS

Alain Lamy, Thierry Martin

CNES, Centre National d’Etudes Spatiales, Toulouse, France

ABSTRACT

CelestLab is a free and open source spaceflight dynamics Scilab toolbox developed by CNES that is particularly suited to spaceflight dynamics mission analysis. The toolbox contains about 250 functions related to coordinate systems (IERS 2010 conventions), orbit propagation, geometry and events, force models, orbit properties and more. Lots of examples, demos, help pages and tutorials are provided which makes it easy for new users to get started.

The first version was put on ATOMS website (website from which Scilab modules can be downloaded) at the end of 2009 (see http://atoms.scilab.org/toolboxes/celestlab). The module is one of the most popular as the number of downloads for all versions now approaches 40000.

CelestLab is completed by an extension module called CelestLabX [2]. CelestLabX contains low level interfaces to available software. The last version of CelestLab / CelestLabX provides utilities related to TLEs (interface to C code for the propagation of Two-Line Elements from: http://celestrak.com/software/vallado-sw.asp) and to STELA (Semi-analytic Tool for End of Life Analysis, tool used at CNES and elsewhere for orbit long-term propagation in particular and used in the context of debris mitigation, see: https://logiciels.cnes.fr/content/stela?language=en).

The paper describes CelestLab contents in more details and particularly focusses on recent features. Lots of illustrations and examples taken from actual mission analyses conducted at CNES will also be shown in order to illustrate some typical uses of CelestLab.

Index Terms— CelestLab**,** Spaceflight Dynamics, Freeware, Scilab, Mission Analysis.

1. INTRODUCTION

Mission design requires tools with specific characteristics. There must exist reference (reliable) tools adapted to standard cases, but the tools also have to be flexible to fit particular needs. They have to be used for both recurrent studies and new and innovative ones, for quick analyses when immediate answers are required and for more complicated tasks, for which partial redeveloping is possible.

Meeting all these objectives is not simple.

For several years, a solution based on Scilab [7] has been developed at CNES for advanced studies (that is for projects in phase 0/A). Scilab is similar to Matlab, is free and open-source, and can be downloaded from http://www.scilab.org.

The low level of all the means used to solve mission analysis problems is called CelestLab. CelestLab is a Scilab library containing functions for SpaceFlight dynamics mission analysis. CelestLab is entirely coded in Scilab language. It is complemented with CelestLabX (CelestLab extension) which provides interfaces to existing software (written in other languages than Scilab).

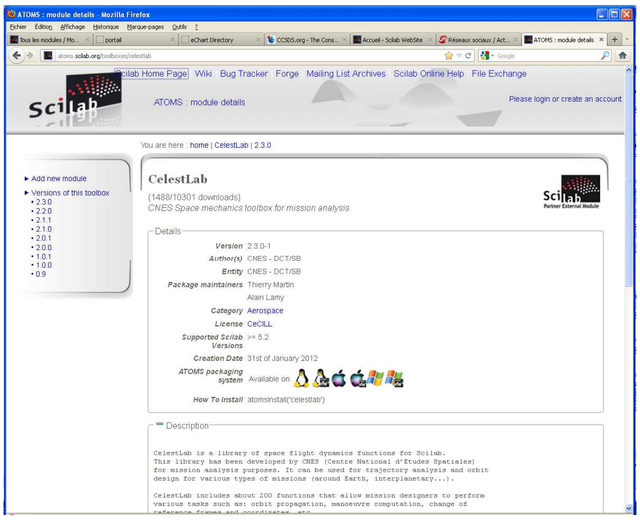
CelestLab can be downloaded from: http://atoms.scilab.org/toolboxes/celestlab.

The *Scilab* Space Mechanics Toolbox

2. CELESTLAB

2.1. Overview

CelestLab was made available to the Scilab community at the end of 2009. The extension module CelestLabX was created more recently, in mid 2015.


ATOMS : module details - Mozilla Firefox
Fichier Édition Affichage Historique Marque-pages Outils ?
Tous les modules / Mo… | portail | eChart Directory | CCSDS.org - The Cons… | Accueil - Scilab WebSite | Réseaux sociaux / Act… | ATOMS : module details
atoms.scilab.org/toolboxes/celestlab
Scilab Home Page | Wiki | Bug Tracker | Forge | Mailing List Archives | Scilab Online Help | File Exchange
ATOMS : module details
Please login or create an account
Add new module
Versions of this toolbox

You are here : home | CelestLab | 2.3.0

- 2.3.0
- 2.2.0
- 2.1.1
- 2.1.0
- 2.0.1
- 2.0.0
- 1.0.1
- 1.0.0
- 0.9

CelestLab

(1488/10301 downloads)

CNES Space mechanics toolbox for mission analysis

Details

Version	2.3.0-1
Author(s)	CNES - DCT/SB
Entity	CNES - DCT/SB
Package maintainers	Thierry Martin
	Alain Lamy
Category	Aerospace
License	CeCILL
Supported Scilab Versions	>= 5.2
Creation Date	31st of January 2012
ATOMS packaging system	Available on
How To Install	atomsInstall('celestlab')

Description

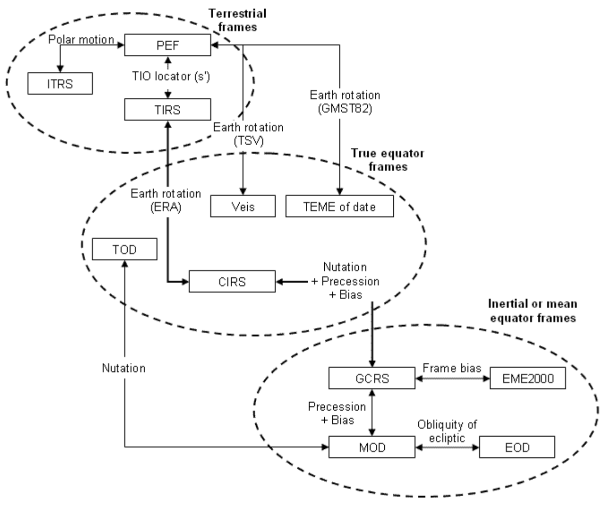
```
CelestLab is a library of space flight dynamics functions for Scilab.
This library has been developed by CNES (Centre National d'Etudes Spatiales)
for mission analysis purposes. It can be used for trajectory analysis and orbit
design for various types of missions (around Earth, interplanetary...).

CelestLab includes about 200 functions that allow mission designers to perform
various tasks such as: orbit propagation, manoeuvre computation, change of
```

CelestLab now contains about 250 functions (that is public interfaces). They are arranged into several categories as shown below:

Topics	Contents
Coordinates and Frames	Dates Reference frames definition and transformation (IERS 2010 conventions), Conversion between of reference systems, Definition of orbital element, Rotations and quaternions
Geometry and Events	Orbital events, Orbital geometry
Interplanetary	Interplanetary transfers, Three-body analysis
Models	Celestial body ephemerides (including DE405), density models, force models…
Orbit properties	Definition of most common orbit properties (sun synchronicity, repeat orbits, frozen orbits…)
Relative motion	Clohessy-Wiltshire formalism
Trajectory and manoeuvres	Orbit propagation (analytical models), Manoeuvre computation, TLE computation, Orbit propagation using STELA
Utilities, Math	Various support functions including for graphics

In order to give a more accurate idea, here are for instance the reference frames defined in CelestLab:



In CelestLab, a few conventions make life easier:

- A reference frame for every day uses is defined. It is called ECI and is identical to CIRF by default.
- A default time scale is also defined, called TREF. It is similar to TAI (that is to say, it is continuous) and is identical to UTC now.
- Many functions have default arguments, in particular for physical constants, which don't have to be specified for typical studies of Earth orbits.

But CelestLab is not just a set of functions. It also contains:

- Classical flight dynamics constants (Earth radius, gravitational constant…),

- Data, for instance IERS series for efficient frame transformation computations, moon, planet ephemerides (DE405), Earth maps…,

- Lots of descriptions and examples: each function comes with one or more example than can easily be copied and tailored according to the user's needs,

- Tutorials: an introductive part gives explanations about the major topics dealt with in CelestLab (reference frames, time scales, force models…), including some useful information like the angular difference between ICRF and EME2000…,

- A "cookbook" section which gives longer examples about particular subjects: frame transformation matrices and quaternions, numerical integration of orbital motion, Jacobian matrices… that can also be easily adapted and reused for a particular analysis.

- Demos (about 100) that show typical uses of CelestLab and that are as many examples that can be used in every day analyses.

CelestLab, as it was originally created is 100% Scilab language, which makes it easy to install as there is no need for compilers for instance. But there are situations where this approach is limiting. This is the case in particular when we would like to use existing (open source) software through Scilab. There are at least 2 main advantages to using existing software: less developing and testing effort is required, and also, the features that are made available are guaranteed consistent with the original versions, considered as standards.

That's why CelestLabX (CelestLab extension module) has been created. It contains low level interfaces to available software. The last version of CelestLab/CelestLabX provides utilities related to TLEs (interface to C code for the propagation of Two-Line Elements from: http://celestrak.com/software/vallado-sw.asp) and to STELA (Semi-analytic Tool for End of Life Analysis, tool used at CNES and elsewhere for orbit long-term propagation in particular, see: https://logiciels.cnes.fr/content/stela?language=en).

To make it simple for users, all public interfaces are made available through CelestLab. CelestLabX only contains hidden interfaces that only CelestLab knows about.

2.2. Why make CelestLab freeware and open source ?

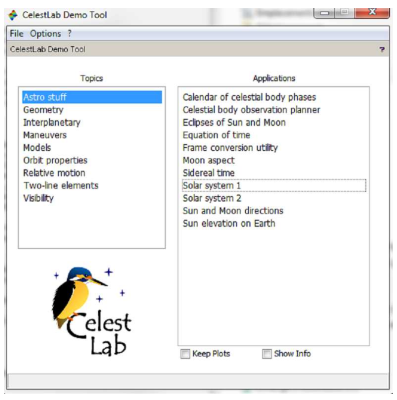
The answers are many. Here are a few:

- First what is made available through CelestLab is not subject to rights issues.
- Making a library open source is a way to share our practices, methods and conventions and make them more standard (and therefore easier to share). It also contributes to making exchanges inside the flight dynamics community easier: instead of data, pieces of code can be exchanged.
- A greater number of users increases the quality of the software.
- The library can also be used (and actually is) for the training of engineers and students in universities.

2.3. Contents by illustration

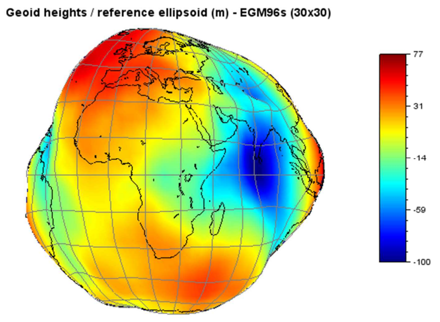
As describing a library is difficult, rather than long sentences and philosophical explanations, this part gives an illustrated overview based on concrete examples.

2.3.1. Demos

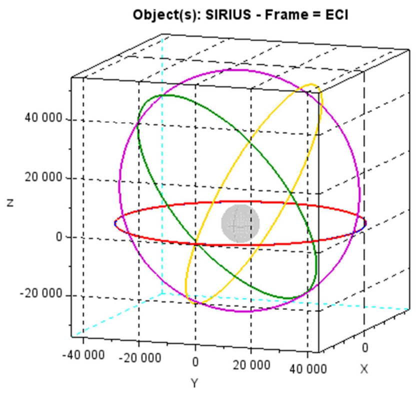


Here is a partial collection of what the demos (and consequently CelestLab) can do:

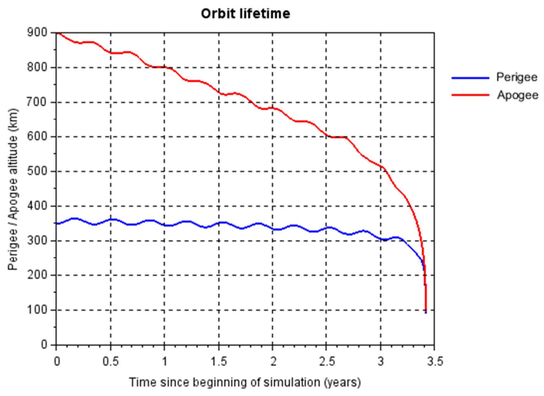
- Geoid computation computed using EGM96s (30x30):



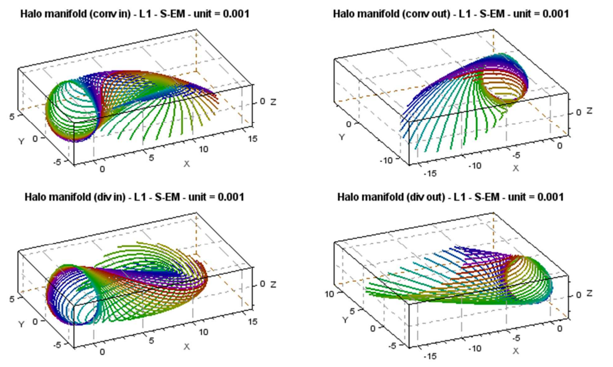
- Computations based on TLEs like the following plot of the "SIRIUS" satellites in an inertial frame:



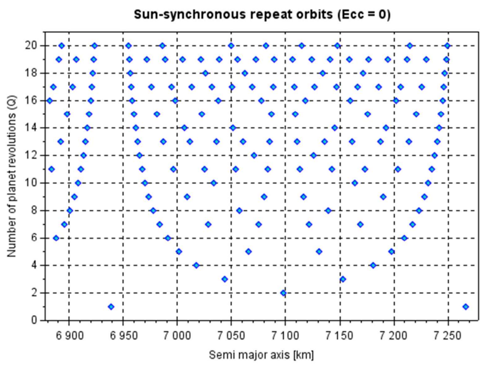
- Long-term propagation using STELA:



- Three-body computations :


Halo manifold (conv in) - L1 - S-EM - unit = 0.001
Halo manifold (conv out) - L1 - S-EM - unit = 0.001
Halo manifold (div in) - L1 - S-EM - unit = 0.001
Halo manifold (div out) - L1 - S-EM - unit = 0.001

- And many other things useful in everyday analyses such as the following plot of sun-synchronous repeat orbits.



2.3.2. Scripts examples

In order to make it even more concrete, it's probably worthwhile to show some lines of codes.

Example 1:

The first example enables the computation of the locations where a satellite (at a given distance from Earth center) is visible from a ground station). This example is directly taken from the help pages.

```
station = [10 * %pi/180; 45 * %pi/180; 0];
azim = linspace(0, 2*%pi, 100);
rsat = 8000.e3;

// Elevation = f(azimuth)
elev = 10 * (%pi/180) * (1+sin(4*azim));
pos_sat = CL_gm_stationVisiLocus(station, azim,
elev, rsat);

scf();
CL_plot_earthMap(color_id=color("grey60"));
CL_plot_ephem(pos_sat, color_id=2);
```

Example 2:

This second example illustrates how STELA can be used through CelestLab. Orbital elements and propagation model are defined (only default values including for the A/M ratio are used here). The orbit is propagated over 60 days, and the inclination is plotted. It's rather compact.

```
// Initial date (cjd, time scale: TREF)
cjd0 = 20000;

// Keplerian mean orbital elements (frame: ECI)
mean_kep0 = [7.e6; 1.e-3; 98*(%pi/180); %pi/2;
0; 0];

// Final dates
cjd = cjd0 + (0:60);

// STELA model parameters (default values)
params = CL_stela_params();

// Propagation
[mean_kep, info] = CL_stela_extrap("kep", cjd0,
mean_kep0, cjd, params, ["m", "i"]);

// Plot inclination (deg)
scf();
plot(cjd, mean_kep(3,:) * (180/%pi));
```

This examples requires CelestLabX to be installed, and STELA to be download and installed as well.

Example 3:

This last example shows some computation with TLEs.
A TLE is defined, parsed and propagated, from which point something else could be done.

```
str = [..
"1 00005U 58002B   00179.78495062  .00000023
00000-0  28098-4 0  4753"; ..
"2 00005  34.2682 348.7242 1859667 331.7664
19.3264 10.82419157413667" ];

tle = CL_tle_parse(str);

cjd = round(tle.epoch_cjd) :
round(tle.epoch_cjd)+5; // TREF

[pos, vel] = CL_tle_genEphem(tle, cjd) // ECI
```

3. CELESTLAB FOR MISSION DESIGN

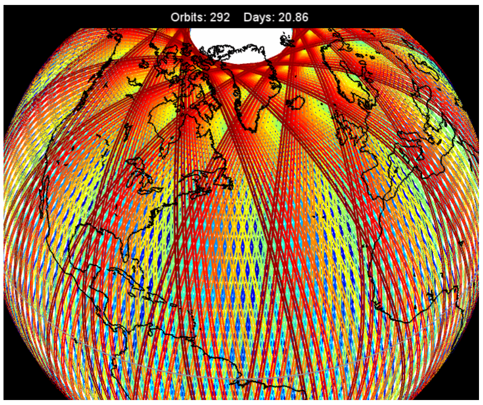
Of course, CelestLab is not only a tool showing demos and tutorials.

It is used every day at CNES for various mission analyses, from recurrent studies to more innovative ones.

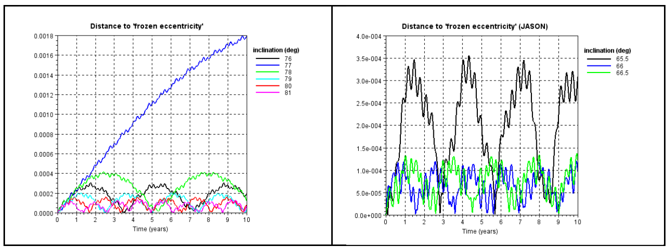
Examples that are given hereafter are taken from the mission analysis for the SWOT mission [4]. SWOT (Surface Water and Ocean Topography) is a joint CNES/NASA mission that extends the series of altimetry missions that are Topex-Poseidon, Jason 1/2/3.... CelestLab has been used for all the mission design until now (and SWOT project is now in phase B).

Of course the mission design includes lots of aspects and it's not possible to show them all. The illustrations that follow only give a quick overview.

Lots of studies concerned the definition of the orbit itself considering various kinds of constraints. It resulted in the repeat orbit as presently defined for which the 3D plot of the swath of the instrument is shown below. The plot was done using Scilab and CelestLab.



The following plots show some results of the perturbations study, more particularly the effect of resonance due to SRP that is function of inclination.
This resonance has in fact limited consequences on the mission.

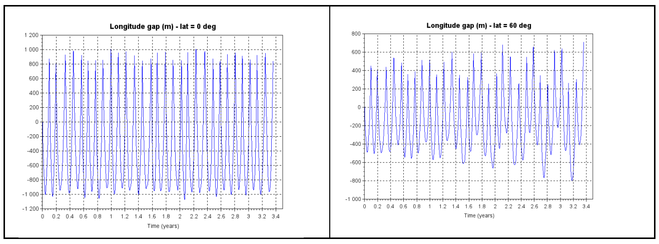


The propagation was done using STELA (CelestLab interface). The orbit is first numerically frozen, then propagated over 10 years. The propagation only takes a few seconds.

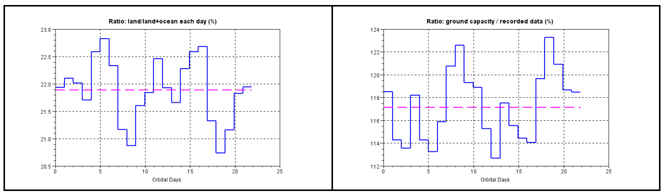
Other analyses related to perturbations were done to establish the DV budget and obtain an estimate of the station keeping maneuver frequency.

But it was not easy to prove that the requirement (keeping the ground track within ±1 km) would be met. A simulation of station keeping had to be performed.

The propagation was done this time again with STELA (thus taking into account all major perturbation effects) extended with an analytical model over less that ½ orbit for the computation of equator crossing longitudes.
It worked fine, and it could be showed that the requirements were met, even without any control of eccentricity (and despite the resonance effect mentioned earlier).



Another important aspect for the mission was the downlink capability: how well can all the measurements be downloaded to the ground (given hypotheses on downlink rates, instruments modes, recording rate…). Typical results are shown below.



Again, all the simulations were based on CelestLab. Comparison of results obtained independently by CNES and JPL showed a very good match.

4. QUALITY ASPECTS

As the library is used more and more by more and more people (even not considering people outside CNES), special attention has to be given to quality aspects. This section will only mention a few ones.

Of course the code is managed using a version control tool, and lots of tests are done to make sure CelestLab is reliable. Comparison with reference CNES libraries or other reference libraries/TOOLS such as SOFA have been extensively done.

All the tests (consisting of about 400 test files) can be run automatically. We also have tools that measure tests coverage. Coverage ratio is measured to be about 70%, which is not bad according to quality people, knowing that emphasis is given to accuracy and quality of numerical results.

Now, CelestLab is even considered as a reference tool for the validation of other software developed at CNES.

In fact, all these quality concerns have had impacts beyond CelestLab and mission analysis. An initiative at CNES resulted in extending language coding rules to Scilab (and Matlab). Scilab Enterprises was also involved and a rule verification tool will be included in the next version of Scilab.

5. CONCLUSION

Scilab and CelestLab have been used at CNES for flight dynamics mission design for a few years now. Lots of examples have been given in this paper. Another example is Rosetta/Philae for which Scilab/CelestLab was used operationally [3].

Scilab appears to be well adapted: recurrent studies are done much more efficiently than before. CelestLab is also widely used in the student community, for instance for the design of nano-satellites.

CelestLab evolves. New topics are included regularly, functions are improved, replaced, or transferred from other libraries. Thanks to Scilab's language interfacing capability (C, Fortran, Java), more and more features become available, through CelestLabX for instance.

The next steps for CelestLab are not completely decided.

One major aspect is to make it compatible with the future major Scilab version (Scilab 6), so that limitations that are sometimes encountered (such as memory limitation) will soon disappear.

We may also think of possible extensions:
- Star catalogs,
- Atmospheric density models (now in other Scilab libraries),
- Radiation belts models,
- …

These features will be included if there is a real need, and also time to do the work properly.

6. REFERENCES

[1] *Set of Tools for Space Flight Dynamics Mission Analysis*, A. Lamy & al, ScilabTec 2012.

[2] CelestLabX: CelestLab's extension module, A. Lamy & al, ScilabTec 2014

[3] Use of Scilab for the Philae landing on comet Churyumov-Gerasimenko, Thierry Martin & al, ScilabTec 2015

[4] Mission Design For The "SWOT" Mission, Alain Lamy, ISSFD 2014

[5] STELA software download site: https://logiciels.cnes.fr/content/stela?language=en

[6] CelestLab download site: https://atoms.scilab.org/toolboxes/celestlab

[7] Scilab web site: https://www.scilab.org