

Overview of CelestLab

Overview of CelestLab

Introduction to CelestLab



**The *Scilab*
Space
Mechanics
Toolbox**



CelestLab is a Scilab toolbox for Space Flight Dynamics. It has been developed by CNES-DCT/SB (Centre National d'Études Spatiales - French Space Agency) for mission analysis purposes.

CelestLab can be used for trajectory analysis and orbit design for various types of space missions. It allows engineers to perform tasks such as: orbit propagation, attitude computation, elementary manoeuvre computation, change of reference frame, change of coordinate systems, ...

The functions are arranged into several categories:

- **Coordinates and frames:** Definition of reference frames, time scales, types of coordinate systems,...
- **Geometry and events:** Computation of orbital geometry and orbital events (ground station visibility, ...)
- **Interplanetary:** Interplanetary and 3-body problem
- **Math:** Mathematical functions
- **Models:** Models for atmospheric density, Sun and Moon position,...
- **Orbit Properties:** Orbit properties (Sun synchronicity, repeating ground tracks, frozen orbits, drift of orbital elements, ...)
- **Relative motion:** Relative trajectory analysis using Clohessy Whiltshire formalism
- **Trajectory and maneuvers:** Orbit propagation and maneuver computation
- **Utilities:** Miscellaneous and multi-purpose functions

Installing and loading CelestLab

CelestLab can be installed in two different (and equivalent) ways:

- **by using ATOMS:**

Simply execute the command:

```
atomsInstall("celestlab");
```

- **manually:**

1) Unzip/copy the celestLab directory to a location of your choice. Let's call *CelestLabRoot* the new CelestLab path.

If the distribution only contains source files, create the library by executing the command:

```
exec(fullfile(CelestLabRoot, "builder.sce"));
```

2) Load CelestLab by executing the command:

```
exec(fullfile(CelestLabRoot, "loader.sce"));
```

(For convenience, you can add this latter line in the `scilab.ini` file.)

Loading CelestLab makes one (global) variable appear: `%CL__PRIV`. It is intended for CelestLab internal use only.

Using CelestLab

All CelestLab functions are directly useable after CelestLab has been loaded; no special action is required. You can browse the help files (in particular the "Introduction" section), and run some of the examples. You can also type "help *function*" (e.g. "help *CL_dat_convert*") if you know what you're looking for (this is not specific to CelestLab).

Another good way to start is to try the demos. Select "demos" in the CelestLab menu, then select one "topic" and one "application". Each demo corresponds to one single script that can be used as an example (although some of them are not that simple).

It is also possible to bring some useful (local) variables into existence (commonly used quantities or conversion factors, values of optional arguments) by just typing `CL_init()`.

Note:

- `CL_init` is only a convenience function. CelestLab is fully functional even if `CL_init` has not been called.
- A call to `CL_init` can be added to the `scilab.ini` file in order to make the variables exist each time celestLab is loaded.

CelestLab's extension module

Some functions in CelestLab require CelestLab's extension module to be installed. This is the case in particular for:

- all the features related to STELA (propagation model or area computation),
- some features related to TLEs (propagation, constants used).

If CelestLabX has not been installed, an error is raised by the corresponding functions.

CelestLabX can be downloaded from Scilab Atoms web page:
<http://atoms.scilab.org/toolboxes/celestlabx>.

More about CelestLab...

To get the latest version of CelestLab or post comments, go to:
<http://atoms.scilab.org/toolboxes/celestlab>.