ICATT 2016

CELESTLAB: SPACEFLIGHT DYNAMICS TOOLBOX FOR MISSION ANALYSIS

Alain Lamy, Thierry Martin, CNES
Alain.Lamy@cnes.fr

16 March 2016

Outline

- **Introduction: what tools for mission design ?**
- **Scilab**
- **CelestLab / CelestLabX, « our » toolbox for mission analysis: what is it used for ?**
- **Various illustrations**
  (Demos …)
- **Conclusion**

Introduction - what tools for mission design ?

What kind of tools do we need for mission design ?

=> Lots of studies of different kinds : recurrent, more advanced, quick evaluations and longer studies

- Not so easy to answer. Compromise between:
  - Flexibility / adaptability (adaptation of tools must be easy)
  - Robustness / stability (reference, reliable tools must be available)
  - Efficiency (simple problems / questions must be answered easily)
  - Consistency between tools

- Scilab appears to fullfill our needs

- Part of the solution is called

The *Scilab* Space Mechanics Toolbox

Scilab ?

- Scilab is free, open Source
  and easilly installable everywhere
- More information: http://www.scilab.org

Some assets:

- The language is simple enough
  Well adapted to engineers: flight dynamics and not programming
- Scilab comes with many functions / libraries
  Maths, graphics
- Links with other langages
  C, Fortran, Java

CelestLab

- Scilab flight dynamics library, pure scilab code
- Open source (same licence as Scilab)
  - ✦ Available on ATOMS web site since end of 2009. Number of downloads ~ 40000
- Contains ~250 functions (~20000 lines of code)
- Functions dealing with main flight dynamics aspects

  Coordinates & Frames, Trajectory and maneuvers, Orbit properties, Interplanetary, Geometry & events, Models, Utilities
- Thoroughly tested
  - ✦ Very few anomalies reported
- Lots of examples, doc, demos (~100), tutorial pages
  - ✦ Can be used as starting points (copy-paste)
  - ✦ Provides immediate answers to common/recurrent questions

=> http://atoms.scilab.org/toolboxes/celestlab

CelestLabX: CelestLab's extension

- CelestLabX
  - Provides interfaces to public (or maybe specific) tools and libraries
  - The additional features are made available through CelestLab (CelestLab then contains either pure Scilab code or calls to functions from CelestLabX)

- Reasons for this extension toolbox:
  - Open-source code / freeware libraries exist and we would like to use them through Scilab
    -> Saves (long) coding / testing time
    -> Useful features become easily available
  - Sometimes efficiency implies calling native code
  - Separate toolbox => keep CelestLab simple

CelestLabX: contents

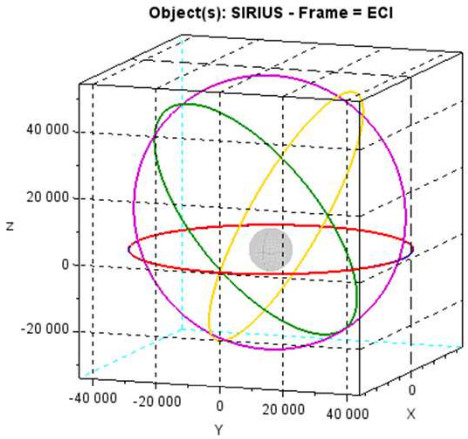
- STELA
  - CNES tool created in the context of satellite end of life regulations.
    Used for orbit long-term propagation (prediction of satellite positions over many years, usually up to 100)
  - Semi analytical propagation: propagation of « mean orbital elements » instead of true orbital elements => much less time consuming
  - Can be downloaded from **https://logiciels.cnes.fr/content/stela?language=en**
  - CelestLabX contains interface to STELA java code

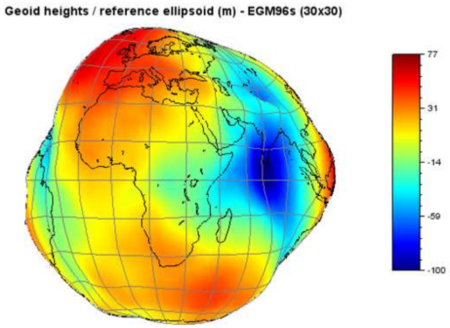
- Two Line Elements
  - Interface to C code from **http://celestrak.com/software/vallado-sw.asp**
  - Usual functions : propagation + various utilities (example later)

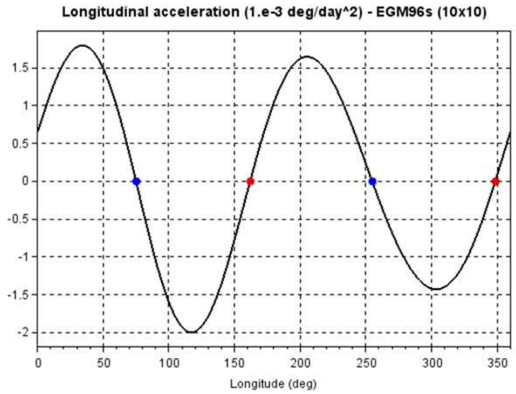
CelestLab : main topics

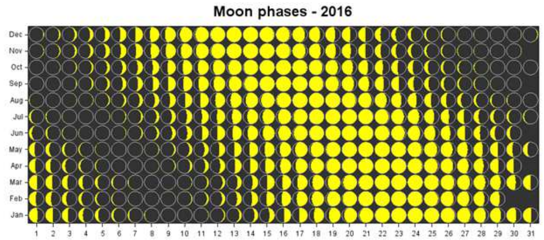
Topics	Examples
Coordinates and Frames	**Dates** **Reference frames** definition (IERS 2010 conventions), Conversion between of reference systems, Definition of **orbital elements**, **Rotations** and quaternions
Geometry and Events	Orbital **events**, Orbital **geometry**
Interplanetary	**Interplanetary** 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

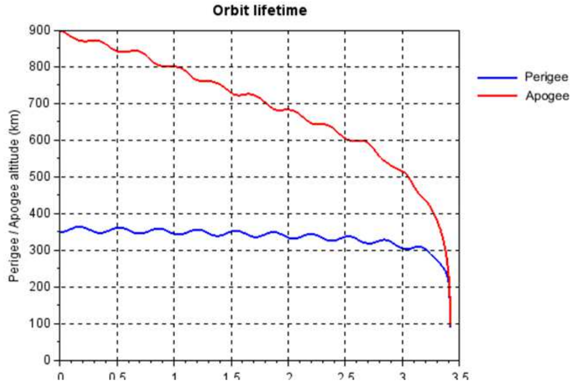
CelestLab: illustrations (demos)

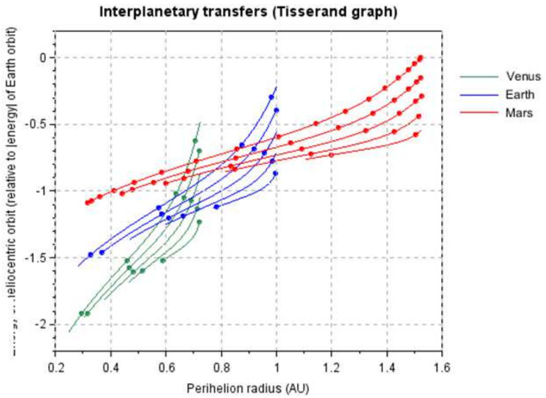


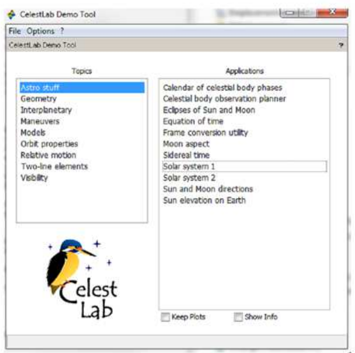


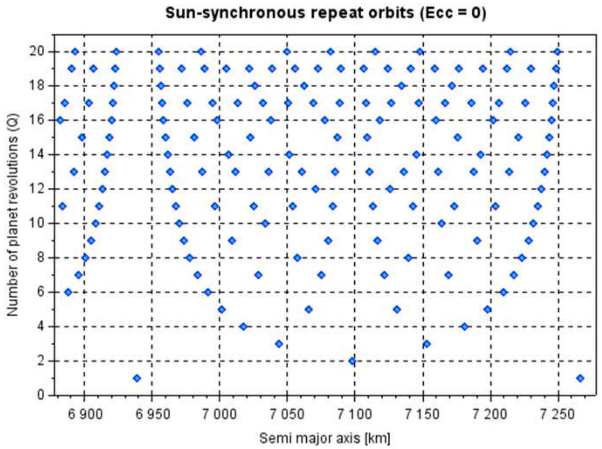


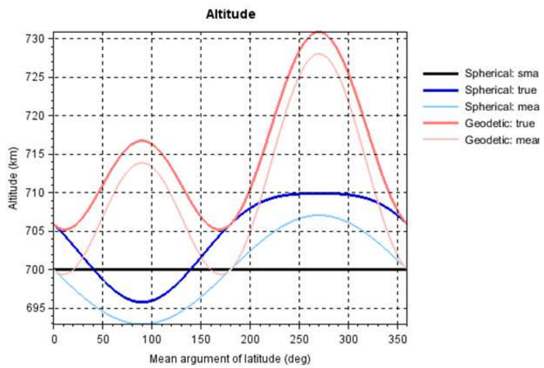












CelestLab Demo Tool

CelestLab: illustrations (help pages, tutorials)

Numerical integration of orbital motion

Numerical integration

Example1: Orbit around Earth (central force)

Here is a simple example of how to use the force functions (plus a few other ones) defined in Celestlab to integrate the motion.

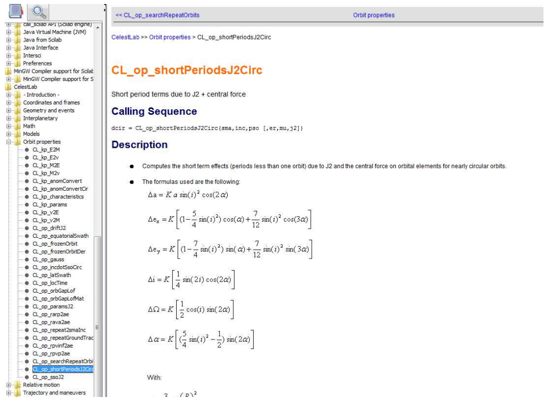
Only the central force is considered. The trajectory is compared with a purely Keplerian trajectory.

```
// -----------------------------------------------------
// Utilities
// -----------------------------------------------------
// State evolution: dy/dt = f(t, y).
// y = [pos; vel] (state vector)
// t in seconds
// model = central force
function [ydot]=fct(t, y)
  pos = y(1:3,:);
  vel = y(4:6,:);
  ydot = zeros(y);
  ydot(1:3,:) = vel;
  ydot(4:6,:) = CL_fo_centralAcc(pos);
endfunction

// -----------------------------------------------------
// Initializations, integration
// -----------------------------------------------------
// initial state
t0 = 0;
pos0 = [7000.e3; 0; 0];
vel0 = [0; 5.e3; 5.e3];
y0 = [pos0; vel0];

t = (0:180:10000); // time instants (seconds)

// integration
rtol = 1.e-12 * [1;1;1;1;1;1];
atol = 1.e-6 * [1;1;1;1.e-3;1.e-3;1.e-3];

y = ode(y0, t0, t, rtol, atol, fct);
pos = y(1:3,:);
vel = y(4:6,:);

// Comparison with Keplerian motion
kep0 = CL_oe_car2kep(pos0, vel0);
kep = CL_ex_kepler(0, kep0, t/86400);
[pos_k, vel_k] = CL_oe_kep2car(kep);

// -----------------------------------------------------
// Plots
// -----------------------------------------------------

scf();
plot(t/86400, CL_norm(pos-pos_k),"k");
```

CelestLab

- - Introduction -
  - Overview
    - Overview of CelestLab — Overview of CelestLab
    - Configuration — Configuration of CelestLab
    - CelestLab data — Overview of data available in CelestLab
  - Flight dynamics
    - Dates and time scales — Dates and time scales
    - Reference frames — Reference frames
    - Local frames — Local frames
    - Orbital elements — Description of orbital elements
    - Ephemerides for celestial bodies — Ephemerides for solar system bodies
    - Force models — Description of force models (acceleration and potential)
    - Orbit propagation models — Description of orbit propagation models available in CelestLab
    - STELA propagation model — Description of STELA (long-term) orbit propagation model
    - Two Line Elements — Two-Line Elements in CelestLab
  - Usage and conventions
    - Functions — Functions
    - Data types — Data types used in CelestLab
    - Flight dynamics conventions — Flight dynamics conventions
  - Cookbook
    - Frame conversions — Frame conversions
    - Numerical integration of orbital motion — Numerical integration
    - Rotations — Use of quaternions and frame transformation matrices
    - Simple orbital simulation — Calculation of classical orbital characteristics and
    - Interplanetary cruise — Interplanetary trajectory calculation
    - Jacobian and covariance matrices — Jacobian and covariance matrices
- Coordinates and frames
  - CL_co_car2ell — Cartesian coordinates to elliptical coordinates

```
// ==============================================================
// Definition of leap seconds (TAI - UTC)
//
// Syntax :
// data = [ tb te nbsec
//           ...           ]
//
// tb: beginning of interval (julian days from 1950.0 0h UTC)
// te: end of interval (julian days from 1950.0 0h UTC)
// nbsec: number of leap seconds between UTC and TAI (s)
//       = time of an event in the TAI time scale minus
//         time of the same event in the UTC time scale
//
// (scilab syntax)
//
// Note:
//
//      day D, 23h, 59mn      day D+1, 0h, 0mn
//           60 sec utc            00 sec utc
//           60+N sec tai          00+(N+1) sec tai
//  -----------|--------------------|---------------> utc
//  <------------------ N ----------[                 tai-utc
//                                  [------ N+1 ---->
//
//                                  ^
//                                  tb
//
// ==============================================================

data.TAI_UTC_DATA = [-%inf,  8035,  00.0
                      8035,  8217,  10.0
                      8217,  8401,  11.0
                      8401,  8766,  12.0
                      8766,  9131,  13.0
                      9131,  9496,  14.0
                      9496,  9862,  15.0
                      9862, 10227,  16.0
                     10227, 10592,  17.0
                     10592, 10957,  18.0
                     10957, 11504,  19.0
                     11504, 11869,  20.0
                     11869, 12234,  21.0
                     12234, 12965,  22.0
                     12965, 13879,  23.0
                     13879, 14610,  24.0
```



CL_man_dvBiElliptic

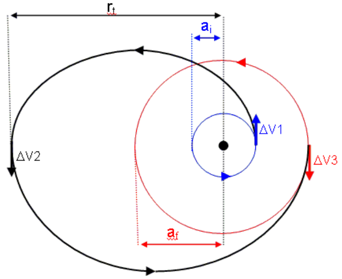
Bi-elliptic transfer

Calling Sequence

```
[deltav,dv1,dv2,dv3,anv1,anv2,anv3] = CL_man_dvBiElliptic(ai,af,rt [,mu])
man = CL_man_dvBiElliptic(ai,af,rt [,mu], res="s")
```

Description

- Computes the maneuvers of a bi-elliptical transfer from a circular orbit with semi-major axis **ai** to a circular orbit with semi-major axis **af**.
  The apogee radius of the elliptical transfer orbit is **rt**.
  deltav is the sum of the norms of the velocity increments.
  Velocity increments are expressed in cartesian coordinates in the "qsw" local frame.
  If the argument **res** is present and is equal to "s", all the output data are returned in a structure.



Parameters

ai :
: Semi-major axis of initial circular orbit [m] (1xN or 1x1)

af :
: Semi-major axis of final circular orbit [m] (1xN or 1x1)

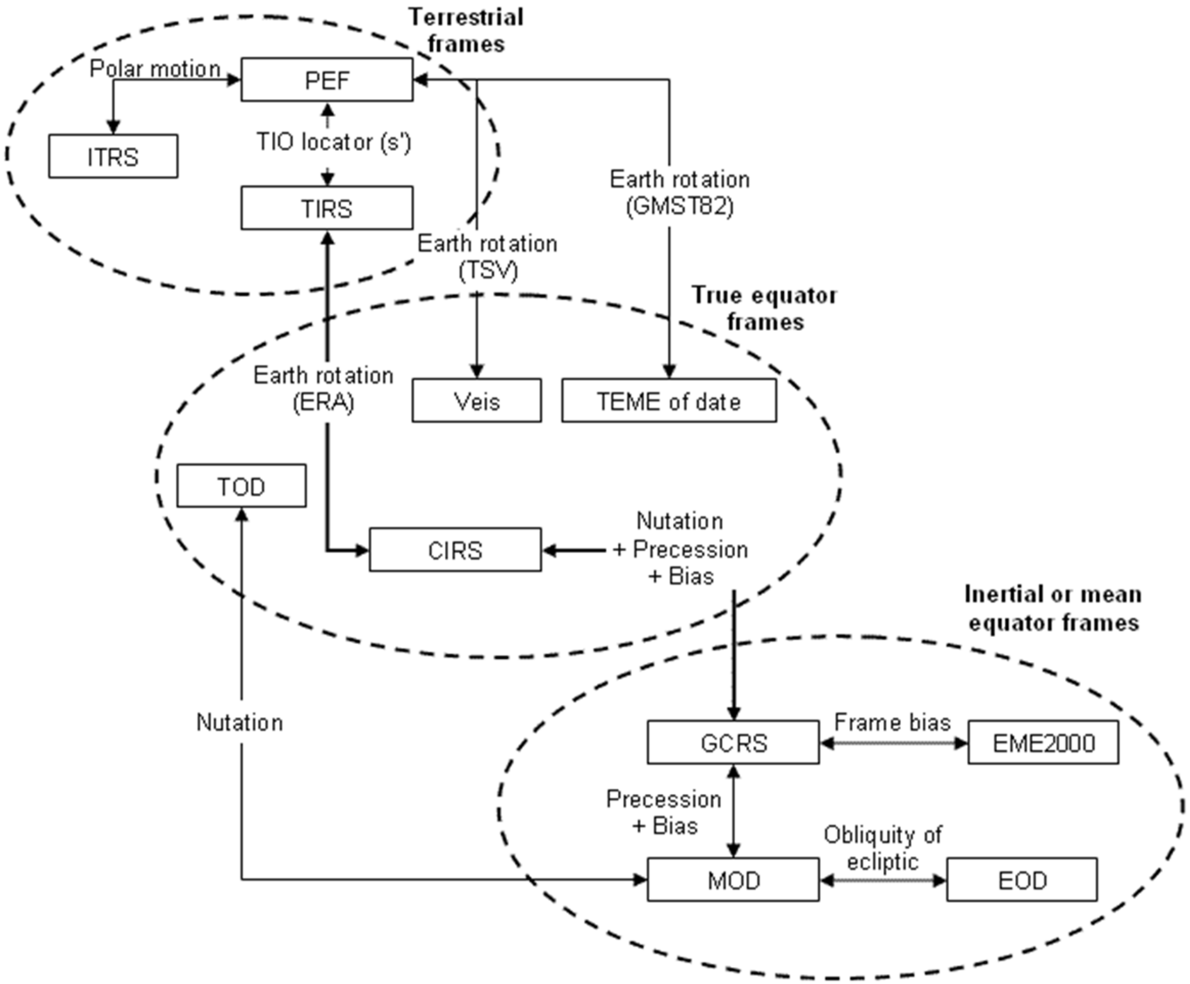
rt :
: Radius at the position of the second maneuver [m] (1xN or 1x1)

mu :
: (optional) Gravitational constant [m^3/s^2] (default value is %CL_mu)

res :
: (string, optional) Type of output: "d" or "s" for . Default is "d".

cnes

CelestLab: Reference frames



Reference frames in CelestLab : IERS 2010

CelestLab: what we do with it

- Initially created for phases 0/A
  - Orbit definition, study of perturbations…
  - DV budget, simulation of mission performance…

- Now used for all phases and even operations:

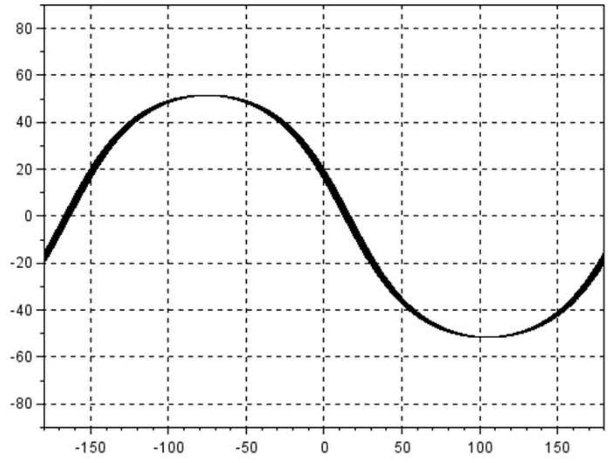
  - Mission analysis for SWOT mission (now: phase B), Rosetta/Philae, …
  - Miscelaneous studies: orbit prediction, taylor algebra, …

- Building blocks for higher level tools / libraries

- Exchange with specialists of other domains (ex: for RF interference studies)

Use of TLEs through CelestLab

```
// Define TLE (ISS)
str = [ ..
"1 25544U 98067A   16069.57438447  .00010945  00000-0  17329-3 0  9998"; ..
"2 25544  51.6422 194.8005 0001583 253.9754 212.9290 15.53974450989487"];

// String to TLE structure
tle = CL_tle_parse(str);

// Propagation dates (=> TREF = UTC)
cjd0 = CL_dat_cal2cjd(2016, 3, 9, 1, 0, 0.0);
cjd = cjd0 + (0 : 60 : 86400) / 86400;

// Propagation (=> frame = ECI = CIRF)
[pos, vel] = CL_tle_genEphem(tle, cjd);

// Plot inertial trajectory
scf();
CL_plot_ephem(pos);
```



Why make CelestLab open-source ?

- No rights issues in CelestLab

- Make things standard

  => A way to share our methods, conventions

- Make exchanges inside the flight dynamics community easier
  - Instead of data => exchange of lines of code
  - Description of methods or algorithms => answer can be « see CelestLab »
  - Can be used for training of engineers / students in the flight dynamics domain

- Contributions
  - Detections of errors is more efficient if there are many users
  - Users may have ideas for extensions or create useful tools based on CelestLab (that we would not have time to develop)

Quality aspects

- Version control tool

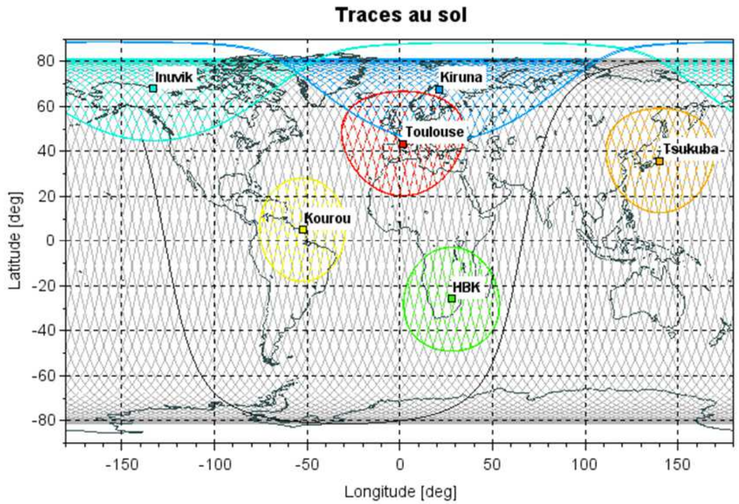
- Tests
  - ~400 test files
  - Comparison with reference tools / libraries

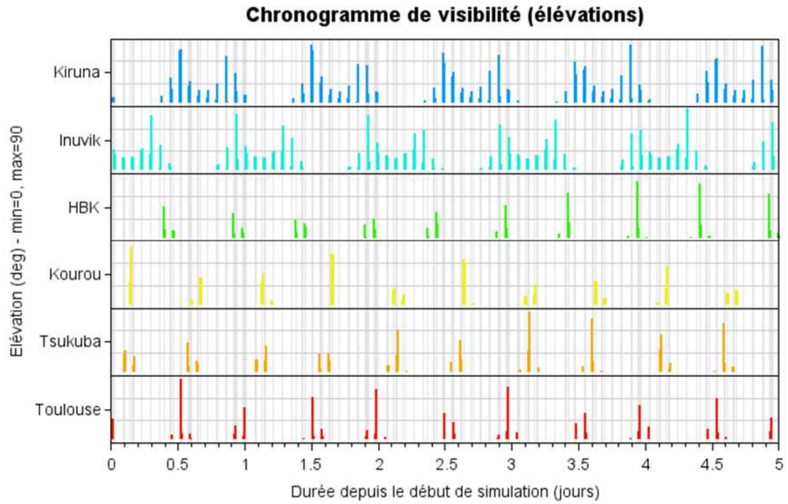
- Coverage
  - Coverage of automatic tests => 70%

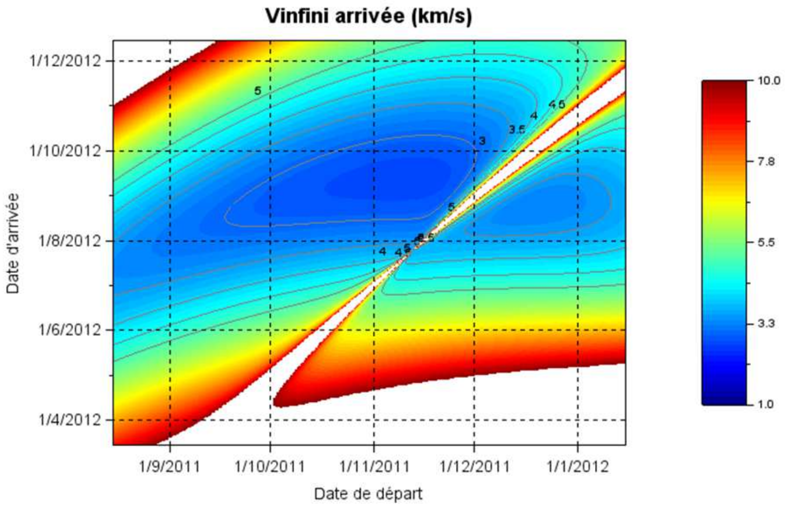
- Initiative beyond CelestLab and Mission analysis
  - Extension of CNES language coding rules document to Scilab
  - Scilab enterprises involved => verification tool included in the next version of Scilab

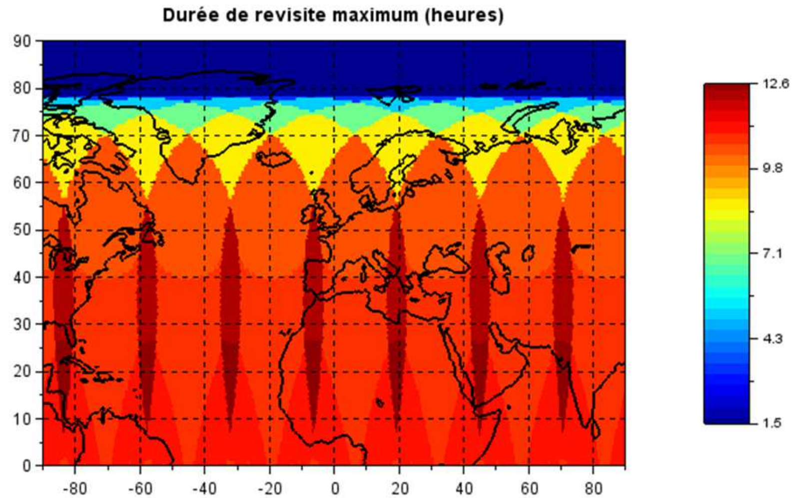
cnes

Other applications based on CelestLab









cnes

Summary / conclusion

- Scilab has been used extensively at CNES for flight dynamics mission analysis for a few years now. 1st CelestLab version: end of 2009.
- Useful features are available through CelestLabX: TLEs, STELA (interface to C or Java code).
- Scilab appears to be well adapted: mission analyses are now done much more efficiently.
- Used even outside flight dynamics domain.
- CelestLab evolves: functions are continuously improved, new features are added as needed…

For more information...

- For more information on CelestLab:
  http://atoms.scilab.org/toolboxes/celestlab