

MULTI PURPOSE MISSION ANALYSIS DEVELOPMENT FRAMEWORK

MUPUMA

Felipe Jiménez ⁽¹⁾, Francisco Javier Atapuerca ⁽²⁾, José María de Juana ⁽³⁾

⁽¹⁾ GMV AD., Isaac Newton 11, 28760 Tres Cantos, Spain, e-mail: fjimenez@gmv.com

⁽²⁾ GMV AD., Isaac Newton 11, 28760 Tres Cantos, Spain, e-mail: fjatapuerca@gmv.com

⁽³⁾ EUMETSAT, Eumetsat Allee 1, D-64295 Darmstadt, Germany, e-mail: Jose.DeJuana@eumetsat.int

ABSTRACT

Multi Purpose Mission Analysis development framework (*mupuma*) is an advanced and generic development framework tool for Mission Analysis that covers LEO, MEO, GEO, GTO and LEOP missions and includes a wide range of Flight Dynamics and Mission Analysis functions.

The aim of this tool is not just to perform a fixed set of predefined computations and plots but to permit the user to dynamically and easily create any possible combination of outputs required.

mupuma has been developed by the Flight Dynamics and Mission Analysis teams at GMV, under a study contracted by EUMETSAT to investigate new modern and easily accessible technologies for the rapid development of generic Mission Analysis tasks.

1. INTRODUCTION

Mupuma shall be understood not only as a Mission Analysis tool that allows the user to compute a certain number of Mission Analysis computations, but as a flexible framework that permits the development of new functionalities in a very easy way.

Some of the main unique features of this new Mission Analysis tool and framework are:

- Supporting of any kind of satellite mission, including LEO, MEO, GEO, GTO and LEOP and covering orbit, attitude and manoeuvre aspects.
- Usage of NASA Open Source tool GMAT as propagation engine, together with a simplified analytical propagator developed by GMV.
- A complete Flight Dynamics and Mission Analysis library developed by GMV is available through the tool. It covers basic astrodynamics computations such as date and reference frame conversions, interface with the propagation functions, analytical Sun ephemeris computation, station coverage

events, memory, power and ageing computations, graphic and plots generation etc.

- Powerful and flexible 2D and 3D plotting capabilities.
- Current development frameworks, as well as further developments are written in standard Matlab scripting language without special packages or toolboxes required.
- Portable to most commonly used hardware and operating systems.

This paper describes in detail the main capabilities of *mupuma* and demonstrates some of the scenarios and missions that it can easily cover with a reduced effort of customization by the user.

2. ARCHITECTURE

mupuma Mission Analysis tool architecture is based on the development of a set of basic Flight Dynamic System and Mission Analysis libraries, easily extensible and easy to be used. Together with these libraries, an interface with NASA's General Mission Analysis Tool (GMAT) has been implemented, with the purpose of using GMAT propagator as the propagation engine of the tool.

Then, any needed Mission Analysis task can be easily developed as a Matlab module. The Matlab module calls GMAT propagator by means of the implemented interface between *mupuma* and GMAT.

Finally GMAT makes the requested orbit propagations and then, Matlab module developed for the task will post-process the propagation information by using the *mupuma* FDS and Mission Analysis libraries.

Here below it can be found a figure representing the way *mupuma* architecture works:

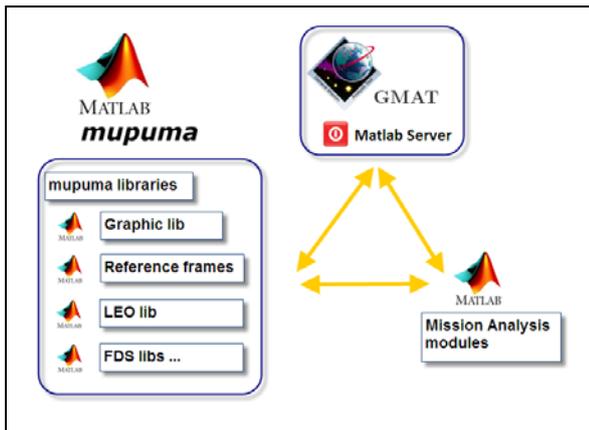


Figure 1: *mupuma* architectural design

3. DEVELOPMENT PROCESS

It is quite interesting to remark how the development process took place for the different functionalities that are currently available at *mupuma* and that will be briefly detailed in next sections.

It shall be noticed that most of the effort was spent in the development of the general architecture of the tool, i.e. in the generic *mupuma* libraries and the tool configuration aspects and the interface with GMAT.

The development of specific Mission Analysis tasks was completed in a short period of time. This is one of the key and strong points of the tool: The easiness of development of new tasks due to the flexible and powerful Flight Dynamics and Mission Analysis library available through the tool, together with the possibility of using a strong propagation engine such as GMAT by means of a simple and stable interface. Also the fact of using Matlab as coding language, facilitates the development process due to its power and also because it is widely used by the scientific and space community.

4. INTERFACE WITH NASA'S GENERAL MISSION ANALYSIS TOOL (GMAT)

One of the main advantages of *mupuma* Mission Analysis tool is the possibility of using NASA'S GMAT General Mission Analysis Tool as propagation engine. GMAT uses a numerical propagator that results from the combination of an integrator and a force model. A large variety of numerical integrators and physical models can be selected following the user needs, and therefore any kind of Mission can be supported by the propagator provided in GMAT.

The following integrators are available in GMAT to perform spacecraft propagation computations:

- RungeKutta89
- RungeKutta68
- RungeKutta56
- PrinceDormand45
- PrinceDormand78
- BulirschStoer
- AdamsBashforthMoulton

The following force models are also supported by GMAT propagator:

- Point mass and non-spherical gravity
- Earth atmospheric drag
- Solar radiation pressure

All the details about these integrators and force models used in GMAT propagation are detailed in GMAT Mathematical Specifications [1].

Through an interface implemented with GMAT, *mupuma* Mission Analysis tool is able to ask for any needed propagation data for one or several satellites of any kind of orbit (LEO, MEO, GEO, GTO, etc.). Then, *mupuma* functions will post-process the propagation information, computing the Mission Analysis products that the user may require.

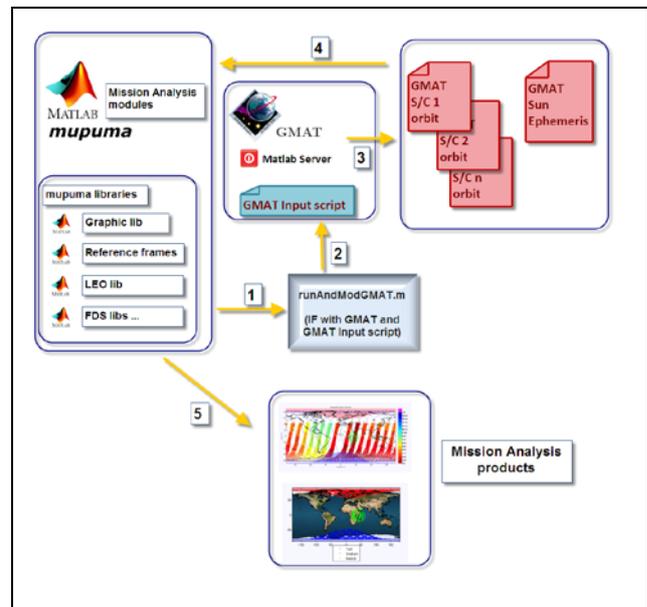


Figure 2: *mupuma* Interface with GMAT

As it can be observed in the figure above, *mupuma*

module can modify and define as needed the inputs of the propagation in GMAT as well as defining the propagation outputs to be generated by GMAT application. This is done by means of a Matlab routine which modifies the GMAT input script with the input and output options desired for the propagation. The script also starts the GMAT execution. Then, GMAT generates a series of output report files with the results of the propagation. This output reports will be finally post-processed by GMAT module and libraries for the computation of the Mission Analysis products.

The interface between *mupuma* and GMAT can be used in two different ways. The first one that is called Matlab to GMAT, in which the program manager is a *mupuma* Matlab function. Firstly program inputs are defined in the Matlab function. Then GMAT propagator is called by means of the corresponding interface function. Finally Matlab post-processes the propagation output reports generated by GMAT.

The second approach (GMAT to Matlab) consists of using GMAT and its Man Machine Interface (MMI) as the program manager where all the different functions and input variables are defined and called. For instance, GMAT can firstly perform a requested propagation, and then it can pass this propagation information by means of the arguments of a *mupuma* Matlab function that GMAT calls from its MMI, and so on. This approach permits to use the user friendly GMAT MMI to define all the program inputs and also to run the program itself.

Here below it is shown how the GMAT MMI looks like:

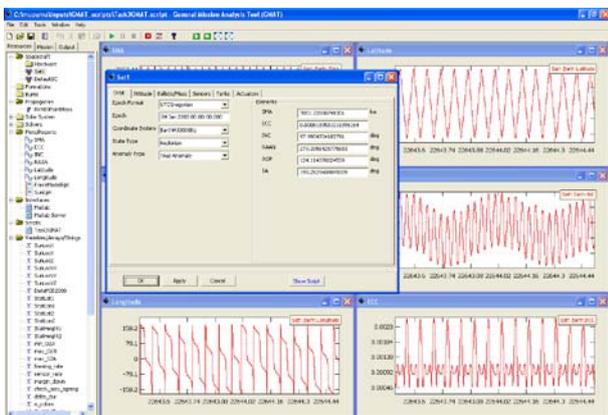


Figure 3: GMAT Man Machine Interface

This approach permits to combine the extremely powerful and versatile GMAT propagator with the flexibility and easy to use libraries developed in *mupuma* for computing Mission Analysis tasks.

5. PLOTTING CAPABILITIES

mupuma contains an extremely powerful and flexible graphical library that easily allows to create 2D and 3D plots and even animations, including satellite orbits, ground tracks, station coverage circles, satellite sensor coverages etc.

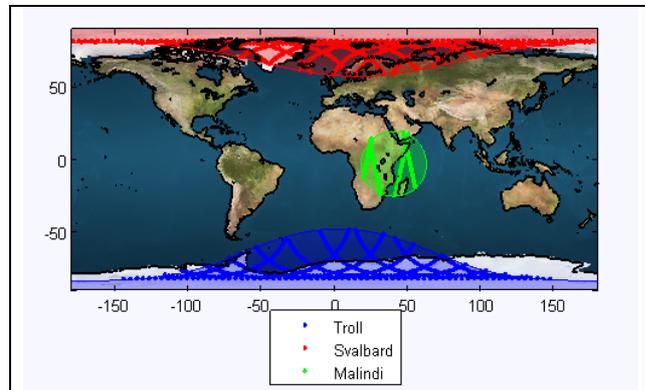


Figure 4: Station coverage 2D plot

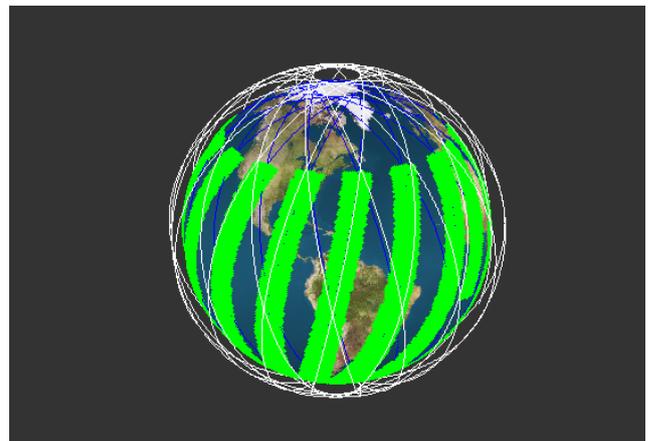


Figure 5: 3D plot of satellite ground track and observation instrument coverage

The way this graphic library has been designed is in order to create the desired plot just within just a few Matlab sentences. The following plotting functions are available through this graphic library amongst others:

- A function to create a two dimensional cylindrical projection of the Earth in order to represents ground track plots and coverage on top of it.
- A function to plot a satellite sensor coverage swath in a two dimensional cylindrical representation of the Earth.
- A function that plots the ground track for a given satellite orbit. This ground track is plotted on top of a two dimensional cylindrical representation of the Earth.

- A function that creates a three dimensional representation of the Earth useful to represent ground track plots and coverage.
- A function that plots a satellite orbit over a three dimensional representation of the Earth.
- A function completely analogous to the one plotting coverage swath in two dimensions but in three dimensions.
- A function that plots the satellite ground track for a given satellite orbit. This ground track is plotted on top of a three dimensional representation of the Earth.
- A function that plots in two dimensions, the station coverage for a given satellite. This coverage plot is displayed on top of a two dimensional cylindrical representation of the Earth.
- A function that plots the station visibility circle for a given altitude.
- A function to plot in two dimensions, the data ageing on-board an Earth observation satellite. This ageing plot is displayed on top of a two dimensional cylindrical representation of the Earth.

As it has been mentioned before, it is even possible to create animations that plot in two and three dimensions the evolution in time of a certain Mission Analysis product. For instance, it can be seen in the figure below the evolution in time of an antenna footprint of a GEO satellite in inclined orbit (5 degrees):

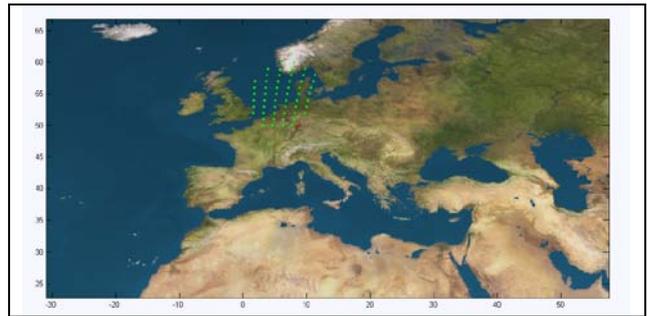
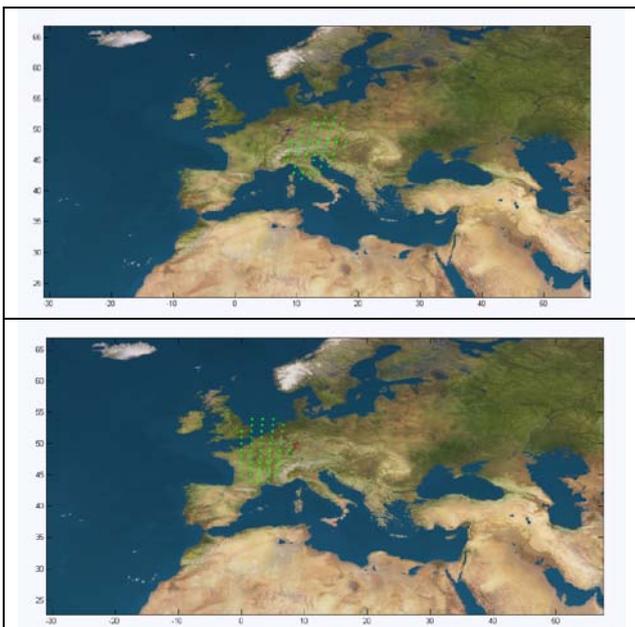


Figure 6: GEO satellite antenna footprint animation at three different parts of the orbit

6. LEO EARTH OBSERVATION SATELLITES COMPUTATIONS

Even if the general architecture of *mupuma* has been designed in order to cover the needs of any kind of satellite mission, current available libraries cover most of the computations and plotting needs of LEO satellite missions.

For instance, the following LEO computations and plots capabilities are covered by *mupuma* libraries:

6.1. GRID COVERAGE OF OBSERVATION SENSORS

This library includes the set of functions that constitute the core of the LEO Mission Analysis tool computations. This bunch of functions allow to compute basic grid computations such as grid coverage and Ground Station contacts that are the basis of any subsequent computation that may be needed specially for Earth Observation satellites.

Here below it can be seen a plot of an observation instrument coverage computed by means of this library and the graphical engine of *mupuma*:

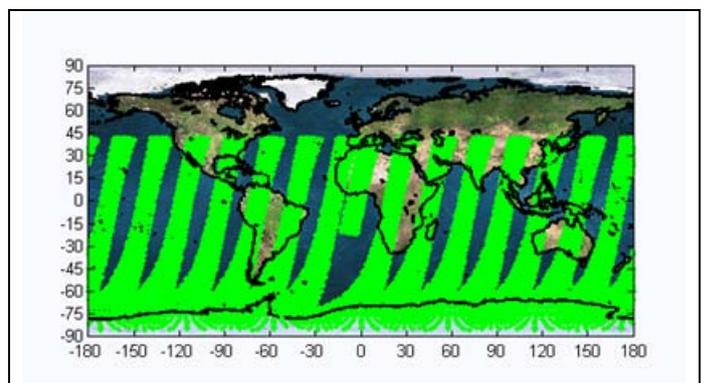


Figure 7: Earth observation sensor coverage

6.2. ON-BOARD MEMORY AGEING

This library includes a set of functions that perform complex computations that make use of previously computed grid products, such as earth coverage or station coverage. For the time being this library computes products related to on-board memory evolution and on board data ageing of an Earth observation satellite, as well as satellite and instruments power evolution. The idea is that this library could be updated and upgraded to cover any new need required by the user.

Here below it can be seen an on-board memory ageing plot computed with this library and using the *mupuma* graphical library:

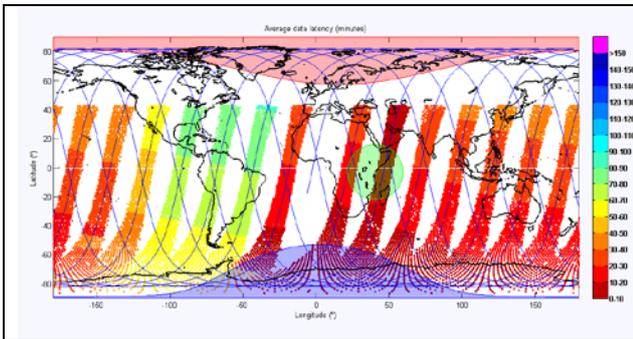


Figure 8: On-board Memory ageing

7. GEO SATELLITES COMPUTATIONS

As it has been remarked on previous section, not only LEO Missions Analysis tasks can be easily covered by *mupuma* but for any kind of mission. This is the case of GEO satellites.

For instance, several plots that are very often used in GEO operations have been implemented in just a few lines of Matlab code:

- Eccentricity plots:

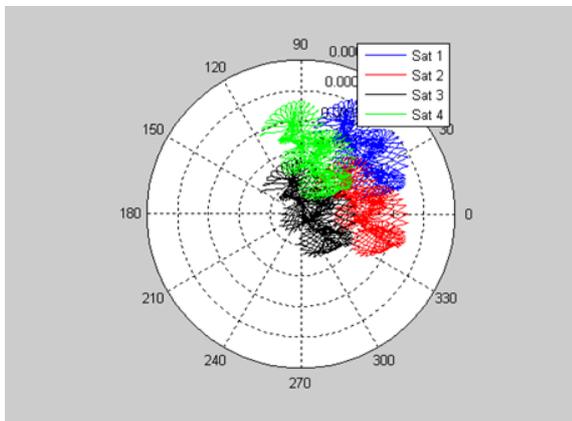


Figure 9: Eccentricity vector for a 4 satellite cluster

- Inclination plots:

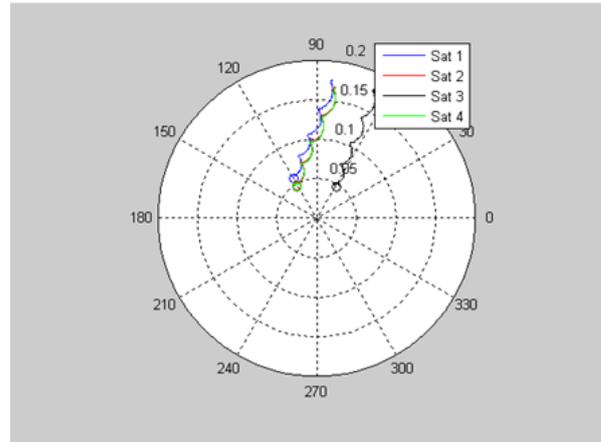


Figure 10: Inclination vector for a 4 satellite cluster

- Longitude plots:

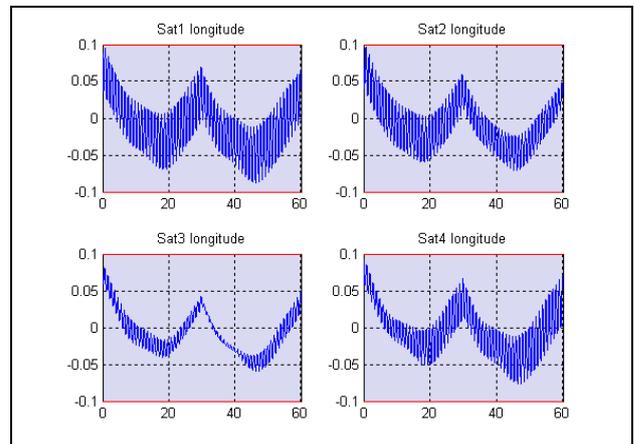


Figure 11: Longitude evolution plots of the cluster

- Inter-satellite distance plots:

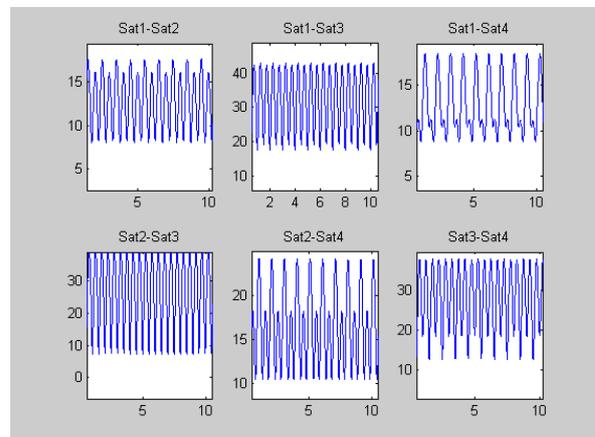


Figure 12: Cluster inter-satellite distances plots

8. LEOP MISSION ANALYSIS FOR GEO SATELLITES

One of the most demanded Mission Analysis tasks for GEO satellites is the analysis of the launch windows, i.e. the periods of time during which satellite can be launched, meeting all the different applicable constraints.

Launch windows constraints are different in each particular mission. For the time being, *mupuma* just contains a simple launch windows analysis tool in which very generic and simple constraints have been considered:

- The launch window is defined just by thermal constraints imposing a solar aspect angle between 70 and 110 degree (angle between sun and s/c attitude defined by its apogee velocity direction)
- Additional constraints specific to the mission requirements can be easily added by means of new encapsulated constraints functions.

Also some simplifications have been considered in order to reduce the computation time:

- The final target orbit it is supposed to be a GEO non-inclined circular orbit.
- The attitude of the spacecraft has been assumed to be aligned with the apogee velocity direction at all times during the LEOP.

Considering these generic constraints applicable to most of the GEO satellites LEOPs, two different launch windows have been computed. The first launch window is computed for a single impulsive manoeuvre taking place at the first apogee opportunity from the GTO orbit to the GEO one. The second launch window is computed for the last opportunity (e.g. at last apogee before 10 days).

Here below it is shown the plot generated by *mupuma* for the computation of the launch windows:

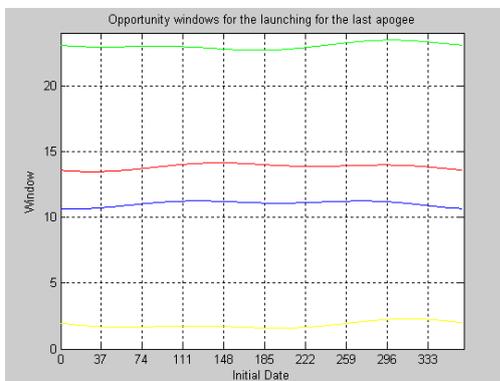


Figure 13: Launch windows representation

9. ANALYTICAL PROPAGATOR

In some cases when Mission Analysis computations are needed to be computed for a big period of time, numerical propagators can have performance problems and therefore computations can take too much time to be completed. This is the reason why a semi-analytical propagator has been developed in *mupuma*. With such propagator, non accurate propagations for big periods of time can be done in just a few minutes.

The implemented analytical propagator computes the averaged variation for the orbital elements influenced by the Earth's oblateness by using the Semi-Analytic Theory of J.J.F.Liu and R.L. Alford for the Motion of a Close-Earth Artificial Satellite [2]. It shall be noticed that only Earth gravitational effects up to J4 term are considered in this model. Atmospheric drag effect together with third bodies perturbations have been considered as second order perturbations and therefore have been omitted.

This propagation function can be used as a propagation stand alone module which generates dedicated products such as propagation plots and reports, computing both osculating and mean elements as it can be seen in the figures below.

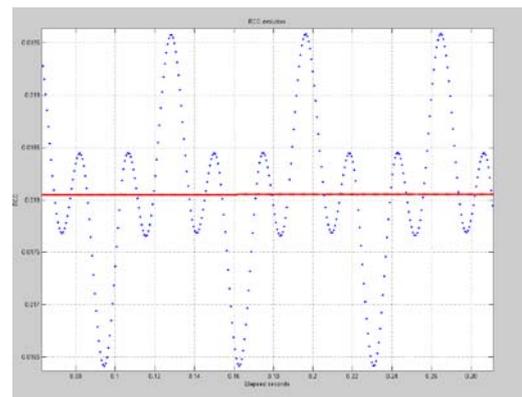


Figure 14: Mean and osculating eccentricity propagation

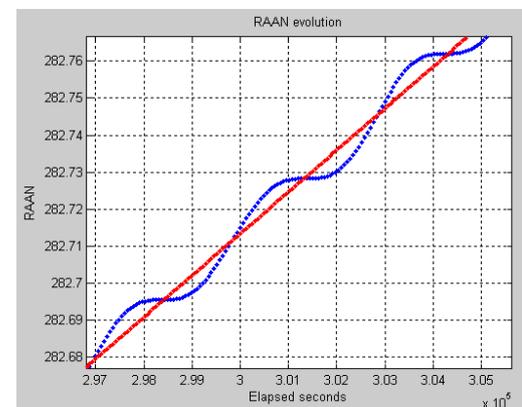


Figure 15: Mean and osculating RAAN propagation

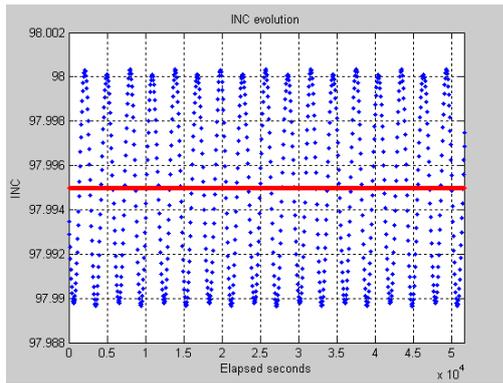


Figure 16: Mean and osculating Inclination propagation

This propagation functions can also be called by other modules in order to compute a needed propagation and then post-process the propagation information. Here below it can be seen how a ground track product was generated by post-processing the propagation information generated by *mupuma* analytical propagator.

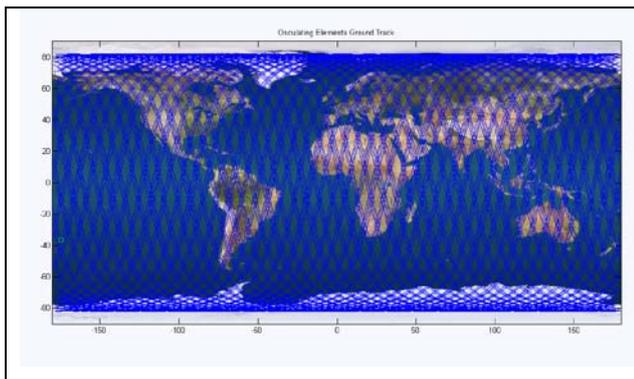


Figure 17: Analytical Ground Track plot

10. FUTURE EVOLUTION

Some of the future developments and enhancements foreseen for *mupuma* Mission Analysis tool are detailed here below:

- Development of new GEO related computations such as Station Keeping manoeuvre computations and long term Mission Analysis simulations.
- Development of a tracking and orbit determination evaluation tool based on *mupuma* solution. The idea is to evaluate the orbit determination performed by an external tool (the one to be evaluated) comparing its determined orbit and the tracking data used with respect to simulated tracking data computed by a new function based on ephemerides from the orbit

determination software or generated by *mupuma* depending on the type of analysis. The objective would be to analyze the orbit determination performance (biases, noises) or detect anomalies in the tracking used, to provide feedback to the original orbit determination process.

- Development of a propagation validation tool based on *mupuma* solution: The idea is to develop new functions in *mupuma* that will automatically compare GMAT propagation reports with other propagation products from different sources (such as ESA's NAPEOS propagation function, STK propagation outputs, external TLE orbits, CCSDS orbit formats (OPM, OEM, OMM ...), etc.)
- There may be additional enhancements opportunities by incorporating certain levels of standardization (like CCSDS formats but not necessarily limited to that) that would allow the easy conversion from non-standard formats and back. This would make *mupuma* get closer to more real operational environments.

11. REFERENCES

1. Steven P. Hughes, *General Mission Analysis Tool (GMAT) Mathematical Specifications*.
2. Liu, J. J. F.; Alford, R. L., *A semi-analytic theory for the motion of a close-earth artificial satellite with drag*. AIAA PAPER 79-0123