# EFFICIENT ALL VS. ALL COLLISION RISK ANALYSES

**Diego Escobar, Alberto Águeda, Luis Martín, Francisco M. Martínez**
*GMV, Isaac Newton 11, 28760 Tres Cantos, Spain, +34 91 807 21 00*
*email:* descobar@gmv.com, aagueda@gmv.com, lmpolo@gmv.com, fmartinez@gmv.com

***Abstract:*** *This paper describes the different changes implemented in a conjunction assessment and collision risk evaluation tool with the aim of reducing drastically the computational cost to ensure that a scenario where all space debris objects are analyzed against each other can be carried out in a short period of time. Improvements at algorithm level and parallelization techniques are used to shorten the time needed for the process of conjunction assessment. In the case of the collision risk evaluation, an approach for the propagation of the state covariance is presented based on the Simplified General Perturbations theory commonly used to propagate Two Line Elements.*

***Keywords:*** *space debris, collision risk, conjunction, parallelization, smart sieve*

## 1. Introduction

In recent years, the space debris environment has gained a lot of attention due to the increasing amount of uncontrolled man-made objects orbiting the Earth. This population poses a significant and constantly growing threat to operational satellites, as proven by the collision of the satellite Iridium-33 with the decommissioned spacecraft Cosmos-2251. Major space organizations have developed their own systems to assess the collision risk and evaluate the need to manoeuvre their satellites in order to avoid collision events with other orbiting objects (see Ref. [1], Ref. [2] and Ref. [3]). Commercial operators have also raised their concerns and demand collision risk assessment and mitigation tools and services.

In order to face this threat in an independent manner, ESA has launched an initiative for the development of a European Space Situational Awareness (SSA) System and all the precursor and preparatory activities. GMV participates in many of these activities ranging from the implementation of ESA´s tools for Collision Risk Assessment and Orbit Determination, CRASS (see Ref. [4] and Ref. [11]) and ODIN respectively, to the development of a catalogue maintenance simulator, SSASIM, and various other studies in the field of space debris tracking.

As part of the activities in this field, GMV has developed ***closeap***, a tool for:

- efficient conjunction assessment of the full USSTRATCOM catalogue population,
- collision probability prediction and alert issuing and
- collision avoidance manoeuvre computation

ESA´s NAPEOS (Navigation Package for Earth Orbiting Satellites) framework is recognized as one of the most reusable and accurate systems for space dynamics. This makes it the best choice as computational engine and numerical propagator for ***closeap***.

At the same time, ***closeap*** makes use of the same trajectory computation, conjunction assessment and collision risk algorithms implemented in CRASS. The orbits of the space debris are computed by propagating Two Line Elements (TLE) with the Simplified General Perturbations (SGP) theory (see Ref. [7]) and the trajectories of the payloads are read interpolating operationally computed ephemerides. On the other hand, the conjunction events are detected by means of a Smart Sieve technique (see Ref. [5]) consisting of a series of fast and robust filters whereas the collision probability is computed based on propagated state covariances with Alfriend's algorithm (see Ref.[6]). These algorithms are being used operationally by ESA to monitor the collision risk of satellites like ENVISAT and ERS-2 as described in Ref. [12].

In the field of collision risk assessment, the most computationally demanding scenario is the one in which all catalogued objects are analyzed against each other - all vs. all scenario - over a typical forecast time span of one week.

This paper describes the performance improvements implemented in ***closeap*** at algorithm level to ensure that the most time demanding scenarios can be analyzed in a reasonable amount of time with commercial-off-the-shelf hardware. Operational use and robustness are also key topics described in the paper.

The latest release of ***closeap*** incorporates several improvements to speed up the analysis of all vs. all scenarios. In the implementation of these improvements, two main principles have been applied:

- minimize the number of calls to the SGP propagator by benefiting from the symmetries of the problem (A vs. B is the same analysis as B vs. A with respect to conjunction detection) and by improving the numerical root finder of the fine conjunction detection process
- carry out as many computations as possible in the TLE reference frame, assumed to be TEME (True Equator Mean Equinox), to reduce the transformations from the TLE reference frame to the reference frames of other objects

The amount of space debris increases steadily due to the human activities. In fact, in 1990 there were around 7000 objects in space catalogued by USSTRATCOM and currently there are more than 14000 objects - a factor of 2 in 20 years. Moreover, objects of smaller size are likely to be catalogued, tracked and included in collision risk analyses in the future. Currently there are more than 600,000 objects larger than 1 cm in orbit according to the ESA MASTER-2005 model. And each collision, such as the Chinese anti-satellite weapon test in 2007, generates a considerable amount of new debris in space. Thus, the number of objects involved in a full collision assessment is expected to increase notably and, consequently, the computational cost, which scales as $n^2$ where n is the number of objects, will increase as well. Additionally, orbit propagation algorithms more sophisticated and time consuming than the SGP theory might be needed in the near future to predict more accurately the trajectories of the space debris objects.

All in all, it is reasonable to assume that the computational cost for an all vs. all scenario will grow significantly. In order to cope with such computational needs, the next natural step in the development of collision assessment tools is the use of parallelization techniques. In this paper we investigate the implementation of these techniques in the conjunction detection process. The computational memory requirements in an all vs. all scenario are of the order of 1GB and thus, the OpenMP parallelization standard, which is specifically designed for shared memory architectures, seems to be an adequate choice. In order to parallelize the SGP computations, an important prerequisite is the use of a multi-instance (object oriented) implementation of the SGP propagator.

Apart from the computation of the trajectories, the covariances of the different objects have to be propagated over time together with the orbital state as part of the collision risk evaluation. So far, ***closeap*** implemented a numerical integrator to propagate the covariance of all objects. However, this approach has two main disadvantages. Firstly, it introduces an inconsistency in the formulation as the SGP theory is used for the orbit computation while the covariance is numerically integrated. And secondly, the time needed for the numerical integration is also very significant.

Alternatively, we investigate in this paper the use of the SGP theory for the propagation of the covariance by means of numerical differentiation for objects whose orbit is generated from a TLE. This method has several advantages over the previous method: it yields consistent orbital states and covariances; it reduces the computational cost; and it allows parallelizing the covariance propagation very easily with the same approach employed for the computation of the trajectories.

The paper is organized as follows. In Section 1, this introduction is provided. Section 2 is devoted to a brief summary of the main features of ***closeap***, with special attention to the filters used for the detection of close conjunctions. A description of the main improvements at algorithm implementation level is given in Section 3, while in Section 4 a complete analysis of the parallelization technique used and the performance gains obtained is done. An efficient algorithm for covariance propagation is tested in Section 5. Finally, Section 6 is for conclusions.

## 2. Conjunction assessment and collision risk evaluation with ***closeap***

### 2.1. Main features of ***closeap***

***closeap*** is an application for close conjunction detection, collision risk assessment and collision avoidance manoeuvre optimization (see Ref. [10]) featuring:

1. catalog filtering, conjunction assessment and collision risk algorithms inherited from CRASS
2. NAPEOS as computational core and orbit propagation library
3. the latest implementation (Ref. [8]) of the SGP theory to compute orbits using TLE sets
4. full integration within ***focussuite***

***closeap*** has been built on the basis of existing and reliable software packages fully compatible with ESA infrastructures and practices yet extensible to other operational environments. Figure 1 depicts the elements and algorithms used from different sources. Details can be found in Ref. [9]

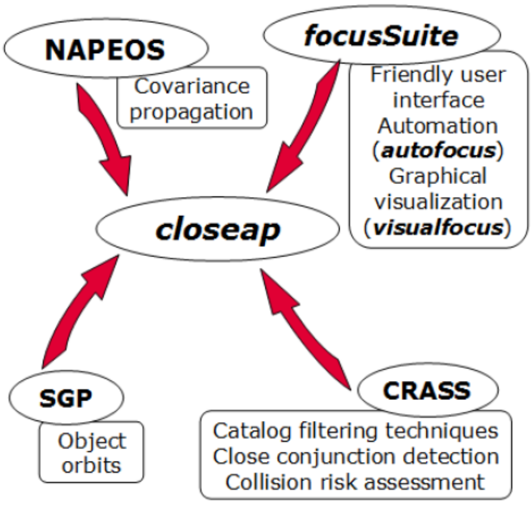
NAPEOS
Covariance propagation
focusSuite
Friendly user interface
Automation (autofocus)
Graphical visualization (visualfocus)
closeap
SGP
Object orbits
CRASS
Catalog filtering techniques
Close conjunction detection
Collision risk assessment

**Figure 1: Conceptual view of algorithms and libraries used in *closeap***

### 2.2. Filters for conjunction detection

Due to the large amount of space debris in space, the propagation of all objects along a period of time on the order of several days is a computationally intensive task. Therefore, those chaser objects whose orbital properties make it impossible for them to collide with the target objects are filtered out reducing the number of pairs to be analyzed in later stages of the filtering process.

In first place, an epoch filter removes those objects whose TLE sets have a generation epoch too old compared with the time span under analysis. Decaying and decayed objects are also removed from the problem. Once these preliminary filters have been applied, pairs of objects which could potentially have a close conjunction are detected at the beginning of the whole process with the classical and very efficient apogee-perigee filter. Basically, this filter removes from the analysis pairs of objects based on the difference between the apogee and the perigee radii of both objects. Afterwards, three different consecutive filters based on the relative position and velocity between target and chaser are used consecutively for each time step: a) the so called Smart Sieve presented in Ref. [5], b) a fine conjunction detection and c) a safety ellipsoid criterion.

The Smart Sieve technique consists of a series of filters based on very simple astrodynamics principles and is designed to minimize the computational cost in a safe and conservative manner. The fine conjunction detection basically consists of: a) a linear search algorithm to detect two points in time where the relative radial velocity of the pair has opposite signs, and b) a numerical root finder to compute the time of closest approach and the corresponding miss distance. Finally, the ellipsoid criterion allows rejecting conjunctions with a miss distance above some thresholds. More details about the conjunction assessment process are described in detail in [1].

## 3. Speed-up of the conjunction detection process

### 3.1. Reduction of the computational cost of the filters

The three filters that are applied at each time step can be optimized in order to reduce as much as possible the time needed for the whole process. In this optimization process, two general principles are applied: a) reduce the number of calls to the SGP propagator, b) use the TEME reference frame as the working reference frame when possible. Based on these simple principles, the following list of improvements have been derived and implemented (in order of importance):

- all computations except for the safety ellipsoid criterion are carried out in the TEME frame

- all filters make explicit use of the symmetries of the analysis (A vs. B is the same conjunction as B vs. A except for the safety ellipsoid criterion)

- the fine conjunction detection filter analyses at the same time all conjunctions passing the Smart Sieve at a given time step rather than each conjunction individually. In this manner, the linear search algorithm does not compute the state vector of one object several times at the same time step

- a Regula-Falsi method is used as the root finder of the fine conjunction detection

- the safety ellipsoid criterion is preceded by a safety sphere criterion whose radius is equal to the largest semi-major axis of the safety ellipsoid. In this way, the more time-consuming computations related to the safety ellipsoid are only carried out when strictly necessary

Thanks to all these improvements it is possible to reduce the computational cost in ***closeap*** by a factor of 24 with respect to CRASS. Table 1 compares the time required by CRASS and ***closeap*** to analyze an all vs. all scenario with 8000 objects and a time span of 1 day. In both cases the very same operating system (SUSE Linux), machine (1 CPU@2.66GHz) and compiler (Intel Compiler) is used to make the comparison of the algorithms meaningful.

**Table 1: CRASS & *closeap* performance for all vs all scenario with 8000 objects over one day**

| Tool | Time needed (mins) |
|---|---|
| CRASS | 121.1 |
| ***closeap*** | 4.7 |

### 3.2. Optimization of the weight of each filter

The weight of each one of the three filters of the process is controlled mostly by two variables: the time step of the Smart Sieve, $dt_1$, and the time step of the fine conjunction detection process, $dt_2$. These two variables can be modified without affecting the results to put more weight on the most efficient filters. Actually, this optimization was carried out early in the development phase of CRASS. However, after all the different changes in ***closeap***, it is interesting to redo the analysis. A case with 14000 objects and 1 day of prediction is used as reference. In this case, the original values used in CRASS of the time steps are $dt_l$=180s and $dt_2$=6s. Some considerations can be done before the actual optimization. First, the results show that conjunctions with miss distances of up 300 km pass the Smart Sieve. Thus, it is expected that decreasing $dt_l$ might reduce the overall computational cost since the conjunctions with large miss distances would be filtered by the Smart Sieve directly. Second, the linear search algorithm of the fine conjunction detection is less efficient than the numerical root finder. Therefore, it makes sense to increase $dt_2$ to let the faster converging Regula-Falsi method compute the time of closest approach starting from a wider time interval.

Table 2 presents the time needed to analyze an all vs. all case with 14000 objects over one day with different settings for $dt_l$ and $dt_2$. In all cases the results in terms of close conjunctions detected and miss distance are the exactly the same. From Tab. 2, the best configuration corresponds to the case with $dt_l$=120s and $dt_2$=60s. This result is in agreement with the two previous considerations.

**Table 2: *closeap* performance for the reference case for different values of dt$_1$ and dt$_2$**

| dt$_1$(s) | | dt$_1$(s) | | dt$_1$(s) | | dt$_1$(s) | |
|---|---|---|---|---|---|---|---|
| 132 | | 120 | | 108 | | 96 | |
| **dt$_2$(s)** | **time(mins)** | **dt$_2$(s)** | **time(mins)** | **dt$_2$(s)** | **time(mins)** | **dt$_2$(s)** | **time(mins)** |
| 6 | 11.4 | 6 | 11.0 | 6 | 10.7 | 6 | 10.6 |
| 16.5 | 9.7 | 15 | 9.6 | 13.5 | 9.7 | 12 | 9.9 |
| 33 | 9.3 | 30 | 9.2 | 27 | 9.3 | 24 | 9.6 |
| 66 | 9.1 | 60 | 9.1 | 54 | 9.3 | 48 | 9.5 |

In order to understand the weight that each of the filters has in the whole conjunction detection process, Table 3 presents a summary of the time spent in each of the tasks for the fastest configuration of Tab. 2. It is clear that most of the time of the process is spent on the Smart Sieve, although the contributions from other tasks are not negligible.

**Table 3: time spent on each one of the tasks for the case with dt$_1$=120s and dt$_2$=60s**

| | |
|---|---|
| Smart Sieve filter | 59.1% |
| TLE propagations for Regula-Falsi method | 12.6% |
| TLE propagations for linear search algorithm | 2.7% |
| TLE propagations for Smart Sieve | 1.6% |
| Conjunction definition | 1.1% |
| Linear search algorithm | 1.0% |

If one uses the improved time steps for the case of Tab.1, the time required is reduced to **2.7 minutes**. So it can be concluded the time required by ***closeap*** has been reduced by almost two orders of magnitude with respect to the original performance of CRASS.

### 3.3. Performance characterization

After describing all the improvements, it is interesting to characterize the actual performance of the tool in real life scenarios. To this end, starting from the reference case with 14000 objects over a time span of one day, a parametric analysis has been carried out modifying independently the number of objects involved in the analysis and the size of the time span under evaluation. As expected, the dependence of the required time with the number of objects is quadratic ($n^2$) and with the size of the time span linear. Note that for a modern and truly operational all vs. all scenario (e.g. 14000 objects and 7 days of prediction) the time required by ***closeap*** for the conjunction detection process is only 63.2 minutes (SUSE Linux, 1CPU@2.66GHz, Intel Compiler).

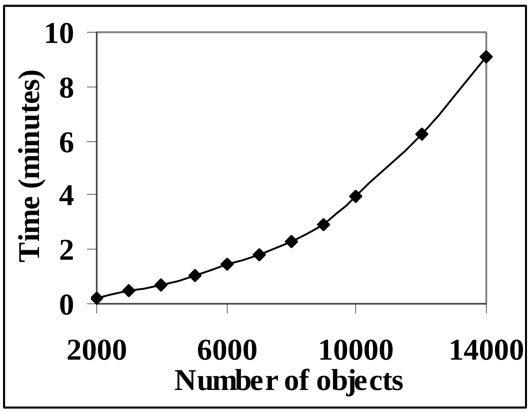

**Figure 3: Variation of the computational time needed by *closeap* with the number of objects (1 day of prediction)**

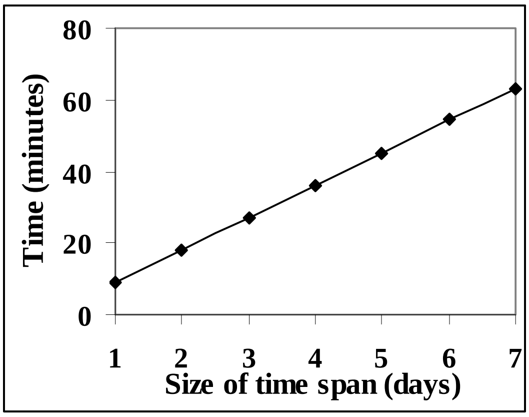

**Figure 4: Variation of the computational time needed by *closeap* with the time span (14000 objects under analysis)**

## 4. Parallelization

### 4.1. Multi-thread processing paradigm selection

During the development of CRASS, the possibility of parallelization was considered as an extension for the near future and possible parallel processing methods were identified at that point. Out of them, the most promising technique was Parallel Virtual Machine (PVM). The overall objective of a PVM system is to permit a collection of heterogeneous computers on a network to be viewed as a general purpose concurrent computation resource. The PVM system provides a set of user interface primitives that can be used for process invocation, message transmission and reception, broadcasting, synchronization, mutual exclusion and shared memory.

Over time two standards for parallel computations have been defined internationally, MPI (Message Passage Interface) and OpenMP (Open Multi-Processing). While MPI is also intended for distributed memory architectures and shares many commonalities with PVM, OpenMP targets shared memory architectures. Both standards are supported by several hardware and software vendors. The main difference is that MPI makes profit of a network of individual computers and OpenMP uses all available cores of a single machine. Modern COTS hardware includes several multi-core CPUs allowing a single machine to have 8 or more cores. There are two important reasons to prefer OpenMP over MPI or PVM for the conjunction assessment process. First, the hardware needed by OpenMP is simpler, easier to maintain and more adequate for a highly critical operational system than MPI or PVM since there is no cluster of computers involved. And secondly, the RAM memory requirements of ***closeap*** for an all vs. all scenario with 14000 objects are well within the 2GB limit. This value fits perfectly in a single modern machine which can easily come with 32GB of RAM installed.

### 4.2. Parallelization of the filters

As already mentioned, after all the improvements described in the previous section, for a typical all vs. all scenario the time spent on each filter and on the SGP propagations is of the same order of magnitude. Thus, in order to make full profit of the multi-thread capability, as many processes as possible have to be parallelized. This includes not only the SGP related computations, but also the Smart Sieve and the fine conjunction detection process. All these algorithms are well suited for parallelization since they reduce to carry out the same computations for a group of elements. The unit of parallelization is an object in the case of SGP propagations and a conjunction between two objects in the case of the Smart Sieve and the fine conjunction detection. An important prerequisite is that the SGP implementation is parallelizable. Apart from that requirement, the number of changes in the software in order to use OpenMP is minimal. In terms of possible performance gains, Figure 5 shows the variation of the time needed with the number of cores used. In particular, with 6 cores under use, the needed time is drastically reduced to **15 minutes**. Note that there is a small overhead coming from the parallelization which explains why the speed-up factor is not perfect.

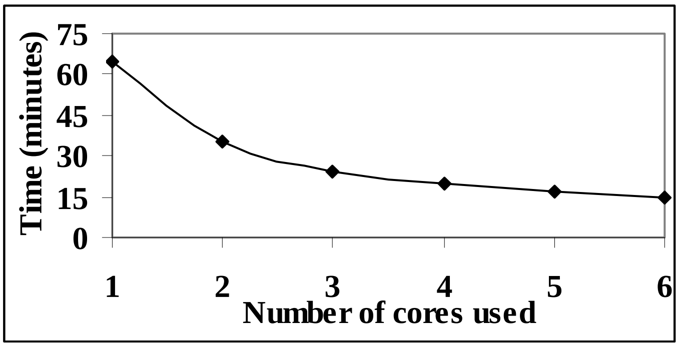

**Figure 5: Variation of the time needed by *closeap* with the number of threads for an all vs. all scenario with 14000 objects and 7 days of prediction (2 Quad-Core@2.66GHz)**

## 5. Covariance propagation

So far the focus has been on optimizing the conjunction assessment process, this is, the detection of close conjunctions. However, the second step of the whole process is the collision risk evaluation of all close conjunctions. Both ***closeap*** and CRASS propagate the covariance of the objects involved in a close conjunction with a numerical propagator. This implies that the orbital state propagated with the SGP theory and the state covariance might not be fully consistent since the dynamical models used are different. Moreover, the time required by the numerical propagator can be as demanding as the conjunction detection. Thus, it makes sense to look for possible alternatives to propagate the covariance of the space debris and the SGP theory seems a good candidate as already pointed out in Ref. [15].

Deriving the analytical or semi-analytical expressions to compute the state transition matrix consistent with the SGP theory is a cumbersome and error-prone process. So in this case it has been decided to use a numerical differentiation method. The main idea behind this formulation is that the evolution of the state covariance is mostly driven by the Earth's central gravity with small contributions from the $J_2$ term, solar and lunar gravity, atmospheric drag, solar radiation pressure and other perturbations. The SGP solves exactly for the central gravity (Kepler's equation) and includes the main perturbations approximately. Thus, it is expected that the models included in the SGP theory are good enough for the computation of collision risks.

Let us assume that the SGP implementation can be expressed in the following functional form:

$$\vec{x}_{TEME} = SGP(\Delta t, \vec{p}, \vec{q}) \tag{1}$$

where $\Delta t$ is the time to propagate with respect to the epoch of the TLE, $\vec{p}$ is a vector of 6 orbital parameters (orbit inclination, right ascension of the ascending node, eccentricity, argument of perigee, mean motion and mean anomaly), $\vec{q}$ is a vector of 3 additional parameters (B* drag term and 1st and 2nd time derivatives of the mean motion) and $\vec{x}_{TEME}$ is the resulting state vector in TEME reference frame. The state vector in inertial frame, $\vec{x}$, is obtained by a reference frame transformation from TEME to J2000.

Based on that functional expression, and by numerical differentiation with respect to each one of the elements of $\vec{p}$, it is possible to obtain the matrix, $\partial\vec{x}/\partial\vec{p}$, at any point in time. In particular, at $\Delta t = 0$, the process yields the square matrix $\partial\vec{x}_0/\partial\vec{p}$, which can be inverted to obtain $\partial\vec{p}/\partial\vec{x}_0$. By composition of both matrices, the state transition matrix, $\partial\vec{x}/\partial\vec{x}_0$, at a given time is:

$$\frac{\partial\vec{x}}{\partial\vec{x}_0} = \frac{\partial\vec{x}}{\partial\vec{p}}\frac{\partial\vec{p}}{\partial\vec{x}_0} \tag{2}$$

Once the transition matrix is obtained, the state covariance matrix at a given time $Cov(\vec{x},\vec{x})$ is:

$$Cov(\vec{x},\vec{x}) = \left(\frac{\partial\vec{x}}{\partial\vec{x}_0}\right) Cov(\vec{x}_0,\vec{x}_0) \left(\frac{\partial\vec{x}}{\partial\vec{x}_0}\right)^T \tag{3}$$

where $Cov(\vec{x}_0,\vec{x}_0)$ is the initial state covariance. Both CRASS and ***closeap*** use a look-up table to compute the initial state covariance associated to a given object as deeply explained in Ref. [13] and Ref. [14].

This formulation can be extended to account for uncertainties in other parameters. In particular, in CRASS and ***closeap*** it is normally assumed that the solar radiation pressure and drag coefficients have a given uncertainty. This effect can be captured in the current formulation with an uncertainty in the B* drag term which is directly proportional to the drag coefficient.

In order to test the validity of the SGP-based covariance formulation, the following test is defined with the European satellite ENVISAT. In first place, a fixed initial state covariance matrix is propagated over seven days with a numerical propagator using different dynamical models, ranging from very precise, which is used as reference, to extremely simple. The results allow us comparing the effect of the different dynamical models on the evolution of the state covariance over time. Secondly, the formulation presented above is used to compute the evolution of the state covariance according to the SGP theory. The result is then compared against the previously obtained results.

Figure 6 shows the results of the test in the form of the temporal evolution of the position sigma in radial, along-track and cross-track directions. And more interestingly the accuracy with which the SGP-based method reproduces the reference case (30x30) is fairly good. Both the final value of the along track position sigma, and the spread of the cross-track position sigma are captured in the SGP-based covariance propagation. Those two properties of the reference case are not captured in the Kepler-motion case. It is clear that the effect of the degree and order of the Earth gravity potential is very small as long as the first few terms of the expansion (central gravity, $J_2$ effect…) are considered. Therefore, at first glance this method seems suitable to be implemented in the collision risk evaluation process, although more tests need to be done.

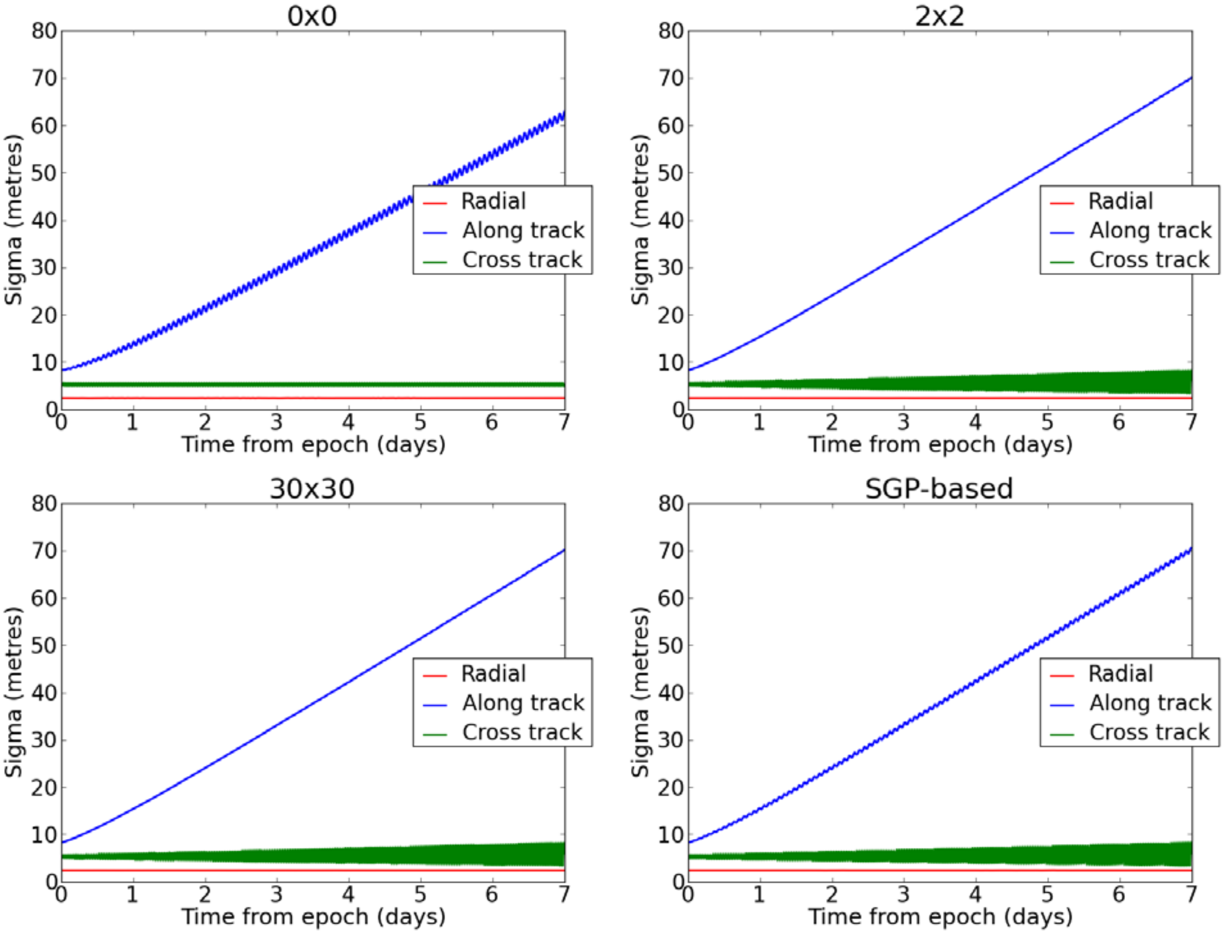

**Figure 6: Evolution of the position sigma in along-track, cross-track and radial direction for different dynamical models. Only central gravity (upper left plot), 2x2 gravity potential (upper right plot), 30x30 gravity potential (lower left plot), SGP-based (lower right plot)**

## 6. Conclusions

In first place, important performance gains have been presented for the conjunction assessment of all vs. all scenarios with a large number of objects involved. Improvements at algorithm level and parallelization techniques have allowed reducing the computational time by 2 orders of magnitude. And secondly, a method to propagate the covariance of objects based on the SGP theory has been tested. The positive results make it suitable for an operational implementation in order to reduce the time required for the collision risk evaluation process.

## 7. References

[1] Klinkrad, H., "On-orbit-risk reduction – collision avoidance", J. Aerospace Engineering, 221(6), 955-962, 2007.

[2] Laporte, F. and Sasot, E., "Operational management of collision risks for LEO satellites at CNES", Proceedings of the Space-Ops 2008 conference, AIAA-2008-3409.

[3] Kelso., T. S., Alfano, S., "Satellite orbital conjunction report assessing threatening encounters in Space (SOCRATES)", Proceedings of the 5th European Conference on Space Debris, 2009.

[4] Alarcón-Rodríguez, J.R., Martínez-Fadrique, F.M. and Klinkrad, H., "Development of a collision risk assessment tool", Advances in Space Research. 34(5), 1120-1124, 2004.

[5] Alarcón-Rodríguez, J.R., Martínez-Fadrique, F.M. and Klinkrad, H., "Collision risk assessment with a Smart Sieve method", Proceedings of Joint- ESA-NASA Space Flight Conference, ESA-SP-486, 159-164, 2002.

[6] Alfriend, K., Akella, M., Lee, D., Frisbee, J. and Foster, J., "Probability of collision error analysis". Space Debris, 1(1), 21–35, 1999.

[7] Hoots, F. R. and Roehrich, R. L, "Spacetrack report No. 3. Models for propagation of the NORAD Element Sets", US Air Force Aerospace Defense Command, Colorado Springs, CO, 1980.

[8] Vallado, D., Crawford, P., Hujsak, R. and Kelso, T.S, "Revisiting space-track report #3", American Institute of Aeronautics and Astronautics. AIAA 2006-6753, 2006.

[9] Escobar Antón, D., Pérez Cambriles, A., Martínez, F. M., Águeda, A., "***closeap***: GMV's conjunction assessment tool", Proceedings of the 5th European Conference on Space Debris, 2009.

[10] Patera, R. P. and Peterson, G. E., "Space vehicle maneuver method to lower collision risk to an acceptable level", J. Guidance, Control and Dynamics, 26(2), 233-237, 2003.

[11] Alarcón-Rodríguez, J.R., Martínez-Fadrique, F.M., Klinkrad, H., Rudolf, A. and Bosquillon de Frescheville, F., "Conjunction event predictions for operational ESA satellites", Proceedings of the Space-Ops 2004 Conference, 2004.

[12] Flohrer, T., Krag. H. and Klinkrad, H., "ESA´s process for the identification and assessment of high risk conjunction events", Advances in Space Research, 44, 355-363, 2009.

[13] Krag. H., Klinkrad, H., Alarcón-Rodríguez, J.R., "Assessment of orbit uncertainties for collision risk predictions at ESA", Proceedings of Second IAASS Conference Space safety in a Global World, Chicago, USA, 2007.

[14] Flohrer, T., Krag. H. and Klinkrad, H., "Assessment and categorization of TLE Orbit Errors for the US SSN Catalogue", Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference, Wailea Beach, Maui, United States, 2008.

[15] Escobar, D., Águeda. A., Martínez, F. M. and García, P., "Implementation of conjunction assessment algorithms in modern space mechanic systems", Proceedings of the 4th International Conference in Astrodynamic Tools and Techniques, Madrid, Spain, 2010.