IMPLEMENTATION OF AN OPEN-SCENARIO, LONG-TERM SPACE DEBRIS SIMULATION APPROACH

Bron Nelson[3], Fan Yang Yang[2], Roberto Carlino[4], Andres Dono Perez[2], Nicolas Faber[1], Chris Henze[5], Arif Göktuğ Karacalıoğlu[4], Conor O'Toole[6], Jason Swenson[7], Jan Stupl[1]

[1]SGT Inc. / NASA Ames Research Center, [2]MEI / NASA Ames Research Center,
[3]Computer Sciences Corporation / NASA Ames Research Center,
[4] STC / NASA Ames Research Center, [5]NASA Ames Research Center, [6]University College Dublin / NASA Ames Research Center, [7]LMCO Space OPNS / NASA Ames Research Center

ABSTRACT

This paper provides a status update on the implementation of a flexible, long-term space debris simulation approach. The motivation is to build a tool that can assess the long-term impact of various options for debris-remediation, including the LightForce space debris collision avoidance concept that diverts objects using photon pressure [9].

State-of-the-art simulation approaches that assess the long-term development of the debris environment use either completely statistical approaches, or they rely on large time steps on the order of several days if they simulate the positions of single objects over time. They cannot be easily adapted to investigate the impact of specific collision avoidance schemes or de-orbit schemes, because the efficiency of a collision avoidance maneuver can depend on various input parameters, including ground station positions and orbital and physical parameters of the objects involved in close encounters (conjunctions). Furthermore, maneuvers take place on timescales much smaller than days. For example, LightForce only changes the orbit of a certain object (aiming to reduce the probability of collision), but it does not remove entire objects or groups of objects. In the same sense, it is also not straightforward to compare specific de-orbit methods in regard to potential collision risks during a de-orbit maneuver. To gain flexibility in assessing interactions with objects, we implement a simulation that includes every tracked space object in Low Earth Orbit (LEO) and propagates all objects with high precision and variable time-steps as small as one second. It allows the assessment of the (potential) impact of physical or orbital changes to any object. The final goal is to employ a Monte Carlo approach to assess the debris evolution during the simulation time-frame of 100 years and to compare a baseline scenario to debris remediation scenarios or other scenarios of interest. To populate the initial simulation, we use the entire space-track object catalog in LEO. We then use a high precision propagator to propagate all objects over the entire simulation duration. If collisions are detected, the appropriate number of debris objects are created and inserted into the simulation framework. Depending on the scenario, further objects, e.g. due to new launches, can be added. At the end of the simulation, the total number of objects above a cut-off size and the number of detected collisions provide benchmark parameters for the comparison between scenarios. The simulation approach is computationally intensive as it involves tens of thousands of objects; hence we use a highly parallel approach employing up to a thousand cores on the NASA Pleiades supercomputer for a single run.

This paper describes our simulation approach, the status of its implementation, the approach to developing scenarios and examples of first test runs.

1. INTRODUCTION

This paper describes our progress towards implementing a flexible, open-scenario and long-term space debris simulation framework. The goal of this effort is to produce a tool that can project the long-term development of the debris environment under various assumptions, e.g. scenarios that include debris mitigation and remediation approaches, as well as intentional collisions.

Space debris is a growing problem for active spacecraft. A 2010 study calculated cost increases for several representative model satellite constellations due to debris impact [1]. For the specific examples, replacement of satellites that are damaged or destroyed through collisions with debris would raise the cost of operating and maintaining by 4 to 14 percent, translating to hundreds of millions of dollars over the assumed operational period of 20 years. Long-term projections predict an increase in the debris population, leading to a further increase of risk and cost [2]. The increasing numbers are the result of new launches, spontaneous explosions, and fragmentations through collisions between space objects. The latter leads to a cascading effect that was originally predicted by Kessler [3] and was also confirmed in various other studies [4]. This Kessler cascade would increase the number of debris objects in orbit and increase the collision risk further, even if no more launches occur [2].

Various methods have been proposed to improve this situation, or at least stabilize the number of debris objects in orbit. In order to classify these methods, the first distinction is between debris mitigation and remediation. In the space debris community, the term mitigation includes activities that fall under more responsible behavior, e.g. preventing accidental explosions and de-orbiting spacecraft at the end of their mission. Activities grouped under the term remediation deal with actively engaging existing debris objects. One class of remediation methods focuses on the active removal of a number of heavy debris objects from orbit. As they are potential sources of additional debris, their active removal can help stabilize the number of debris objects, as shown in simulations. The second class of remediation activities are active, externally induced collision avoidance measures, where debris trajectories are influenced just in time to avoid collisions [5]. Just in time collision avoidance has the potential to be both of short-term and long-term use: on the one hand, such a capability can be used to protect active spacecraft; on the other hand, each avoided collision reduces the number of additional debris objects created.

Some specific methods for active debris removal include debris removal by sending disposal spacecraft to grab objects and de-orbit them, or attaching electrodynamic tethers to debris objects and using the Earth's magnetic field to induce a force that slows debris and leads to deorbit. Methods for active, externally-induced collision avoidance include: deploying drag-enhancing mist along the debris trajectory, or the LightForce method that our group has been investigating extensively [6,7,8,9]. LightForce uses photon pressure from ground-based lasers to modify the in-track velocity of space objects. The idea is to apply LightForce in reaction to conjunction warnings and ensure that both objects involved in a potential collision do not appear in the same place at the same time.

Assessing the utility and efficiency of a specific remediation method on long time scales is not straightforward. The state-of-the-art simulation tool is NASA LEGEND [10,11]. Within a Monte Carlo framework, LEGEND propagates every object over a 10cm size cutoff, using large time steps of several days (the default is 5) [11]. In between these time steps, a statistical evaluation returns the probability of a collision event which then leads to a random decision whether to insert new fragments into the model. Over multiple Monte Carlo runs, this approach provides an average projection of the future debris environment. Investigating active debris removal (ADR) in this framework is comparably easier than collision avoidance methods, because for ADR a given group of objects is removed from the environment at a given time. Investigating collision avoidance, however, is more complicated. Whether a specific collision avoidance engagement is successful or not will depend on the specifics of each close encounter between objects (a conjunction). For example, success of a LightForce engagement will depend on the masses of the objects involved, their orbits, position of laser ground stations and ground station capabilities. The probability of success will depend on these and other inputs that cannot be easily derived and hence cannot easily be introduced into a tool like LEGEND that uses time steps that are orders of magnitude larger than the actual engagement.

Motivated by our research of the LightForce collision-avoidance scheme and promising results on the 85% fraction of conjunctions that can be mitigated in the short term [9], we decided to work towards a long-term debris simulation tool that uses time steps on the order of seconds and hence is flexible enough to assess various debris remediation schemes, including just-in-time collision avoidance, and can also assess other events and scenarios of interest. The following sections describe our simulation approach, the current status of implementation (including examples of outputs), and list the planned next steps.

2. SIMULATION APPROACH

2.1 Introduction and general overview of the simulation approach

This section outlines the planned implementation of our tool. We start with the general approach and then describe the details of the physics and numerical implementation, as well as the approach to data analysis.

The main challenge for long-term debris projections is that single events can determine the state of the entire environment. For example, combined, the destruction of the Chinese Fengyun spacecraft in 2007 and the 2009 collision between Iridium 33 and COSMOS 2251 doubled the population of fragmentation debris in orbit. While the first event was probably intentional, the second was one out of tens of thousands of close encounters (conjunctions) between objects that occur during a year. The simulation of each event is of a highly statistical nature, as minimal changes in initial positions and velocities of space objects, as well as changes in external forces, lead to significant changes in future positions that are several orders of magnitude larger than the object sizes. Similarly, changes in position on the order of the object size influence the likelihood of a collision. Hence, a Monte Carlo approach is needed to produce a projection of a likely future, randomizing initial object states, as it is done in LEGEND.

In order to gain the capability to assess the effect of arbitrary changes to individual space objects onto the environment, we choose a simulation approach where every object's position and velocity are simulated over time in high resolution (on the order of seconds, not days). Nikolaev et al. coined this as a *brute force* approach when they implemented a similar approach [12]. This gives the ability to manipulate arbitrary objects as needed to simulate collision-avoidance engagements, and also gives further options to simulate other scenarios, e.g. the deployment of drag devices or de-orbit maneuvers. Since these events happen on a timescale of seconds, this approach is computationally intensive, but allows high precision and maximum flexibility to simulate various engagements and events. In addition, the code needs to track and assess all conjunctions (close encounters between objects). In the case of collisions, new objects are inserted into the simulation. If objects decay into the Earth's atmosphere they are removed. New launches are added at the appropriate time.

The described work is an extension of the simulation software that was developed for the LightForce efficiency assessment described in 2013 and 2014 AMOS papers [8, 9]. The following summarizes the approach described in these papers, focusing on the changes in the current software.

2.2 Physics and numerical implementation

2.2.1 Propagation of space objects

The simulation framework is initialized by reading in a list of object data, consisting of area-to-mass ratio, area, drag coefficient, and the object state vector (position and velocity) at a specific time. Future positions and velocities are calculated using a high-precision propagator. The scheme used by the propagator for the numerical integration is a 4th/5th order Runge-Kutta scheme with variable time step [13,14]. The forces taken into account during the propagation include Earth's gravitational field, the gravitational perturbations from the Moon and the Sun, atmospheric drag, and the solar radiation pressure. Space weather and planetary ephemerides are fed into the model via input files, as described in section 3.2.4 (scenario development). The numerical implementation is built around the NAIF SPICE Toolkit [15] and the physical model used for each force is referenced in Table 1.

We validated our propagator against STK's HPOP, an industry standard. The propagator is configured to dynamically adjust its integration time steps to maintain a specified relative error tolerance. That means that the propagator is "running free", producing a new position/velocity point for an object whenever it is appropriate. These calculations are done in parallel on multiple processors.

Table 1: Forces taken into account in the dynamical modeling of the debris and the models used for their numerical implementation.

Force	Numerical implementation	Reference
Earth's Gravitational Field	Earth Gravitational Model 1996 (EGM96)	[16]
Luni-Solar Perturbations	NASA JPL Planetary Ephemerides	[17]
Atmospheric Drag	NRL-MSISE-00 model	[18]
Solar Radiation Pressure	Debris modeled as a sphere, eclipses taken into account	[19]
Laser Radiation Pressure	In-house model	[6]

In addition to standard propagation in a gravity field with atmospheric drag, the propagator can add additional forces. For example, LightForce is already implemented and can be activated as needed for a specific scenario. LightForce adds photon pressure from a laser ground station, whenever access is available, and the additional force is able to reduce the probability of an upcoming collision. For future revisions of the software, one could force the propagator to follow a predefined trajectory for an object, e.g. to implement specific de-orbit maneuvers or foul play. Another option to be implemented is the capability to change object properties at given times, e.g. to account for the deployment of drag-enhancing devices at the end of a spacecraft's mission.

To simulate collisions, spontaneous explosions, and new launches, there is a requirement to add and delete objects dynamically during an ongoing simulation run. The simulation framework is designed to allow for addition and deletion of objects that either occur at a certain time as defined by input files, or internally. Internal deletions happen when objects deorbit below a cut-off altitude, or when explosions or collisions occur. Spontaneous explosions are triggered randomly, as described in section 3.2.3. The original object will be deleted and replaced by a debris cloud predicted by the NASA standard breakup model (see section 2.2.3 for details). The same approach for object deletion and creation is taken for object fragmentations that are caused by collisions between objects.

2.2.2 Conjunction analysis

In order to detect collisions in the simulation, the code performs an all-on-all conjunction analyses. That means, object positions are checked against each other as time progresses. The analysis time-step size determines the time interval that the code is checking for conjunctions and is independent of the propagator time-steps, which differ for each object. During each analysis time step, the propagator produces a limited time series of state vectors for each object, referred to as anchor points. With the anchor points in hand, each CPU core uses spline interpolation to determine the position of all objects for the beginning of the current time-step of the conjunction analysis. These positions are then sorted along one axis of the coordinate frame. The code then does collision detection. Each core assigns itself a subset of the objects, and decides if an object might possibly interact with some other object during the current time step. If that is the case, the distance at the time of closest approach is calculated.

To calculate both the distance and the time of closest approach we use an algorithm that uses spline interpolation between the state vectors of the two objects at times t_1 and t_2 [19, p. 919-937]. Acquiring these state vectors is not trivial: As the propagator is using variable time steps, the high-precision anchor points appear at different times for each object (see figure 1). It is extremely unlikely that the propagator produces anchor points for the same times. Hence we use the last anchor point outside the time step of interest to choose t_1. That state vector could either belong to the first or second object. We then generate the state vector of the second object at t_1 using spline interpolation. For t_2, we chose an existing anchor point as well. The state vector of the second object at t_2 is determined by spline interpolation again. The resulting four state vectors (two for each object) are used to determine the time of closest approach, the miss distance and state vectors at that time. The code also returns the probability of collision, using covariances that can be provided by the input files. As all related calculations are computationally expensive, the code first uses a couple of simple tests to try and (cheaply) eliminate the possibility that the two objects could interact during the investigated time step.

Note that the method for producing the t_1 and t_2 state vectors is independent of the size of the analysis time-step, depending only on the anchor points produced by the propagator. This enables the simulation to dynamically vary the size of an analysis time-step for optimal speed, yet still maintain exact bitwise reproducibility.

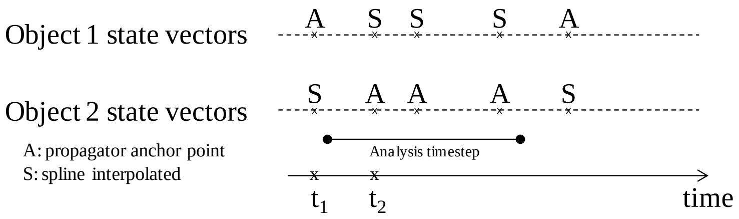

Figure 1: High precision state vectors produced by the propagator (anchor points) and interpolated state vectors during conjunction analysis

2.2.3 Fragmentation through Collisions and Explosions

Fragmentation can occur either through spontaneous explosions or collisions. The process of randomly triggering explosions is dependent on the input scenario, as described in section 3.2.3. To trigger collision fragmentation, we use the (miss) distance between objects at time of closest approach as the final benchmark to decide whether a collision has occurred. If the distance is closer than the combined object radii, fragmentation is initiated. This method is the same as used by Nikolaev [12]. It has the advantage that it is independent of state covariances, which might not be accurate. While (even with perfect initial states) the propagator is not able to predict an object position with an accuracy that is comparable to the object size, this method still gives reasonable results in combination with the Monte Carlo method, because uncertainties are modeled through the inherent randomness of the method [12].

The inserted debris clouds for both explosions and collisions are created using the approach outlined in the NASA EVOLVE 4.0 breakup model [20, 21]. We make the following three implementation choices: 1) EVOLVE provides area-to-mass distributions for fragments that are smaller than 8 cm and larger than 11 cm, but not between 8 cm and 11 cm. In order to close the gap we chose randomly between one of the two distributions. 2) Velocity change for the objects in the fragment cloud is assigned using the distributions given by EVOLVE. The direction of the velocity change is randomly assigned. 3) For explosions, we determine the number and size of pieces using a scaling factor S=1, which is valid for upper stages between 600-1000 kg.

2.3 Data products and analysis

During a simulation run, all conjunctions above a user-definable probability of collision and miss distance threshold are tracked. The probability of collision [22] and the state vectors of the objects involved are stored. The simulation code also tracks the number of objects in orbit, divided by object class, i.e. debris, spacecraft and rocket bodies. New debris objects that are created during a simulation run are categorized by debris objects that are created by explosions and by collisions. In addition, state vectors of all objects involved in breakups are stored at the time of the breakup event. Finally, the simulation periodically stores state vectors of all objects with additional data including potential de-orbit time and the parent objects of newly created fragments.

The combined data products enable various analyses: On a global level, the development of object numbers over time gives insight into the development of populations and the influence of debris mitigation or remediation measures. The number of conjunctions gives a first insight into the development of collision risk. Tracking the probability of collision for each conjunction enables more sophisticated analyses, e.g. tracking the cumulative value. More detailed analyses for a class of spacecraft, a specific constellation or a specific spacecraft are possible as well, always comparing different scenarios or a different time frame in a specific scenario.

A Monte Carlo analysis of multiple simulation runs is the preferred method to gain information on simulation uncertainties. However, a single run does represent one possible future and can also provide insights in a specific scenario, e.g. the number of expected conjunctions given a specific population on orbit.

3. STATUS OF IMPLEMENTATION AND SCENARIO DEVELOPMENT

3.1 Current status of software implementation

The simulation framework is currently utilizing the NASA Ames Pleiades supercomputer. As of August 2015, the Pleiades supercomputer consists of a total of 207,501 cores, distributed over a mix of the following types of Intel Xeon processors: E5-2680v3 (Haswell), E5-2680v2 (Ivy Bridge), E5-2670 (Sandy Bridge), and X5670 (Westmere). The system is an SGI ICE cluster connected with InfiniBand in a dual-plane hypercube technology.

The software is implemented in C and C++ and uses the standard Message Passing Interface (MPI) for parallelization. Production of (high precision propagator) anchor points is done fully in parallel, with each MPI rank doing propagation on a subset of the satellites, and then exchanging the results with the other ranks. When complete, each MPI rank has the full set of information about every satellite. Conjunction analysis is conducted only partly in parallel, as described in section 2.2.2.

Currently, the software is ready to produce data of single runs. A Monte Carlo wrapper has not yet been implemented. The data shown in what follows thus represents “one possible future”, but not an average projection with error bounds.

3.2 Scenario development

3.2.1 Initial population

The first step to develop a scenario is to generate a population of space objects at the beginning of the simulation period. The code accepts a list of objects that includes object state (position and velocity), area-to-mass ratios, area, ballistic coefficient and reflectivity, the state covariance and the epoch for that data. For statistical purposes, we also provide the object type (rocket body, spacecraft or debris).

To create that list we are currently using the following four step process:

1) **Create object population from TLEs:** For a given date (in this case June 15, 2015) we utilize the publicly available two-line element (TLE) orbital data from the Joint Space Operations Center (JSpOC) and select all objects with a perigee below 1800 km [23]. For each object the catalog provides a unique identifier, the object type and the orbital elements at a given epoch. Unfortunately, the catalogue does not directly provide an area-to-mass ratio. Also, single TLEs are error prone and have limited accuracy. To improve the quality, we use least-squares fitting of TLE data as described by Levit and Marshall [24] to obtain an improved state vector, an area-to-mass ratio guess and a covariance uncertainty matrix. The method is described in more detail in [8]. Out of 12725 records for objects in LEO, a successful fit was provided for 10256 objects. For the remaining objects we use a direct translation from TLE to state vector and assign a covariance matrix corresponding to the average covariance of the successfully fitted objects.
2) **Correct area-to-mass ratio:** The area-to-mass ratio guesses from step 1 are not accurate enough to result in reasonable object life times for approximately 30% of the objects. To increase the accuracy of the area-to-mass ratio over the method described in [8], we use a list of area-to-mass ratios provided by our colleague Wang Ting (Princeton). He uses on average 500 TLEs to determine the area-to-mass ratio through fitting the semi-major axis decay of an object and also provides an RMS error of that fit. From his list, we removed 275 entries where the relative RMS error of the fit is larger than one, as well as 113 entries for negative area-to-mass ratios, the latter indicating maneuverable spacecraft. In total, we can match 11050 objects from the 12725 objects from step 1. Out of the remaining 1675, we provide values for about 70 of the 113 maneuvering spacecraft from a 2008 snapshot of ESA DISCOS data [25], using the average between the provided minimum and maximum area and the mass at beginning of life. For the rest we draw random area-to-mass ratios from objects belonging to the same class (rocket body, spacecraft or debris) as the object with the missing data. For example, for each spacecraft with unknown area-to-mass ratio we assign the area-to-mass ratio of a random spacecraft with known data. Using that approach we preserve the original area-to-mass distributions.
3) **Translate RCS to area:** The cross-sectional areas of the objects are assigned using radar cross-sections (RCS) in a first step. Space-track provided that data until July 2014. We convert RCS to a physical cross-section using the method given in [26]. After July 2014, space-track has been providing only classifications into “small”, “medium” and “large” RCS. These groups correspond to defined RCS bins. For all new objects with only classifications but no specific RCS data, we draw one existing RCS out of the appropriate group of “small”, “medium” and “large” objects, and translate that to a physical size.
4) **Correct area for intact objects:** The method described in [26] to translate RCS to physical size was developed for debris (using samples from hypervelocity impacts) and is not necessarily accurate for intact spacecraft and rocket bodies. Hence we replace the cross section for intacts with data from the 2008 DISCOS snapshot [25]. We find matching data for 2416 out of 3481 entries for rocket bodies and spacecraft. We translate the average dimension into an area, only omitting the IMAGE and Picosat 1&2 spacecraft. IMAGE is equipped with 4 250m radial antennas, which would translate into an unrealistic area. Picosat 1&2 are tethered. If no data is available, we keep the data derived from RCS data.

For ballistic coefficient we assign 2.2 and 1 for reflectivity. Future work would include incorporating updated DISCOS data, as well as other available data for known objects.

3.2.2 Future launches

On top of the current population of objects, future launches will put additional rocket bodies and spacecraft into orbit. The code accepts a list for future launches that is mostly identical to the one provided for the initial population. Instead of the epoch time for the fit, this file provides a date and time that specifies when the object is to appear in the simulation.

There are several options to create a future launch scenario. The simplest option is to set future launches to zero. A common practice for long-term space debris simulations to define a period of time (e.g. the last 10 years before the start of the simulation) and use the list of objects launched within this period recurrently as a forecast until the end of the simulation period. Although this method may provide reasonable future scenarios in a stable environment, the current satellite market is going through a transformation where commercial “new space” companies have started to dominate the numbers of newly launched spacecraft. We chose to develop a scenario based on the assumption that these new companies thrive.

To develop an optimistic “new space” scenario, we systematically gather available data on future launches and collect it in a database. We aim to build a database that covers all the publicly available launch related information regarding the companies which intend to launch satellites into Low Earth Orbit (LEO) between 2015-2030. These companies/constellations include, but are not limited to, Blacksky, CICERO, EROS, Landmapper, Leosat, Northstar, O3b, OmniEarth, OneWeb, Orbcomm, OuterNet, PlanetIQ, Planet Labs, Radarsat, RapidEye Next Generation, Sentinel, Skybox, SpaceX, and Spire. Data is gathered either through direct contact with the company or from online resources (i.e. company press releases, published interviews). Information collected includes statements on the number of launches for each year between 2015-2030, information on the number of orbital planes the constellation will be distributed to, as well as apogee, perigee, inclination, spacecraft mass and area. Whenever data are not available, estimations were made considering the constellation’s purpose and company’s previous missions, if any. The database also takes into account possible newcomers into commercial earth observation and telecommunication markets as well as the replenishment launches of the current and upcoming constellations. Figure 9 below shows a summary graph generated from that information for the time between 2015-2030.

Launched rocket bodies are strongly correlated to the number of spacecraft. However, another trend is that more spacecraft are being carried as secondary payloads or are deployed from the International Space Station. Therefore we decided to use the historical positive-sloped trend for launched rocket bodies as a baseline and add approximately 30 launches per year on top of that baseline starting from 2017, to account for the new constellations. Not all of these rocket bodies decay naturally. Our analysis of the online satellite catalog (SATCAT) data [23] shows that between 2007-2015 around 25% of the upper stages in LEO decayed within the first 10 days after their launch date.

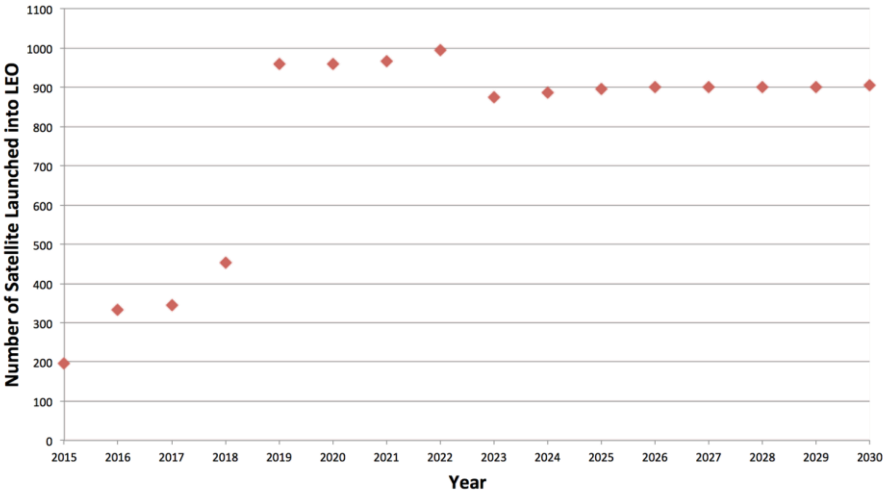

Figure 2: Optimistic “New Space” spacecraft launch scenario data built on compiled data on announced satellite launches to LEO

Assuming a positive trend, with more strict rules and potential use of reusable launchers; we build the scenario around an assumption that 30% are de-orbited and 70% of rocket bodies naturally decay. The total number is plotted in figure 3. The apogee, perigee, and inclination data for these objects were estimated in correlation with the information gathered on announced spacecraft launches and the historical trends. For modeling 2030 onward, available information is increasingly sparse. Therefore we chose to build the scenario extrapolating the 2015-2030 data on yearly basis.

The compiled list of predictions for satellites and rocket bodies with their annual number estimates, as well as their apogee, perigee, inclination, mass, and area information needs to be transformed into an input file for the orbit propagator. An automated script was built to pull the necessary parameters from the launch database and convert them into a suitable format for the simulation.

The script converts the estimated Keplerian elements of each object into Cartesian position and velocity state vectors at the predicted launch epoch. The database does not provide details on the orbits, hence, the argument of periapsis, eccentricity, true anomaly and Right Ascension of the Ascending Node (RAAN) were assigned randomly using uniform distributions within the domains of each element. For Sun-Synchronous orbits, the script calculates the required inclination depending on the altitude and eccentricity of the object.

Launch epoch dates were assigned randomly within the launch year for each constellation from the database. The approach assigned a maximum of 15 objects for the same launch. Additional parameters, i.e. area-to-mass ratios, drag coefficient and reflectivity are assigned to each object according to their physical specifications. The positional covariance matrix was set to an average for all new objects. In order to calculate this average, the full set of objects from the June 2015 initial population is used.

3.2.3 Spontaneous on orbit explosions

Another source of orbital debris is spontaneous in-orbit explosions. In order to set a probability for those events, we evaluate the list of explosions given in the 14th edition of NASA's "History of On-Orbit Satellite Fragmentations" report [27]. On a yearly basis, this comprehensive list is correlated with the number of objects on orbit to obtain a measure for the probability of such an explosion. Figure 4 shows a general trend to a lesser probability of explosion per object over time for both rocket bodies and spacecraft. The x-axis shows the time in 5-year blocks, where the first period refers to the slot between 1963-1967, the second to 1968-1972, and so on. We fit that data to an exponential function. This function is used in the scenario to randomly trigger explosions, using the time-dependent number of rocket bodies and spacecraft as input.

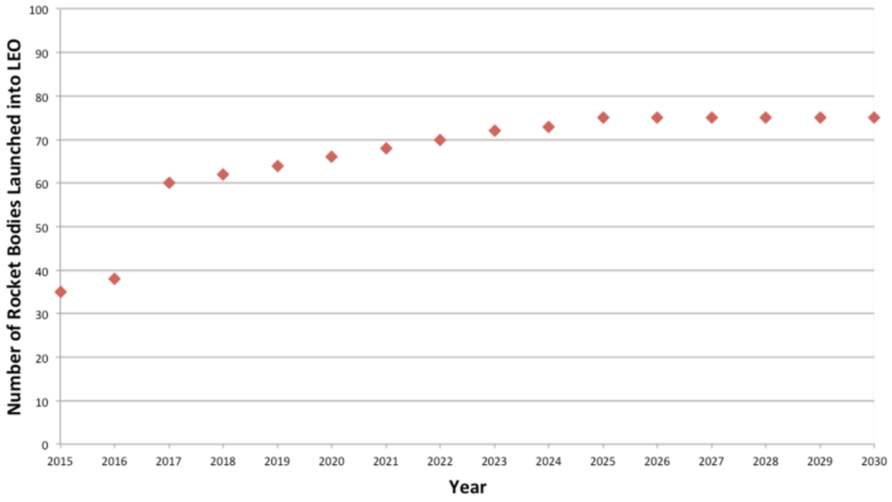

Figure 3: Optimistic "New Space" scenario for launched rocket bodies

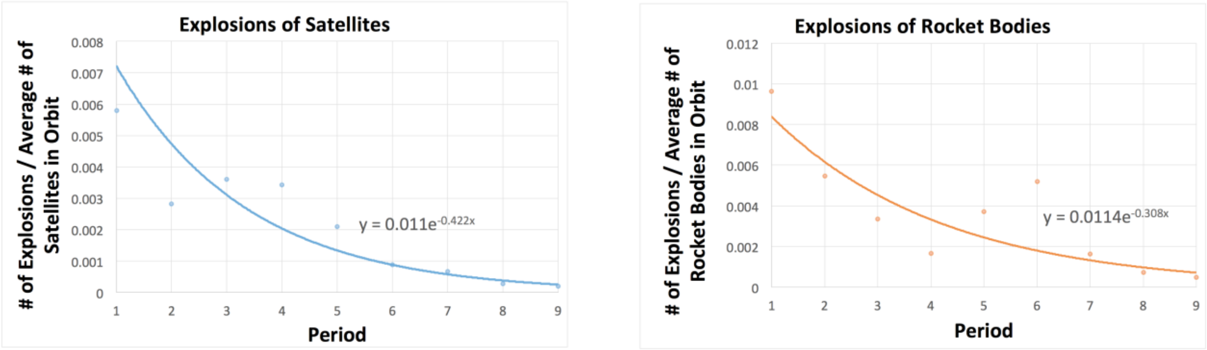

Figure 4: Number of explosions divided by objects on orbit over time for satellites and rocket bodies. Period 1 referes to the slot between 1963-1967, period 2 to 1968-1972, ... Exponential fit added.

3.2.4 Space Weather and SPICE files

In order to allow long term simulations over a 100 year time frame, both the space weather input and the NAIF SPICE input data need to be provided.

The space weather data, which is needed to model the atmosphere's response to the Sun's activity, has to be projected up to July 2115. Our method of extending this data is purely statistical, not physical, and so will not capture all of the intricacies of the Sun. However, it should still capture the general trends, such as periodic variability with the Solar cycle, along with the large degree of day-to-day variation seen in observational data.

The method implemented here makes use of a significant amount of observational data, obtained from [28]. This data extends from January 1st 1957 to July 10th 2015, covering 4 complete cycles (minima and maxima). The data necessary for the NRL-MSISE-00 is the daily value of the 10.7cm flux (F10.7), along with the 81-day centered average of F10.7, and the daily values of the planetary index A_p in 3 hour intervals and a daily average.

In order to generate reasonable data for F10.7, we first determine its period by fitting a cosine function to the data:

$$f(x) = a_1 \cos(a_2 x + a_3) + a_4 \qquad .$$

The period is 10.79 years. The observational data includes four complete cycles, which we use. For the future date of interest, we use the corresponding dates in all four of these cycles and consider a 6 month interval centered on these dates. Combining these intervals, we have a distribution of F10.7 values, from which we select a value at random for our future date of interest. The use of such large intervals is an attempt to mirror the short-term variability of F10.7 by providing as many potential values as possible. The results of this technique are shown in Figure 12. The 81-day centered averages are produced from that dataset we create. At the end of the dataset the input interval for each if those averages is reduced to the remaining values.

For the planetary index A_p we find no obvious repeating pattern. Hence, we use the observational data to create a probability distribution (table 3). According to this distribution an appropriate number of integers are randomly drawn out of each interval.

It should be noted that the use of daily values would be the ideal method of implementing this data. However, to reduce strain during data lookup, we approximate the data by using the values on the first day of each month as constant throughout that month. This still produces reasonable results, with the average altitude of the same object after a year differing by only tens of meters using the two methods.

In addition to the space weather, the NAIF SPICE toolkit needs planetary ephemerides and Earth orientation input data, which were provided by the JPL/NAIF team.

Table 2: Observed historical distribution of the planetary index A_p and average difference of the implemented data to the historical observations

Interval	Proportion in Observations	Average Difference from Observed Proportion
0-10	52.732%	3.66%
10-20	28.372%	6.95%
20-40	14.265%	3.741%
40-80	3.741%	0.082%
80-120	0.614%	0.033%
120-160	0.171%	0.017%
160-200	0.067%	0.01%
200-240	0.024%	0.003%
240-280	0.009%	0.006%

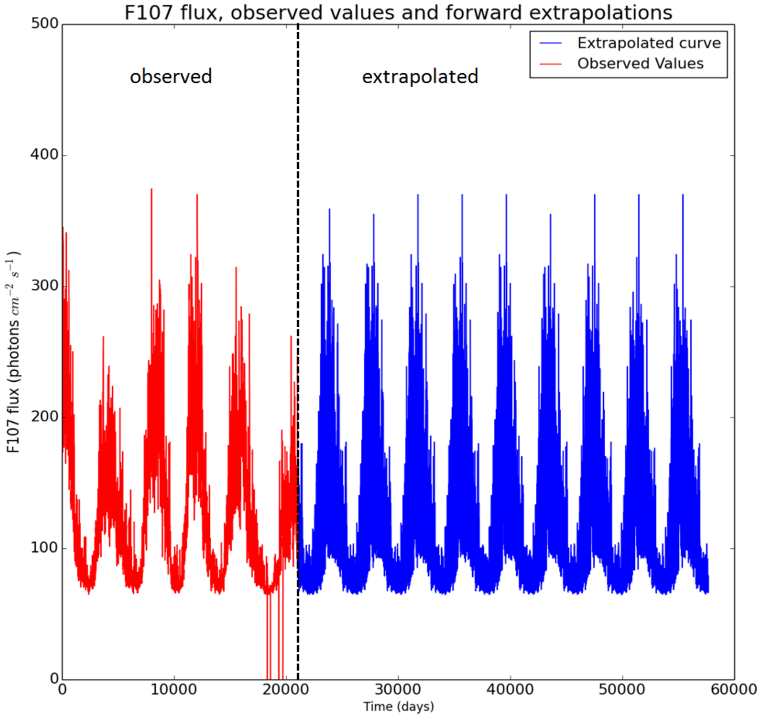

Figure 5: Predicted values of F10.7 up to December 31st 2115. Note our curve appears “thicker” than the observed data. This is likely due to the fact that our data varies more than the observational data, given that it was produced by a statistical technique. For instance, the value of F10.7 on any given day has no bearing on the value the following day, which is unlikely to be the case in reality. For a similar reason, our data contains no relatively quiet Solar maxima, despite two appearing in the observational data (including the current one). This is because values from more active maxima are more likely to be selected. So even if a value from a quiet maximum is selected for one day, the next day, a value from an active maximum will likely be drawn.

3.3 Example simulations

In the following section we provide three examples of data products generated by the current software implementation. For all cases, we use the environmental data as described in the last section.

Example 1: Decay of the June 2012 debris population, ignoring any future collisions or explosions

Figure 5 shows a plot of the debris population over time. The initial input was the June 2012 debris population from the spacetrack catalogue [23], using steps one to three from section 3.2.1 for data conditioning. Collision and explosion functionality had been disabled for this specific simulation run. Hence, figure 6 shows the natural decay of debris objects. The periodic changes in decay rate are a result of the influence of the solar cycle on the drag caused by the upper atmosphere.

Example 2: LEO population for optimistic "New Space" scenario including collisions and explosions

For the data plotted in figure 7, full collision and explosion functionality of the code are enabled. The initial population is using the space-track catalogue from June 2015 and full data conditioning as detailed in steps 1 to 4 in section 3.2.1. On top of that initial population we are introducing additional objects to the population over time, based on the results of the projections described in section 3.2.2. This assumes a thriving "New Space" economy, fulfilling currently announced plans for new satellite constellations and the required replenishing launches to keep them operational. This accounts for approximately 900 launched spacecraft per year. In this (singular) simulation run, the first collision occurs in late 2023. It involves two rocket bodies at 850 km, producing over 5000 pieces of debris. Further major collisions occur in late 2029, early 2030, 2034 and 2035.

Example 3: Number of LEO conjunctions for optimistic"New Space" scenario

This example uses the same scenario as in example 2. Figure 8 shows the number of conjunctions with a probability of collision larger than 10^{-4} in each year. There is an obvious increase over the years. The absolute maximum in 2023 is caused by follow up conjunctions after the initial collision.

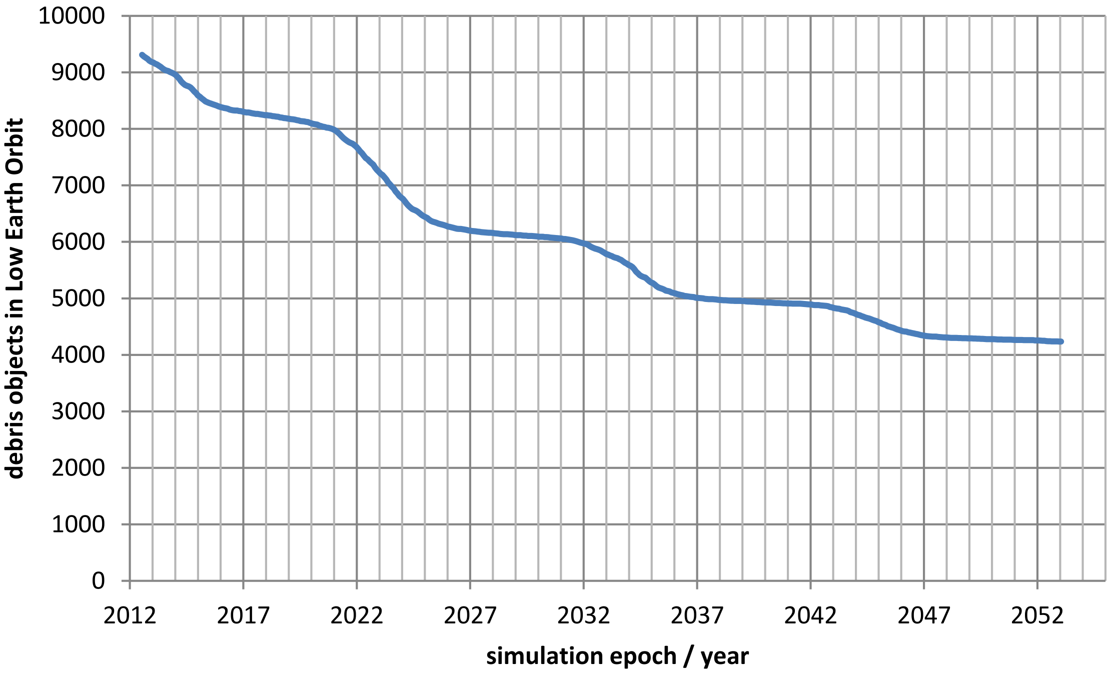

Figure 6: Number of debris objects in Low Earth Orbit over time. Collisions and explosions are disabled.

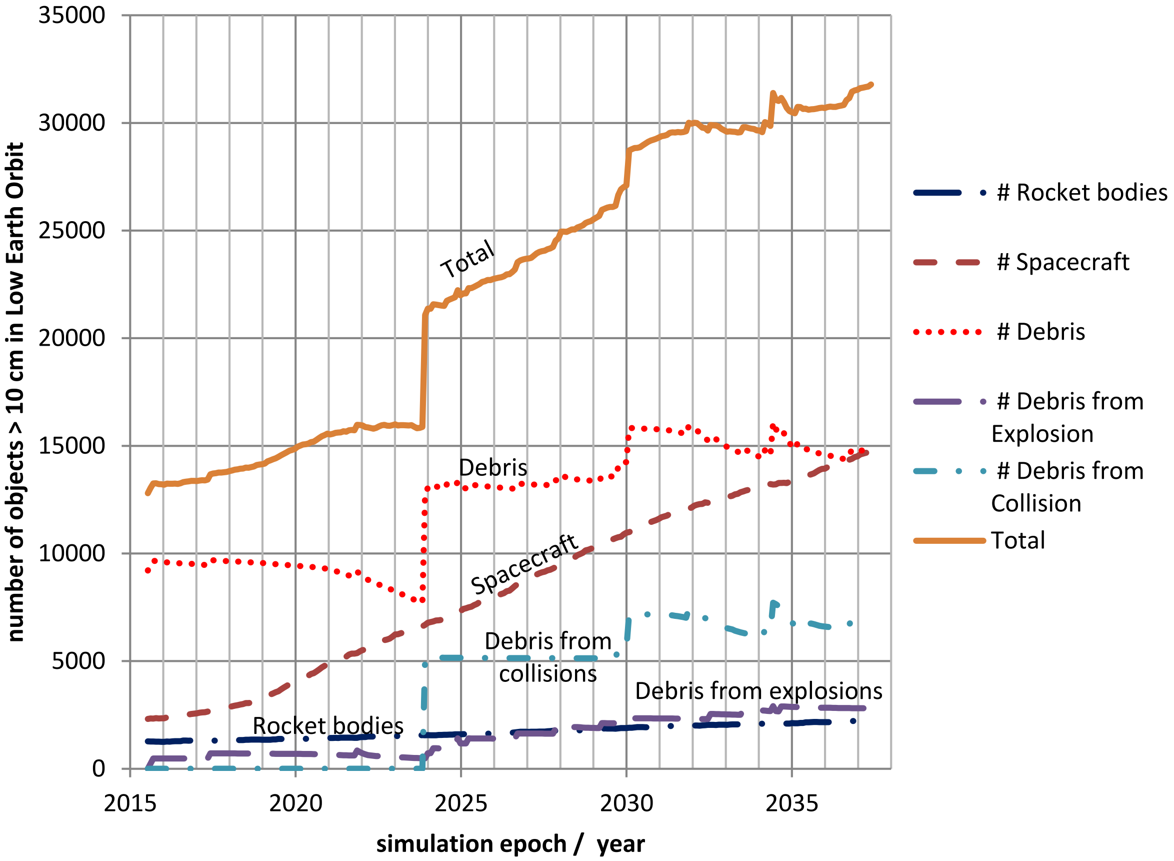

Figure 7: Number of debris objects in Low Earth Orbit over time for a single simulation run using the optimistic "New Space" scenario.

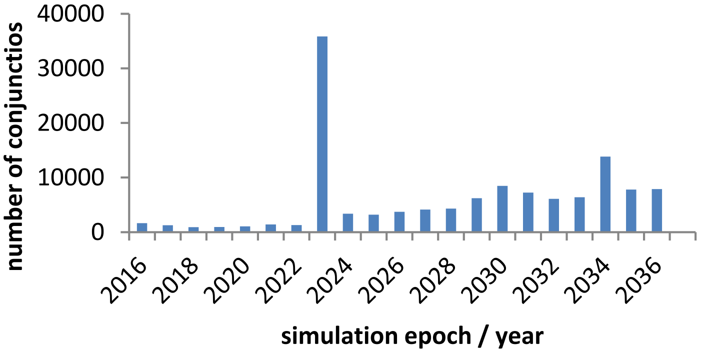

Figure 8: Number of conjunctions with a probability of collisions larger than 10^{-4} for one run of the optimistic "New Space" scenario.

4. NEXT STEPS

While single runs already provide interesting information, e.g. about the total number of conjunctions in a certain debris environment, error bounds and average projections can be obtained with a full Monte Carlo treatment. Implementing that will be our main task for the future.

Further steps aim to increase flexibility in scenario development. We want to:

- Enable execution of predefined trajectories (e.g. for launches or de-orbit maneuvers),
- Enable changing object properties at given times (e.g. to simulate attachment of tethers or drag enhancement devices), and
- Increase the number of objects that can be simulated to enable the assessment of an environment with more particles or with broader knowledge of smaller particles.

5. CONCLUSION

We described our chosen methodology for a flexible, open scenario space debris simulation software, as well as its current status of implementation. We are using a highly parallel simulation approach that utilizes NASA's Pleiades supercomputing cluster. The current implementation propagates all objects in the simulation with a high precision propagator. Time steps are chosen dynamically, and determined by the chosen relative tolerance setting. We further described our approach for the development of input scenarios that includes both the current space environment, including spacecraft, rocket bodies and debris and also projects future launches. For the future launches we detailed one specific scenario that assumes a successful development of the current "New Space" economy. We provided examples of data outputs of the current software suite.

6. Acknowledgments

We would like to thank our colleagues and the center management at NASA Ames Research Center for continuing support. We also would like to thank the NAIF SPICE team at JPL, especially Nat Bachman for providing a long term version of their Earth orientation file. We thank Cyrus Foster both for his past contributions, implementing the original propagator, and for useful discussions over the last months. Special thanks go to Wang Ting (Princeton) for sharing his area-to-mass ratio database and for providing insight in a past implementation of the EVOLVE breakup model. For useful discussions we would like to thank Gene Stansbery, Paula Krisko, Carsten Wiedemann, Jonas Radtke and Holger Krag. We thank Andrew Shacker for providing useful input for future versions of the software.

7. REFERENCES

1. Ailor, W.; Womack, J.; Peterson, G.; Lao, N." Effects of Space Debris on the Cost of Space Operation." *61st International Astronautical Congress*; Prague; Czech Republic, 29. Sept-1 Oct. 2010; IAC-10.A6.2.10, 7p.
2. Liou, J.-C. and Johnson, N. "Instability of the present LEO satellite populations." *Adv. Space Res.*, 41, 1046-1053, 2008.
3. Kessler, D. and Cour-Palais, B. "Collision frequency of artificial satellites: The creation of a debris belt." *J. of Geophys*. Res., 83(A6), 2637–2646, 1978.
4. Inter-Agency Space Debris Coordination Committee, "Stability of the Future LEO Environment", IADC-12-08, Rev. 1, January 2013
5. McKnight, D.S; Di Pentino, F; Kaczmarek, A.; Dingman, P.,"Sstem Engineering Analysis of Derelict Collision Prevention Options", 63rd International Astronautical Congress, Naples, IT, 2012l IAC-12-A6.5.2.
6. Mason, J.; Stupl J.; Marshall, W.; Levit, C. "Orbital debris–debris collision avoidance." *Advances in Space Research* Vol. 48 No. 10 (2011): 1643-1655.
7. Stupl, J.; Mason, J.; Marshall, W.; Levit, C.; Smith, C.; Olivier, S.; Pertica. A.; De Vries, W."LightForce: An Update on Orbital Collision Avoidance Using Photon Pressure." *63rd International Astronautical Congress*; Naples, Italy; 1-5 Oct. 2012; IAC-12.A6.5.11.p1.x14662.
8. Stupl, J.; Faber, N.; Foster, C.; Yang Yan, F.; Levit, C.; "LightForce Photon-Pressure Collision Avoidance: Efficiency Assessment on an Entire Catalogue of Space Debris"AMOS 2013,10-13 September 2013,Wailea, HI

9. Stupl, J.; Faber, N.; Foster, C.; Yang Yang, F.; Nelson, B.; Aziz, J.; Nuttall,A.; Henze, C.; Levit; C., “LightForce Photon-Pressure Collision Avoidance: Updated Efficiency analysis Utilizing a Highly Parallel Simulation Approach”, AMOS 2014, September 10-12 2014,Wailea, HI

10. Liou, J. C., Hall, D. T., Krisko, P. H., & Opiela, J. N. (2004),” LEGEND–A three-dimensional LEO-to-GEO debris evolutionary model”, Advances in Space Research, 34(5), 981-986.

11. Liou, J. C., “Collision activities in the future orbital debris environment”, Advances in Space Research, 38(9), 2102-2106.

12. Nikolaev, S.; Phillion, D.; Springer, H.K.; de Vries,W.; Jiang,M.; Pertica, A.; Henderson, J.; Horsley, M.; Olivier, S. “Brute Force Modeling of the Kessler Syndrome”, *AMOS 2012*.

13. Dormand, J. R.; Prince, P. J. (1980), "A family of embedded Runge-Kutta formulae", *Journal of Computational and Applied Mathematics* 6 (1): 19–26

14. Hairer, E.; Nørsett, S.P.; Wanner, G. “Solving ordinary differential equations I: Non stiff problems.” New York 2008.

15. Acton, C.H. "Ancillary data services of NASA's Navigation and Ancillary Information Facility." *Planetary and Space Science* 44.1 (1996): 65-70.

16. Lemoine, F.G.; Smith, D.E.; Kunz, L.; Smith, R.; Pavlis, E.C.; Pavlis, N.K.; Klosko, S.M.; Chinn, D.S.; Torrence, M.H.; Williamson, R.G.; Cox, C.M.; Rachlin, K.E.; Wang, Y.M.; Kenyon, S.C.; Salman, R.;Trimmer, R.; Rapp, R.H.; Nerem, R.S."The development of the NASA GSFC and NIMA joint geopotential model." *Gravity, Geoid and Marine Geodesy*, International Association of Geodesy Symposia Nr 117 (1997): 461-469.

17. Folkner, W.M.; Williams, J.G.; Boggs, D.H. "The planetary and lunar ephemeris DE 421." IPN Progress Report (2008): 42-178. http://tmo.jpl.nasa.gov/progress_report/42-178/178C.pdf

18. Picone, J. M.; Hedin, A.E; Drob, D.P.; Aikin, A.C. "NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues." *Journal of Geophysical Research: Space Physics* (1978–2012) 107.A12 (2002): SIA-15.

19. Vallado, D.A. “Fundamentals of Astrodynamics and Applications”, New York 2013, 4th edition.

20. Johnson, N. L., Krisko, P. H., Liou, J. C., & Anz-Meador, P. D. (2001), “ NASA's new breakup model of EVOLVE 4.0”, Advances in Space Research, 28(9), 1377-1384.

21. Krisko, P., "Proper implementation of the 1998 NASA breakup model.", Orbital Debris Quarterly News 15.4 (2011): 1-10.

22. Patera, R.P. "*General method for calculating satellite collision probability*." *Journal of Guidance, Control, and Dynamics* 24.4 (2001): 716-722.

23. www.space-track.org

24. Levit, C. and Marshall,W. "Improved orbit predictions using two-line elements." *Advances in Space Research* Vol. 47 No.7 (2011): 1107-1115.

25. ESA/ESOC, “ESA register of Objects in Space EsaROS”, editor H. Klinkrad, Darmstadt 2008

26. Barton, D.K; Brillinger, D.; El-Shaarawi, A.H.; McDaniel, P.; Pollock, K.H.; Tuley, M.Y. “Final Report of the Haystack Orbital Debris Data Review Panel”, NASA Technical Memorandum 4809, Feb. 1998.

27. NASA Orbital Debris Program Office, “HISTORY OF ON-ORBIT SATELLITE FRAGMENTATIONS”, 14th Edition, June 2008, NASA/TM-2008-214779.

28. http://celestrak.com/SpaceData