

# Development and Analysis of the Automated Object Reentry Survival Analysis Tool Parametric Study Wrapper

Andrew N. Smith<sup>(1)</sup>, Benton R. Greene<sup>(2)</sup>

<sup>1</sup>Jacobs ESSCA Contract, NASA MSFC, 620 Discovery Drive NW, Suite 140, Huntsville, AL 35808, USA,  
andrew.n.smith@nasa.gov

<sup>2</sup>Jacobs JETS Contract, NASA Johnson Space Center, Mail Code XI4-9E, Houston, TX 77058, USA

## ABSTRACT

The NASA Orbital Debris Program Office (ODPO) Safety Group at the Johnson Space Center analyzes reentering spacecraft at the end of life. The program primarily used by ODPO in this effort is the Object Reentry Survival Analysis Tool (ORSAT). ORSAT utilizes shape primitives as well as a variety of other parameters (material, size, thickness, aerodynamic mass, orbit inclination, etc.) to simulate the reentry process and ultimately, to determine if a spacecraft could be hazardous to the population on the ground.

The NASA ODPO plans to automate the ORSAT process to run multiple ORSAT input files either concurrently or consecutively. This type of automation program will provide several benefits. First, there is a need to run large parametric studies for ORSAT analysts to gain a greater understanding of reentering object's sensitivity to certain input variables. Secondly, a database of pre-run ORSAT cases will be used to develop a survivability model, which could be made available to spacecraft developers as a design for demise (D4D) tool.

The recently completed Automated Object Reentry Survival Analysis Tool (AutoORSAT) Wrapper is currently being used to build a survivability database, the first step in developing a survivability model. Already, the data that AutoORSAT has produced provides a greater understanding of the sensitivity of variables such as the initial temperature of the spacecraft, spacecraft breakup altitude, and the aerodynamic mass of spacecraft.

## 1 Introduction

The NASA Orbital Debris Program Office (ODPO) located at the NASA Johnson Space Center in Houston, Texas, actively works on analyzing reentering spacecraft at the end of their mission. The ultimate goal of this work is to determine if the object will demise during reentry or survive and pose a risk to the population on the ground. All spacecraft launched or operated by U.S. entities must demonstrate that the ground casualty risk from uncontrolled reentry of the spacecraft is below 1:10,000. The Object Reentry Survival Analysis Tool (ORSAT) was developed to aid in the reentry analysis. ORSAT assesses the effects of trajectory, atmosphere, aerodynamics, aeroheating, and temperature/ablation to determine which, if any, spacecraft components survive to the ground with a dangerous impact kinetic energy [1]. If an object does not fully demise in the reentry process, ORSAT calculates a predicted debris casualty area and the footprint of the impact to be used in a risk assessment calculation.

ORSAT models a spacecraft as a collection of shape primitives (box, cylinder, sphere, etc.) that are all released at a standard breakup altitude of 78 km. The external dimensions, thickness, material, and aerodynamic mass of each fragment model are included in an ASCII input file, along with the initial reentry conditions (altitude, velocity, inclination, etc.) of the parent spacecraft. Table 1 provides a list of the variables included in the ORSAT input file.

Table 1. ORSAT Spacecraft Input Variables

Parameter	Domain	Notes
Shape	Spacecraft	*Sphere, Cylinder, Box, Flat Plate, Disk, Cone, Frustum, Ring
# Materials	Spacecraft	Defines number of material layers for each object
Material	Spacecraft	There are 80 predefined materials, additional materials can have properties read in
Aero. Dia.	Spacecraft	Aerodynamic Diameter of object, used for trajectory calculations
Therm. Dia.	Spacecraft	Thermodynamic Diameter of object, used for heating calculations
Aero. Length	Spacecraft	Aerodynamic Length of object, used for trajectory calculations
Therm. Length	Spacecraft	Thermodynamic Length of object, used for heating calculations
Aero. Height	Spacecraft	Aerodynamic Height of object, used for trajectory calculations
Therm. Height	Spacecraft	Thermodynamic Length of object, used for heating calculations
Inner Radius	Spacecraft	Inner radius of object, effectively defining thickness of the object
Aero. Mass	Spacecraft	Cumulative mass of the shape primitive and any objects inside of it
Therm. Mass	Spacecraft	Mass of the shape primitive
Initial Temp.	Spacecraft	Temperature shape is at when it is initially exposed to flow
Altitude	Flight Path	Initial altitude for when re-entry "begins" (typically 122 km)
Velocity	Flight Path	Initial Orbital Velocity of spacecraft
Flight Path Angle	Flight Path	Angle of spacecraft's velocity vector relative to the local horizon at the beginning of reentry
Inclination	Flight Path	Orbital Inclination of Spacecraft
Ascending/Descending	Flight Path	Specifying if spacecraft is in ascending or descending portion of orbit
Latitude	Flight Path	Initial latitude at time of reentry
Longitude	Flight Path	Initial longitude at time of reentry
# Children	Spacecraft	How many objects are inside the parent object

\* Some shapes have multiple reentry orientations (ex. End-On Spinning and Random Tumbling)

ORSAT calculates both trajectory and heating profiles through the reentry process using the input fragment geometry and initial trajectory parameters. If the accumulated heating exceeds the heat of fusion of the material plus the sensible heat required to raise the temperature to the melting point, it is assumed that the object demises, subjecting the children components inside that object to flow. This process continues until either all of the spacecraft has demised during reentry, or some objects impact the ground. If the object demised, ORSAT will store the trajectory data for how far downrange, and at what altitude each object demised. For objects that impact ORSAT stores how far downrange the object impacted, as well as its impact velocity, impact energy and debris area. Objects with impact kinetic energy greater than 15 J are considered hazardous and are included in the total debris casualty area for the spacecraft. Currently, ORSAT is limited to a single nesting layer with a parent object and a list of child objects. For spacecraft with multiple nesting components, each sub-level component must be manually copied into a separate ORSAT run.

While ORSAT is a powerful tool, enclosing the relatively simple ORSAT program within an automated wrapper provides many advantages. The current work involves developing just such an automated wrapper, called AutoORSAT. One key area of improvement would be the speed at which multiple variations of an ORSAT input file could be run. Prior to the AutoORSAT wrapper development, ORSAT analysts had to manually create each input file and run them individually. While this process is acceptable when only a few runs must be completed, the process gets extremely tedious when many ORSAT runs must be completed at once. For this reason, systematic parametric studies for all ORSAT input variables have never been run.

The ability to run parametric studies of arbitrary ORSAT variables has allowed NASA ODPO to begin development of two new capabilities based on AutoORSAT: Monte Carlo reentry risk analysis for spacecraft and a demisability database for use as an aid in design for demise (D4D).

## 2 AutoORSAT Structure

The AutoORSAT process begins with the Wrapper Input file. This file is an Excel spreadsheet-based input deck that defines the iterations for each variable that ORSAT will run. The input file is shown below in Fig. 1.

# AutoOrsat Wrapper Input Data

## Control Variables and Summary

DROP DOWN AND SINGULAR VARIABLES				
Variable	Value	Value 2	Value 3	Description
TSTART	0	-	-	Click for Info
TSTOP	99999	-	-	Click for Info
IPRT	100	-	-	Click for Info
NFRAG	0	-	-	Click for Info
DT	0.005	-	-	Click for Info
ISCREEN	1	-	-	Click for Info
IPRTHD	0	-	-	Click for Info
IPTRAJ	1	-	-	Click for Info
IPHEAT	1	-	-	Click for Info
IPTEMP	1	-	-	Click for Info
IPAERO	1	-	-	Click for Info
IPATM	0	-	-	Click for Info
IDHEAT	0	-	-	Click for Info
IDTEMP	0	-	-	Click for Info
IBATCH	0	-	-	Click for Info
IENG	0	-	-	Click for Info
ITARG	0	BLANK	BLANK	Click for Info
IASC	1	BLANK	-	Click for Info
Nesting Type	2	-	-	Click for Info
GMAT Input	0	-	-	Click for Info
Calculate Breakup	0	-	-	Click for Info
Preheating	None	-	-	Click for Info
Auto Parent Mass	1	-	-	Click for Info

RANGE VARIABLES				
Variable	Max	Min	Iterations	Description
ALT	122000	122000	1	Click for Info
VEL	7356.1	0	2	Click for Info
GAMMA	-0.1	-0.1	1	Click for Info
HAP	0	0	1	Click for Info
HP	0	0	1	Click for Info
INC	0.1	0.1	1	Click for Info
LAT	0	0	1	Click for Info
LON	0	0	1	Click for Info
DWNRANG	0	0	1	Click for Info

SUMMARY	
Total ORSAT Runs	2
Nesting Levels	0
Nesting Strings	0

Apply Multiple Range Values

Apply

Min Value

Max Value

Iterations

Variable

Apply values to selected:

Fragment Sheet Controls

Reset Fragments

Fig. 1. AutoORSAT wrapper input spreadsheet. Shows the main control worksheet with atmospheric entry conditions and computation flags. Worksheet also includes VBA macros to automate certain data entry tasks.

Each ORSAT input variable is represented within the spreadsheet. The first page, shown in Fig. 1, contains the spacecraft entry conditions and the computation flags that enable or disable items like higher-fidelity heating models or which atmospheric model to use. Each spacecraft component is then defined using a separate worksheet of the format shown in Fig. 2. Variables that have only a small number of discrete values, like component material and certain computation flags, have up to five slots to input different values for the variable. Variables that have a numeric value like mass and dimensions are defined by a range denoted by a maximum and minimum value and a number of points between them.

Parent	
Name:	Parent Object

RANGE VARIABLES				
Variable	Max	Min	Iterations	Description
ASTOP	78000	78000	1	Click for Info
KKMAX	1	1	1	Click for Info
INN(1)	1	1	1	Click for Info
IMAT Property 1(1)	0	0	1	Click for Info
IMAT Property 2(1)	0	0	1	Click for Info
THICKNESS (1)	0.01	0.01	1	Click for Info
DAERO	1	1	1	Click for Info
DHEAT	1	1	1	Click for Info
LAERO	0	0	1	Click for Info
LHEAT	0	0	1	Click for Info
HAERO	0	0	1	Click for Info
HHEAT	0	0	1	Click for Info
MASS	100	100	1	Click for Info
THMASS	0	0	1	Click for Info
FACT	1	1	1	Click for Info
TAU	0.5	0.5	1	Click for Info
TINIT	300	300	1	Click for Info
CL	0	0	1	Click for Info
IAERO Property 2	0	0	1	Click for Info

MULTI MATERIAL RANGE VARIABLES				
Variable	Max	Min	Iterations	Description
INN(2)	0	0	1	Click for Info
IMAT Property 1(2)	0	0	1	Click for Info
IMAT Property 2(2)	0	0	1	Click for Info
THICKNESS (2)	0	0	1	Click for Info
INN(3)	0	0	1	Click for Info
IMAT Property 1(3)	0	0	1	Click for Info
IMAT Property 2(3)	0	0	1	Click for Info
THICKNESS (3)	0	0	1	Click for Info
INN(4)	0	0	1	Click for Info
IMAT Property 1(4)	0	0	1	Click for Info
IMAT Property 2(4)	0	0	1	Click for Info
THICKNESS (4)	0	0	1	Click for Info
INN(5)	0	0	1	Click for Info
IMAT Property 1(5)	0	0	1	Click for Info
IMAT Property 2(5)	0	0	1	Click for Info
THICKNESS (5)	0	0	1	Click for Info
INN(6)	0	0	1	Click for Info
IMAT Property 1(6)	0	0	1	Click for Info
IMAT Property 2(6)	0	0	1	Click for Info
THICKNESS (6)	0	0	1	Click for Info

DROP DOWN VARIABLES						
Variable	Value 1	Value 2	Value 3	Value 4	Value 5	Description
ITYPE	1	BLANK	BLANK	BLANK	BLANK	Click for Info
IMAT Material (1)	5	BLANK	BLANK	BLANK	BLANK	Click for Info
IMAT Structure (1)	0	BLANK	BLANK	BLANK	BLANK	Click for Info
IMAT Material (2)	0	BLANK	BLANK	BLANK	BLANK	Click for Info
IMAT Structure (2)	0	BLANK	BLANK	BLANK	BLANK	Click for Info
IMAT Material (3)	0	BLANK	BLANK	BLANK	BLANK	Click for Info
IMAT Structure (3)	0	BLANK	BLANK	BLANK	BLANK	Click for Info
IMAT Material (4)	0	BLANK	BLANK	BLANK	BLANK	Click for Info
IMAT Structure (4)	0	BLANK	BLANK	BLANK	BLANK	Click for Info
IMAT Material (5)	0	BLANK	BLANK	BLANK	BLANK	Click for Info
IMAT Structure (5)	0	BLANK	BLANK	BLANK	BLANK	Click for Info
IMAT Material (6)	0	BLANK	BLANK	BLANK	BLANK	Click for Info
IMAT Structure (6)	0	BLANK	BLANK	BLANK	BLANK	Click for Info
IAERO Property 1	0	BLANK	BLANK	BLANK	BLANK	Click for Info
IATMOS	1	BLANK	BLANK	BLANK	BLANK	Click for Info
ITHM	1	BLANK	BLANK	BLANK	BLANK	Click for Info
IRAD	0	BLANK	BLANK	BLANK	BLANK	Click for Info
IFAYRID	0	BLANK	BLANK	BLANK	BLANK	Click for Info
IOX	1	BLANK	BLANK	BLANK	BLANK	Click for Info
IRR	1	BLANK	BLANK	BLANK	BLANK	Click for Info

Fig. 2. AutoORSAT wrapper input spreadsheet: spacecraft parent and fragment definition sheets.

Another capability offered by AutoORSAT is the ability to simulate multiple nested components within a spacecraft. Fig. 3 illustrates the three configurations that AutoORSAT provides. The Un-nested case is the standard ORSAT configuration with a single parent object and multiple child objects that are all released at the same time upon demise of the parent object. The Linear Nested mode assumes that each successive child object is not exposed to the reentry environment until the previous object demises, resulting in a spacecraft that resembles a Russian nesting doll. Finally, the Complex Nested mode allows the definition of an arbitrarily complex combination of parent and child objects.

AutoORSAT will take the nesting relationships of all of the components as input, and for each object, take the parent object's demise position and velocity and supply those as input conditions. This process is repeated for each nesting level of the spacecraft model.

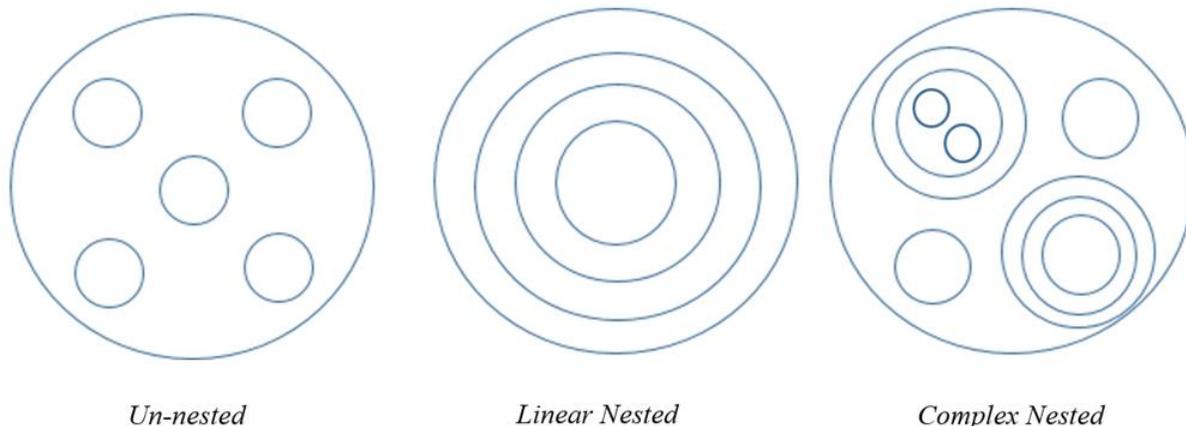


Fig. 3. Different Functionalities of ORSAT Wrapper.

To speed up the computation process, AutoORSAT uses the multiprocessing Pool Python module to run each instance of ORSAT as a separate computation thread. The user can then specify the number of concurrent ORSAT instances to run. In general, the optimum number for computation speed appears to be between 1 and 1 1/2 times the number of available processor cores on the machine.

At the end of a run, AutoORSAT collects all of the input and output information for each unique instance of each fragment, and outputs it to a series of CSV files. Each CSV file contains all of the possible instances for a given fragment defined by the value ranges given for each fragment variable.

For large AutoORSAT simulations, this CSV formatted output can be difficult to post-process due to the large size. A more efficient SQLite output format is being developed that will more efficiently store the information and provide a fast and efficient means of querying and analyzing the results of the simulation.

### 3 AutoORSAT Performance

With AutoORSAT, the time required to modify an input file is automatically done within the wrapper, with the parameters for how each input is to be varied pre-defined within the AutoORSAT input file. This reduces the time between ORSAT runs to just a few milliseconds of processor time to generate the new input file and parse the output of the previous ORSAT run. The wrapper then automatically compiles the output from all parameter variations into a standard set of CSV files to easily correlate parametrized inputs to program outputs.

Typically, an ORSAT run for a single fragment takes about 1 to 5 seconds of computation time depending on whether and how quickly the fragment demises. The latest stable version of AutoORSAT running in a single thread can process 700 to 3000 variations per hour. Using the multiprocessing module of python, AutoORSAT can run an arbitrary number of independent threads. Tests run on the ODPO computing cluster show that two to three threads per processor core provide an optimum computation speed given the relatively intensive disk usage of standard ORSAT. Using a single node of the computer cluster with 16 cores and 32 parallel threads, up to 126,000 ORSAT runs per hour have been achieved, with a typical run computing about 72,000 ORSAT runs per hour on average.

The recently implemented ORSAT 6.2.1 update also includes multiple speed improvements and enhanced AutoORSAT compatibility. In addition to an increase in the computation speed of ORSAT itself, almost all input/output has been moved to the *stdin* and *stdout* interfaces, allowing communication between ORSAT and AutoORSAT without the use of input and output text files, almost eliminating disk usage during computation. Testing of the latest version of AutoORSAT designed to take advantage of these speed improvements shows a doubling in the average number of ORSAT runs per second.

## 4 Survivability Database

### 4.1 Database Structure

The survivability database currently exists as a comma-separated value text file containing the input conditions and final state of each fragment scenario. As the number of fragment scenarios calculated increases; however, this method of data storage becomes increasingly cumbersome. Work is ongoing to implement the database in SQLite, which will improve searchability, data storage efficiency, and portability.

The output of AutoORSAT is added to the database using a separate script that reads the output file structure of AutoORSAT and copies the relevant fields into the database file.

Each entry in the database represents a fragment with a given starting geometry, mass, altitude, latitude, longitude, velocity, and heading. All of the inputs to the ORSAT run are included, and the demise altitude, impact velocity, impact kinetic energy, final mass, cross-sectional area, and debris casualty area are all saved as outputs.

### 4.2 Preliminary Results

When finished, the Survivability Database will provide a convenient tool for determining the sensitivity of a given fragment's reentry survival to various parameters such as ballistic number, material, release altitude, release flight path angle, and release velocity. With the limited initial data available in the database currently, some broad conclusions can be made regarding the sensitivity of reentry survivability to some select variables.

Currently, the database contains 109,000 entries for fragments made of aluminum, titanium, copper, acrylic, and stainless steel. A breakdown of the representation of each type of material is shown in Fig. 5. Fig. 4 shows the distribution of starting altitude and starting velocity for every fragment in the database.

Currently, the altitude of 122 km is highly over-represented because it is the standard "entry" altitude. A typical spacecraft in a near-circular orbit with an altitude below this value will typically reenter in less than one full orbit. Because entries are made in the database from full AutoORSAT runs of parent spacecraft and child fragments, the parent spacecraft, which by standard practice starts at 122 km altitude, comprise a significant fraction of the database. For purposes of the preliminary analysis later in this section, these entries are filtered out.

Fig. 4 also shows the distribution of initial velocity for each of the fragment entries in the database. The velocities are clustered between 6 km/s and 8 km/s. This is the result of the natural deceleration of the parent body before the fragments are released.

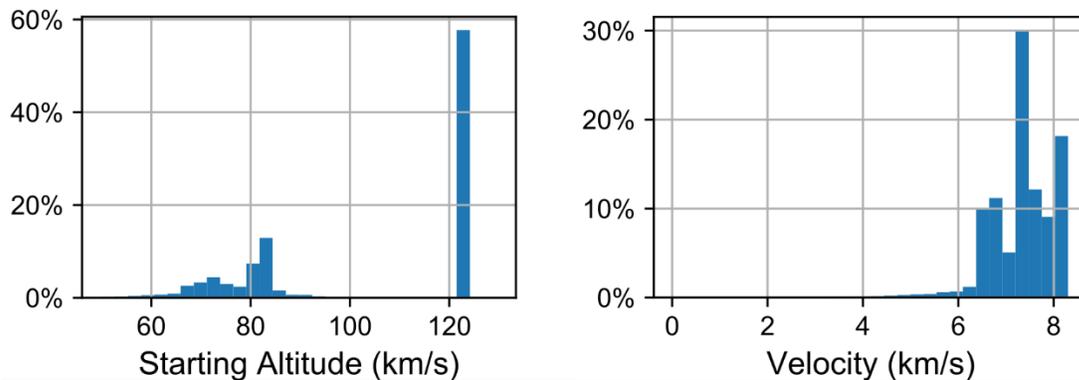


Fig. 4. Histogram of starting altitude and starting velocity for fragments in survivability database.

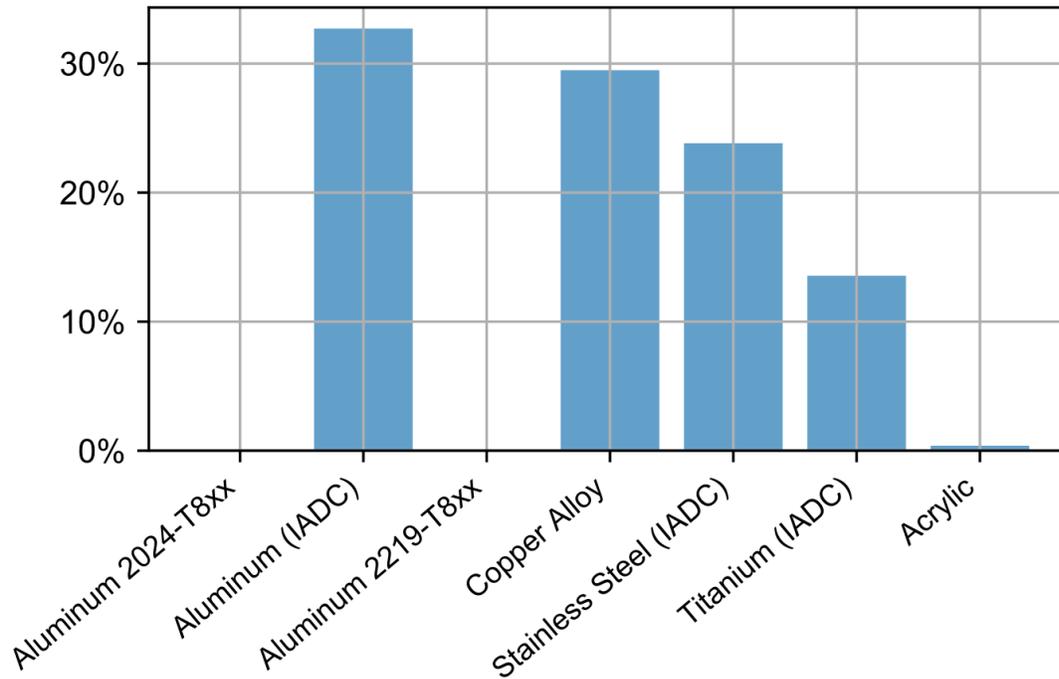


Fig. 5. Current representation of material types in survivability database.

The authors have performed some preliminary analysis of the fragments currently represented in the database. These results are very limited, as the fragment sizes included in the database are currently restricted to objects of approximately 0.2 m in diameter. Additionally, only an initial orbit inclination of 35 degrees is represented.

Given these caveats, a general trend of fragment demisability vs starting altitude can be extracted for fragments made of copper and aluminum, given in Fig. 6 and Fig. 7, respectively. As expected, the demisability of a fragment made of either material decreases as the release altitude decreases, with very few fragments demising when released below 70 km. However, the figures show that copper fragments demise more readily than aluminum fragments of comparable size. A possible explanation for this effect is the high density of copper. This increases the ballistic coefficient, and therefore the length of time the fragment is exposed to peak heating.

A bi-modal trend can also be seen in Fig. 7, with two distinct trends in fragment survival vs release altitude. This is due to different initial temperatures for the fragments. A higher initial temperature results in a lower fragment survival rate.

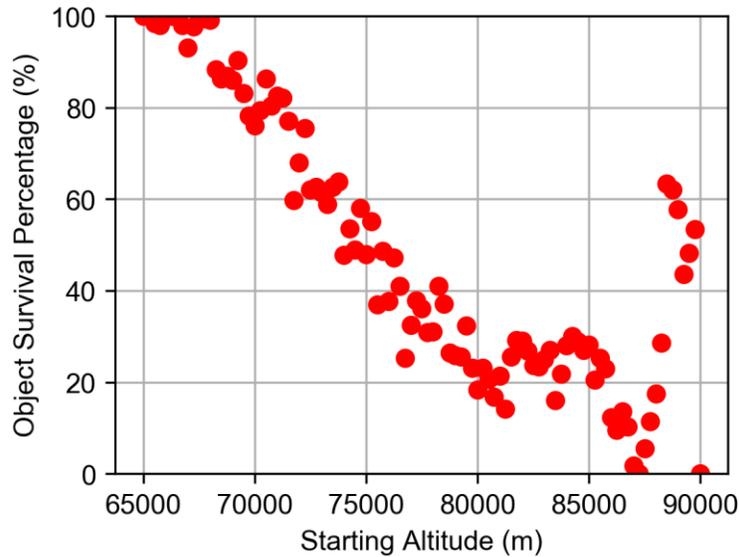


Fig. 6. Effect of release altitude on survival of copper fragments.

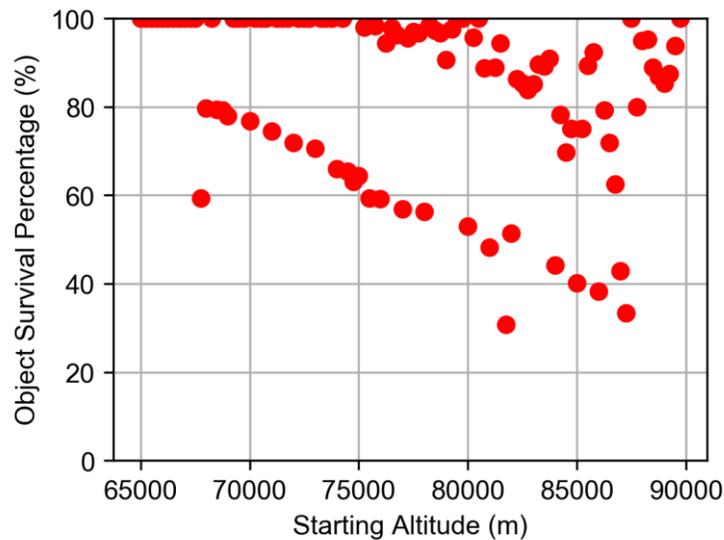


Fig. 7. Effect of release altitude on survival of aluminum fragments.

## 5 Monte Carlo Risk Analysis

Traditionally, ORSAT is run on a single case basis, using a set of standard inputs, to determine a value for the ground casualty risk of a given spacecraft. Some variation in input parameters can be studied either by using the limited built-in parametric study feature of ORSAT 6.0 or by manually editing the input text file each time to run multiple cases. This is a tedious process and only allows for simple one- or two-variable parametric studies for a small number of spacecraft fragments. These sorts of parametric studies therefore are used only when something in the risk model is marginal and needs more study to solidify the ORSAT analyst's confidence in the model.

In recent years, work by the ODPO has shown that variation in time of year and latitude of reentry can have a significant effect on the ground casualty risk of a reentering spacecraft [2-3], and that atmospheric effects can bias reentries toward the equator and greater average population density, increasing the ground casualty risk [4].

Using the AutoORSAT wrapper, large parametric studies of spacecraft demisability models can be implemented and run in only a few hours. This vast increase in capability has allowed the ODPO to investigate the use of Monte Carlo

analysis to obtain a clearer and more accurate picture of the real ground casualty risk of a reentering spacecraft. Rather than using a single standard entry assumption, AutoORSAT can use a realistic statistical model of the entry conditions to calculate a ground casualty risk that incorporates uncertainty and small biases like those identified in references 2, 3, and 4.

Development of appropriate statistical models for inputs to the Monte Carlo analysis and ways to meaningfully interpret the output of the analysis are still being developed, and must be well-characterized and approved before they can be used in official reentry risk analysis.

## **6 Conclusion**

The NASA ODPO at the Johnson Space Center utilizes ORSAT to analyze re-entering spacecraft and the likelihood of spacecraft fragments surviving to impact. While ORSAT is a very powerful analysis tool, it historically has not been able to be run quickly, which limits the capabilities of the analysis, and makes parametric studies difficult. There was a need to automate large-scale parametric studies that would aide in the development of a survivability database. AutoORSAT was developed to fill this role.

The AutoORSAT input is a “construction” of the spacecraft’s components from shape primitives, as well as the relevant orbital parameters. From this point, AutoORSAT develops all of the necessary ORSAT input files, and through multi-threading can run a large number of ORSAT cases at once. On the ODPO’s Beowulf cluster, over 100,000 ORSAT runs per hour have been achieved, allowing for the possibility of Monte-Carlo reentry analysis of spacecraft and large-scale sensitivity studies of various spacecraft reentry parameters.

AutoORSAT currently is used to develop a fragment survivability database for use in D4D. When the database is adequately populated, it will be possible to determine the likelihood of an objects demise when only a handful of parameters of that spacecraft are known. This will allow a spacecraft designer to incorporate D4D into the early stages of the design process and thereby improve the reentry safety of future missions.

As a new tool for the NASA ODPO, AutoORSAT’s computation capabilities will significantly improve our understanding of the spacecraft reentry process.

## **7 Future Work**

Work on the Demisability Database is ongoing, and a relational SQLite database structure is under development to replace the current CSV file format, which will allow much easier correlation of fragment inputs to demisability. Once the SQLite database is complete, efforts will be redirected toward expanding the database content. In parallel with this effort, an easy-to-use graphical front-end will be written for the database to allow a spacecraft designer to easily gauge the demisability of a part during the design phase of a mission, allowing for better integration of D4D into the spacecraft design process.

In addition to generating a demisability database, AutoORSAT is being used to study the feasibility of employing a Monte Carlo risk analysis model to determine ground casualty risk for a given spacecraft. Rather than use a single standard input, a realistic range of input variables can be used to cover the most probable spacecraft entry and breakup conditions and arrive at the most likely ground casualty risk. Research is ongoing into the most probable range of entry conditions [2-3], as well as into the most appropriate way to interpret the probabilistic results of the analysis.

## **Acknowledgments**

The authors would like to thank Mr. Christopher Ostrom, Mr. Christopher Sanchez, Dr. John Bacon, Dr. Heather Cowardin, Dr. Jer-Chyi Liou and the rest of the NASA ODPO at Johnson Space Center. Their support and expertise in Orbital Debris reentry were invaluable to the development of the AutoORSAT wrapper.

## **8 References**

1. Orbital Debris Program Office. “User’s Guide for Object Reentry Survival Analysis Tool (ORSAT) – Version 6.0”, JSC-62861, NASA JSC, Houston, Texas, 2006.
2. Ostrom, Chris. “Effect of Latitude Bias in Entry Angle on Ground Casualty Risk from Naturally Decaying Space Objects”, *10<sup>th</sup> IAASS Conference: Making Safety Happen*, El Segundo, CA. May 2019.

3. Bacon, John B. "Seasonal- and Beta-Angle-Dependent Latitude Bias Variations in Natural Decays", *10<sup>th</sup> IAASS Conference: Making Safety Happen*, El Segundo, CA. May 2019.
4. Bacon, John B. and Matney, Mark. "Statistical Issues for Calculating Reentry Hazards", *8<sup>th</sup> IAASS Conference: Safety First, Safety for All*, Melbourne, FL, May 2016.