Application of density-based propagation to fragment clouds using the Starling suite

S. Frey[(1)], **C. Colombo**[(1)], **and S. Lemmens**[(2)]

[(1)]Politecnico di Milano, Via La Masa 34, 20156 Milano, Italy, stefan.frey/camilla.colombo@polimi.it
[(2)]Space Debris Office, ESA/ESOC, Robert-Bosch-Strasse 5, 64293 Darmstadt, Germany; stijn.lemmens@esa.int

ABSTRACT

The Starling suite estimates the non-linear evolution of densities in orbit. These densities can be probability density functions describing uncertainties about states or distributions of particle clouds modeled as continua. In this work, the Starling suite is applied to satellite fragmentations in a circular as well as highly eccentric orbit around Earth. The former case is challenging because the fragment distribution is bounded. Such boundedness is tackled using feature spaces defined relative to propagated reference hypersurfaces. The latter case is challenging as assumptions of the randomization in the node and argument of perigee do not hold over the mid-term. The suite also allows to estimate the impact rate from a fragment cloud on missions in any orbital region. Estimating the consequences of fragmentations is important as it can be used to rate planned space missions in terms of risks towards the future space environment.

1 INTRODUCTION

More than 500 on-orbit fragmentations have so far occurred due to various reasons including explosions and collisions [1]. Explosions of spacecraft and upper stages and collisions with other objects potentially release hundreds of thousands of fragments with a diameter larger than 1 mm [2]. To estimate the long-term consequences of such fragment clouds on active space missions in terms of impact rates and collision risk, a diverse set of tools were developed. Engineering tools, such as NASA's LEGEND [3], rely on Monte Carlo sampling to estimate the number of collisions from space debris. They are non-invasive and thus work with any existing propagator in any force model for a wide range of orbit geometries. However, to derive an accurate density estimate in $6+$ dimensions over the full domain requires a prohibitively large number of samples to be drawn and propagated. Other tools directly propagate densities through volume mapping [4] or application of the continuum equation [5]. Their analytical implementations [6] require simplified force models and restricted orbital geometries. Numerical solutions [7, 8, 9] can deal with any dynamics but the result is known only following the characteristics of the system. Solutions based on finite-differences [10, 11, 12, 13], instead, model the distribution over the full domain but are difficult to extend into more than 3 dimensions due to restrictions in terms of computational power and memory. Hence, a method combining the best of all worlds to be fast, applicable for many force models and orbital geometries and extendable into many dimensions is desirable.

Here, a novel tool, the Starling suite, is introduced to estimate evolving continua subject to non-linear dynamics. The suite is being developed at the Politecnico di Milano and funded by the COMPASS European Research Council project and the European Space Agency. The name is inspired by the murmurations of starlings, a spectacle of nature where millions of birds fly together in perfect synchrony, seemingly as one ever-changing continuum. It is based on the continuity equation and uses numerical propagation to allow incorporation of accurate force models and modeling of densities in any orbit. To estimate the distribution across the full domain without requiring a large number of propagations, Starling fits a surrogate model to the characteristics. The suite was designed to allow working in any dimensionality and with easy transitioning between feature spaces. An additional plug-in estimates the rate of impacts an evolving fragment cloud has on targets in any orbital regime. Two examples of fragmentation cloud evolution are given in 2 and 5 dimensions, respectively. The former shows the usage of reference hypersurfaces to normalize the feature space and facilitate the fitting of the surrogate model. It also shows the application of the impact rate estimation tool. The later shows the potential of the method of propagating clouds in many dimensions while keeping the computational resources required low.

This work is organized as follows: Sec. 2 and 3 give an overview of the theoretical background of the method and discuss limitations. The tool itself is described in Sec. 4. Use cases of the tool, propagation of a fragment clouds and estimation of the collision risk, are presented in Sec. 5. Finally, conclusions are drawn in Sec. 6.

2 CONTINUUM FORMULATION

The uncertainty and/or fragment cloud distribution are modeled as density functions. Uncertainties are probability density functions, i.e. integration over the full domain results in unity. Fragment distributions, instead, are particle density functions. Integration over the full domain results in the number of fragments, N, that are present in the cloud. The method presented here does not distinguish between the two types of densities, and both are simply referred to as density function, $n_{\boldsymbol{x}}$, in the D-dimensional phase space, $\boldsymbol{x} \in \mathbb{R}^D$.

The following subsections discuss the initial condition and transformation (Sec. 2.1), the propagation of the continuum (Sec. 2.2), how to find surrogate models at different snapshots in the future (Sec. 2.3), as well as current limitations to the approach (Sec. 2.4).

2.1 Initial distribution and transformation

Here, the main focus lies in the long-term propagation of fragmentation clouds. To speed up the integration of the trajectories, the particles are propagated in averaged orbital elements. The initial condition is required to be given in the same element set. Often, however, the initial distribution is not given in the appropriate phase space. The initial fragment distribution given by the NASA standard breakup model (SBM) [2] is defined as a function of the magnitude of the velocity, Δv, relative to the parent orbit, and fragment characteristics, $\boldsymbol{b}^T = (L\ A/m)$, with the characteristic length, L, and the area-to-mass ratio, A/m. The distribution, $n_{\boldsymbol{v},\boldsymbol{b}}$, that can be obtained from the NASA SBM is defined in the velocity, $\boldsymbol{v} \in \mathbb{R}^3$, and $\boldsymbol{b}$, and can be extended to include uncertainties also in radial direction, $\boldsymbol{r}$ [14].

For propagation, the distribution is transformed into Keplerian elements, $\boldsymbol{\alpha}^T = (a\ e\ i\ \Omega\ \omega\ f)$, with the semi-major axis, a, the eccentricity, e, the inclination, i, the right ascension of ascending node, Ω, the argument of perigee, ω, and the true anomaly, f. This can be achieved by application of the method of change of variables [15, 14] as

$$n_{\boldsymbol{\alpha},\boldsymbol{b}} = \frac{n_{\boldsymbol{s},\boldsymbol{b}}(\boldsymbol{\varphi}^{-1}(\boldsymbol{\alpha}))}{|\det \boldsymbol{J}|} \qquad J_{ij} = \frac{\partial \varphi_i}{\partial s_j} \tag{1}$$

with the point transformation, $\boldsymbol{\varphi}$, from $\boldsymbol{\alpha}$ to $\boldsymbol{s}^T = (\boldsymbol{r}^T\ \boldsymbol{v}^T)$ and the Jacobian of the transformation, $\boldsymbol{J}$. For semi-analytical propagation, randomization of f is assumed.

2.2 Continuum propagation

The distribution is modeled as a continuum. To predict its evolution, the general continuity equation is applied [5]

$$\frac{\partial n_{\boldsymbol{x}}}{\partial t} + \nabla \cdot (n_{\boldsymbol{x}} \boldsymbol{F}) = 0 \qquad \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{F} \tag{2}$$

with the density, $n_{\boldsymbol{x}}$, given in the phase space, $\boldsymbol{x}^T = (\boldsymbol{\alpha}^T\ \boldsymbol{b}^T) \in \mathbb{R}^D$ in D dimensions, the independent variable time, t, the dynamics, $\boldsymbol{F}$, and ignoring sources and sink terms. This partial differential equation can be converted into an ordinary differential equation using the method of characteristics [5], giving an additional equation for the density evolution

$$\frac{\mathrm{d}n_{\boldsymbol{x}}}{\mathrm{d}t} = -n_{\boldsymbol{x}} \operatorname{tr}\left(\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}\right) \tag{3}$$

with the trace, tr. The characteristics of the system follow the trajectories. They are propagated using the *PlanODyn* suite [16] as it provides the Jacobian of the averaged dynamics with respect to the mean elements for all the major perturbations. The effect of drag is modeled according to [17]. The fragment characteristics, $\boldsymbol{b}$, are assumed to be constant.

The drawback of the method of the characteristics is that the solution to the equation is known only along the characteristics. To obtain the solution over the full domain, interpolation between the characteristics is required (see next Subsection). The initial characteristics are selected through the Metropolis-Hastings algorithm [18], given the initial NASA SBM distribution.

2.3 Surrogate modeling

The characteristics are propagated until the time or times of interest for evaluation of the number of impacts on active missions. At each of these snapshots, a Gaussian Mixture Model (GMM) [19] is fitted to the characteristics in order to

obtain a fast to evaluate surrogate model. Please see Ref. [9] for an in depth discussion of the fitting procedure, which is based on the minimization of a cost function.

If the underlying distribution exhibits strongly non-linear behavior, fitting a GMM might not be computationally practical as a large number of kernels is required. Furthermore, the breakup distribution can exhibit boundedness (e.g. eccentricity is bounded to be positive). The problem is tackled by introducing one or multiple reference hyperplanes, defined through samples that are propagated using the same force model as the characteristics. A continuous definition of the hyperplane is found through interpolation of the reference samples. The cloud characteristics are then fitted in the Δ-space, e.g. in the $\Delta e = e - e_r$, where e_r is the reference hyperplane. As the reference samples are subject to the same non-linear deformations as the cloud characteristics, the resulting Δ-space shows less non-linear behavior and is thus easier to fit using a small number of kernels. Sec. 5 explains the concept using examples.

2.4 Current limitations

The method was successfully tested on a large number of orbital configurations, in varying dimensions subject to atmospheric drag and the oblateness of Earth (J_2), albeit neglecting the J_2-drag coupling.

At this stage, the method is not suitably able to accommodate forces that lead to resonances on a small subset of the feature space. Perturbations from third bodies or solar radiation pressure potentially induce resonances. Such resonances tend to separate, or branch out, parts of the distribution from the main bulk of the characteristics. Additionally, it renders the reference hypersurfaces to be a function of all the elements rather than just a few decoupled ones. This introduces inaccuracies in the definition of the surfaces and hence leads to faulty Δ-space distributions.

Currently, the possibility of performing domain splitting [20] for the cases where the distribution branches out and prohibits convergence of the fitting is investigated. If the branch can be isolated and separated such that both the branch and the remaining samples can be well represented in their respective subdomains, the GMMs could be fitted separately.

3 IMPACT ESTIMATION

For mid- to long-term propagation, assumptions can be made on the distribution of LEO fragmentations after a certain amount of time [21]. Shortly after the fragmentation, given even small differences in the orbit energy and area-to-mass ratio, the fragments will randomize over the mean anomaly, M, and form a torus. After a few years and due to the oblateness of the Earth, the fragments will further distribute into a band, i.e. randomize in both Ω and ω and limited in latitude by the parent inclination, i_0, which is assumed to be constant.

These assumptions can be used if the distribution in the full Keplerian element set, $n_{\boldsymbol{\alpha}}$, is to be recovered from a simpler definition in less elements. Note that in this section the dependence on the fragment characteristics, $\boldsymbol{b}$, is dropped for better readability. It can either be integrated out, or carried through along the derivations to get an impact rate dependent on the fragment characteristic.

If the distribution is given as a function of the spatial variables a and e only, the dependence on i can be recovered by assuming a fixed $i = i_0$ for all fragments, i.e.

$$n_{a,e,i} = \delta(i - i_0) n_{a,e} \tag{4}$$

where δ is the Dirac delta function, defined as

$$\delta(x) = \begin{cases} 1, & \text{if } x = 0 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

The dependence on the angles, Ω, ω, and M, is introduced by assuming uniform distribution over $[0, 2\pi)$

$$n_{a,e,i,\Omega,\omega,M} = \frac{1}{(2\pi)^3} n_{a,e,i} \tag{6}$$

Using the following transformations between M, eccentric anomaly, E, and f [22]

$$M = E - e \sin E \qquad \cos E = \frac{e + \cos f}{1 + e \cos f} \qquad \sin E = \frac{\sqrt{1 - e^2} \sin f}{1 + e \cos f} \tag{7}$$

the derivative of f with respect to M is found as

$$\frac{\mathrm{d}f}{\mathrm{d}M} = \frac{(1+e\cos f)^2}{(1-e^2)^{3/2}} \tag{8}$$

Thus, the distribution in the full Keplerian elements set, $\boldsymbol{\alpha}$, using the method of change of variables as

$$n_{\boldsymbol{\alpha}} = \left|\frac{\mathrm{d}f}{\mathrm{d}M}\right|^{-1} n_{a,e,i,\Omega,\omega,M} = \frac{(1-e^2)^{3/2}}{(1+e\cos f)^2} n_{a,e,i,\Omega,\omega,M} \qquad \forall f \in [0, 2\pi) \tag{9}$$

To estimate the number of impacts, the distribution of fragments, $n_{\boldsymbol{s}}$, in Cartesian coordinates, $\boldsymbol{s}$, is required. This can be obtained by rearranging Eq. 1 as

$$n_{\boldsymbol{s}} = |\det \boldsymbol{J}|\, n_{\boldsymbol{\alpha}} \tag{10}$$

The fragment density, $n_{\boldsymbol{s}} = n_{\boldsymbol{s}}(\boldsymbol{r}, \boldsymbol{v})$, is a function of both the radius, $\boldsymbol{r}$, and velocity $\boldsymbol{v}$. Integration over $\boldsymbol{v}$ yields the spatial density, $n_{\boldsymbol{r}}$, however, the directional information is important for the estimation of the impact rate. To calculate the rate of impact, $\dot{n}_i(\boldsymbol{r}, \boldsymbol{v}^*)$, at any given point in space and target velocity, $\boldsymbol{v}^*$, integration of the product of the cross-sectional area, A, $n_{\boldsymbol{s}}$ and the magnitude of the relative velocity, $||\Delta\boldsymbol{v}^*|| = ||\boldsymbol{v} - \boldsymbol{v}^*||$, over all the fragment velocities is required

$$\dot{n}_i(\boldsymbol{r}, \boldsymbol{v}^*) = \int\int\int A\left(\frac{\Delta\boldsymbol{v}^*}{||\Delta\boldsymbol{v}^*||}\right) n_{\boldsymbol{s}}\left(\boldsymbol{r}, \boldsymbol{v}\right) ||\Delta\boldsymbol{v}^*||\, \mathrm{d}\boldsymbol{v} \tag{11}$$

where it is assumed that $n_{\boldsymbol{s}}$ is constant over A and that the area of the chaser is negligible compared to the target. If A is independent of the direction of impact, i.e. if the target is a sphere, it can be taken out of the integration. If the inclination of the distribution is assumed constant, the volume integral reduces to two area integrals. Finally, the number of impacts can be calculated by integrating over time, t, as

$$n_i(t) = \int_{t_0}^{t} \dot{n}_i\left(\boldsymbol{r}(t), \boldsymbol{v}^*(t)\right)\, \mathrm{d}t \tag{12}$$

4 TOOL STRUCTURE

The Starling suite consists of several independent routines, which can be found in Fig. 1. The majority of the routines, shown in boxes with blue shades, are implemented in the PYTHON language. Input files, shown in boxes with transparent background, are formatted as JSON or comma separated value files. The role of the interpreter that is required throughout all the subroutines is briefly introduced in Sec. 4.1. The subroutines and their respective inputs are discussed in Sec. 4.2.

4.1 Interpreter

One of the main objectives of Starling is to seamlessly increase/decrease the dimensionality of the problem and switch between different feature spaces. The interpreter takes the role of understanding and transforming the characteristics using symbolic mathematics and the SYMPY package [23]. It is itself defined through the interpreter settings, a JSON file containing

`"variables"`: a dictionary of all the variables present, with the following items
- `"value"`: symbolic representation of the feature as a function of the states
- `"default"`: default value in case it is not part of the feature space but only used during integration
- `"variable"/"unit"`: required for labeling of the figure axes

`"data"`: list of strings describing the full element set, i.e. mix of values and default values required for integration
`"states"`: list of strings describing subset of the data required for the feature space
`"weight"`: string describing the density around the state
`"features"`: list of strings describing features to be used for model fitting, defined through symbolic representation as a function of the states

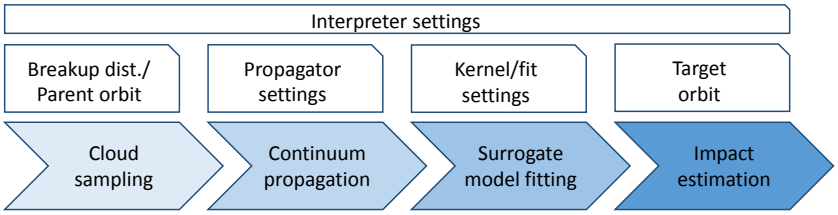


Fig. 1: Schema of the Starling suite. Routines are depicted in blue shades, while input files are shown with no filling.

```
{ "data": ["sma", "ecc", "inc", "raan",
           "aop", "mano", "amr", "dens"],
  "states": ["sma", "ecc"],
  "weight": "dens",
  "features": ["rper", "rapo"],
  "variables": {
    "sma": {
      "value": "(rper+rapo)/2",
      "variable": "$a$",
      "unit": "m" },
    "ecc": {
      "value": "(rapo-rper)/(rper+rapo)",
      "variable": "$e$",
      "unit": "$-$" },
    "inc": {
      "default": 1.5708,
      "variable": "$i$",
      "unit": "rad" },
    "raan": { ... },
    "aop": { ... },
    "mano": { ... },
    "amr": { ... },
    "dens": { ... },
    "rper": {
      "value": "sma*(1-ecc)",
      "variable": "$r_p$",
      "unit": "m" },
    "rapo": {
      "value": "sma*(1+ecc)",
      "variable": "$r_a$",
      "unit": "m" }
  }
}
```

Fig. 2: Code snippet of interpreter settings for a 2-dimensional features space in perigee and apogee radius.

The snapshot files, stored in `.csv` format, contain only the required state and weight columns. Upon reading up the snapshot file, the interpreter transforms the states and their respective weights into the feature space according to the mathematical description in `"value"`. This gives the user the opportunity to effortlessly explore a large variety of feature spaces. Changing the dimensionality of the problem is done simply by extending the states and features lists. However, if the dimensionality is changed, the cloud needs to be resampled to obtain the appropriate densities. The data list is used whenever the snapshot is to be integrated. The interpreter extracts the states and weight and adds the `"default"` values required for the integration. After propagation, it truncates the data again down to states and weight only. Finally, the interpreter also provides the labels to be used when the features are plotted.

A code snippet for a 2-dimensional case can be found in Fig. 2. It is the description of a perigee/apogee radius feature space.

4.2 Routines

The subroutines can be broadly split into cloud sampling, continuum propagation, surrogate model fitting and impact estimation. The first three subroutines are completely ignorant of orbital dynamics and can be used for propagation of any distribution in any dynamics. The last subroutine is treated as a plug-in, as it serves a clear use case, but using similar input files as the other routines.

The cloud is sampled using an implementation of the Metropolis-Hastings algorithm [18] given any initial distribution and parent orbit. The output file is a `.csv` file containing the states and density defined through the interpreter. In case reference hyperplanes are required, they can be obtained by mapping the cloud samples on the desired hyperplane. As hyperplanes are defined through samples, they have the same format as the cloud samples with densities set to zero.

The continuum propagation routine in Starling is merely providing interfacing functionality to external propagators. The characteristics and hypersurface samples to be propagated are read by the interpreter, transformed into extended states, and sent to the integrator together with the propagator settings. Parallelization is tasked to the integration tool. The returned propagated states are broken up into snapshots, i.e. states at fixed epochs, and saved in the same format as the initial characteristics file.

Given the characteristics and reference hypersurfaces at each snapshot, together with the fit settings, the GMM is fitted.

Different snapshots can be fitted in parallel. The GMM is stored in a JSON file, including all the relevant information to retrieve the full distribution. Additionally, a fit result file is stored containing information about the quality of the fit.

Lastly, the impact estimation tool, a separate plug-in to Starling, requires the GMM file, reference hypersurface samples and a definition of the target orbit to calculate the number of impacts.

5 APPLICATION

Fragment clouds from two different orbital regions are presented. First, the initial fragment cloud of an explosion from a circular parent orbit is presented in 2 dimensions, in a feature space derived from a and e. The impact rate is estimated at different times, assuming a fixed parent inclination and randomization in all angles. This example strives to strengthen the understanding of the usage of the reference hypersurface and shows the estimation of the impact rate. Second, the evolution of a fragment cloud of an explosion from a highly eccentric parent orbit is presented in 5 dimensions, in a feature space derived from a, e, Ω, ω, and A/m. This example shows the potential of the continuum approach in terms of computational efficiency in higher dimensions.

Both examples are modeled according to the NASA SBM [2] as a rocket body explosion considering characteristic lengths between 1 mm and 1 m, resulting in $N_0 = 3.8 \times 10^5$ fragments.

5.1 Cloud evolution in 2 dimensions and impact rate estimation

The distribution is sampled from a circular parent orbit defined by $a_0 = 7200$ km and $e_0 = 0$. Hence, the radius of fragmentation is, $r_0 = a_0$. As can be observed in Fig. 3a showing the characteristics, the distribution is bounded in perigee radius, $r_p = a(1 - e)$, and apogee radius, $r_a = a(1 + e)$, respectively. The boundary is defined by the initial radius of fragmentation, i.e. no resulting fragment can have a perigee above – or an apogee below – the fragmentation radius. The hypersurface

$$r_{a,r} = r_{a,r}\left(\xi_p = \log_{10}(r_0 - r_p)\right) \tag{13}$$

is used to normalize the target space and remove the boundary. The reference samples are obtained by mapping the cloud samples on $r_p = r_0$ and $r_a = r_0$, respectively (see Fig. 3b). The target space is defined by

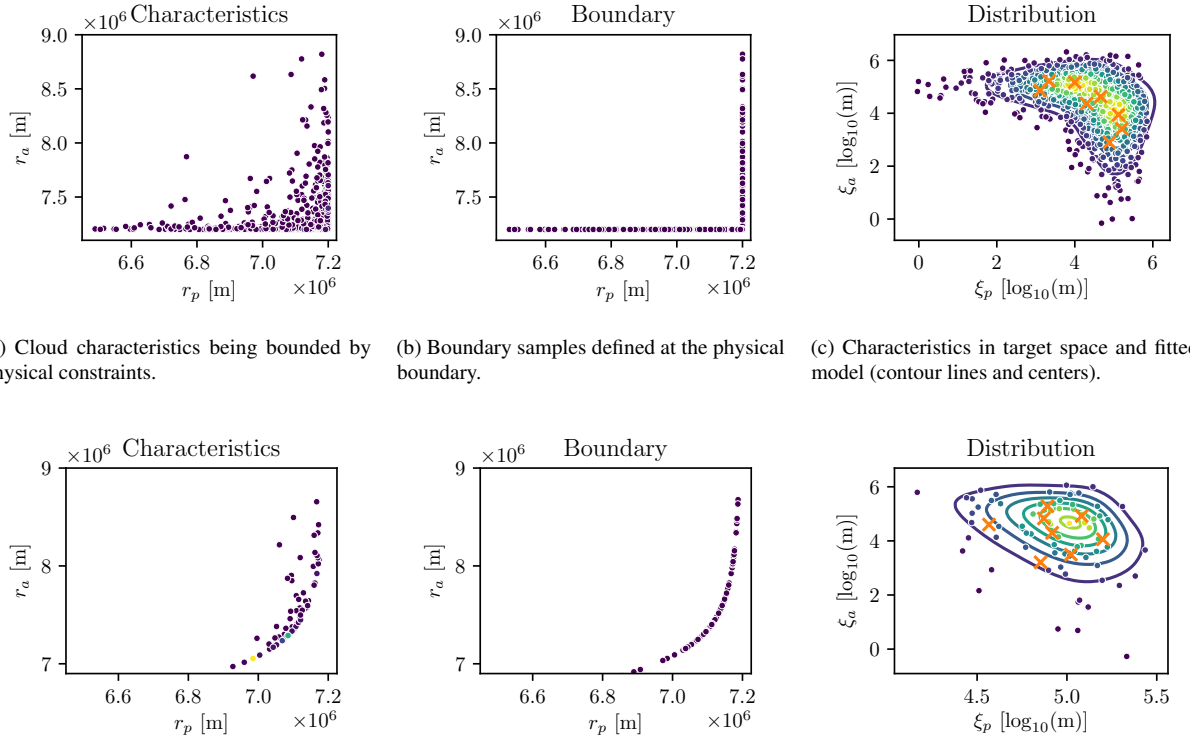
$$(\xi_p, \xi_a = \log_{10}(r_{a,r} - r_a)) \tag{14}$$

Fig. 3c shows the initial distribution in the target space. Fitting can easily be performed in a few seconds using a low number of kernels. The bottom row of Fig. 3 shows the same series of transformations after propagation of the cloud and reference samples for $t = 10$ years assuming a fixed $A/m = 1$ m^2/s and drag forces. Despite the low number of cloud samples remaining ($N = 73$), the fit still performs well. To obtain a density profile with similar accuracy across a 100×100 grid from samples only would require $N = 10^5$ samples to be present at each snapshot [14].

If it is assumed that the continuum inclination is fixed at the parent inclination, $i_0 = 45$ deg, and randomized over all the other Keplerian angles, the impact rate on arbitrary target orbits can be estimated. The impact rate as a function of the true anomaly, f^*, can be found in Fig. 4 for a fixed target orbit at $a^* = 7500$ km, $e^* = 0.1$ and $i^* = 30$ deg, modeling the target as a sphere with a cross sectional area, $A = 1$ m^2. The impact rates right after the fragmentation at $t = 0$ years, after $t = 5$ years and after $t = 10$ years are shown, respectively. Initially, the impact rate is highest when the target object passes through the fragmentation radius, i.e. at $f^* = 72/288$ deg. As the fragment cloud starts decaying, the altitude with peak impact rate is lowered too and reaches 7135 km after 5 years where it remains for the next 5 years. The average impact rate over a full orbit, projected onto a year, is 4.4×10^{-5}/year, 1.1×10^{-5}/year and 3.2×10^{-6}/year after $t = 0, 5, 10$ years.

5.2 Cloud evolution in 5 dimensions

The distribution is sampled for a GTO with $a_0 = 22800$ km, $e_0 = 0.7$, $i_0 = 18$ deg, and $\Omega_0/\omega_0/f_0 = 180/20/45$ deg, respectively, to obtain 2000 characteristics. To avoid collapse of dimensionality of the distribution (by going from 3 Δv components to 4 Keplerian elements), an additional normally distributed uncertainty with zero mean and 5 km standard deviation is introduced in the orbital plane in the direction perpendicular to the velocity vector (see [14] for more information). The cloud is subjected to atmospheric drag and a non-spherical Earth defined by J_2, and is fitted in a



(a) Cloud characteristics being bounded by physical constraints.

(b) Boundary samples defined at the physical boundary.

(c) Characteristics in target space and fitted model (contour lines and centers).

(d) Cloud characteristics still showing boundedness.

(e) Propagated boundary samples.

(f) Characteristics in target space and fitted model (contour lines and centers).

Fig. 3: Representation of characteristics (left column), boundary samples (middle) and fitted distributions (right) for the 2-dimensional example right after fragmentation (top row) and after 10 years of propagation (bottom).

feature space derived from a, e, Ω, ω and A/m. The effects of drag and J_2 on the orbit are modeled to be decoupled. Two hypersurfaces are defined from two sets of reference samples in order to normalize the space in 3 features

$$e_r = e_r(r_p, \chi = \log_{10}(A/m)) \qquad \Omega_r = \Omega_r(r_p, e, \chi) \qquad \omega_r = \omega_r(r_p, e, \chi) \tag{15}$$

The initial samples for the first hypersurface for the derivation of e_r are mapped from the cloud samples to have a fixed $e = e_0$. The initial samples for the second hypersurface for the derivation of Ω_r and ω_r are mapped from the cloud samples with a fixed $\Omega = \Omega_0$ and $\omega = \omega_0$. Hence, both hypersurfaces cut through the sample cloud. Cubic spline interpolation and radial basis function interpolation [24] is used for the 2- and 3-dimensional hypersurfaces, respectively. The target space is

$$(\zeta_p = r_p - (R + h_0), \Delta e = e - e_r, \Delta\Omega = \Omega - \Omega_r, \Delta\omega = \omega - \Delta\omega, \chi) \tag{16}$$

with the Earth radius, $R = 6378$ km, and the final considered altitude, $h_0 = 100$ km. The reason why the logarithm of r_p is taken is to avoid fraying of the samples towards low altitudes.

Fitting a GMM with 20 kernels to a 5-dimensional snapshot – i.e. requiring 420 parameters – takes about 60-120 seconds and can be performed successfully up until $t = 500$ years, after which only 337 of the initially 2000 cloud characteristics still remain. An example fit after 100 years of propagation, and 1153 remaining characteristics, is shown in 5a. As the secondary hypersurface follows the cloud samples (trailing or leading them only in Ω and ω), the resulting fit needs to be compared to a second set of verification cloud samples. Fig. 5b shows the verification samples, their respective densities and the estimated density from the surrogate model. It becomes evident that the Δ-space does not accurately represent the verification samples, because the secondary hypersurface does not interpolate accurately enough across the full domain. Some of the samples are estimated to be far away from the

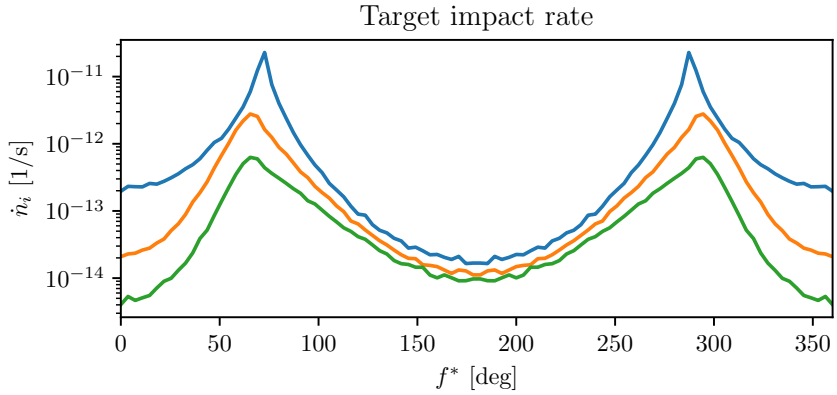


Fig. 4: Impact rate on constant target orbit as a function of the target orbit true anomaly, f^*, right after the fragmentation (top), after 5 years and after 10 years (bottom).

hypersurface (> 1000 deg), even though they ought to remain at a constant distance given by the initial distribution. However, the fit still allows to estimate the extend of randomization of the distribution in Ω and ω as a function of the other features. If, by varying the features within measurement uncertainties, the Ω or ω is displaced by more than 360 deg, it can safely be assume randomized in that angle, and the problem can be studied in a lower dimension. Also, the cloud distribution can still be studied in terms of impact rate, keeping in mind that it will be shifted in some parts in the angles. If impact rates are calculated over a long time span for many objects, these errors in angles are expected to average out.

It was shown in a previous work that even $N = 10^{10}$ samples do not suffice to derive an accurate density profile of a GTO distribution in 5 dimensions [14]. Propagation of such a large number of samples, using the same setup as the one used for the density propagation, would require 4.5 years. Instead, the 2000 cloud characteristics and the 4000 hypersurface samples, sufficient to derive a surrogate model over the full domain, are propagated within less than 2.5 minutes.

6 CONCLUSION

The Starling suite was introduced for the probabilistic propagation of densities modeled as continua. It was shown that using the continuity equation, in tandem with semi-analytical propagation along the characteristics, and fitting of a surrogate model, it is possible to obtain a model of an evolving orbital fragment cloud. Examples in 2 and 5 dimensions show the potential for fast mid- to long-term propagations of fragment clouds, considering perturbations by drag and J_2, requiring only a small number of propagated samples. The fragment densities can subsequently be used to estimate the impact rate on any target orbit from which the collision risk can easily be derived. Starling was designed to be ignorant towards the underlying dynamics and can thus be used for integration of any particle density or probability density function subject to any dynamics.

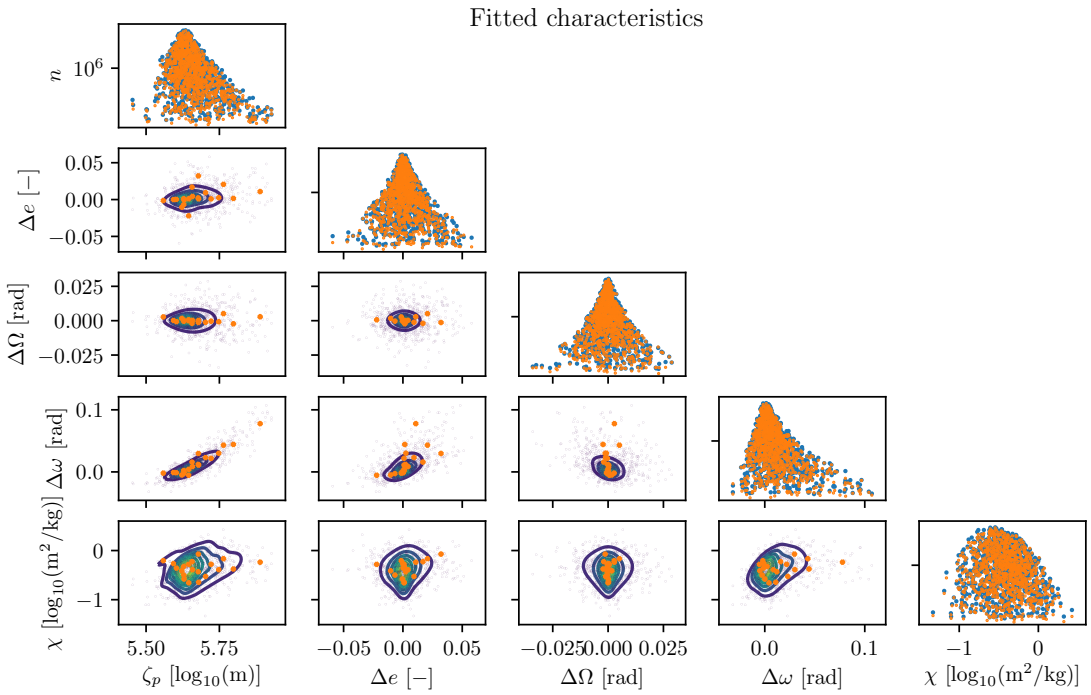
7 ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme as part of project COMPASS (Grant agreement No 679086). The datasets generated for this study can be found at the link http://www.compass.polimi.it/publications.

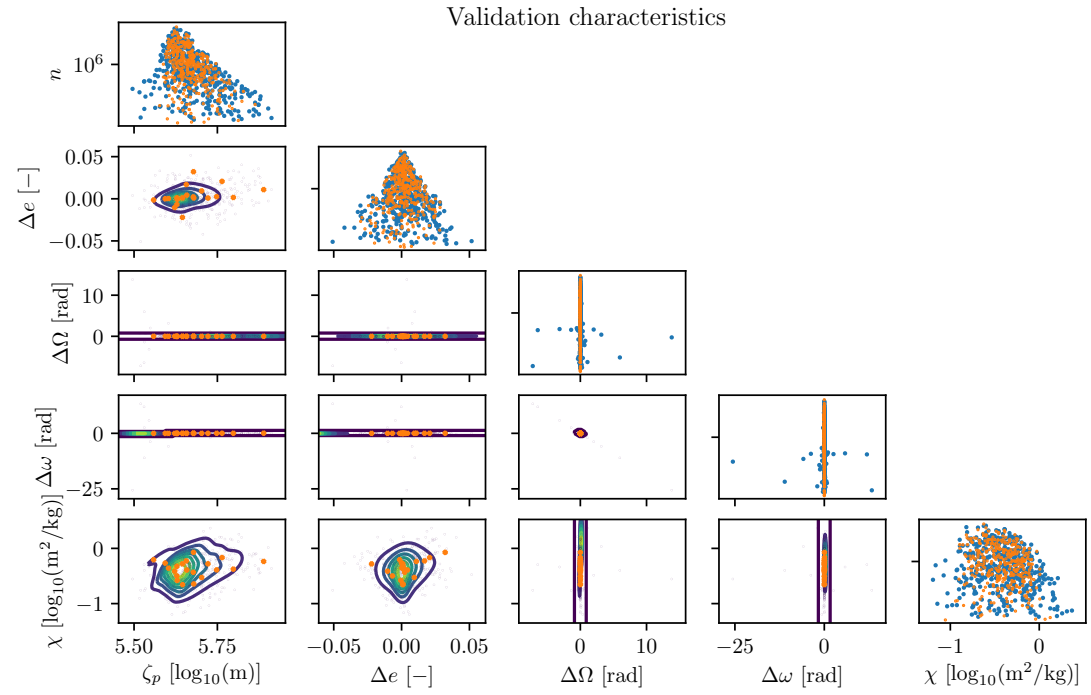
8 REFERENCES

[1] ESA Space Debris Office. ESA's annual space environment report. Technical Report GEN-DB-LOG-00271-OPS-SD, 2019.

[2] N. L. Johnson, P. H. Krisko, J.-C. Liou, and P. D. Anz-Meador. NASA's new breakup model of evolve 4.0. *Advances in Space Research*, 28(9):1377–1384, 2001.



(a) Characteristics upon which the surrogate model was fitted on.



(b) Validation characteristics to check quality of the fit. The hypersurface in Ω and ω struggles to give an accurate measure for some instances.

Fig. 5: 5-dimensional distribution after $t = 100$ years. Diagonal plots: characteristics (blue dots) and fitted distribution (orange dots) showing density in the y-scale. Off-diagonal plots: characteristics (dots), fitted distribution (contours) and kernel centers (orange crosses).

[3] J.-C. Liou, D. T. Hall, P. H. Krisko, and Opiela J. N. LEGEND - a three-dimensional LEO-to-GEO debris evolutionary model. *Advances in Space Research*, (34):981–986, 2004.

[4] W. B. Heard. Dispersion of ensembles of non-interacting particles. *Astrophysics and Space Science*, 43:63–82, 1976.

[5] C. R. McInnes. An analytical model for the catastrophic production of orbital debris. *ESA Journal*, 17:293–305, 1991.

[6] F. Letizia, C. Colombo, and H. G. Lewis. Multidimensional extension of the continuity equation method for debris clouds evolution. *Advances in Space Research*, 57:1624–1640, 2016.

[7] N. N. Gor'kavyi, L. M. Ozernoy, J. C. Mather, and T. Taidakova. Quasi-stationary states of dust flows under poynting-robertson drag: new analytical and numerical solutions. *The Astrophysical Journal*, 488:268–276, 1997.

[8] A. Wittig, C. Colombo, and R Armellin. Long-term density evolution through semi-analyical and differential algebra techniques. *Celestial Mechanics and Dynamical Astronomy*, 128(4):435–452, 2017.

[9] S. Frey, C. Colombo, and S. Lemmens. Interpolation and integration of phase space density for estimation of fragmentation cloud distribution. In *Proceedings of the 29th AAS/AIAA Space Flight Mechanics Meeting*, 2019.

[10] D. L. Talent. Analytic model for orbital debris environmental management. *Journal of spacecraft and rockets*, 29(4):508–513, 1992.

[11] A. I. Nazarenko. The development of the statistical theory of a satellite ensemble motion and its application for space debris modelling. In *Proceedings of the 2nd European Conference on Space Debris*, 1997.

[12] A. Rossi, L. Anselmo, A. Cordelli, P. Farinella, and C. Pardini. Modelling the evolution of the space debris population. *Planetary and Space Science*, 46(11/12):1583–1596, 1998.

[13] F. Letizia. Extension of the density approach for debris cloud propagation. *Journal of Guidance, Navigation, and Control*, 41(12):2650–2656, 2018.

[14] S. Frey and C. Colombo. Transformation of satellite breakup distribution from cartesian coordinates to orbital elements, submitted to the *Journal of Guidance, Control, and Dynamics* in November, 2019.

[15] T. T. Soong. *Fundamentals of probability and statistics for engineers*. John Wiley & Sons, Ltd., 2004.

[16] C. Colombo. Planetary orbital dynamics (PlanODyn) suite for long term propagation in perturbed environment. In *Proceedings of the 6th International Conference on Astrodynamics Tools and Techniques (ICATT)*, 2016.

[17] S. Frey, C. Colombo, and S. Lemmens. Extension of the King-Hele orbit contraction method for accurate, semi-analytical propagation of non-circular orbits. *Advances in Space Research*, 64:1–17, 2019.

[18] S. Chib and E. Greenberg. Understanding the metropolis-hastings algorithm. *The American Statistician*, 49(4):327–335, 1995.

[19] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[20] A. Wittig, P. Di Lizia, R. Armellin, et al. Propagation of large uncertainty sets in orbital dynamics by automatic domain splitting. *Celestial Mechanics and Dynamical Astronomy*, 122(3):239–261, 2015.

[21] D. McKnight and G. Lorenzen. Collision matrix for low earth orbit satellites. *Journal of Spacecraft and Rockets*, 26(2):90–94, 1989.

[22] D. A. Vallado. *Fundamentals of Astrodynamics and Applications*. Hawthorne, CA: Microcosm Press, 4th edition, 2013.

[23] A. Meurer, C. P. Smith, M. Paprocki, et al. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3, 2017.

[24] E. Jones, T. Oliphant, and P. Peterson. SciPy: Open source scientific tools for Python, http://www.scipy.org/, 2001.