

MULTILAYER CLUSTERED SAMPLING TECHNIQUE (MLCS) FOR NEAR-EARTH ASTEROID IMPACT HAZARD ASSESSMENT

Javier Roa* and Davide Farnocchia[†]

Because of planetary encounters, the motion of near-Earth asteroids is chaotic and small differences in the initial conditions tend to diverge exponentially. Linear approximations for propagating orbital uncertainties can lead to inaccurate estimates of the probability of an Earth collision. We present a novel fully nonlinear strategy for estimating the probability of an asteroid impact using sequential Monte Carlo layers. The method first explores a low-resolution layer to locate potentially relevant regions. Then, we conduct localized searches on deeper layers with higher resolution. The method retains the accuracy of brute-force Monte Carlo sampling while reducing the computational cost by only sampling relevant regions.

INTRODUCTION

The most robust method for estimating the probability of an asteroid impacting the Earth is Monte Carlo (MC) sampling.¹ Random samples are drawn from a given probability distribution of initial conditions and propagated forward in time while recording the number of impacts. The method owes its accuracy and robustness to making no simplifying assumptions on how the orbital uncertainties are mapped into the future. However, MC is a very computationally expensive technique because it requires propagating a large number of samples, typically of the order of the inverse of the target probability resolution. As a result, MC sampling can become unpractical when the probability is small (below one per several thousands).² For asteroid impact hazard assessment, we generally estimate probabilities of about 10^{-6} and even lower, and that is why MC is not widely used in automatic impact monitoring systems.

Current asteroid impact monitoring systems like Sentry³ at the Jet Propulsion Laboratory (JPL) and NeoDyS⁴ at the University of Pisa implement a semi-linear method for propagating orbital uncertainty to estimate the impact probability. The mapping strategy is based on the concept of the *Line of Variations*^{5,6} (LOV), which takes advantage of the way uncertainties tend to stretch along a one-dimensional manifold along the orbit. For example, this behavior can be caused by the uncertainty in semimajor axis that causes a runoff in longitude that increases with time. Instead of sampling the initial conditions from the entire 6D space, the LOV method samples initial conditions along a one-dimensional subspace. The resulting samples are propagated with the fully nonlinear model while the orthogonal directions can be explored using a linear approximation. Sampling along the LOV results in important computational savings compared to MC methods. However,

*Navigation Engineer, Solar System Dynamics Group. Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109-8099, USA. javier.roa@jpl.nasa.gov

[†]Navigation Engineer, Solar System Dynamics Group. Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109-8099, USA.

© 2019 California Institute of Technology. Government sponsorship acknowledged.

the presence of multiple encounters can significantly stretch the LOV and make its geometry complex, thus violating some of the underlying assumptions of the method. In addition, highly uncertain orbits determined from just a few observations might not exhibit any privileged direction when mapping the uncertainty to the target plane. For this reason, we aim at developing a method that captures the nonlinear nature of the problem, and that is robust and efficient enough to run autonomously in an automatic system.

From a more general perspective, estimating small probabilities is a common yet challenging problem in physics, engineering, economics, data mining, and many other fields. Given a random variable $\mathbf{x} \in \mathbb{R}^d$ with probability density function (pdf) $\varphi(\mathbf{x})$, the goal is to estimate the probability of a function $f(\mathbf{x})$ taking values below a certain threshold ε . The set of samples that lead to the rare event $f(\mathbf{x}) < \varepsilon$ is called the failure region,

$$\mathcal{F} = \{\mathbf{x} \in \mathbb{R}^d \mid f(\mathbf{x}) < \varepsilon\}. \quad (1)$$

Correctly describing and sampling \mathcal{F} can be challenging when the failure probability $P(\mathcal{F})$ is small.

In this context, importance sampling⁷ is an interesting technique for estimating probabilities more efficiently because it draws samples according to a distribution that increases the frequency at which important events are detected, thus reducing the number of function evaluations required. An alternative approach is the use of splitting methods.² These methods start by detecting a larger failure region defined by an increased threshold ε . Then, the region is reduced sequentially by lowering the value of the threshold until the actual failure region is detected. Subset simulation⁸ is a well-known example of a splitting method that uses Markov chains to re-sample the increasingly small failure regions. Recently, Losacco *et al.*⁹ (2018) compared the performance of subset simulation for estimating the impact probability of (99942) Apophis during its 2029 close approach against a new technique based on differential-algebra and importance sampling. Importance sampling and splitting methods often require human interaction to interpret the results and assess the quality of the solution, given the intrinsic assumptions and the stochastic nature of the processes. Although techniques like subset simulation capture the nonlinear behavior of the function, the inherent non-deterministic behavior of the intermediate sampling steps (often involving Markov chains) results in certain variability in the solution that complicates their implementation in automatic systems.

MC sampling estimates the impact probability $P(\mathcal{F})$ by mapping $f(\mathbf{x})$ for N samples $\mathbf{x}_1, \dots, \mathbf{x}_N$ and counting how many cases fall in the failure region, $N_{\mathcal{F}}$:

$$P(\mathcal{F}) = \frac{N_{\mathcal{F}}}{N}. \quad (2)$$

The uncertainty in the estimated probability P is given by

$$\sigma^2 = \frac{P(1-P)}{N}. \quad (3)$$

Equation (3) shows that estimating a probability $P = 10^{-6}$ with a 25% relative uncertainty requires $N = 16 \times 10^6$ function evaluations. Nonetheless, there are different ways in which MC sampling can be sped up. Multi-level MC methods¹⁰⁻¹² start from a simplified physical model and increase the accuracy sequentially, and estimate the probability by combining the estimates obtained on each accuracy level. Multi-level MC is faster than direct MC because the expensive, accurate model is only evaluated in the final level. The implementation of MC methods on GPUs has received significant attention in the last few years because MC sampling is conceptually well suited for implementation

in parallel. For orbital mechanics in particular, regularization techniques can potentially speed up the propagation of each sample compared to simple formulations in Cartesian coordinates.¹³ These methods involve either simplifying the dynamical model (introducing potential inaccuracies in sensitive problems) or re-writing production code, a task that is often not practical due to the associated costs of implementing, validating, and testing code with strict requirements.

The present paper introduces a new technique for estimating small probabilities in generic problems, called *multilayer clustered sampling* (MLCS), which we apply to estimating asteroid impact probabilities. We conceived MLCS to retain the robustness and accuracy of direct MC sampling for adequately modeling strongly nonlinear cases while reducing the number of function evaluations. The method is to be implemented in an automatic system, which means that even pathological cases such as orbits with deep planetary encounters must be supported with no human interaction. The main features of MLCS are:

- Transparency to the type of probability distribution that models the orbit uncertainty.
- Fully deterministic once the initial sampling of initial conditions is complete.
- As accurate as MC sampling (no simplifications of the physical model) with speedups of up to three orders of magnitude.
- The smaller the target probability, the greater the speedup compared to MC sampling.
- No need for proposal distributions or stochastic methods for re-sampling.
- Handles disjoint failure regions leading to impact.

INITIAL EXPLORATION

In the context of asteroid impact probability estimation, the random vector \mathbf{x} typically represents a set of orbital elements or Cartesian coordinates, $d_{ca} = f(T, \mathbf{x})$ provides the close-approach distance to Earth within a given time interval T , obtained by propagating the orbit using \mathbf{x} as the initial conditions, and \mathcal{F} is the failure region, i.e. the set of initial conditions that result in an impact with Earth. The failure region needs not be connected and might result in the union of several disjoint subsets

$$\mathcal{F} = \bigcup_i \mathcal{F}_i, \quad (4)$$

where each subset \mathcal{F}_i of initial conditions leading to impact is generally called a *Virtual Impactor*¹⁴ (VI).

The first step consists in locating the dates when close approaches occur, t . Each close approach will then be analyzed separately, looking for impacts in the time interval $T = [t - \Delta t, t + \Delta t]$. The width of the interval is inversely proportional to the velocity of the asteroid relative to Earth at close approach. Mathematically, for each close approach i the goal is to compute the probability

$$f(T_i, \mathbf{x}) < \varepsilon \equiv R_E, \quad (5)$$

where R_E is the equatorial radius of the Earth, 6378 km. In order to identify the relevant dates t_i , we draw a set of $N_1 = 10^4$ initial conditions from the distribution $\varphi(\mathbf{x})$ and propagate them forward in time while recording all close approaches within $d_{ca,lim} = 0.1$ au of the Earth. We assume a

Gaussian distribution in Cartesian space (or orbital element space for longer arcs) as obtained from the least-squares fit to the astrometry, although the method accepts any type of distribution. If we find at least one sample that makes a close approach at t , we flag that date for further exploration.

Assuming that the trail of samples mapped to the target plane is a line that goes through the center of the Earth, the maximum separation between samples on the target plane should be less than $0.2 \text{ au} = 2346D_E$ to detect at least one close approach (D_E is the diameter of the Earth). The integral probability between consecutive samples expressed in Earth diameters can be estimated like $1/N_1 = 10^{-4}$, resulting in the completeness level:⁶

$$P_{\text{cl}} = \frac{10^{-4}}{2346} \simeq 4 \times 10^{-8}. \quad (6)$$

In practice, the completeness level of the method provides the probability P_{cl} of missing a VI after running the initial exploration.

MULTILAYER CLUSTERED SAMPLING (MLCS)

MLCS is a new sampling technique based on the premise that MC sampling becomes inefficient because it propagates too many points far from the region of interest \mathcal{F} . Since the size of the failure region scales with the probability, the lower the probability the fewer points lie inside \mathcal{F} . Consequently, in the ideal case in which the location of \mathcal{F} is known a priori (or at least an approximation is available), estimating a small probability should require the evaluation of just the few samples that are inside or close to \mathcal{F} . In practice, the location of \mathcal{F} is not known a priori and that is the reason why many function evaluations are required. MLCS identifies relevant regions in the parameter space by clustering data into groups, explores each region with increased resolution, and repeats these steps sequentially until \mathcal{F} is found. The trade-off between how much each region is reduced and how much the resolution is increased controls the number of function evaluations.

Multilayer Decomposition for Achieving Variable Resolution

MLCS introduces a special sampling technique based on a series of discrete *layers*. The initial set of N_1 points generated during the exploration phase is the first layer, denoted \mathcal{S}_1 . Then, MLCS generates a second layer with $N_2 = \ell N_1$ points by sampling $N_2 - N_1$ additional points from $\varphi(\mathbf{x})$. The parameter $\ell \in \mathbb{N}$ is a constant factor. We typically choose $\ell = 2$ because low values of ℓ yield a conservative exploration of the uncertainty region. This step is repeated sequentially to generate up to n layers, each layer having $N_{i+1} = \ell N_i$ points. As a result, the i -th layer is a subset of the $(i+1)$ -th layer, i.e. $\mathcal{S}_1 \subset \mathcal{S}_2 \subset \dots \subset \mathcal{S}_n$. The number of samples in the final layer, N_n , should be large enough to capture the minimum expected target probability, P_{min} . N_n can be estimated as $N_n \geq \beta/P_{\text{min}}$ in terms of a parameter β that is chosen depending on the requested level of accuracy. This parameter relates to P_{min} and the required relative uncertainty $\tilde{\sigma} = \sigma/P_{\text{min}}$ as

$$\beta = \frac{1 - P_{\text{min}}}{\tilde{\sigma}^2} \quad (7)$$

In practice, the required number of layers is determined from the number of points in the initial layer, N_1 , the estimate of the minimum target probability, P_{min} , and the value of ℓ :

$$n = \left\lceil \log_{\ell} \left(\frac{\beta}{N_1 P_{\text{min}}} \right) \right\rceil + 1. \quad (8)$$

Here, $\lceil \cdot \rceil$ is the ceiling operator. For example, if the system is required to capture probabilities as low as $P_{\min} = 10^{-7}$ with $\tilde{\sigma} = 0.5$ and $\ell = 2$, a total of 15 layers is required. Each layer has

$$N_i = \ell^{i-1} N_1 \quad (9)$$

points. For MC sampling, targeting P_{\min} requires evaluating all N_n points on the deepest layer whereas the initial exploration of MLCS evaluates only the N_1 points on the first layer. Typically $n > 10$, which means that $N_1 \ll N_n$ and the time required to complete the initial exploration of MLCS is negligible compared to MC sampling. Taking the case $n = 15$ and $\ell = 2$ as an example results in $N_1/N_n = 2^{-14} = 6 \times 10^{-5}$.

Selecting a Refined Subset of Points

Given a set of points \mathcal{C} , MLCS ranks all points in \mathcal{C} according to a certain cost function $J(\mathbf{x})$. Initially, \mathcal{C} is the entire first layer, $\mathcal{C} \equiv \mathcal{S}_1$. In the steps that follow, \mathcal{C} refers to independent clusters of points. The ranking of points allows MLCS to detect the regions in \mathcal{C} that are worth exploring. The simplest strategy is to rank points according to the value of the function $f(\mathbf{x})$, making $J(\mathbf{x}) \equiv f(\mathbf{x})$. To improve the flexibility of the method, we suggest the extended cost function

$$J(\mathbf{x}) = w_1 f(\mathbf{x}) + w_2 \rho(\mathbf{x}) + w_3 r(\mathbf{x}), \quad (10)$$

which takes into account not only the value of the function $f(\mathbf{x})$ but also $\rho(\mathbf{x})$, a measure of the relative separation between neighboring points (inverse of the local density), and $r(\mathbf{x})$, the distance to the centroid (barycenter) of the current subset of points, modulated by the weights w_i . The inverse density $\rho(\mathbf{x})$ is defined as the mean distance to the closest k neighbors, viz

$$\rho(\mathbf{x}) = \text{mean}(\min_k \|\mathbf{x} - \mathbf{x}_j\|), \quad \text{with } \mathbf{x}, \mathbf{x}_j \in \mathcal{C} \text{ and } \mathbf{x}_j \neq \mathbf{x}. \quad (11)$$

In practice, we choose $k = 5$ based on several tests. The distance to the centroid of \mathcal{C} reads

$$r(\mathbf{x}) = \left\| \mathbf{x} - \frac{1}{N_{\mathcal{C}}} \sum_{j=1}^{N_{\mathcal{C}}} \mathbf{x}_j \right\|. \quad (12)$$

The set \mathcal{C} is refined by taking only the top p -percentile of points with lowest value of $J(\mathbf{x})$. We typically use $p = 0.3$. Taking a subset of points according to their associated cost is a common technique of splitting methods.^{2,8}

Clustering and Region Detection

The core of MLCS is its ability to detect relevant regions by clustering the points ranked based on Eq. (10). MLCS uses clusters to generate likely bounds of the different \mathcal{F}_i regions. Each cluster \mathcal{C}_i is reduced sequentially using the technique described in the previous section until it converges to the VI $\mathcal{F}_i \subset \mathcal{C}_i$, i.e. $\mathcal{C}_i \rightarrow \mathcal{F}_i$. A conservative clustering that comprises a large fraction of the state space is robust because important regions are hardly missed but can be inefficient because many points need to be evaluated. Conversely, an aggressive approach to clustering that reduces the search space rapidly results in fewer function evaluations but it can potentially miss relevant regions.

Clustering:

Many different clustering algorithms can be found in the machine-learning and data-mining literature.¹⁵ MLCS implements the single-link hierarchical clustering technique¹⁶ using the Euclidean distance between points i and j , $\|\mathbf{x}_i - \mathbf{x}_j\|$, as a measure of similarity. Figure 1 presents a schematic representation of single-link clustering of ten points resulting in two clusters. The user determines how many clusters will be obtained. Initially, each point belongs to its own cluster. Then, all the intercluster distances are computed and the closest pair is linked. The intercluster distances are computed again, using the distance between the two closest points as a measure of the distance between two clusters. The process is repeated until the desired number of clusters is obtained.

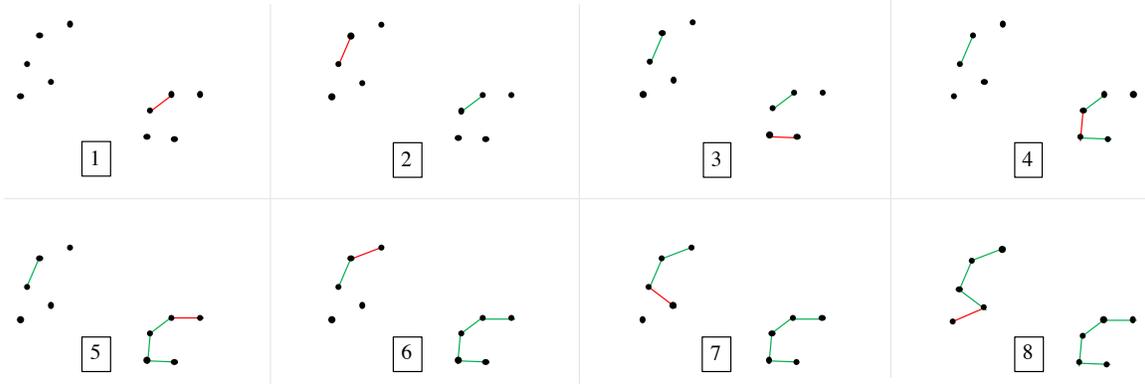


Figure 1: Example of single-link hierarchical clustering. The last link is plotted in red whereas previous links appear in green.

We selected the single link hierarchical clustering technique because of the following reasons:

1. The data to be clustered have already been filtered according to the value of $J(\mathbf{x})$. After the initial filtering and under reasonable assumptions of regularity and continuity of $f(\mathbf{x})$, points that are close lead to similar values of $J(\mathbf{x})$ and define a region of interest. For this reason, we choose to form clusters considering the minimum distance between points.
2. The failure region \mathcal{F} can have any arbitrary shape. In particular, the semimajor axis tends to dominate the dynamics and there is often a privileged direction in orbital element space, which originated the concept of the LOV.⁶ Single-link clustering tends to favor elongated clusters whereas complete-link clustering, for example, yields more spherical clusters.
3. The number of disjoint sets \mathcal{F}_i is unknown, so the number of clusters that should be explored is not known either. Hierarchical clustering allows the user to choose a sufficiently large number of clusters for the initial exploration and then MLCS applies a custom technique that we developed (called *envelope-based clustering*) to merge clusters when needed.

Envelope-based criterion for merging clusters:

MLCS clusters data to find likely bounds of the failure region, using cluster envelopes as described below. To improve the performance of the pure single-link clustering, MLCS implements an ad hoc strategy for merging clusters based on their envelope.

When clustering data, MLCS first generates a relatively large number of clusters, e.g. ten clusters. Then, MLCS checks if a cluster \mathcal{C}_i should be merged with any of the other clusters \mathcal{C}_j . For that, both

clusters are merged momentarily and the number of points in the resulting cluster, n_{i+j} , is compared with the number of points in both clusters, $n_i + n_j$. MLCS finds the cluster \mathcal{C}_j that yields the smallest difference $n_{i+j} - (n_i + n_j)$, and \mathcal{C}_i and \mathcal{C}_j are merged if

$$n_{i+j} < \frac{1+k}{k}(n_i + n_j). \quad (13)$$

We found that $k = 5$ works well in practice. Figure 2a shows an example of how MLCS checks if two clusters should be merged. The points circled in red correspond to the points in \mathcal{C}_{i+j} that are not in \mathcal{C}_i or \mathcal{C}_j , $\mathcal{C}_{i+j} \setminus (\mathcal{C}_i \cup \mathcal{C}_j)$. In this example, $n_i = 12$, $n_j = 5$, and $n_{i+j} = 19$, condition (13) holds, and the two clusters are merged. This criterion ensures that clusters are merged only if the extra cost in terms of additional function evaluations is small.

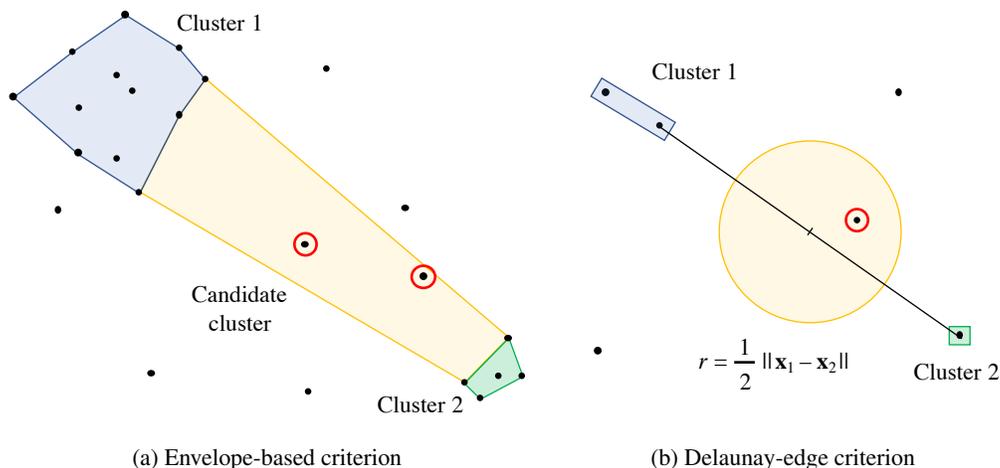


Figure 2: Graphical representation of the merging strategies

When merging two small clusters (using the empirical value $n_i + n_j < 4$), MLCS uses a different technique to evaluate the cost of merging them. The two closest points between each cluster, \mathbf{x}_1 and \mathbf{x}_2 , are identified and MLCS checks if they form a reduced Delaunay edge, which means that there are no points in the circle of radius $\|\mathbf{x}_1 - \mathbf{x}_2\|/2$ centered at $(\mathbf{x}_1 + \mathbf{x}_2)/2$. The example in Fig. 2b does not satisfy this condition so the two clusters are not merged.

Cluster envelope:

Once a cluster \mathcal{C} is detected, MLCS computes its *envelope*, \mathcal{E} . Three definitions of envelope are implemented depending on the dimensions of the system. For one-dimensional systems, the envelope is simply defined by the minimum and maximum values of the set of points \mathbf{x} that form the cluster:

$$\mathcal{E} = [\inf(\mathcal{C}), \sup(\mathcal{C})], \quad \text{for } \mathcal{C} \subset \mathbb{R}. \quad (14)$$

For two-dimensional systems, MLCS uses the α -shape algorithm¹⁷ to determine the non-convex boundary of the cluster. The α -radius r_α controls how tight the boundary is to the set of points in the cluster:

$$\mathcal{E} = \alpha\text{-shape}(\mathcal{C}, r_\alpha), \quad \text{for } \mathcal{C} \subset \mathbb{R}^2. \quad (15)$$

For $r_\alpha \rightarrow 0$, the envelope converges to the set of points itself (zero-area shape), $\mathcal{E} \rightarrow \mathcal{C}$, whereas $r_\alpha \rightarrow \infty$ yields the convex hull of \mathcal{C} . Typical algorithms like MATLAB's boundary function allow

the user to define a *shrink factor* instead of the α -radius. The shrink factor takes values from 0 to 1, with 0 corresponding to the convex hull and 1 corresponding to a zero-area shape. For MLCS we selected a shrink factor of 0.7 based on numerical tests.

The convexity of \mathcal{C} is a strong assumption, especially given the tendency of single-link clustering to provide elongated regions. Nevertheless, if the cluster can be assumed to be convex, MLCS can compute the convex hull instead by simply setting the shrink factor equal to zero. In higher dimensions, the envelope is computed as the convex hull of the cluster,

$$\mathcal{E} = \text{Conv}(\mathcal{C}), \quad \text{for } \mathcal{C} \subset \mathbb{R}^d \quad (d > 2). \quad (16)$$

Forcing the envelope to be convex can potentially lead to more function evaluations when the cluster is non-convex because it would capture points that do not belong to the cluster. However, the convexity assumption does not hinder the robustness of the method; no relevant regions are lost because of this assumption. The advantage of using the convex hull is that it is typically more efficient to compute than the α -shape.

At least $d + 1$ points are required to compute the envelope using the techniques described above. When the number of points in the current cluster is less than $d + 1$, the envelope is simply defined as a sphere centered at the centroid (barycenter) of the cluster.

Increasing the resolution within a cluster:

Initially, MLCS evaluates all points on \mathcal{S}_1 , refining the data according to the cost of each point, and clustering the reduced data. This leads to the first set of clusters \mathcal{C}_{1j} , where the first index refers to the first layer and the index j identifies each cluster. MLCS then computes the envelope of each cluster, \mathcal{E}_{1j} .

The next step consists in increasing the resolution within each envelope to further refine the cluster. The resolution is increased by advancing to the next layer, \mathcal{S}_2 , and finding all points in \mathcal{S}_2 that fall inside each \mathcal{E}_{1j} . Figure 3 sketches how the resolution is increased by considering the points on the next layer. The resulting set of points are ranked according to Eq. (10), we select the top p -percentile of points, and we cluster the refined set of points. This provides a new set of clusters \mathcal{C}_{2k} than might include more or less clusters than the previous set, depending on whether regions were split or merged. The process is repeated sequentially until convergence (see the next section for a definition of convergence).

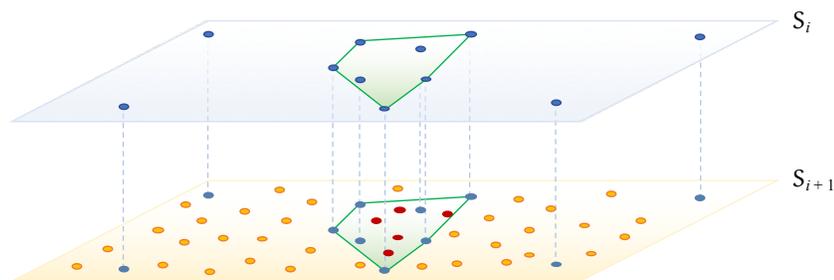


Figure 3: Increasing the resolution in a given region by advancing to a deeper (denser) layer

A cluster \mathcal{C}_{ij} will not be reduced further if:

- The cluster becomes too small;

- The maximum value of $f(\mathbf{x} \in \mathcal{C}_{ij})$ decreased by less than 25% after refining the cluster;
- The cluster stalled, in which case the cluster is no longer explored.

Stalled clusters:

It is possible that the initial exploration of \mathcal{S}_1 results in some clusters that do not contain any VI, $\mathcal{C}_{1j} \cap \mathcal{F} = \emptyset$. Similarly, on a given layer i a cluster might be split and one of the resulting clusters \mathcal{C}_{ij} could not be relevant either. At some point, the algorithm should detect that these clusters do not lead to the failure region but rather introduce unnecessary function evaluations. MLCS implements a criterion to detect when a cluster stalls based on the smallest value of $f(\mathbf{x})$ within the cluster: if it is smaller than the median of the smallest values of all clusters on the previous layer,

$$\inf(f(\mathbf{x} \in \mathcal{C}_{ij})) > \text{median}_k[\inf(f(\mathbf{x} \in \mathcal{C}_{i-1,k}))], \quad (17)$$

then the cluster does not make it to the next layer. As a safety barrier, if the term on the right-hand side becomes smaller than 10ϵ , we use

$$\inf(f(\mathbf{x} \in \mathcal{C}_{ij})) > 10\epsilon \quad (18)$$

in lieu of Eq. (17).

Convergence

Once all clusters on the current layer have been explored, the impact probability is estimated like in Eq. (2) using the total number of samples N_i in the current layer and the number $N_{\mathcal{F}}$ of detected impacts:

$$P(\mathcal{F}) = \frac{N_{\mathcal{F}}}{N_i}. \quad (19)$$

In some cases where the probability is not too small (sufficiently larger than the minimum probability used to determine the number of layers), MLCS can accurately converge to the impact probability without reaching the deepest layer. The test for convergence consists in comparing the relative size of the 95% confidence interval with a certain tolerance,

$$1.96 \sqrt{\frac{1}{N_i} \left(\frac{1}{P} - 1 \right)} < \epsilon_{\text{tol}}. \quad (20)$$

We set the tolerance to $\epsilon_{\text{tol}} = 0.2$. A larger value results in a more efficient search but lower precision in the final result.

Dimension reduction

MLCS can handle problems of arbitrary dimension. However, care should be taken when clustering samples because the failure region might form a clear pattern in a projected subspace of reduced dimension but not in the entire space of initial conditions. For example, consider a problem of dimension $d = 3$ where \mathcal{F} is a cylinder,

$$\mathcal{F} = \{\mathbf{x} \in \mathbb{R}^3 \mid x^2 + y^2 = 1\}. \quad (21)$$

If the samples are clustered in the two-dimensional subspace (x, y) , the algorithm detects the circle $x^2 + y^2 = 1$ and converges to it, capturing all the points in the cylinder. If the algorithm looks for

patterns in three dimensions, the cylinder is divided in several subsets and excessive effort might be invested in merging clusters. On the other hand, if the algorithm clusters data only in one dimension, the algorithm evaluates too many points in the region $x \in [-1, 1]$. Specifying which dimensions should be used for clustering can speed up the overall process. Nevertheless, clustering in too many or too few dimensions does not affect the overall accuracy of the algorithm because the entire region \mathcal{F} is explored in any case.

When no information about the underlying physical model is available, a good rule of thumb is to compute the partial derivatives of the function $f(\mathbf{x})$ at the nominal value of \mathbf{x} to identify the variables that the function is most sensitive to. In general, clustering on these variables yields the best performance of MLCS.

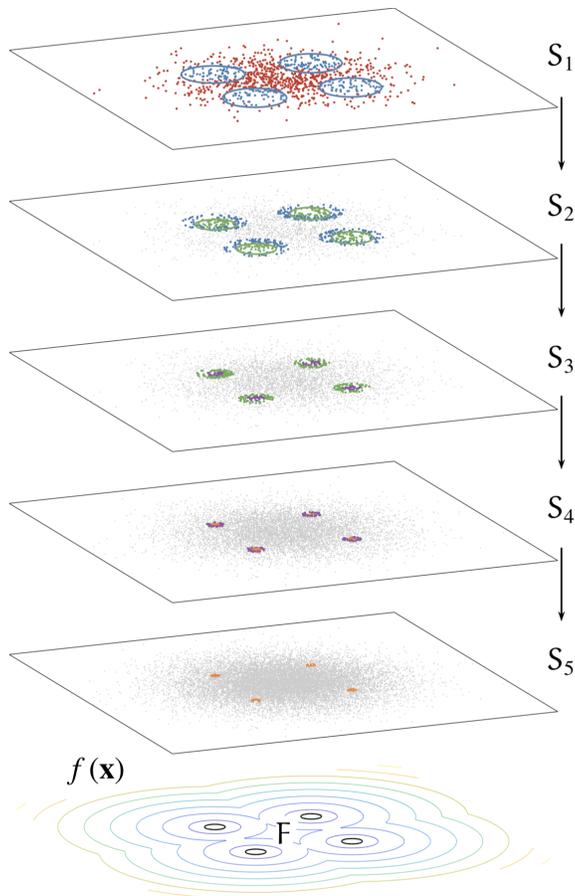


Figure 4: Diagram showing the MLCS algorithm marching through five layers to find the points $(x, y) \in \mathbb{R}^2$ that satisfy $(|x| - 1)^2 + (|y| - 1)^2 < 0.2$.

Figure 4 shows how the MLCS algorithm works with a simple example. The goal is to find the probability $P(\mathcal{F})$ given $\mathcal{F} = \{\mathbf{x} \in \mathbb{R}^2 \mid f(\mathbf{x}) \equiv (|x| - 1)^2 + (|y| - 1)^2 < 0.2\}$, which is $P = 3 \times 10^{-3}$. The contour plot at the bottom of the plot represents the function $f(\mathbf{x})$ and the black circles highlight the failure region \mathcal{F} , defined by four disjoint sets. This example uses $n = 5$ layers sampled from the standard normal distribution with their resolution increasing by a factor $\ell = 2$, starting at $N_1 = 10^3$. The gray dots represent all the points on each layer and the colored points are the samples that are actually evaluated on each layer. The resolution increases from layer S_i to layer S_{i+1} according to

$N_{i+1} = \ell N_i$. During the initial exploration phase, all points on \mathcal{S}_1 were evaluated (red dots in the figure). Using the weights $w_1 = 1$ and $w_2 = w_3 = 0$, the top 30% of points according to Eq. (10) are plotted in blue. MLCS detected four separate clusters whose envelopes appear as solid blue lines. Next, MLCS advanced to \mathcal{S}_2 and found all points that are inside each of the four envelopes. The function $f(\mathbf{x})$ was evaluated at those points and the top 30% of points within each cluster is plotted in green. These steps were repeated in sequence until MLCS arrived at layer \mathcal{S}_5 and found all points in \mathcal{F} , with the resolution dictated by N_5 . In this example, MLCS required 1,628 function evaluations while MC requires $N_5 = 16,000$ evaluations.

The algorithm

Algorithm 1 summarizes how the MLCS method works. After the initialization phase, the algorithm marches through all layers, exploring the regions defined by the envelopes of the clusters from the previous layer.

Algorithm 1 Overview of the MLCS algorithm

```

1: Generate  $n$  layers
2: Map  $f(\mathbf{x})$  over entire first layer # Initial exploration
3: if  $f(\mathbf{x}) < \varepsilon$  then
4:    $N_{\mathcal{F}} = N_{\mathcal{F}} + 1$ 
5: end if
6:  $P(\mathcal{F}) = N_{\mathcal{F}}/N_1$  # Compute probability
7: Check for convergence
8: Select a refined subset of points
9: Detect  $n_{\text{cluster}}$  hierarchical clusters
10: for  $i = 2, \dots, n$  do # Increase resolution
11:   for  $j = 1, \dots, n_{\text{cluster}}$  do # Explore each cluster
12:     Determine envelope of the  $j$ -th cluster
13:     Find all points from  $i$ -th layer that are inside the envelope
14:     Map  $f(\mathbf{x})$  over the points found in the previous step
15:     if  $f(\mathbf{x}) < \varepsilon$  then
16:        $N_{\mathcal{F}} = N_{\mathcal{F}} + 1$ 
17:     end if
18:     Select a refined subset of points
19:     Detect  $n_{\text{cluster}}$  hierarchical clusters
20:   end for
21:    $P(\mathcal{F}) = N_{\mathcal{F}}/N_i$  # Compute probability
22:   Check for convergence
23:   Flag relevant clusters for further exploration and update  $n_{\text{cluster}}$ 
24: end for

```

EXAMPLES

This section evaluates the performance of MLCS for estimating the impact probability of different asteroids. Table 1 lists the orbits of the asteroids at the corresponding epoch of osculation and indicates the JPL orbit solution identification number.*

*<https://ssd.jpl.nasa.gov/sbdb.cgi>

Table 1: Orbits of the sample asteroids

Asteroid	JPL Sol.	Epoch (MJD)	e	a (au)	t_p (MJD)	Ω (deg)	ω (deg)	i (deg)
2017 RH16	3	58021	0.44271	0.87523	57924.39	338.449	226.495	0.619
2013 YB	4	56649	0.42742	1.55106	56685.77	269.032	230.152	0.192
2006 QV89	14	53978	0.22662	1.19650	54034.27	166.220	235.982	1.069
2018 UM1	9	58600	0.17389	1.14370	58793.59	260.610	54.322	1.995
(99942) Apophis	15	53335	0.19125	0.92203	53276.65	204.570	126.199	3.334
	193	54441	0.19108	0.92228	54570.82	204.457	126.394	3.331

We analyze two different close approaches for asteroids 2017 RH16, 2013 YB, and (99942) Apophis, and a single close approach for asteroids 2006 QV89 and 2018 UM1. We selected these asteroids and their close approaches to cover a range of impact probabilities from 10^{-2} down to 10^{-6} , as well as different dynamical regimes. For the particular case of (99942) Apophis, we use two different orbit solutions to analyze its 2029 and 2068 close approaches. We compare the MLCS solution against direct MC sampling to validate the result and to assess the computational savings. The number of layers in these examples is $n = 12$, obtained by setting $P_{\min} = 5 \times 10^{-7}$ in Eq. (8).

In the following examples, MLCS is set to cluster data using the semimajor axis only except for the second close approach of (99942) Apophis, for which data is clustered using the nongravitational parameter driving the Yarkovsky effect.¹⁸ Although using additional variables like the time of periaapsis passage or the eccentricity could potentially reduce the number of function evaluations in certain cases, choosing only one variable proves to be a reliable approach.

Impact probability of asteroid 2017 RH16

According to Sentry,* asteroid 2017 RH16 has an impact probability of 8.3×10^{-4} on 2026-Aug-31, and of 4.9×10^{-6} on 2031-Sep-01. Figure 5 shows how MLCS converged to the failure region in the initial-conditions space for both close approaches (Figs. 5a and 5b, respectively). The initial clusters detected on the first layer appear as orange rectangles, whereas the green rectangles show the clusters that reached the final layer and converged to the impact region. The blue dots are all the samples that were evaluated, and the yellow markers correspond to impacts.

Table 2 compares the solution obtained with MLCS against direct MC sampling. The probability obtained by MLCS, P_{MLCS} , matches exactly the result obtained by running MC sampling on the N_i points of the last layer reached by MLCS, $P_{i,\text{MC}}$. The reference solution is $P_{n,\text{MC}}$, which denotes the MC probability obtained by evaluating all N_n samples of the deepest layer. P_{Sentry} is the impact probability predicted by Sentry. The layer on which MLCS converged is given by the column ‘‘Conv. layer’’. The last two columns of the table show the speedup compared to using MC sampling with all points of the last reached layer and of the deepest layer, N_i and N_n points respectively.

The impact probability on 2026-Aug-31 is relatively large and MLCS converged early on layer 5. The speedup compared to MC sampling on that layer is of one order of magnitude, and over three orders of magnitude compared to running all samples on the deepest layer. The latter corresponds to

*<https://cneos.jpl.nasa.gov/sentry/details.html?des=2017%20RH16>

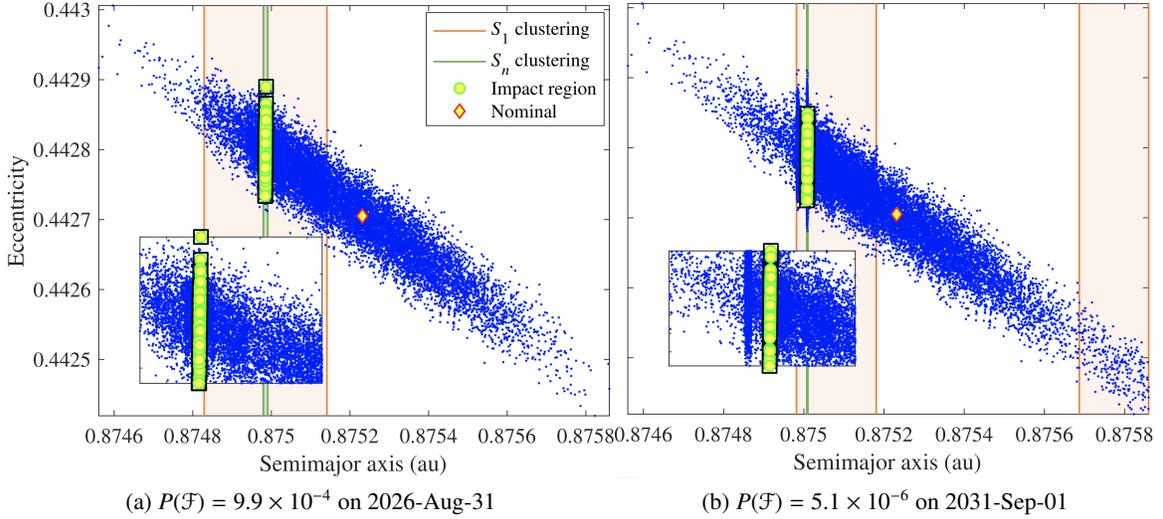


Figure 5: Analysis of the virtual impactors of asteroid 2017 RH16

Table 2: Impact probability data for asteroid 2017 RH16

Date	Accuracy				Conv. layer	N_{eval}	Speedup	
	P_{MLCS}	$P_{i,\text{MC}}$	$P_{n,\text{MC}}$	P_{Sentry}			N_i/N_{eval}	N_n/N_{eval}
2026-Aug-31	9.9E-4	9.9E-4	8.4E-4	8.3E-4	5 / 12	16919	9.5	1210.5
2031-Sep-01	5.1E-6	5.1E-6	5.1E-6	4.9E-6	12 / 12	74307	275.6	275.6

a brute-force MC approach, whereas the former can be regarded as an adaptive MC method. For the second close approach, MLCS reached the deepest layer reducing the computational cost 276 times. In this case, two clusters were detected during the initial exploration phase and only one made it to the last layer. The second cluster stalled on layer 7 according to Eq. (17) and was not passed to the next layer.

Impact probability of asteroid 2013 YB

Figure 6 analyzes the impact probability of asteroid 2013 YB on 2023-Dec-24 and 2024-Dec-23, two close approaches with similar impact probabilities of approximately 2×10^{-5} . In both cases, MLCS initialized the search for the impact regions with four clusters. As the algorithm marched through the layers, three clusters stalled for the 2023 close approach with only one reaching layer 10. This cluster converged to the failure region, defined by a single VI. For the case of the 2024 close approach, two clusters were still active when MLCS converged although only one contained impacts.

The performance of MLCS in this case can be assessed by looking at Table 3. MLCS converged before reaching the deepest layer in both cases, stopping at layer 10. The solution matches exactly the MC solution on layer 10, and the difference with respect to the complete MC sampling of the last layer is bounded by the criterion in Eq. (20). MLCS required more than 30 times fewer function evaluations than MC sampling of layer 10, and 130 times fewer evaluations than the brute-force MC

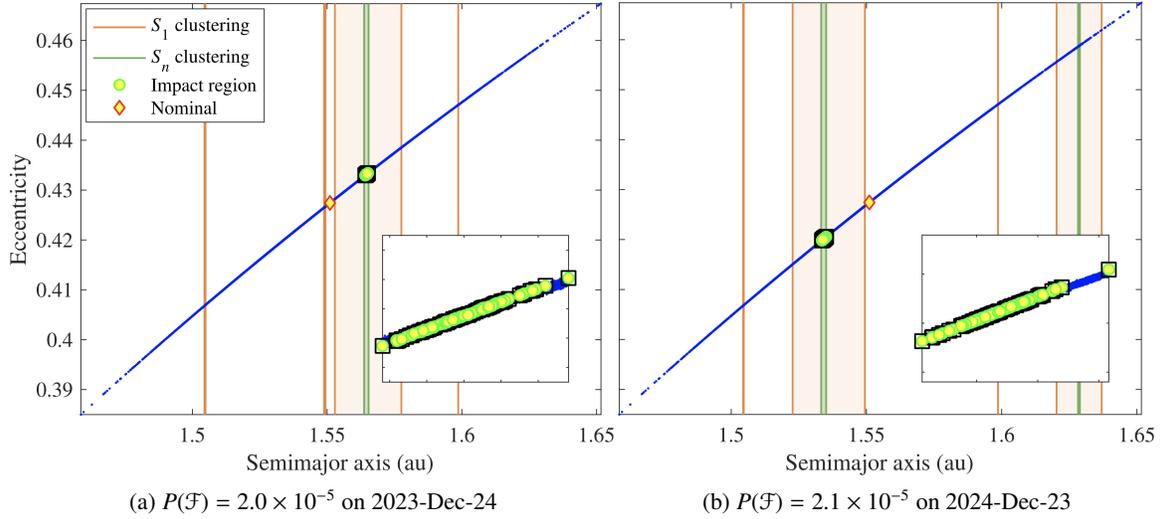


Figure 6: Analysis of the virtual impactors of asteroid 2013 YB

sampling of N_n points.

Table 3: Impact probability data for asteroid 2013 YB

Date	P_{MLCS}	$P_{i,\text{MC}}$	$P_{n,\text{MC}}$	P_{Sentry}	Conv. layer	N_{eval}	N_i/N_{eval}	N_n/N_{eval}
2023-Dec-24	2.0E-5	2.0E-5	1.8E-5	1.9E-5	10 / 12	156443	32.7	130.9
2024-Dec-23	2.1E-5	2.1E-5	2.3E-5	2.0E-5	10 / 12	146418	35.0	129.9

Impact probability of asteroid 2006 QV89

The close approach in September 2023 of Asteroid 2006 QV89 has been selected as an example in which the impact probability (approximately equal to 10^{-6}) approaches the limit estimated by the value of P_{min} used to determine the number of layers. More layers are required to capture smaller probabilities accurately. Figure 7 shows that MLCS initially detected two clusters. The widest cluster stalled and the narrowest cluster converged to the failure region, which is tightly clustered in semimajor axis. MLCS overestimated the size of the final cluster because the top p -percentile of points on the final layer were distributed along a wider region.

The relative speedup in this case is close to a factor of 300 (Table 4). Since the probability is close to P_{min} , MLCS had to reach the final layer in order to locate the impact region.

Table 4: Impact probability data for asteroid 2006 QV89

Date	P_{MLCS}	$P_{i,\text{MC}}$	$P_{n,\text{MC}}$	P_{Sentry}	Conv. layer	N_{eval}	N_i/N_{eval}	N_n/N_{eval}
2023-Sep-08	1.3E-6	1.3E-6	1.3E-6	9.2E-7	12 / 12	70575	290.2	290.2

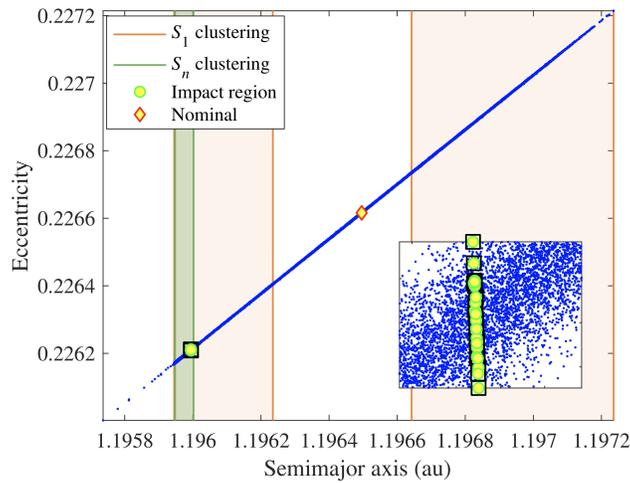


Figure 7: Analysis of the virtual impactor of asteroid 2006 QV89 on 2023-Sep-08, $P(\mathcal{F}) = 1.3 \times 10^{-6}$

Impact probability of asteroid 2018 UM1

Asteroid 2018 UM1 approaches the Earth in 2095, coming closer than 0.1 au. Estimating the impact probability requires propagating its orbit almost 80 years into the future, a long time period after which nonlinear terms dominate the propagation of the orbit uncertainty. Interestingly, this close approach presents three different virtual impactors, distinct regions \mathcal{F}_i in orbital-element space that lead to collisions. Figure 8 shows the cluster sequence converging to the three distinct regions visible in the zoomed view on the bottom-right corner. All three regions are captured by independent clusters that originated from a single cluster from the initial sampling. Two additional clusters reached the final layer despite containing no VIs.

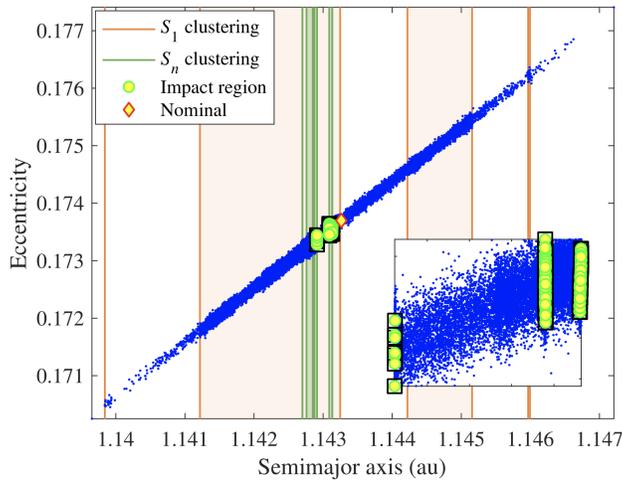


Figure 8: Analysis of the virtual impactors of asteroid 2018 UM1 leading to impacts on 2095-Jun-09

The impact probability and nominal impact date associated with each VI is listed in Table 5. First, it is worth noticing that MLCS predicts an impact probability that is closer to the MC result than Sentry's prediction. The speedup in this case is of a factor of 15 when comparing the number of

function evaluations with the last reached layer and of 61 when considering all N_{12} samples.

Table 5: Impact probability data for asteroid 2018 UM1

Date	P_{MLCS}	$P_{i,\text{MC}}$	$P_{n,\text{MC}}$	P_{Sentry}	Conv. layer	N_{eval}	N_i/N_{eval}	N_n/N_{eval}
2095-Jun-09.08	1.4E-6	1.4E-6	1.6E-6	6.6E-7	10 / 12			
2095-Jun-09.39	1.8E-5	1.8E-5	1.6E-5	2.1E-5	10 / 12	334041	15.3	61.4
2095-Jun-09.40	1.6E-5	1.6E-5	1.6E-5	2.0E-5	10 / 12			

Impact probability of asteroid (99942) Apophis

In December, 2004, the probability of asteroid (99942) Apophis impacting Earth on April 13, 2029, reached almost 3%. Such a high probability triggered numerous observations of the asteroid and, although the impact probability in 2029 was reduced as the orbit was refined, this asteroid continues to be a common case-study of impact monitoring systems.

The JPL orbit solution 15 presented in Table 1 yields the maximum impact probability in 2029. MLCS converged after running only the $N_1 = 10^4$ samples in the first layer. The estimated impact probability is 2.88×10^{-2} on 2029-Apr-13 (see Table 6). In this simple case where the probability is so large that only one layer is required, MLCS is equivalent to direct MC sampling. Figure 9 depicts the region in orbital-elements space that leads to impacts.

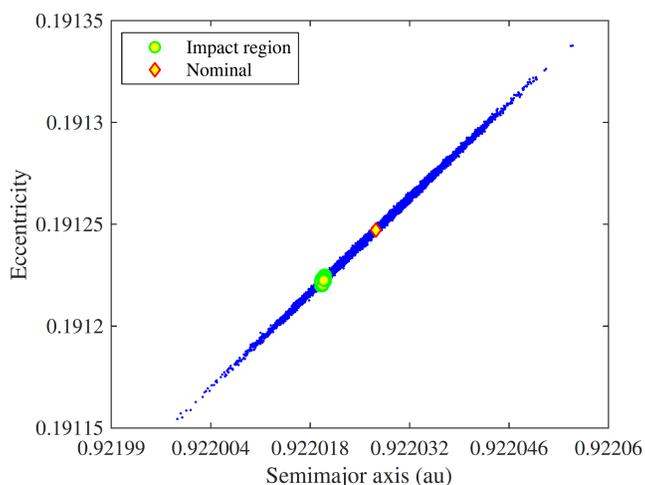


Figure 9: Analysis of the virtual impactors of asteroid (99942) Apophis leading to impacts on 2029-Apr-13

Although later observations ruled out the possible impact, the asteroid still suffers a very close encounter with Earth in 2029, a phenomenon that complicates orbit predictions beyond that date. The resulting dynamical divergence leads to an exponential growth of the uncertainty region, making the problem strongly nonlinear. Figure 10 depicts the nominal orbit of the asteroid before and after the encounter. In addition, recent studies proved that the Yarkovsky effect plays a significant role in the long-term evolution of the asteroid.^{19,20} Farnocchia *et al.*¹⁹ show that a keyhole during the 2029 close approach yields a 7×10^{-6} impact probability in April, 2068, using the JPL orbit

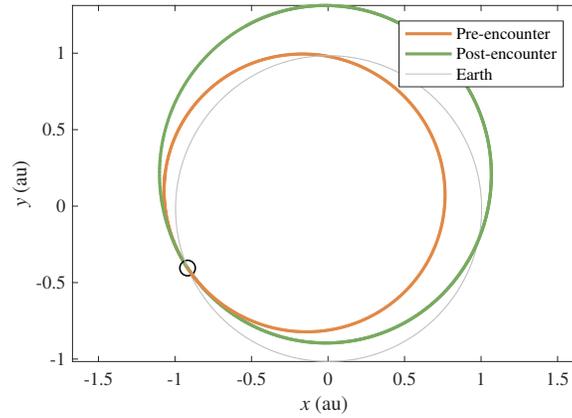


Figure 10: Orbit of asteroid (99942) Apophis before and after its encounter with Earth on 2022-Apr-12

solution 193 from Table 1. The nominal value of the A_2 parameters used to model the Yarkovsky effect is -3.42×10^{-14} .

Since the dynamics of the asteroid is driven by the Yarkovsky effect, MLCS uses the parameter A_2 to cluster the data. Figure 9 shows the performance of MLCS when clustering data using this parameter. Initially, MLCS detected three clusters. The left-most cluster was divided into three smaller clusters that eventually stalled, like the right-most cluster. The central cluster was reduced sequentially until it converged to \mathcal{F} on layer 12.

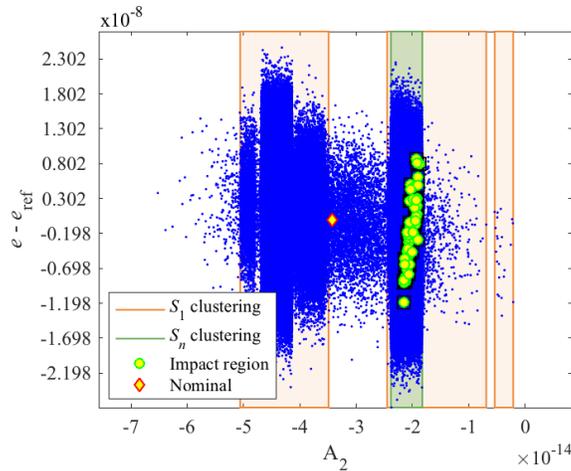


Figure 11: Analysis of the virtual impactors of asteroid (99942) Apophis leading to impacts on 2068-Apr-12, driven by the Yarkovsky effect. The vertical axis plots the difference with respect to the nominal value of the eccentricity

Table 6 presents more details about the impact probability estimation using MLCS. The algorithm reached the last layer and arrived at the same result as direct MC sampling, with more than one order of magnitude less function evaluations.

Table 6: Impact probability data for asteroid (99942) Apophis

Date	P_{MLCS}	$P_{i,\text{MC}}$	$P_{n,\text{MC}}$	P_{Sentry}	Conv. layer	N_{eval}	N_i/N_{eval}	N_n/N_{eval}
2029-Apr-13	2.9E-2	2.9E-2	2.9E-2	–	1 / 1	10000	1.0	1.0
2068-Apr-12	2.5E-6	2.5E-6	2.5E-6	–	12 / 12	1687687	12.1	12.1

CONCLUSIONS

Multi-layer clustered sampling (MLCS) is a useful technique for estimating small probabilities in strongly nonlinear problems. Its accuracy is due to not making any simplifying assumptions about the dynamics, matching exactly the results obtained with direct Monte Carlo (MC) sampling. MLCS is typically several orders of magnitude faster than direct MC sampling thanks to an adaptive sampling technique that focuses only on the regions of interest.

MLCS does not rely on intermediate proposal distributions to resample each refined subdomain. Instead, it generates several layers of samples with increasing resolution at once using the initial probability distribution. As a result, the intermediate sampling problem reduces to a discrete exploration of the sets of pre-generated samples.

The nature of MLCS allows the algorithm to easily identify disjoint failure subregions, i.e. different virtual impactors. The initial subset of samples is grouped into clusters, each of which is resampled, refined, and reclustered again until the cluster no longer improves the solution or it detects impacts with Earth. Since each cluster evolves independently from the rest, it is easy to identify each virtual impactor.

After proving that the MLCS concept works in practice, we will explore more efficient ways to locate the virtual impactors reducing computational time. Optimizing the algorithm requires a better understanding of the impact of each of the configuration parameters, as well as maximizing the convergence rate of individual clusters. For the latter task, we plan to investigate different techniques to predict the size, shape, and orientation of the clusters.

ACKNOWLEDGMENTS

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

REFERENCES

- [1] Farnocchia, D., Chesley, S., Milani, A., Gronchi, G., and Chodas, P., “Orbits, long-term predictions, impact monitoring,” *Asteroids IV*, 2015, pp. 815–834.
- [2] Rubino, G. and Tuffin, B., *Rare event simulation using Monte Carlo methods*, John Wiley & Sons, 2009.
- [3] Chamberlin, A., Chesley, S., Chodas, P., Giorgini, J., Keesey, M., Wimberly, R., and Yeomans, D., “Sentry: an automated close approach monitoring system for near-Earth objects,” *Bulletin of the American Astronomical Society*, Vol. 33, 2001, p. 1116.
- [4] Chesley, S. and Milani, A., “NEODYs: an online information system for near-Earth objects.” *Bulletin of the American Astronomical Society*, Vol. 31, 1999, p. 1117.
- [5] Milani, A., “The asteroid identification problem: I. recovery of lost asteroids,” *Icarus*, Vol. 137, No. 2, 1999, pp. 269–292.
- [6] Milani, A., Chesley, S. R., Sansaturio, M. E., Tommei, G., and Valsecchi, G. B., “Nonlinear impact monitoring: line of variation searches for impactors,” *Icarus*, Vol. 173, No. 2, 2005, pp. 362–384.

- [7] Geweke, J., “Bayesian inference in econometric models using Monte Carlo integration,” *Econometrica: Journal of the Econometric Society*, Vol. 57, No. 6, 1989, pp. 1317–1339.
- [8] Au, S.-K. and Beck, J. L., “Estimation of small failure probabilities in high dimensions by subset simulation,” *Probab. Eng. Mech.*, Vol. 16, No. 4, 2001, pp. 263–277.
- [9] Losacco, M., Di Lizia, P., Armellin, R., and Wittig, A., “A differential algebra-based importance sampling method for impact probability computation on Earth resonant returns of near-Earth objects,” *Mon. Not. R. Astron. Soc.*, Vol. 479, No. 4, 2018, pp. 5474–5490.
- [10] Giles, M. B., “Multilevel Monte Carlo path simulation,” *Operations Research*, Vol. 56, No. 3, 2008, pp. 607–617.
- [11] Giles, M. B. and Waterhouse, B. J., “Multilevel quasi-Monte Carlo path simulation,” *Advanced Financial Modelling, Radon Series on Computational and Applied Mathematics*, 2009, pp. 165–181.
- [12] Giles, M. B., “Multilevel Monte Carlo methods,” *Monte Carlo and Quasi-Monte Carlo Methods 2012*, Springer, 2013, pp. 83–103.
- [13] Roa, J., *Regularization in Orbital Mechanics: Theory and Practice*, Vol. 42, Walter de Gruyter GmbH & Co KG, 2017.
- [14] Milani, A., Chesley, S. R., Boattini, A., Valsecchi, G. B., and Giovanni, B., “Virtual impactors: Search and destroy,” *Icarus*, Vol. 145, No. 1, 2000, pp. 12–24.
- [15] Jain, A. K., Murty, M. N., and Flynn, P. J., “Data clustering: a review,” *ACM computing surveys (CSUR)*, Vol. 31, No. 3, 1999, pp. 264–323.
- [16] Sneath, P. H., Sokal, R. R., et al., *Numerical taxonomy. The principles and practice of numerical classification.*, Freeman, London, 1973.
- [17] Edelsbrunner, H., Kirkpatrick, D., and Seidel, R., “On the shape of a set of points in the plane,” *IEEE Transactions on information theory*, Vol. 29, No. 4, 1983, pp. 551–559.
- [18] Farnocchia, D., Chesley, S., Vokrouhlický, D., Milani, A., Spoto, F., and Bottke, W., “Near Earth asteroids with measurable Yarkovsky effect,” *Icarus*, Vol. 224, No. 1, 2013, pp. 1–13.
- [19] Farnocchia, D., Chesley, S. R., Chodas, P. W., Micheli, M., Tholen, D., Milani, A., Elliott, G., and Bernardi, F., “Yarkovsky-driven impact risk analysis for asteroid (99942) Apophis,” *Icarus*, Vol. 224, No. 1, 2013, pp. 192–200.
- [20] Vokrouhlický, D., Farnocchia, D., Čapek, D., Chesley, S. R., Pravec, P., Scheirich, P., and Müller, T. G., “The Yarkovsky effect for 99942 Apophis,” *Icarus*, Vol. 252, 2015, pp. 277–283.