# Meteoroid Ejecta of Lunar Secondaries Engineering Model

*Anthony M. DeStefano*
NASA, MSFC, EV44

August 20, 2021

## Contents



## List of Figures

## List of Tables

# 1 Battleship Monte Carlo

The random sampling of the ejecta initial state (speed, zenith angle, and azimuth angle) to find a hit can be extremely laborious, especially when the asset is far from the point-of-impact. For simple cases, the set of hits can be found analytically. However, for the general case of complicated asset geometry or an asset that is above the lunar surface (e.g., in lunar orbit), it becomes difficult to solve these directly. Therefore, a Monte Carlo type method is needed to abstract away these complications.

In this section, the Battleship Monte Carlo (BMC) method is introduced. Various ideas in the common literature are used, such as importance sampling and simulated annealing, but are done so in a dynamic way based on previously sampled points. The simple way Battleship Monte Carlo works is there are two strategies working together, a search method and a destroy method. The search method draws from a uniform distribution that covers the entire domain, for each dimension. Once hits are found, the destroy method kicks in to scan near previous hit locations, using a localized uniform distribution. As more hits are found, there are more choices for the destroy method to choose from in order to make a new shot. On average, the destroy method will find hits more efficiently than the search method, but may struggle with expanding to unknown territories. Hence, there should be a balance with searching and destroying. One possible technique to take advantage of both methods would be to start off with heavily searching and then over time switch to the destroy method once a sufficient number of hits are found over the entire domain in order to quickly fill in any valid hits (i.e., ejecta initial states).

## 1.1 Standard Importance Sampling

The difficulty with changing the probability distribution function (PDF) every iteration, in terms of importance sampling, is that there needs to be a concise way to track this PDF and keep it normalized correctly. Starting with the definition of importance sampling, and without loss of generality, only working with 1 dimension,

$$\int_{x_{min}}^{x_{max}} f(x)dx \sim \sum_{i=0}^{N-1} \frac{f(x_i)}{\rho(x_i)} + \mathcal{O}(N^{-1/2}), \tag{1.1}$$

where $x_i$ is pulled from the probability density function $\rho(x)$, given that

$$\int_{x_{min}}^{x_{max}} \rho(x)dx = 1. \tag{1.2}$$

## 1.2 Dynamic Importance Sampling with Annealing

Next, the set of probability densities that had a hit are given by $\rho_k^H$, multiplied by the lifetime mask $L_{k,i}^H$ and domain mask $D_k^H(x_i)$ such that the effective probability density for the BMC method is given by

$$\rho(x_i) = \begin{cases} 1 \text{ , if } N_H = 0, \text{ else} \\ \alpha\rho_0 + \frac{1-\alpha}{N_H - k_0} \sum_{k=\max(0, N_H - N_p)}^{N_H} \rho_k^H L_{k,i}^H D_k^H(x_i) \end{cases} , \tag{1.3}$$

where $N_H$ is the total number of hits so far, $N_p$ is the maximum number of hits to consider (to avoid biasing the first hits), and the individual probability densities are given by

$$\rho_k^H = \frac{1}{\triangle x_k}. \tag{1.4}$$

The probability density of the entire domain is given by

$$\rho_0 = \frac{1}{x_{max} - x_{min}}. \tag{1.5}$$

The lifetime mask is defined as

$$L_{k,i}^H = \begin{cases} 1 \text{ , if } L_k < i - i_k \text{, else} \\ 0 \end{cases}, \tag{1.6}$$

where $i_k$ is the iteration at when the $k$-th hit was made, and the domain mask is defined as

$$D_k^H(x_i) = \begin{cases} 1 \text{ , if } |x_k - x_i| \leqslant \triangle x_k/2 \text{, else} \\ 0 \end{cases}, \tag{1.7}$$

where $\triangle x_k$ is the width of the uniform PDF, centered on $x_k$.

When pulling a new sample $x_i$, first a uniform random number $r_0$ from $0$ to $1$ is pulled and compared with $\alpha$.

- If $r_0 \leqslant \alpha$ (i.e., the search method), then $x_i$ is sampled from a uniform distribution that spans the entire domain, $x_{min} \leqslant x_i \leqslant x_{max}$.
- Otherwise, if $r_0 > \alpha$ (i.e., the destroy method), then a previous hit location $x_k$ is chosen at random with equal probability, such that $x_i$ is sampled from a uniform distribution[1] spanning $x_k - \triangle x_k/2 \leqslant x_i \leqslant x_k + \triangle x_k/2$.

The new lifetime $L_{N_H+1}$ can be computed as

$$L_{N_H+1} = \beta L_k, \tag{1.8}$$

where $0 \leqslant \beta \leqslant 1$ is the lifetime reduction rate. This helps to not bias a certain early population of hit points by setting the new lifetime as the max lifetime. If the new hit originated from the search method, then the lifetime would be set as the max. Any hits found from the destroy method would follow Equation (1.8).

Here, $\alpha$ could in principle change by iteration i.e., simulated annealing. One strategy could be to start with a larger $\alpha_0$, giving more searching, and slowly bring the percentage to a final value $\alpha_f$, changing to more destroy, multiplying by a rate $\gamma$ following

$$\alpha_i = \gamma(\alpha_{i-1} - \alpha_f) + \alpha_f, \tag{1.9}$$

such that for large $i$, $\alpha_i \sim \alpha_f$.

[1] Both a triangular and parabolic region were considered. Difficulties arise when the boundaries of $D_k^H$ go outside of the overall main domain. In this case, $D_k^H$ must be reduced to fit inside of the main domain, which causes the normalization of a triangular region amounts to solving a quadratic equation and the parabolic region amounts to solving a cubic equation. In general, this can be computationally expensive, so a uniform distribution is trivial when finding the normalization constant, which is just the inverse of the width of the region.

# 2 Geometry and Coordinates

In this section, the geometry of the asset and Moon are discussed as well as the coordinated systems used. The asset is made of a combination of basic shapes, instead of a surface of arbitrary polygons. Using basic shapes provides a quicker way at computing collisions[2], especially when various curvatures are needed. The basic shapes used so far are 1-parameter spheres (Section 2.1.1), 2-parameter cylinders (Section 2.1.2), and 3-parameter rectangular prisms (Section 2.1.3). The asset can be made of any number of shapes equal to or greater than one (Section 2.3), where the shapes can overlap.

There are three levels of coordinate systems, at the individual shape level (Section 2.2), at the asset level (Section 2.4), and at the planetary level (Section 2.6). The ejecta secondaries are transported in planetary level coordinate system, so when computing collisions, the ejecta position must be translated into the coordinate system of each shape separately. To avoid computing collisions at each time step for each shape, a bounding radius is defined around the asset (see Section 2.3 for more details). If the ejecta position is outside of the bounding radius, collision detection is skipped for that trajectory time step.

## 2.1 Shapes

### 2.1.1 Sphere

### 2.1.2 Cylinder

### 2.1.3 Rectangular Prism

## 2.2 Shape Coordinates

## 2.3 Asset Geometry

## 2.4 Asset Coordinates

## 2.5 Planetary Geometry

## 2.6 Planetary Coordinates

[2] I did not want to spend a lot of time on game engine dynamics...

# References