DimSun Insolation File

Saptarshi Bandyopadhyay, Pedro Neves

October 27, 2025

User Guide

1.1 Requirements

To run the DimSun Insolation File (link), the user must have the following components installed:

- MATLAB software with the following toolbox: Symbolic Toolbox (for solving dynamical equations);
- MuSCAT software with the corresponding Supporting Files (section 1.2).

1.2 Overview and Installation of MuSCAT

From the MuSCAT page on GitHub (https://github.com/nasa/muscat):

> Multi-Spacecraft Concept and Autonomy Tool (MuSCAT) is an open-source simulation software offering an integrated platform for conducting low-fidelity simulaitions of spacecraft mission concepts and testing autonomy algorithms. Whether designing single or multiple spacecraft missions, MuSCAT provides comprehensive simulation capabilities for both cruising and orbiting spacecraft.
>
> MuSCAT employs a dual-loop architecture that efficiently handles different timescales for spacecraft dynamics, allowing for accurate simulation of both fast-changing attitude dynamics and slower-evolving orbital mechanics in a computationally efficient manner.

For the installation procedure, follow this link.

As an alternative to `git clone`, the user may download MuSCAT as a *.zip* file. Ensure that both the *MuSCAT_Matlab_v2* and *MuSCAT_Supporting_Files* folders are within the same parent directory. The supporting files folder contains essential resources such as the SPICE kernels. Verify that all directory paths set in the script are correct and that MATLAB has the necessary permissions to read files.

1.3 Code setup

The insolation file provided analyzes the transmission of solar rays from the Sun to the Earth, accounting for the influence of a dust cloud placed in the Sun-Earth L1 Lagrangian point (SEL1).

First and foremost, the code starts by initializing several constants, such as the masses and radii of the Earth and Sun, for example.

After defining these constants, the code configures the file paths for the SPICE kernel data and MuSCAT resources. It is **critical** to ensure that the `Insolation_File_v4.m` file and the MuSCAT and MuSCAT_Supporting_Files folders are on the same parent directory. In case the user prefers to have the MuSCAT and MuSCAT_Supporting_Files folders in a different parent directory, the local paths should be updated accordingly in the code.

Following this, the code loads three SPICE kernels for the planetary ephemerides, leap seconds and planetary constants.

After this, **the user should set a specific UTC date and time** (`time_utc`) **and the total number of rays** (`num_rays`) for the simulation.

Overall, the two variables set by the user are: `time_utc` and `num_rays`.

1.4 Simulation

The first computations start by converting the selected UTC time to Ephemeris Time (ET), since all SPICE functions use this time system. Then, the code retrieves the Earth's position relative to the Sun in the J2000 reference frame, with the distance between the Sun and the Earth being calculated using the norm of this position vector.

1.4.1 Computation of the SEL1 point

The SEL1 point is computed by solving the restricted three-body problem equation.

$$\frac{G \cdot M_{\odot}}{(SEL1)^2 \cdot 10^6} - \frac{G \cdot M_{\oplus}}{(d_{\odot-\oplus} - (SEL1))^2 \cdot 10^6} - \omega_{\oplus}^2 \cdot (SEL1) \cdot 10^3 = 0 \tag{1.1}$$

The solution gives the location of the SEL1 point relative to the Sun and, subsequently, the distance between Earth and SEL1.

$$d_{\oplus-SEL1} = d_{\odot-\oplus} - d_{\odot-SEL1} \tag{1.2}$$

1.4.2 Rotation Matrix for the Ray

In order to align the mathematical axes of the simulation with the true Sun-to-Earth vector the chosen epoch, a rotation matrix is built, with the primary vector used for the ray-tracing being set in the positive x direction. Then, this is aligned to match the orientation of the Earth using a rotation matrix.

First, the normalized vectors are defined as:

$$u = \frac{[d, 0, 0]}{\|[d, 0, 0]\|} \tag{1.3}$$

$$w = \frac{\mathbf{r}_E}{\|\mathbf{r}_E\|} \tag{1.4}$$

where d is the distance between the Sun and the Earth and $\|\mathbf{r}_E\|$ the Earth position vector.

Depending on the relationship between u and v, the code analyzes possible different scenarios. If these vectors are parallel, no rotation is required, making $R = I_3$.

On the other hand, if these vectors are opposite ($u \cdot v = -1$), an auxiliary vector t (`third_vector`) independent of u is chosen to avoid collinearity and the axis is computed as

$$v = \frac{u \times t}{\|u \times t\|} \tag{1.5}$$

leading to the corresponding rotation matrix

$$R = -I_3 + 2vv^T \tag{1.6}$$

In case neither of these applies, Rodrigues' rotation formula is applied, where

$$v = u \times w \tag{1.7}$$

with the skew-symmetric matrix

$$\hat{v} = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \tag{1.8}$$

and the rotation matrix becoming

$$R = I_3 + \hat{v} + \frac{\hat{v}^2}{1 + u \cdot w} \tag{1.9}$$

For further information on this topic and ray-tracing in general, the reader is referred to [1].

1.4.3 Intersection of Ray with Sphere

Within the main Monte Carlo loop, which iterates over each simulated solar ray, the code generates random starting points on the Sun's disc and random points within a circle at the SEL1 location, representing the dust cloud region. For each ray, calculations check whether the ray's starting point and path intersect the dust cloud.

Our ray passes those three points:

- $[x_1, y_1, z_1]$ at the disc of the Sun, passing through the center of the Sun
- $[x_2, y_2, z_2]$ at the location of the Sunshade (like Sun-Earth Lagrangian-1 point)

- $[x_3, y_3, z_3]$ at the disc of the Earth, passing through the center of the Earth

Our objective is to find the intersection of this ray with the sphere of Earth. Hence we will represent the ray as:

$$[x, y, z] = t * [x_2, y_2, z_2] + (1 - t) * [x_3, y_3, z_3]\,, \tag{1.10}$$

where t is a variable.

t is a fractional distance along the ray, having different interpretations:

- $t = 0 \rightarrow$ the point is at P_3 (Earth center);
- $t = 1 \rightarrow$ the point is at P_2 (dust cloud at L_1);
- $0 < t < 1 \rightarrow$ the point lies between the dust cloud and the Earth center;
- $t < 0$ or $t > 1 \rightarrow$ the point lies outside the ray path segment

The equation of the sphere is given by:

$$(x - d_{S,E})^2 + y^2 + z^2 = r_E^2\,, \tag{1.11}$$

where $d_{S,E}$ is the distance from Sun to Earth, and r_E is the radius of Earth.

Substituting Eq (1.10) into Eq. (1.11) gives:

$$(tx_2 + (1-t)x_3 - d_{S,E})^2 + (ty_2 + (1-t)y_3)^2 + (tz_2 + (1-t)z_3)^2 = r_E^2\,, \tag{1.12}$$

$$(t(x_2 - x_3) + (x_3 - d_{S,E}))^2 + (t(y_2 - y_3) + y_3)^2 + (t(z_2 - z_3) + z_3)^2 = r_E^2\,, \tag{1.13}$$

$$\begin{aligned} &t^2(x_2 - x_3)^2 + 2t(x_2 - x_3)(x_3 - d_{S,E}) + (x_3 - d_{S,E})^2 + t^2(y_2 - y_3)^2 \\ &\qquad + 2t(y_2 - y_3)y_3 + y_3^2 + t^2(z_2 - z_3)^2 + 2t(z_2 - z_3)z_3 + z_3^2 = r_E^2\,, \end{aligned} \tag{1.14}$$

$$\begin{aligned} &At^2 + Bt + C = 0\,, \qquad \text{where} \\ &A = (x_2 - x_3)^2 + (y_2 - y_3)^2 + (z_2 - z_3)^2 \\ &B = 2(x_2 - x_3)(x_3 - d_{S,E}) + 2(y_2 - y_3)y_3 + 2(z_2 - z_3)z_3 \\ &C = (x_3 - d_{S,E})^2 + y_3^2 + z_3^2 - r_E^2 \end{aligned} \tag{1.15}$$

The solution for t is given by the solution of the quadratic equation Eq. (1.15):

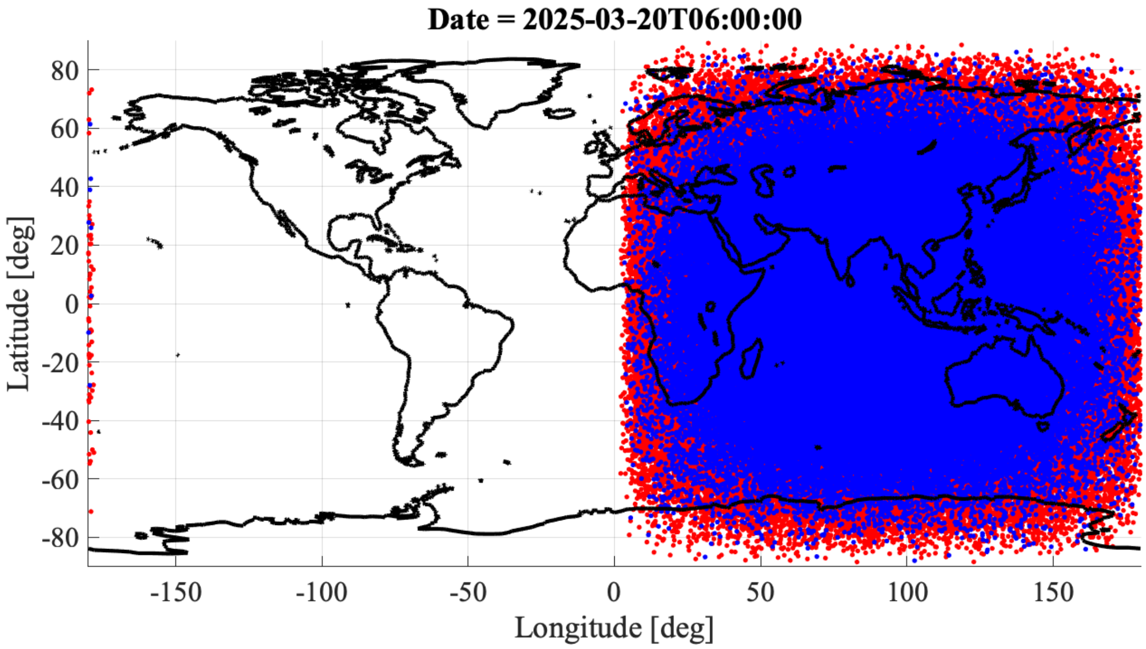
$$t = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}\,. \tag{1.16}$$

In the code, the first solution, t_+, is computed (from the Sun towards Earth). If this solution is negative, meaning the interception point lies behind the intended direction of the ray, the code uses t_- instead. This ensures that the intersection takes place in the intended direction. After t is computed, it is substituted into (1.10) to determine the exact coordinates of the intersection point on the Earth's surface.

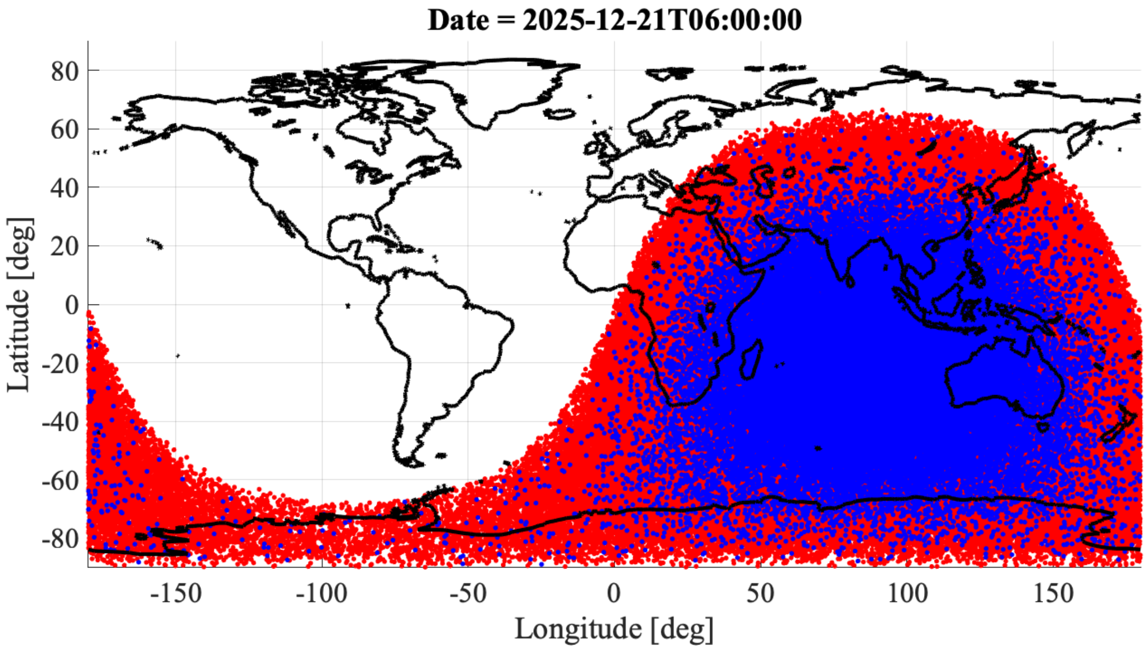
Overall, the code is generating a large number of rays and, for each iteration, a ray is randomly initialized at the surface of the Sun and taken towards a random location near SEL1 and then towards the Earth's surface. By doing this, the code takes into account rays that may or may not intersect the dust cloud and, if they do, the ray gets flagged as having been attenuated. After this, the code performs a check on whether the attenuated ray intersects the Earth's surface or not. If it does, that intersection point is stored, together with the ray's initial position on the Sun and its intersection position with the dust cloud at L1. All the data is saved on a .mat file in the local folder.

1.5 Visualization

After all these computations, for all the rays that intersect the Earth, the coordinates of intersection are converted into latitude and longitude, using functions from the SPICE toolkit. This is followed by visualization in a global map (`coastlines` in the code), with the red dots plotted meaning the rays reached Earth directly and the blue dots that they passed through the dust cloud first. Two example figures of the expected output are shown below (figure 1.1).



(a) March 20th, 2025 at 06:00:00 UTC.



(b) December 21st, 2025 at 06:00:00 UTC.

Figure 1.1: Ray tracing map (example output)

Bibliography

[1] Andrew S. Glassner et al. *An Introduction to Ray Tracing*. Academic Press, 1989. ISBN: 0-12-286160-4. URL: https://www.realtimerendering.com/raytracing/An-Introduction-to-Ray-Tracing-The-Morgan-Kaufmann-Series-in-Computer-Graphics-.pdf.