Generating Random Surfaces for Thermal Emission Studies

ABHISHEK,[1,*], GUNESHWAR THANGJAM,[1,+]

[1]National Institute of Science Education and Research, Jatni, Khordha 752050, Odisha, India
[*]abhishek.2019@niser.ac.in
[+]thangjam@niser.ac.in

Abstract: This report presents different methods to generate random surfaces for thermal emission studies. The methods include the generation of random surfaces of several shapes and sizes using the Python programming language. Most of the methods are employed from the two sources [1] and [2]. My part of the work is to implement the methods in Python and generate the random surfaces. In addition, the report also presents the solution of the 1D heat equation for the moon's surface.

1. Introduction

The Moon Mineralogy Mapper (M^3) [3] radiance spectra on the lunar surface include a thermal emission component that becomes significant around 2.2 μm, particularly in the 2.8–3.0 μm range where water and hydroxyl (OH) absorption features are present. To interpret these spectra, compensation for the local surface temperature is required.

Clark et al. (2011) [4] proposed a complex method for temperature estimation that involves comparing different parts of the reflectance spectrum. However, this method can be imprecise, particularly in regions with strong pyroxene absorptions, and it often fails to fully account for thermal emissions in the OH absorption range.

Wöhler et al. (2014) [5] suggested a simpler method, which combines a standard lunar sample spectrum and a blackbody spectrum. However, their estimates are often too low for surface temperatures below 250–280 K.

Li and Milliken (2016) [6] introduced another approach using a thermal emission model and a reflectance model to estimate temperature. While this method is more accurate, it still has difficulties in areas rich in certain minerals and at high incidence angles.

In response to these issues, the researchers developed two new methods. One method calculates solar energy per surface element and models thermal energy flow within lunar soil. The other method derives surface properties from $\mathbf{M^3}$ measurements.

2. Effects of Surface Roughness on Thermal Emission

A smooth surface emits radiation uniformly in all directions, and its temperature corresponds to the observed brightness temperature. However, a rough surface, with its uneven angles and shadows, exhibits significant temperature variations across its extent. Consequently, the observed radiation is not uniformly distributed and does not correspond to a single temperature. Instead, it depends on the incidence of sunlight and the viewing geometry. If sunlight predominantly strikes directly and we observe those regions, the observed radiation is higher, a phenomenon known as beaming. Conversely, if there are many shadows, the observed radiation is lower. Surface roughness significantly influences the emitted radiation.

Under normal circumstances, we can estimate the surface temperature from the emitted radiation using Planck's law:

$$B(\lambda, T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda kT}} - 1} \tag{1}$$

where $B(\lambda, T)$ is the spectral radiance at wavelength λ and temperature T. However, the emission spectra of rough surfaces comprise several components emitting a superposition of several wavelengths due to heat diffusion and self-heating of the surface. As a result, it is not possible to accurately determine the temperature using Planck's law.

3. Different Topographies of the Moon's Surface

3.1. Concave Spherical Segments

A 1 m^2 area in the $x - y$-plane is divided into a quadratic grid with 1024 squares. Each node on the grid is assigned a z-value, which is zero outside a circular rim centered on $x = y = 0$ with radius $r = 0.437$ m, to produce a crater coverage of $f = 0.6$. Within the rim, the z-coordinate is given by

$$z = R - h - \sqrt{R^2 - x^2 - y^2} \tag{2}$$

This gives the terrain within the rim the shape of a concave spherical segment. Here, h is the crater depth and $R = \frac{r^2+h^2}{2h}$ is the curvature radius of the spherical segment. By varying h, we obtain different degrees of roughness. The four nodes on the terrain surface corresponding to a particular square in the xy-plane are used to form two triangular facets. The total number of triangular facets on the surface is thus 2048. For each facet, we calculate the surface area, the normal vector, and its tilt with respect to the z-axis. These are used to calculate the RMS slope angle s according to Eq. (1). There are several types of spherical segment terrains with $s = 5°$, 20°, and 35° by using $h = 0.0359$ m, 0.151 m, and 0.313 m, respectively. In this study, only the $s = 35°$ terrain is considered and reconstructed.

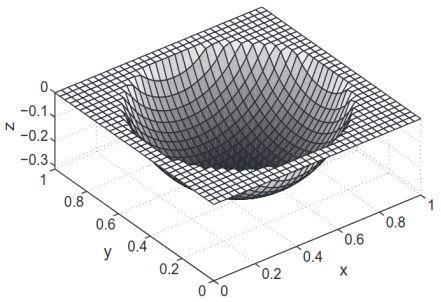

Fig. 1. Concave spherical segment with $s = 35°$ [2].

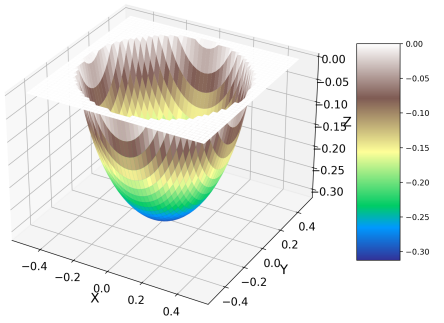

Fig. 2. Reconstructed concave spherical segment with $s = 35°$.

In Figure 1, a topography of concave spherical segment with $s = 35°$ is shown which is takem from [2]. The reconstructed concave spherical segment with $s = 35°$ is shown in Figure 2.

3.2. Parallel sinusoidal trenches

To create parallel sinusoidal trenches, we use the same quadratic grid as in Section 3.1, but with z-coordinates determined by $z = C_t + C_t \sin(2\pi(N_{\text{trench}} + 0.5)x)$, resulting in three trenches parallel to the y-axis. The amplitude C_t controls the roughness level. We construct 2048 triangular surface facets based on Cartesian node positions and calculate s as for spherical segments. To achieve $s = 5°$, 20°, and 35°, we set $C_t = 0.00575$ m, 0.02438 m, and 0.04912 m, respectively. In this study, only the $s = 35°$ terrain is considered and reconstructed.

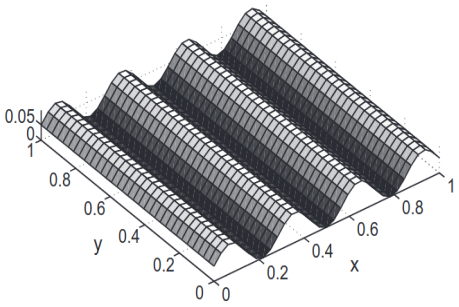

Fig. 3. sinusoidal trenches with $s = 35°$ [2].

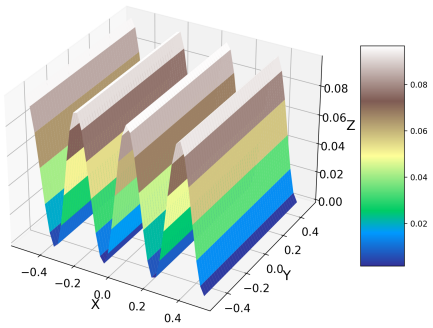

Fig. 4. Reconstructed sinusoidal trenches with $s = 35°$.

In Figure 5, a topography of sinusoidal trenches with $s = 35°$ is shown which is reconstructed as shown in Figure 6.

3.3. Random Gaussians

Random numbers are required to generate random surfaces. The random numbers are generated using the linear congruential generator (LCG) method that was taught in the course.

- The function `lcg(modulus, a, c, seed)` is defined to implement a Linear Congruential Generator (LCG). The LCG is a type of pseudorandom number generator algorithm that yields a sequence of pseudorandom numbers. The LCG uses the equation $X_{n+1} = (aX_n + c) \mod m$ to generate the next number in the sequence, where:
  - X_n is the current number in the sequence (the seed for the first number),
  - a is the multiplier,
  - c is the increment,
  - m is the modulus.
- The parameters for the LCG are defined as follows:
  - `modulus = 2**32`: The modulus m is set to 2^{32}.
  - `a = 1103515245`: The multiplier a is set to 1103515245.
  - `c = 12345`: The increment c is set to 12345.
  - `seed = int.from_bytes(os.urandom(4), byteorder="big")`: Sets the seed X_0 to a random 32-bit integer from the OS's random number source.
- The function `lcg` is a generator function that, when called, returns an iterator that yields pseudorandom numbers in the range $[0, 1)$.

3.3.1. 2D random Gaussian surface using correlation length

This method is employed from the source [7]. A random surface is defined by the function $h = f(x, y)$, where h denotes the height at a given position (x, y). This height, represented by the variable h, is subject to randomness with a specific mean and standard deviation. The characteristics of the surface are determined by two key parameters: the root mean square (rms) of the height distribution, denoted as h_{rms} (equivalent to the standard deviation of h), and the correlation length c_l, which governs the spatial frequency of surface variations or the roughness

of surface is defined using this correlation length. In our isotropic surface, both the correlation lengths in the x and y directions are equal and denoted as $c_{lx} = c_{ly} = c_l$.

Following are the steps to generate the random Gaussian surface using correlation length:

- Generate a random 2D matrix of size $N \times N$ with random numbers.
- Perform a 2D Fourier transform of the random matrix.
- N can not be chosen arbitrarily, it should satisfy the Nyquist criterion [8].

$$\frac{N}{rl} < \frac{2}{cl}$$

  Where N is the number of points on the x and y axis individually, and rl is the length of the x and y sides of the square. cl is the correlation length.

- The final marix is obtained by convolving the random matrix with a Gaussian filter.

$$G = \exp\left(\frac{x^2 + y^2}{\frac{cl^2}{2}}\right)$$

  Here cl is the correlation length. x and y are the coordinates of the matrix for 2D matrix.

- The final matrix is obtained by convolving the random matrix with the Gaussian filter.

$$\mathbf{H_f} = \frac{2L}{\sqrt{\pi} \cdot N \cdot cl} \, \mathrm{FFT}^{-1} \left(\mathrm{FFT}(\mathbf{H}) \cdot \mathrm{FFT}(\mathbf{G})\right) \tag{3}$$

Here $\mathbf{H}$ is the initial 2D N×N matrix, $\mathbf{G}$ is the Gaussian filter, and H_f is the final matrix.

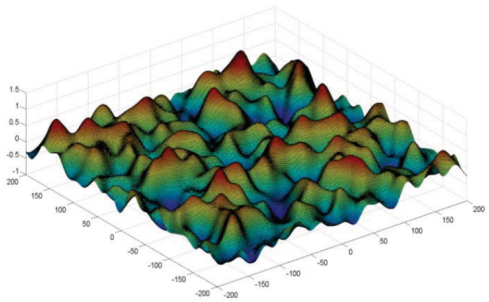

Fig. 5. 2D Gaussian random surface taken from the source [7].

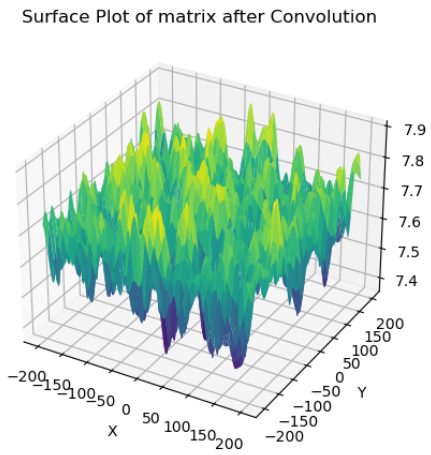

Fig. 6. Reconstructed 2D Gaussian random surface.

3.3.2. 2D random Gaussian surface using superposition of multiple Gaussians

We generate a rough surface using random Gaussians on the same x-y grid as mentioned in Section 3.1. Each node's z-coordinate is determined by the superimposition of Gaussian functions, which are randomly generated based on amplitude, width, and location.

The z-coordinate of the (x, y) node is given by following equation:

$$z(x,y) = z_0(x,y) + A_{\text{gauss}} \exp\left(-\frac{(y-y_0)^2 + (x-x_0)^2}{2\sigma_g^2}\right)$$

Here, $z_0(x,y)$ is the z-coordinate before the addition of the new Gaussian. The other parameters are defined as:

$$\begin{aligned} A_{\text{gauss}} &= C \cdot rn[-1,1] \\ \sigma_g &= 2.25 + 0.75 \cdot rn[0,1] \\ x_0 &= 16 \cdot rn[-1,1] \\ y_0 &= 16 \cdot rn[-1,1] \end{aligned}$$

In these definitions, $rn[-1,1]$ and $rn[0,1]$ are real random numbers uniformly generated within the intervals $[-1,1]$ and $[0,1]$, respectively. C is a constant that varies based on the required roughness level. In these simulations, we use $C = 0.65$ to create surfaces with $s = 35°$.

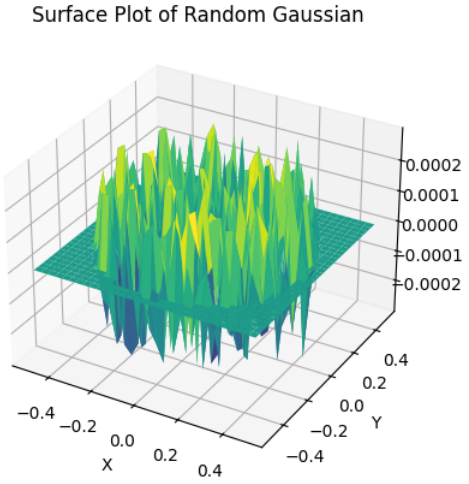

Fig. 7. 2D Gaussian random surface reconstructed using the superposition of multiple gaussians.

4. Assumptions and 1D Heat Conduction

Many researchers have assumed that the size scale of topographic features D is significantly larger than the thermal skin depth L of the surface material. The thermal skin depth is defined as $L = \sqrt{\frac{j}{qcx}}$, where j, q, and c are the heat conductivity, mass density, and specific heat capacity, respectively, while $x = \frac{2\pi}{P}$ is the angular velocity, with P being the the rotational period of the object under consideration. The surface temperature fluctuates during rotation, and L is the depth under the surface where the amplitude has decreased by a factor $e \approx 2.7$ compared to the surface value. Thus, L measures the distance over which heat conduction starts to become ineffective in transmitting temperature changes from one part of the surface to another. The assumption that $D >> L$ is used to justify considering only 1D heat conduction for the smallest resolution elements of the models. Therefore, these researchers neglect lateral heat conduction and the more advanced models use a 1D heat equation solver for each facet in the plate model that resolves the topographic features.

4.1. 1D Heat Equation using Finite Difference Method

The finite difference method is a numerical method used for solving differential equations. It approximates derivatives by finite differences. In the context of the 1D heat equation, which is a parabolic partial differential equation, the finite difference method can be used to approximate the solution at discrete points in space and time.

Property	**Value**
Density	1800 kg/m^3
Heat Capacity	840 J/(kg.K)
Thermal Conductivity	9.30×10^{-3} W/(m.K)

Table 1. Properties of Moon Regolith [9]

$$\alpha = \frac{9.3 \times 10^{-3} \times 1800}{840} \approx 0.020\,\mathrm{m}^2/\mathrm{s} \tag{4}$$

The 1D heat equation is given by:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2} \tag{5}$$

where u is the temperature, t is time, x is the spatial coordinate, and α is the thermal diffusivity.

To solve this equation using the finite difference method, we discretize the spatial domain into N points and the time domain into M points. We can then approximate the second derivative in space using the central difference scheme and the first derivative in time using the forward difference scheme. This leads to the following finite difference equation:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \alpha \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} \tag{6}$$

where u_i^n is the temperature at spatial point i and time point n, and Δt and Δx are the time and space step sizes, respectively.

This equation can be rearranged to give an explicit formula for the temperature at the next time step in terms of the temperature at the current and neighboring spatial points at the current time step:

$$u_i^{n+1} = u_i^n + \frac{\alpha \Delta t}{\Delta x^2}(u_{i+1}^n - 2u_i^n + u_{i-1}^n) \tag{7}$$

This formula can be used to iteratively update the temperature at each point in space and time, starting from an initial temperature distribution at time $t = 0$.

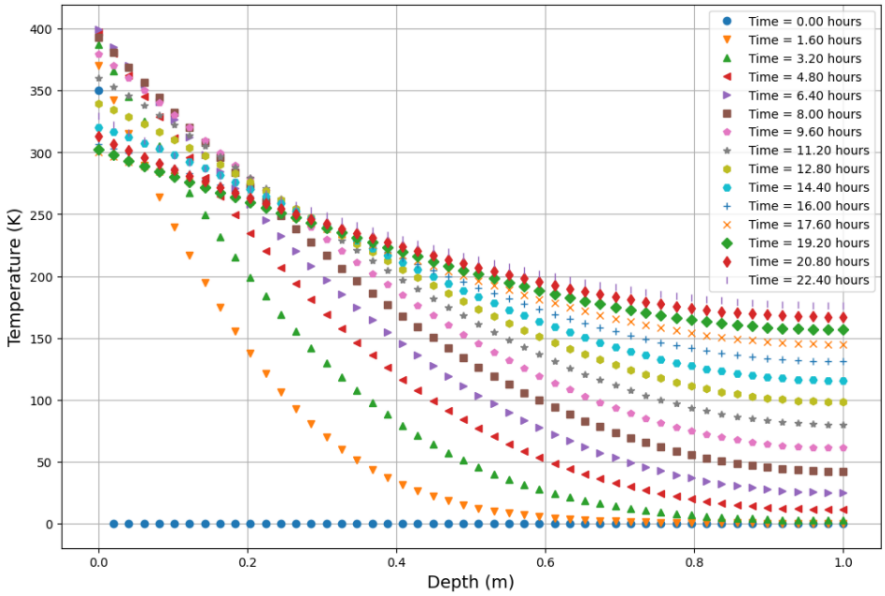

Fig. 8. Solution of the 1D heat equation for the moon's surface.

Figure 8 shows the solution of the 1D heat equation for the moon's surface. The temperature distribution is shown at different time steps, illustrating how the temperature evolves over time due to heat conduction.

4.2. Summary

This report presents the implementation of various methods to generate random surfaces for thermal emission studies, particularly focusing on the lunar surface. The methods employed are primarily derived from two sources [1] and [2], and are implemented using the Python programming language.

The report discusses different types of topographies and the corresponding numerical methods used to generate these surfaces. A key component of this work is the development of a random number generator, which is utilized to create Gaussian random surfaces.

In addition to the generation of random surfaces, the report also presents the solution of the one-dimensional heat equation for the moon's surface. This is achieved through an iterative process over various time steps, providing a detailed understanding of the heat conduction within the lunar regolith.

References

1. C. Wöhler, A. Grumpe, A. A. Berezhnoy, *et al.*, "Temperature regime and water/hydroxyl behavior in the crater boguslawsky on the moon," Icarus **285**, 118–136 (2017).
2. B. J. Davidsson, H. Rickman, J. L. Bandfield, *et al.*, "Interpretation of thermal emission. i. the effect of roughness for spatially resolved atmosphereless bodies," Icarus **252**, 1–21 (2015).
3. NASA Science, "Chandrayaan-1," https://science.nasa.gov/mission/chandrayaan-1/ (2020). Accessed: April 27, 2024.
4. R. N. Clark, C. M. Pieters, R. O. Green, *et al.*, "Thermal removal from near-infrared imaging spectroscopy data of the Moon," J. Geophys. Res. **116**, E00G16 (2011).
5. C. Wöhler, A. Grumpe, A. Berezhnoy, *et al.*, "Integrated topographic, photometric and spectral analysis of the lunar surface: Application to impact melt flows and ponds," Icarus **235**, 86–122 (2014).
6. S. Li and R. E. Milliken, "An empirical thermal correction model for Moon Mineralogy Mapper data constrained by laboratory spectra and Diviner temperatures," J. Geophys. Res. Planets **121**, 2081–2107 (2016).
7. M. Mrnka, "Random gaussian rough surfaces for full-wave electromagnetic simulations," IEEE (2017).

8. “Nyquist–shannon sampling theorem,” https://en.wikipedia.org/wiki/Nyquist%E2%80%93Shannon_sampling_theorem.
9. K. Durga Prasad, V. K. Rai, and S. V. S. Murty, “A comprehensive 3d thermophysical model of the lunar surface,” Earth Space Sci. **9**, e2021EA001968 (2022). E2021EA001968 2021EA001968.