

# Simulating Non Gaussian Surfaces

April 30, 2021

## 1 3D-Fractal Surface Generation

**Author:** Sebastian Korsak

### 1.1 The method

The method is describes on the paper: *Numerical Simulation of 3D Rough Surfaces and Analysis of Interfacial Contact Characteristics*, by Guoqing Yang, Baotong Li, Yang Wang and Jun Hong. Here we have some of the steps as are written in this paper.

#### 1.1.1 Simulations of Gaussian rough surfaces

The simulation of Gaussian rough surfaces is the process of producing 2D phase sequences from white noise sequences and then updating for generating Gaussian rough surfaces. In order to make the target Gaussian rough surfaces satisfy the input ACF,  $\mu, \sigma, S_{kz}$  and  $K_{uz}$ , the discrete method for the input ACFs and the definition of phase sequences as well as the computation of the PSD constants of white noise sequences are presented in detail.

The concrete steps for the simulation of Gaussian rough surfaces are listed as follows:

1. The discrete form of ACF  $R(m, n)$  is extracted from the given ACF  $f(x, y)$ .
  - The input ACF should be defined at first. For example, the following exponential form of ACF  $f(x, y)$  is widely quoted.

$$f(x, y) = \sigma^2 \exp \left[ -2.3 \sqrt{\left( \frac{x'}{\beta_x} \right)^2 + \left( \frac{y'}{\beta_y} \right)^2} \right] \quad (1)$$

where

$$x' = x \cos \phi + y \sin \phi, y' = -x \sin \phi + y \cos \phi \quad (2)$$

where  $\sigma$  is the standard deviation of height sequence;  $\beta_x$  and  $\beta_y$  stand for the auto-correlation lengths in x and y directions, respectively;  $\phi$  stands for the prescriptive orientation of surface texture.

- An equal discrete spacing in  $x$  and  $y$  directions needs to be specified (for example,  $\Delta x = \Delta y = 1 \mu m$ ) and then, discretize the given ACF into sequence  $R(m+1, n+1)$  within the range of  $-m/2 \leq x \leq m/2$  and  $-n/2 \leq y \leq n/2$ .

$$R\left(g + \frac{m}{2} + 1, h + \frac{n}{2} + 1\right) = f(g, h) \quad (3)$$

where  $m$  and  $n$  are numbers of surface points in  $x$  and  $y$  directions;  $g = -m/2, -m/2 + 1, \dots, m/2$ ;  $h = -n/2, -n/2 + 1, \dots, n/2$ .

- One of the repeated rows  $m/2 + 1$  or  $m/2 + 2$ , and the repeated columns  $n/2 + 1$  or  $n/2 + 2$  in the sequence  $R(m+1, n+1)$  is required to be deleted to generate the sequence  $R(m, n)$  of the same size with the target rough surface.
2. The PSD and TF can be obtained from  $R(m, n)$ ,
- FFT method is applied here to get the PSD  $P(I, J)$ .
  - Since the PSD of white noise is a constant  $C$ , assuming  $C = 1$ , the transfer function  $H(m, n)$  can be obtained:

$$H = \sqrt{\frac{P}{C}} = \sqrt{P} \quad (4)$$

3. The phase sequence  $\Phi(m, n)$  can be produced from the white noise sequence  $\eta(m, n)$  which needs to be updated with inverse fast Fourier transform (IFFT) method.

- The white noise generator `randn(m, n)` is used to obtain the white noise sequence  $\eta(m, n)$ .
- The phase sequence  $\Phi(m, n)$  is generated from the sequence  $\eta(m, n)$  with

$$\Phi(k+1, l+1) = 2 \tan^{-1} \left( \frac{-\sum_{r=0}^{m-1} \sum_{s=0}^{n-1} \eta(r+1, s+1) \sin\left(\frac{2\pi kr}{m} + \frac{2\pi ls}{n}\right)}{\sum_{r=0}^{m-1} \sum_{s=0}^{n-1} \eta(r+1, s+1) \cos\left(\frac{2\pi kr}{m} + \frac{2\pi ls}{n}\right)} \right) \quad (5)$$

where all the elements in the phase sequence  $\Phi$  range from 0 to  $2\pi$ , and the computation of  $\Phi(m, n)$  can be conducted with FFT method to improve the efficiency.

- The white noise sequence  $(m, n)$  is updated from  $\Phi(m, n)$  with IFFT method.
  - FFT method is then introduced to the updated the white noise sequence  $\eta(m, n)$ .
4. The Gaussian sequence  $z_0(m, n)$  is generated from the dot-product of  $A$  and  $H$  with IFFT method.
5. Repeat steps 3 4, if the skewness and kurtosis values of  $z_0$  approach 0 and 3, respectively, otherwise, repeat the step 2.
6. The Gaussian sequence  $z_0$  is required to be scaled and translated to obtain the target Gaussian height sequence.

### 1.1.2 Non-Gaussian Translator System

Here we choose the Johnson SU distribution

$$z_2 = \xi + \lambda \sinh \frac{\eta - \gamma}{\delta} \quad (6)$$

where  $\eta$  is the white noise sequence and  $z_2$  is the non-Gaussian sequence;  $\gamma$  and  $\delta$  are shape parameters;  $\lambda$  and  $\xi$  are proportional coefficient and position parameter, respectively.

The main problem we had here it was that we want to simulate this distribution using the first four moments. Therefore, we need a function that is able to correspond the four moments to parameters of our distribution. Unfortunately, there is no library in python that can do that. *However, I want to special thank Max Pierini who translated MATLAB's toolbox that is able to do that and upload it in the repository [https://github.com/maxdevblock/j\\_johnson\\_M](https://github.com/maxdevblock/j_johnson_M).*

Note that the restriction  $K_{uz} - S_{kz}^2 - 1 \geq 0$  must be satisfied.

### 1.1.3 Non\_Gaussian Transformation

The complicated filter process is eliminated and height sequences with any input  $S_{kz}$  and  $K_{uz}$  in the whole skewness-kurtosis plane ( $K_{uz} - S_{kz} - 1 \geq 0$ ) can be obtained by carrying out the following steps.

1. The white noise generator `randn(m,n)` is utilized to obtain the white noise sequence  $\eta(m,n)$  and transform it to a non-Gaussian sequence  $z_2(m,n)$  with Johnson's translator system in terms of the input first four moments; If  $S_B$  or  $S_U$  are unable to converge, use the Pearson's translator system to obtain the non-Gaussian sequence  $z_2(m,n)$ , instead.
2. The skewness and kurtosis of the sequence  $z_2(m,n)$  is calculated and repeat steps 1 3 if the accuracy cannot meet the satisfaction.
3. The sequence  $z_2(m,n)$  needs to be scaled and translated with  $z_3 = \sigma \cdot z_2 / \sigma_1 - \mu_1 + \mu$  ( $\sigma_1$  and  $\mu$  stand for the standard deviation and the mean of the sequence  $z_2$ ) to update the non-Gaussian sequence to satisfy the given first four moments.

## 1.2 Import Libraries

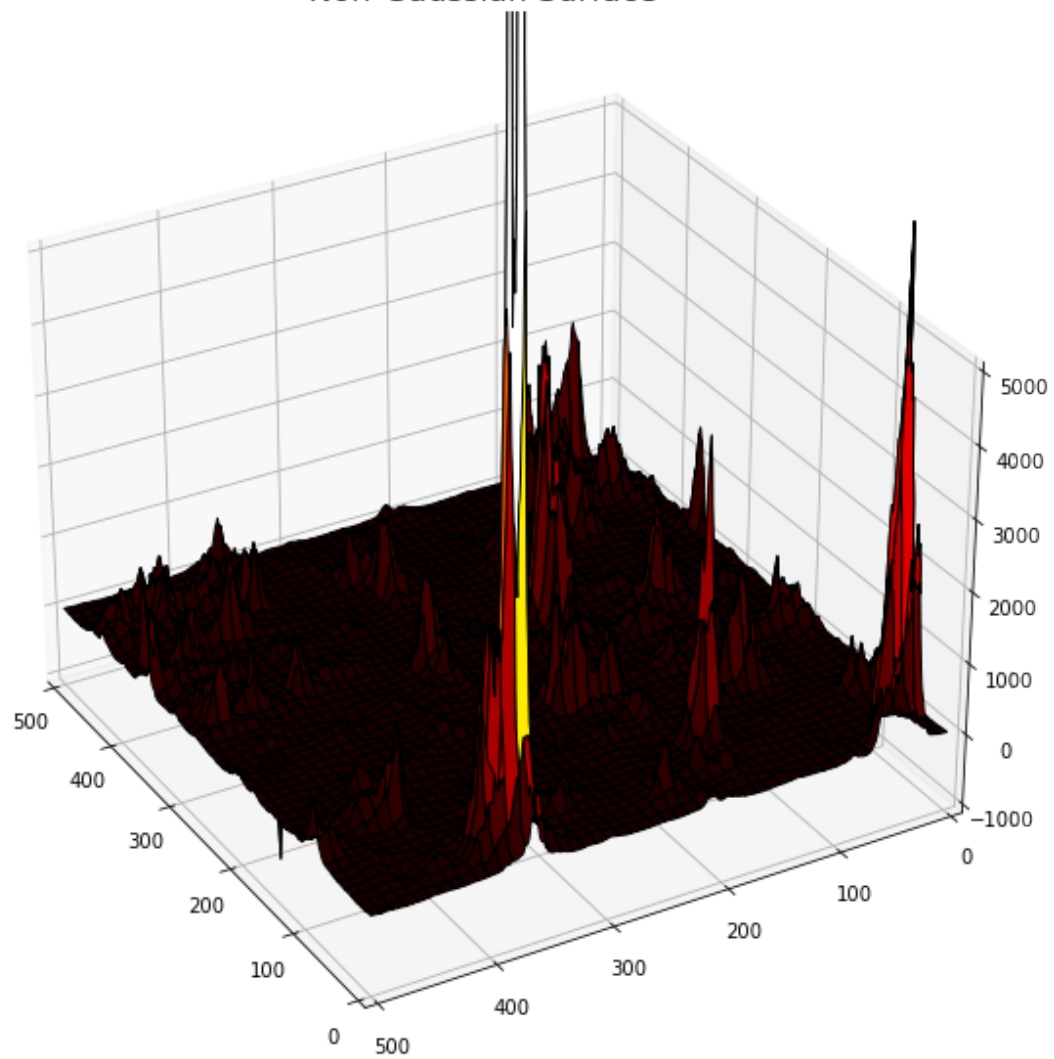
```
[1]: import non_gaussian_surfaces as seb
```

## 1.3 How to use the function

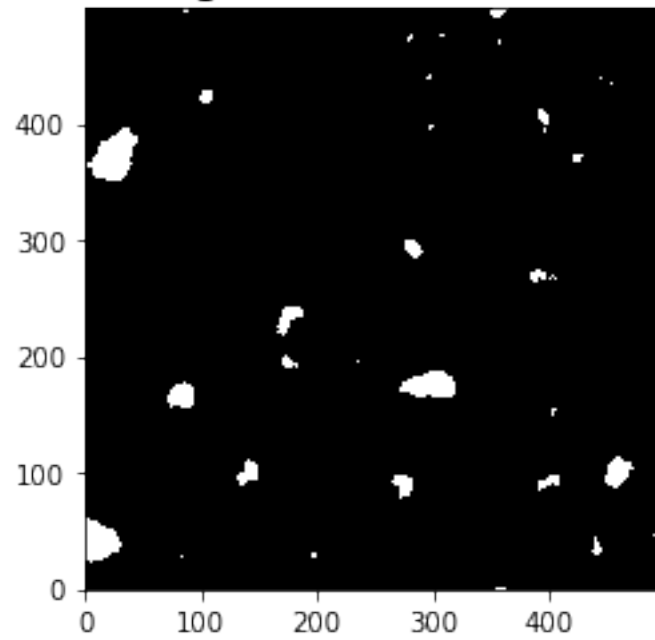
```
[2]: seb.SAimage_fft_2(500,3,1,3,20,20,0.9,800.0,corr=True)

(500, 500)
```

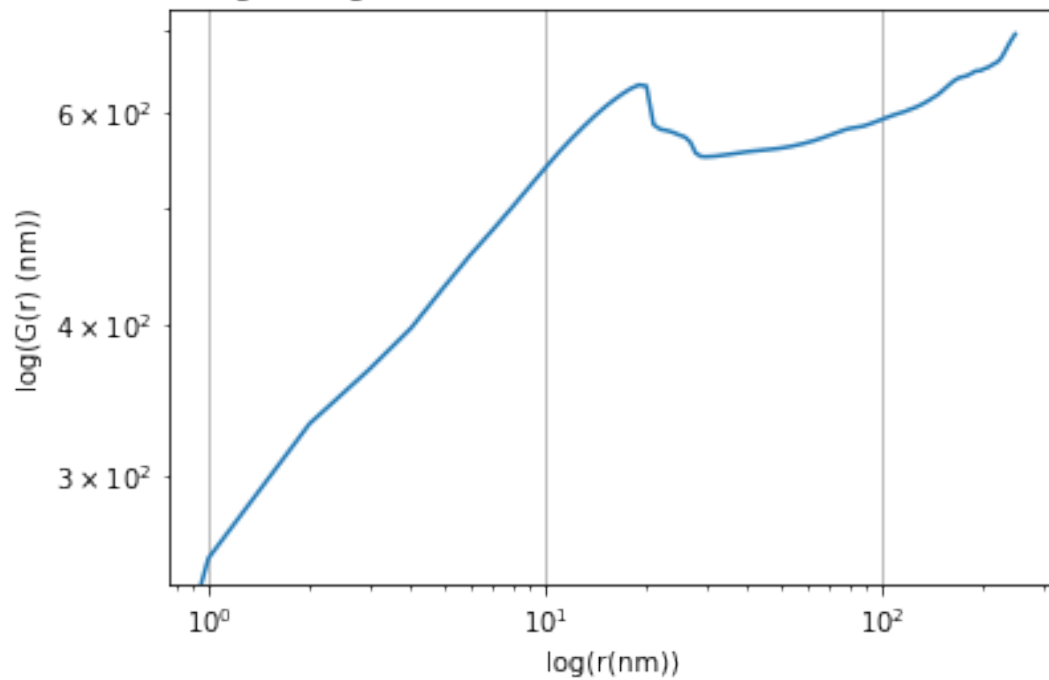
Non-Gaussian Surface



Binarized Image from Non-Gaussian Surface



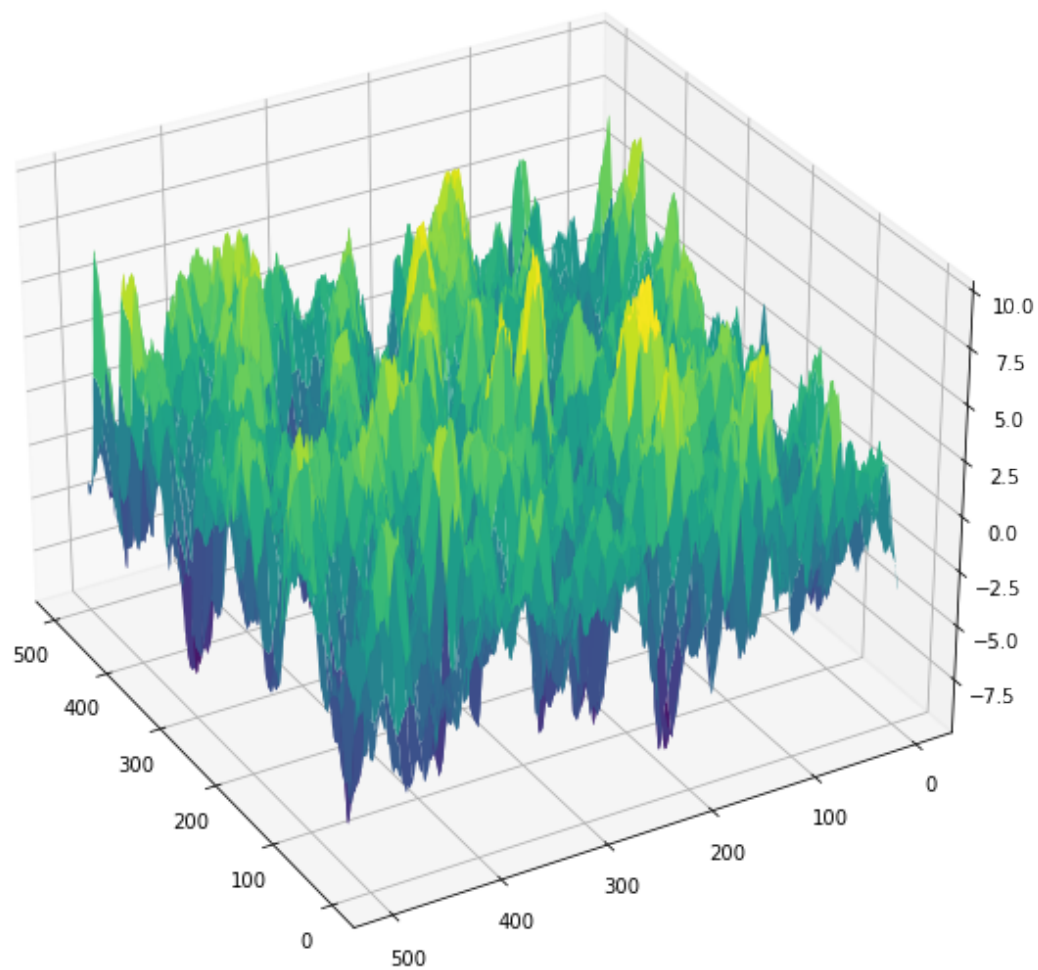
1-D height-height correlation function (non-Gaussian surface)

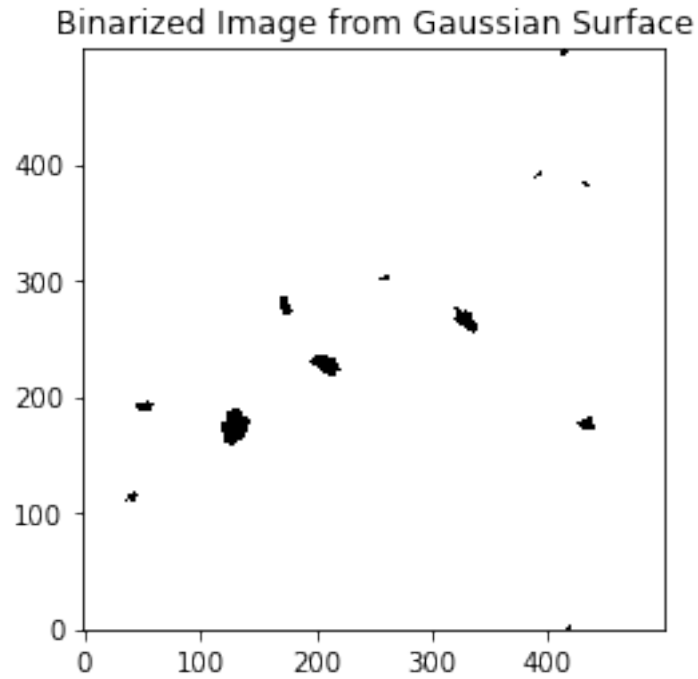


```
[3]: seb.SAimage_fft_2(500,3,1,3,20,20,0.9,th=7.5,non_Gauss=False,corr=False)
```

(500, 500)

Gaussian Surface





## 2 References

- *Numerical Simulation of 3D Rough Surfaces and Analysis of Interfacial Contact Characteristics*, by Guoqing Yang, Baotong Li, Yang Wang and Jun Hong, Tech Science Press, Vol.103, No.4, 2014, pp.251-279, doi:10.3970/cmcs.2014.103.251.