# Numerical generation of fractal surfaces:

a flexible and controllable approach



Webinar Presented By

#### **INVITED SPEAKERS**

PROF. ANDREAS ALMQVIST Luleå University of Technology DR. FRANCESC PEREZ-RÁFOLS

Universitat Politècnica de Catalunya

30 JUNE 2023





## Outline

- Motivation
  - Why do we need generated surfaces
  - Height probability function
  - Power Spectrum
- Algorithm
  - Description
  - A note of caution
- Implementation example



Article

F. Pérez-Ràfols, A. Almqvist, *Generating randomly rough surfaces with given height probability distribution and power spectrum*, Tribology International, Volume 131, pp. 591-604 (2019)

• Code

https://www.mathworks.com/matlabcentral/fileexchange/129469-fractalsurface-generator







### **Motivation – Measured vs. Generated**

#### **Measured surfaces**

- Close to reality
- Uncontrollable features
- High uncertainty
- Expensive to obtain





#### **Generated Surface**

- Idealization
- Controlled if done right
- Moderate uncertainty
- Readily available





### **Motivation – Measured vs. Generated**

#### **Measured surfaces**

- Close to reality
- Uncontrollable features
- High uncertainty
- Expensive to obtain



 $\begin{array}{ccc} 0.5 & 1 & 1.5 & 2 \\ x_1 \text{-direction (mm)} \end{array}$ 



It is hard to do parametric studies with measured surfaces

It is more fruitful to develop the theory on generated surfaces and then test it on measured ones

We want the generated surfaces to be as close as possible to the real ones

#### **Generated Surface**

- Idealization
- Controlled if done right
- Moderate uncertainty
- Readily available





### Average height

**RMS roughness:** 

$$R_q = \sqrt{\frac{1}{N}\sum(z-\bar{z})^2}$$

Mean absolute value:

$$R_a = \frac{1}{N} \sum |z - \bar{z}|$$



Both are quite similar and give a sense of how large roughness is



### **Other height parameters**

The three profiles have the same  $R_q$ , but they are extremely different





#### **Other height parameters**

The three profiles have the same  $R_q$ , but they are extremely different

#### Skewness:

:::

UPC

$$R_{sk} = \frac{1}{R_q^3} \left( \frac{1}{N} \sum (z - \bar{z})^3 \right)$$

 $R_{sk}$  characterises symmetry of the HPD  $R_{sk} > 0$ : more peaks than valleys  $R_{sk} < 0$ : more valleys than peaks





#### **Other height parameters**

This two profiles have the same  $R_q$  and  $R_{sk}$ , but they are still quite different









#### **Other height parameters**

This two profiles have the same  $R_q$  and  $R_{sk}$ , but they are still quite different

Higher order moments:

$$R_{ku} = \frac{1}{R_q^4} \left( \frac{1}{N} \sum (z - \bar{z})^4 \right)$$
$$m_5 = \frac{1}{R_q^5} \left( \frac{1}{N} \sum (z - \bar{z})^5 \right)$$





• • •





## **Motivation – Height Probability Density (HPD)**



The area indicates the probability that a given point lays between two heights

It can also be thought of as the fraction of points within that height range

It contains all information on height; height parameters summarize its information



## Motivation – Height Probability Density (HPD)





UPC



- The two profiles have the same HPD
- Nonetheless, we see that they are **notably different**
- **Controlling height is not enough** to specify a surface
- Lateral information is also needed







The Power Spectral Density (PSD) indicates the amplitude of a wave with a given wavelength.



The Power Spectral Density (PSD) indicates the amplitude of a wave with a given wavelength. The PSD of z is given by  $|\mathcal{F}\{z\}|^2$ , the square of the amplitude Fourier transform.



The Power Spectral Density (PSD) indicates the amplitude of a wave with a given wavelength.

The PSD of z is given by  $|\mathcal{F}\{z\}|^2$ , the square of the amplitude Fourier transform.

For the algorithm it is convenient to work with  $\sqrt{PSD} = |\mathcal{F}\{z\}|$ 



### **Motivation – Self Affine Surfaces**



As we zoom in a surface, we see ever more rough details In most cases, the smaller scales "look the same" as the larger ones







A common way to study these kind of surfaces is to define them as Self Affine

These have a PSD of the form  $PSD(q) \propto q^{-2(1+H)}$ , where *H* is the Hurst exponent





A common way to study these kind of surfaces is to define them as **Self Affine** 

These have a PSD of the form

 $PSD(q) \propto q^{-2(1+H)}$ ,

where H is the Hurst exponent

It is readily understood in an averaged sense





In practice, self affinity will only be considered in a limited region of wave numbers, i.e.,

$$PSD(q) = \begin{cases} 0 & q < q_0 \\ Cq^{-2(H+1)} & q_0 < q < q_1 \\ 0 & q_1 < q \end{cases}$$

where

:::

UPC

$$q_0 = \frac{2\pi}{\lambda_0}$$
 indicates longest wavelength present,  
 $q_1 = \frac{2\pi}{\lambda_1}$  indicates shortest wavelength present





In surfaces we have

$$\vec{q} = [q_x, q_y]$$
$$q = \sqrt{q_x^2 + q_y^2}$$





### **Motivation – Conventional Surface Generation**







### **Motivation – Non-Gaussian Surfaces**

#### Weibull HPD

:::

...

UPC





#### Weibull HPD





#### Example of application with Weibull HPD



Pérez-Ràfols, F., Almqvist, A. On the stiffness of surfaces with non-Gaussian height distribution. *Sci Rep* **11**, 1863 (2021). https://doi.org/10.1038/s41598-021-81259-8







Step 1: Normalize the inputs

<u>Step 2</u>: Correct the power spectrum

Step 3: Correct the height distribution

Step 4: Compute error

Check exit condition





#### **<u>Step 1</u>**: Normalize the inputs

<u>Step 2</u>: Correct the power spectrum

Step 3: Correct the height distribution

Step 4: Compute error

Check exit condition

Both the PSD and HPD encode  $\bar{z}$  and  $R_q$ 

Normalization guarantee that they are equal

$$Z = \frac{z - \bar{z}}{R_{q}}$$



<u>Step 1</u>: Normalize the inputs

Step 2: Correct the power spectrum -

Step 3: Correct the height distribution

Step 4: Compute error

Check exit condition

From the Fourier transform of z, we read:

- Modulus:  $|\mathcal{F}(z)|$ : What we defined as  $\sqrt{PSD}$
- Phase:  $\angle \mathcal{F}(z)$ : Related to HPD

We correct the PS by

$$\mathcal{F}(z)^{n+1} = \mathcal{F}(z)^n \frac{\sqrt{PSD}}{|\mathcal{F}(z)^n|}$$

This, however, distorted the HPD





<u>Step 1</u>: Normalize the inputs

<u>Step 2</u>: Correct the power spectrum

**Step 3**: Correct the height distribution

Step 4: Compute error

Check exit condition

ĕĕĞ

UPC

We want to change the height distribution while minimizing the total change of the surface

- Sort *z* and *h*
- Replace, in height order, the value of z by those in h

 $h_1 h_2 h_3 h_4 h_5$ 





<u>Step 1</u>: Normalize the inputs

<u>Step 2</u>: Correct the power spectrum

Step 3: Correct the height distribution

Step 4: Compute error

d'Enginyeria de Manresa

We want to change the height distribution while minimizing the total change of the surface

- Sort z and h
- Replace, in height order, the value of z by those in h





<u>Step 1</u>: Normalize the inputs

<u>Step 2</u>: Correct the power spectrum

Step 3: Correct the height distribution

Step 4: Compute error

We want to change the height distribution while minimizing the total change of the surface

- Sort z and h
- Replace, in height order, the value of z by those in h





<u>Step 1</u>: Normalize the inputs

<u>Step 2</u>: Correct the power spectrum

Step 3: Correct the height distribution

Step 4: Compute error

We want to change the height distribution while minimizing the total change of the surface

- Sort *z* and *h*
- Replace, in height order, the value of z by those in h





<u>Step 1</u>: Normalize the inputs

<u>Step 2</u>: Correct the power spectrum

Step 3: Correct the height distribution

Step 4: Compute error

We want to change the height distribution while minimizing the total change of the surface

- Sort *z* and *h*
- Replace, in height order, the value of z by those in h





<u>Step 1</u>: Normalize the inputs

<u>Step 2</u>: Correct the power spectrum

Step 3: Correct the height distribution

#### Step 4: Compute error

Check exit condition

$$err_h = \sum_{hins} (n_p - n_h) w_b$$

Error in height distribution

 $z_h$  has the perfect HPD, we compare it with  $z_p$ 

- Construct a histogram
- Compare the number of points in each bin





<u>Step 1</u>: Normalize the inputs

<u>Step 2</u>: Correct the power spectrum

<u>Step 3</u>: Correct the height

Step 4: Compute error

Check exit condition

$$\operatorname{err}_{p} = \sqrt{\frac{1}{N_{S}} \sum_{q \mid \mathcal{F}\{z_{p}\}(q) \neq 0} \left(\frac{|\mathcal{F}\{z_{h}\}(q)| - |\mathcal{F}\{z_{p}\}(q)|}{|\mathcal{F}\{z_{p}\}(q)|}\right)^{2}}$$

Error in the power spectrum

We use the rms value of the relative error

We must be careful with points that have a prescribed value of zero







<u>Step 1</u>: Normalize the inputs

<u>Step 2</u>: Correct the power spectrum

<u>Step 3</u>: Correct the height

Step 4: Compute error

Check exit condition

$$\operatorname{err}_{p_{0}} = \frac{\sqrt{\frac{1}{N_{S_{0}}} \sum_{q \mid \mathcal{F}\{z_{p}\}(q)=0} |\mathcal{F}\{z_{h}\}|^{2}}}{\frac{1}{N_{S_{0}}} \sum_{q \mid \mathcal{F}\{z_{p}\}(q)=0} |\mathcal{F}\{z_{p}\}|}$$

the inputs <u>E</u>

Error in the power spectrum

We use the rms value of the relative error

We must be careful with points that have a prescribed value of zero





<u>Step 1</u>: Normalize the inputs

<u>Step 2</u>: Correct the power spectrum

<u>Step 3</u>: Correct the height

Step 4: Compute error

**Check exit condition** 

Three exit conditions are used

- <u>Convergence reached</u> One of two conditions are fulfilled:
  - $\operatorname{err}_h < \operatorname{tol}$
  - $\operatorname{err}_p < \operatorname{tol} \operatorname{and} \operatorname{err}_{p_0} < \operatorname{tol}$
- <u>Slow speed</u>

Convergence is not reached but the surface does not change significantly

• <u>Maximum number of iterations</u> The algorithm has failed



### **Results – Weibull HPD**



#### **Results – Weibull HPD**



d'Enginyeria de Manresa

40/52

UNIVERSITY OF TECHNOLOGY

### **Results – Weibull HPD**

#### Minimum error attainable







### **Results – Artefacts**







#### Giving sufficient flexibility for HPD and PSD to match







EPSEM – Escola Tècnica Superior d'Enginyeria de Manresa



#### Giving sufficient flexibility for HPD and PSD to match









EPSEM – Escola Tècnica Superior d'Enginyeria de Manresa

#### Giving sufficient flexibility for HPD and PSD to match



d'Enginyeria de Manresa





#### Giving sufficient flexibility for HPD and PSD to match









EPSEM – Escola Tècnica Superior d'Enginyeria de Manresa

#### Giving sufficient flexibility for HPD and PSD to match







EPSEM – Escola Tècnica Superior d'Enginyeria de Manresa



#### Giving sufficient flexibility for HPD and PSD to match







EPSEM – Escola Tècnica Superior d'Enginyeria de Manresa



#### Giving sufficient flexibility for HPD and PSD to match



The code is too powerful!

If you ask it to give a perfect HPD with  $\lambda_0 = L$ , it will often provide, but artefacts might appear





UPC

#### Giving sufficient flexibility for HPD and PSD to match



The code is too powerful!

If you ask it to give a perfect HPD with  $\lambda_0 = L$ , it will often provide, but artefacts might appear







- Generated surfaces can be a powerful tool to study the effect of roughness
- These are even more useful if they can be flexibly generated, controlling both power spectrum and height probability distribution
- We have presented an algorithm that achieves this flexibility and is easy to operate
  - Caveat: one must be careful with cut-off wavelengths





1. Go to: https://bit.ly/surfgenFPRAA









- 1. Go to: https://bit.ly/surfgenFPRAA
- 2. Use the Download button





EPSEM – Escola Tècnica Superior d'Enginyeria de Manresa

...

- 1. Go to: <u>https://bit.ly/surfgenFPRAA</u>
- 2. Use the Download button
  - and the Zip option







 Go to: <u>https://bit.ly/surfgenFPRAA</u>
 Use the Download - button

> Toolbox Zip

and the Zip option

3. Unzip (to preferred directory)







1. Go to: https://bit.ly/surfgenFPRAA Use the Download -2. button Toolbox Zip

and the Zip option

- Unzip (to preferred directory) 3.
- 4. Start MATLAB

:::

UPC

5. Run mySurfGenApp.mlappinstall from the directory chosen

• • •	< > mySurfGenApp						
avorites	Name	~ 0	Date Added				
Documents	> mySurfGenApp 1	0	8 Jun 2023 at 13:01				
Downloads	fftbandfilter.m	0	8 Jun 2023 at 13:01				
	🔚 fractal surf generator.m	0	8 Jun 2023 at 13:01				
Downloads	fractal surf hu tonder anisotropic.m	0	8 Jun 2023 at 13:01				
AirDrop	license.txt	0	8 Jun 2023 at 13:01				
Recents	mainSGFunction.m	0	8 Jun 2023 at 13:01				
	mydiffx.m	0	8 Jun 2023 at 13:01				
MATLAB_Drive	mydiffy.m	0	8 Jun 2023 at 13:01				
🔁 Skidor	myRdist.m	0	8 Jun 2023 at 13:01				
- C0004M	mySurfGenApp_1.prj	$\bigcirc$	8 Jun 2023 at 13:01				
	1 mySurfGenApp.mlapp	$\bigcirc$	8 Jun 2023 at 13:01				
M00XXM	🚰 mySurfGenApp.mlappinstall	$\bigcirc$	8 Jun 2023 at 13:01				
M0012T	mySurfGenApp.prj	$\bigcirc$	8 Jun 2023 at 13:01				
M7022T	🔚 mySurfParam.m	$\bigcirc$	8 Jun 2023 at 13:01				
M70331	myWebAppfprintf_1col.m	$\bigcirc$	8 Jun 2023 at 13:01				
M7018T	Realization_2023-6-8_1254315202.txt	0	8 Jun 2023 at 13:01				
Yinhu Xi (Postdoc)	Realization_2023-6-8_1255104028.txt	$\bigcirc$	8 Jun 2023 at 13:01				
	👚 results.mat	$\bigcirc$	8 Jun 2023 at 13:01				
Joakim Sandberg (PhD)	魡 splash.fig	$\bigcirc$	8 Jun 2023 at 13:01				
🗎 Jamal Choudry (PhD)	ल splash.png	0	8 Jun 2023 at 13:01				
🗎 Ali Usman (Postdoc)			ROUGHNESS GENER				
🗂 De Huang (PhD)	1 of 20 selected, 1,52 TB available on iCloud, uploadin						



56/52

Avalutat Frances Dar

- 1. Go to: https://bit.ly/surfgenFPRAA
- 2. Use the Download button

and the Zip option

- 3. Unzip (to preferred directory)
- 4. Start MATLAB

Toolbox

Zip

- 5. Run mySurfGenApp.mlappinstall from the directory chosen
- 6. Locate mySurfGe...on the APPS tab and click it!





1. Go to:

#### https://bit.ly/surfgenFPRAA

2. Use the Download - button

Toolbox Zip

and the Zip option

- 3. Unzip (to preferred directory)
- 4. Start MATLAB
- 5. Run mySurfGenApp.mlappinstall from the directory chosen
- 6. Locate mySurfGe...on the APPS tab







- 1. Go to: https://bit.ly/surfgenFPRAA
- 2. Use the Download button

and the Zip option

- 3. Unzip (to preferred directory)
- 4. Start MATLAB

Toolbox

Zip

- 5. Run mySurfGenApp.mlappinstall from the directory chosen
- 6. Locate mySurfGe...on the APPS tab
- 7. Click it and then

:::

...

UPC









- Now play with the App as you prefer!
- Notice that you also can use the functions: mainSGFunction.m mySurfParam.m
  - mySurfGenApp.mlappinstall
  - standalone

• •	•			Editor -	/Users/almqvist/	Documer	nts/_SIMULA	TION_/MATL	AB/ROUG	HNESS_GENERATION	l/mySurf
E	DITOR		PUBLISH	VIEW							
New	Open	Save FILE	🔄 Compare 🔻	Со То Со То	<ul> <li></li></ul>	Refactor	% ‰ % 	🔂 Profiler 🚽 Analyze ANALYZE	Run Section	E Section Break Run and Advance Run to End SECTION	Run
1 function [X,Y,Z,HRc,HPDc,qs,PSDs,minq,maxq] =											
2	2 mainSGFunction(surfHPD,surfPSD,varargin)										
3											
4	%	5 Inp	outs								
5	if nargin == 9 % Hurst exponent or exp coefficient provided										
6	HPD = surfHPD; % Gaussian Bi-Gaussian Weibull										
/	<pre>PS = SUFTPSU; % 'EXPONENTIAL' if stromp(PS 'Self-affine')</pre>										
9		Hr = varargin{1}: & Hurst exponent for self affine DS									
10		elseif stromp(PS, 'Exponential')									
11		bEPS = varargin{1}: % Beta coefficient for exponential PS									
12		end									
13		af = varargin{2}; % Anisotropy coefficient									
14		<pre>if strcmp(HPD,'Gaussian')</pre>									
15		bWB = 1;									
16		cBG = 1;									
17		<pre>elseif strcmp(HPD,'Weibull')</pre>									
18		<pre>bWB = varargin{3}; % b coeficient in Weibull</pre>									
19			% A CC	erric	lent in Wei		Tixed to	D) (1,1/hup)			
20		a = sqrt(1^2/(gamma(1+2/DWB)-gamma(1+1/DWB)^2));									
21		$cBG = varargin{3}{3}: & Coefficient for hi-Gaussian surface$									
23		er	nd -	tara	.g(5), 0		Lerence in	0. DI 000		5411466	
24		Ls	s = varar	ain{4	}; % Sho	rt wav	elength (	cut-off			
25		LI	l = varar	gin{5	}; % Lon	gest w	avelengt	h present			
26						-	5	-			
27		М	= varar	gin{6	}; % Num	erical	resolut	ion			
28		Ν	= varar	gin{7	}; % Num	erical	resolut	ion			
29	e	nd									
30											



:::

UPC

