World Academy of Science, Engineering and Technology
International Journal of Mechanical and Mechatronics Engineering
Vol:17, No:5, 2023

The Use of Fractional Brownian Motion in the Generation of Bed Topography for Bodies of Water Coupled with the Lattice Boltzmann Method

Elysia Barker, Jian Guo Zhou, Ling Qian, Steve Decent

Abstract—A method of modelling topography used in the simulation of riverbeds is proposed in this paper which removes the need for datapoints and measurements of a physical terrain. While complex scans of the contours of a surface can be achieved with other methods, this requires specialised tools which the proposed method overcomes by using fractional Brownian motion (FBM) as a basis to estimate the real surface within a 15% margin of error while attempting to optimise algorithmic efficiency. This removes the need for complex, expensive equipment and reduces resources spent modelling bed topography. This method also accounts for the change in topography over time due to erosion, sediment transport, and other external factors which could affect the topography of the ground by updating its parameters and generating a new bed. The lattice Boltzmann method (LBM) is used to simulate both stationary and steady flow cases in a side-by-side comparison over the generated bed topography using the proposed method, and a test case taken from an external source. The method, if successful, will be incorporated into the current LBM program used in the testing phase, which will allow an automatic generation of topography for the given situation in future research, removing the need for bed data to be specified.

Keywords—Bed topography, FBM, LBM, shallow water, simulations.

I. Introduction

SHALLOW water flows, such as rivers, streams and even flash floods, are used to predict and simulate the movement and behaviour of a fluid over time which informs the best course of action for flash flood prevention measures, property development, resource management and countless other important factors. There are many areas of limitation, however, and this paper focuses on a limitation which arises when considering the topography of a surface such as a riverbed.

Typically, a riverbed will be modelled by taking a series of points at set intervals, or by using expensive and specialised equipment to gather detailed data. However, this equipment is not available to everyone and has its own limitations such as areas where satellite imaging cannot penetrate; and using a series of points is not an accurate representation of real life topography. The task of overcoming this issue with the constraints of making the new method widely available for use without expensive or specialised equipment seemed, initially, ambitious.

An effective way to model a realistic structure would be to take inspiration from the way in which large stretches of topography are procedurally generated in video games. Instead of using expensive equipment or estimated data, a fractal-like structure is created using a generalization of Brownian Motion known as FBM to form a non-linear, realistic looking topography. In video games, however, there is no need for this to represent a real section of topography, so this is where the use in these cases ends. This is taken a step further in this paper by including a series of constraints and rules to control the FBM and contort the realistic looking topography to become a suitable, accurate representation of a real topography without the need for specialised equipment.

A major advantage of this method over set interval estimation is that little pre-existing data about the area being modelled are required. This is particularly useful for areas in which it is difficult to gather accurate data, due either to dangerous or inaccessible location or because of external factors such as difficulties in satellite imagery penetrating the area, layers of thick ice and sediment build up as well as dynamic changes due to erosion, sediment and particulate build up or geographical changes arising from recent earthquakes or flash floods.

II. Related Work

This paper will describe how FBM has been used to generate bed topography. There have been other studies which have similarly used FBM for natural phenomena. For example, Mandelbrot sets out to show that the natural world can be ”efficiently and beautifully modelled by mathematical objects” [1], which focuses on FBM as a primary source of doing so. Nature and the natural world is chaotic. Its ingrained characteristics are unpredictable and uncontrollable in ways that are far more complex than this paper can explain. From the way water moves and behaves to the geometry of mountains or the patterns we find in trees, everything contains elements of chaos and nonlinearity. So this led to the question; is there more in common between Chaos and Natural Phenomena than first thought? Many areas of research have implemented the use of chaos theory and fractal geometry already, including research into the behaviour of water. This idea is fascinating and prompts thinking not just about the way water behaves but how the surface the water interacts with affects the behaviour. This began the development of a method which uses fractals to more accurately model real world riverbeds to see the impact this would have on the behaviour of water.

Elysia Barker is with The Centre for Mathematical Modelling and Flow Analysis, Manchester Metropolitan University, Manchester, United Kingdom, M15 6BH (corresponding author, phone: +44 7972042596, e-mail: Elysia.G.Barker@stu.mmu.ac.uk.)

J. G. Zhou and L. Qian are with Manchester Metropolitan University.

Steve Decent is with Glasgow Caledonian University.

In other work, the exact, properly adjusted fractal dimension is shown to not inhibit results in any noticeable manner [2], which is a property the method developed in this paper has chosen to build upon.

The inception of FBM was by Andrei Kolmogorov [3], who, at the time, was studying spiral curves in Hilbert space. Recognising its usefulness and relevance to the study of hydrology the use of FBM to model terrain was first proposed by Benoit Mandelbrot [4] following the initial proposal, in which he and Van Ness first coined the term ”Fractional Brownian motion.” [5] It was in this paper that Mandelbrot and Van Ness outlined the many important properties which makes the study and use of fractals so unique and interesting.

Observing FBM fractals many scholars have noted the striking resemblance to a mountainous horizon [6], and considered how fractals could be used in the generation and simulation of landscapes [7]. Further study has led to in depth discussion of the fractal dimensions of the range of alternative landscapes which can be generated from different fractal types.

When considering the use of fractals for generating terrain [8], we can see how modern video games already make use of fractals to render realistic looking landscapes which attempt to replicate certain natural phenomena and erosion [9], albeit with artistic and gameplay concerns overriding realism - such as in Skyrim, Minecraft, and other well known video games.

Fig. 1 FBM Generated Terrain Using Indigo Renderer [10]

Work by Milne used computerized fractal designs to simulate ecological studies of foraging animals which he predicted would enhance both the aesthetics of the studies and the ecological relevance [11]. This method has similar values to that of Milne’s work as the ultimate aim is to enhance work in the field and allow for areas previously restricted to become available.

III. Generation of Bed Topography Using Fractional Brownian Motion

A. Fractional Brownian Motion

The Fractional Brownian Motion (FBM) model is a popular fractal regular stochastic process [12].

Brownian Motion can be described as a form of diffusion: particles in a basin diffuse from an area of high concentration to low concentration independent of one another. FBM is a generalization of Brownian Motion which, unlike classical Brownian Motion, the increments of FBM need not be independent. This means that the previous data influence the current data (e.g. if there is an increasing trend, the next datapoint will likely also be increasing from the previous).

The aim of this method is to generate a fractal like topography structure. This has already been utilised in landscape design, games and cinema special effects [13] which creates realistic, life-like surfaces. Taking inspiration from this could optimise the way we model bed topography.

Fractional Brownian Motion (FBM) has three main characteristic features:

1) It is a continuous Gaussian process,
2) It is self-similar,
3) It has stationary increments

A process X is called self-similar if there exists a positive number H such that the finite-dimensional distributions of $\{T^{-H}X(Tt), t \geq 0\}$ do not depend on T. We denote by B_H the FBM with index of self similarity H. The stationarity of the increments implies that $E[\ |B_H(t) - B_H(s)|^2]\ = |t-s|^{2H}$, and this relation determines the covariance function [14].

$$E[\ B_H(s)B_H(t)]\ = \frac{1}{2}(s^{2H} + t^{2H} - [\ s-t]\ ^{2H}), \qquad (1)$$

As the variance of the sum of N variables with variance 1 cannot exceed N^2, the parameter H (known as the Hurst parameter) is smaller than 1.

To introduce the sequence of increments $G_H(j) = B_H(j) - B_H(j-1), j = 1, 2, \ldots,$ that constitute what is sometimes called fractional Gaussian noise, we note that they are strongly correlated $(H \neq \frac{1}{2})$. More precisely,

$$\begin{aligned} E[\ G_H(j)G_H(j+k)]\ &= \frac{1}{2}((k+1)^{2H} - 2k^{2H} + (k-1)^{2H}) \\ &\underset{k\to\infty}{\sim} H(2H-1)k^{2H-2} \end{aligned} \qquad (2)$$

Observations of the process for $H < \frac{1}{2}$ show the increments being negatively correlated which corresponds to chaotic behaviour, while $H > \frac{1}{2}$ is positively correlated between the increments which corresponds to a more disciplined behaviour [14].

B. Generation of Topography

1) FBM in MATLAB: MATLAB has a built in FBM algorithm which was first implemented in [15] and was then adapted by [16] to overcome the abundance of high-frequency components.

The algorithm builds a biorthoganal wavelet depending on a given orthogonal one and is adapted to the Hurst parameter, starting with the expression of the FBM process as a fractional integral of the white noise process. The generated sample path is then obtained by the reconstruction using the new wavelet starting from a wavelet decomposition at a given level. Details’ coefficients are independent random Gaussian realizations and approximation coefficients arising from a fractional ARIMA [17] process [15]. The algorithm was then improved upon by downsampling the obtained sample by a factor of 10. Two internal parameters ($\delta = 10$ the downsampling factor, and a threshold $prec = 1e-4$) to evaluate series by truncated sums, can be modified for extreme values of the Hurst parameter [18].

2) Evaluating the Hurst Parameters: Hurst parameters describe the raggedness of the resultant motion. The smaller the Hurst parameter, the more ragged the motion will become. An appropriate Hurst parameter must be chosen for the method to generate realistic bed topography.

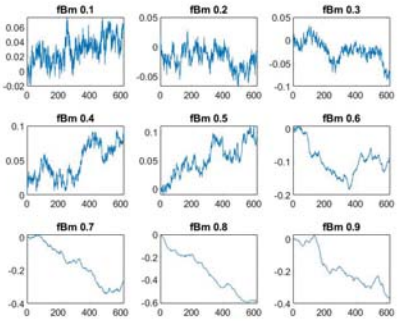


Fig. 2 Plots of Hurst parameters from 0.1 to 0.9

The real bed topography chosen as a test case to test the FBM generated topography against is topography from [19] and shown in Fig. 3.

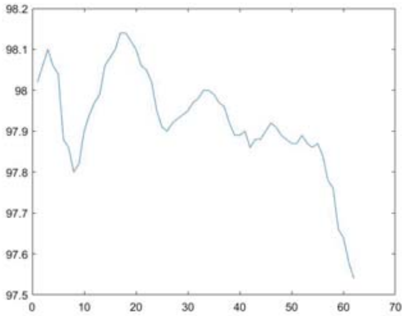


Fig. 3 Plot of Real Bed Topography from Test Case

Upon visual inspection, it appears that $H = 0.9$ bears the closest resemblance to the Real bed topography, however, the results comparing the spacing between the increments show that $H = 0.6$ has the most similar increment spacing to the Real bed topography.

From the above data collected from Fig 2, a program was created to find the average distance between the increments on 100 different versions of each of the Hurst parameter graphs then compared against the average spacing between the real topography to find the Hurst parameter which is most appropriate.

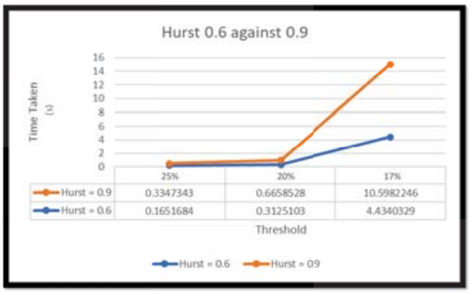
TABLE I
INCREMENTAL DIFFERENCES OF TEST CASE AGAINST HURST PARAMETERS

x	**Real**	**H=0.1**	**H=0.2**	**H=0.3**	**H=0.4**	**H=0.5**	**H=0.6**	**H=0.7**	**H=0.8**	**H=0.9**
1	0.02820	0.09558	0.07227	0.05359	0.03792	0.03291	0.02451	0.02316	0.02064	0.02245
2	0.02820	0.09115	0.07142	0.05198	0.04133	0.03135	0.02584	0.02169	0.01956	0.01818
3	0.02820	0.09051	0.07240	0.05070	0.04213	0.03137	0.02573	0.02223	0.01825	0.01877
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
97	0.02820	0.09989	0.07215	0.05451	0.03905	0.03226	0.02561	0.02229	0.02192	0.01780
98	0.02820	0.09917	0.06992	0.05429	0.03923	0.03183	0.02551	0.02165	0.01975	0.01858
99	0.02820	0.09425	0.06665	0.05218	0.03940	0.03110	0.02532	0.02182	0.01938	0.02114
100	0.02820	0.09415	0.06853	0.05385	0.04008	0.03021	0.02651	0.02216	0.01998	0.01954
Avg	0.02820	0.09511	0.06947	0.05230	0.03995	0.03155	0.02563	0.02173	0.01977	0.01953

To further support the decision to use $H = 0.6$ over $H = 0.9$, a program was created to determine the percentage difference between the Real bed and the FBM bed and the run time was noted down. For the purposes of testing the percentage difference against time taken of the different Hurst parameters, the fully developed method was not used as this was to decide the most appropriate Hurst parameter in the development.

$$D_i = 100 \left| \frac{|F_i - R_i|}{(F_i + R_i)/2} \right|, \qquad i = 1, 2, \ldots, n \tag{3}$$

where F_i is the FBM generated bed; R_i is the Real bed; D_i is the percentage difference between the corresponding data points; and n being the number of data points. In this case, $n = 62$ as the Real bed topography has only 62 data points to compare against.



	25%	20%	17%
Hurst = 0.9	0.3347343	0.6658528	10.5982246
Hurst = 0.6	0.1651684	0.3125103	4.4340329

Fig. 4 Percentage Similarity Against Time Taken To Run

Table I shows the results of comparing both the time taken and the average differences between using $H = 0.6$ and $H = 0.9$.

The method caps the percentage at a certain point which forces the program to restart if a bed is generated at a difference greater than the cap. This early stage algorithm used to decide the most appropriate Hurst parameter employs a Brute Force technique to judge which option will optimise time taken for improvements in the more developed method. It can be observed that as the cap is decreased, the program takes longer to run. However, $H = 0.6$ remains faster and more efficient throughout the tests.

Each of the Hurst parameters was tested 10 times at different caps and an average was found. $H = 0.6$ is seen to be twice as fast as $H = 0.9$ until the final cap, in which there is only 11 seconds difference.

C. FBM Bed Topography Program

The FBM generated bed topography program was written in MATLAB using the built in FBM command which this method utilises and builds upon. There are two different routes the method can take:

1) Route A: Bed topography of the area currently exists, $(R_{exists} = 1)$
2) Route B: Bed topography of the area does not exist, $(R_{exists} = 0)$

A parameter R_{exists} is used and set to either 0 or 1 depending on whether data for the area in question exist. If data exist, the data are manually uploaded to a data file. If the data file contains information, the parameter $R_{exists} = 1$ which allows the program to follow Route A. Conversely, if the data file is empty, the parameter is set to 0 and therefore follows Route B.

1) Route A: Data from the real bed topography are used and reduced to sea level which overcomes issues in boundary conditions when using the LBM: $R^{sl} = datafile - datafile_{min}$, where R^{sl} is the real data reduced to sea level; $datafile$ is the real data read from the data file; and $datafile_{min}$ is the minimum value read from the data file.

Further parameters are defined from the data file such as: $minmax = R^{sl}_{max} - R^{sl}_{min}$, $a_i = mean(\frac{|R^{sl}_i - R^{sl}_{i+1}|}{|minmax|})$, $r_{trend} = \nabla(R^{sl})$, $nr = number\,of\,datapoints$, where $minmax$ is the maximum difference between the peak and trough of the existing data at sea level; a_i is the mean difference between the individual datapoints (for ease of reading, this will be referred to as a_i despite being a fixed parameter); r_{trend} is the inclination of the bed (the gradient of the line of best fit); and nr is the number of datapoints in the data file. Additionally, the Hurst parameter, H is calculated using the program described for (3) and Table I.

2) Route B: Route B offers the user of the method the option to manually input data for the desired topography where no data exist ($R_{exists} = 0$). The only required knowledge is the inclination of the desired bed, and an approximation of the roughness of the terrain. The following data are an example used when attempting to replicate the test case without using its raw data aside from the previously mentioned known information (inclination and roughness of the plane): $minmax = 0.6$, $a_i = 0.25$, $nr = 62$, $r_{trend} = -1$, $H = 0.55$,

To summarise, the $minmax$ value suggests a gentle slope, and the r_{trend} value is negative indicating declination. Additionally the a_i value shows only a slight deviation between points which indicates that the surface is not rough, and the H is the optimal Hurst parameter - as found from the previous section in which the Hurst parameter for the test case was analysed. This will be used as a standard for this method.

D. The Algorithm

There are four stages to the algorithm of the method which alters and manipulates the generated dataset, conforming it to an appropriate and accurate representation of the desired topography. The stages are described below:

1) Stage 1: This is the shortest stage in the algorithm which generates an initial FBM base using the defined Hurst parameters specified in either Route A or Route B. This initial bed contains ten times the amount of datapoints as the parameter nr specifies. Each point is then reduced by a factor of ten to resemble the minute changes in topography since MATLAB produces a large scale FBM which requires reduction to a more appropriate scale.

The dataset generated will be referred to using the parameter fbm.

2) Stage 2: This stage orients the generated dataset in the correct inclination specified by r_{trend}.

If the orientation of the generated data matches with that of r_{trend}, the stage is complete. However, if the orientation does not match, the dataset is reversed. This results in an inclining dataset becoming a declining dataset and vice versa.

3) Stage 3: To ensure that the roughness of the plane does not exceed that of the approximate or real roughness, the algorithm compares the incremental values of the generated dataset to the a_i parameter.

$$\forall i \in nr \cdot 10, \frac{|fbm_i - fbm_{i+1}|}{|fbm_{max} - fbm_{min}|} \leq a_i \qquad (4)$$

If an increment exceeds the threshold value, a_i, the fbm_{i+1} datapoint is increased or reduced accordingly so it falls within the acceptable range. This process is repeated for each individual datapoint.

4) Stage 4: The final stage, similarly to Stage 3, examines the generated data according to the threshold value, $minmax_{high} = minmax + 0.25\%$, which provides a basis for steepness of the gradient of the line of best fit.

The datapoints are tested to ensure that there is never a gap exceeding the $minmax_{high}$ value to avoid a steeper gradient than required. This may be a defined parameter known only due to the fact that the elevation (metres above sea level (MSL)) is known/estimated at both the beginning and end of the desired section of topography (Route B), or calculated from real data (Route A).

If there is an instance which exceeds the $minmax_{high}$ threshold, the line of best fit is altered and the individual datapoints are adjusted accordingly, remaining the same distance from the line, thus reducing the steepness. Likewise, a minimum threshold is automatically generated to ensure there is no zero gradient which is adjusted in the same way. This parameter is known as $minmax_{low} = minmax - 0.25\%$.

The $minmax \pm 0.25\%$ allows for a slight margin of error which has proven to be invaluable in optimising runtime of the program.

E. Testing Method

To test both the Real and the FBM generated bed topographies, a program running the LBM was used [20], and adapted for more complex boundary conditions with an updated elastic collision scheme [21], and simulations were taken after 1, 10, 100 and 500 time steps with both a stationary flow case and a steady flow case using initial velocity and discharge, respectively, as $u_0 = 0.2$ and $Q_0 = 2.2$.

IV. Results and Discussion

The test case was used as a basis for comparison with the generated bed using the FBM method developed in this paper. For the purposes of testing the extremes of the method, no bed data of the test case were entered into the $datafile$. Instead, the parameters mentioned in Route B were used, which forced the method to assume no real data exist, yet the test case was attempted to be replicated by using parameters loosely fitting that of the test case.

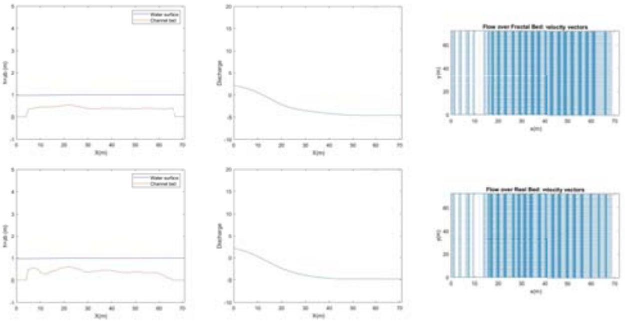


Fig. 7 FBM (a) Against Test Case (b) For a Steady Flow Case after 500 Time Steps

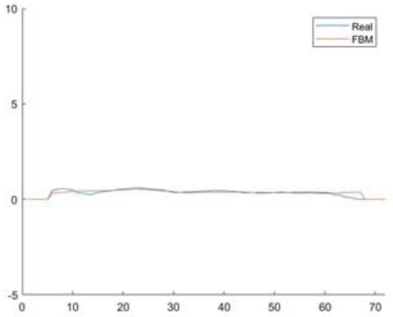


Fig. 5 Comparison of the Test Case (Blue) against the FBM Bed (Red)

It can be seen that, while there is slight deviation around the beginning and end of the generated bed, the method has replicated the test case to a satisfactory degree, with the difference between the generated bed and the test case being only 14.8%.

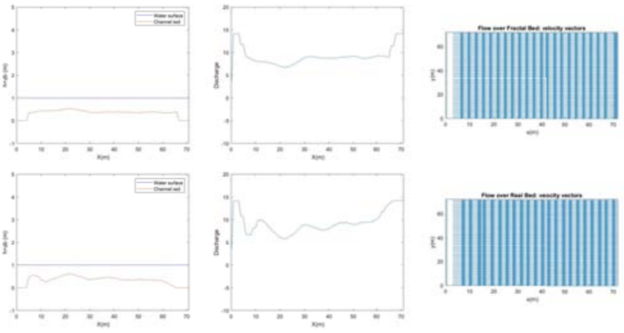


Fig. 6 FBM (a) Against Test Case (b) For a Steady Flow Case after 1 Time Step

Fig. 6 shows side by side comparisons of the FBM generated bed, and the test case at its initial state after only one time step. A graph of the depth, discharge, and velocity vectors, respectively, for each of the time steps has been plotted. The FBM generated bed graphs are shown along the top row whilst the Real bed is shown along the bottom.

Fig. 7 shows side by side comparisons of the FBM generated bed, and the test case at its initial state after five hundred time steps.

TABLE II
PERCENTAGE DIFFERENCE OF WATER DEPTH, FLUID DISCHARGE, AND VELOCITY OF THE TEST CASE AGAINST THE FBM GENERATED BED

Time Steps	Difference Discharge	Difference Depth	Difference Velocity
1	6.85811%	5.95213%	10.26145%
10	5.90483%	5.41447%	22.12619%
100	4.85524%	8.06631%	24.46693%
500	3.39998%	9.63622%	26.57885%

It can be seen that the difference between the discharge is 3.4%, which is promising for the method. Ideally, the differences should be below 5%, however below 10% is also acceptable. The differences in depth are below 10% which is, again, promising for the method, however, this can be improved upon. The greatest difference is with the velocity vectors, which show a larger difference of 26.6% which needs to be improved upon.

Overall, the method is showing promising results, which is ideal for the continuation of research into this area. Although there is a large difference between the velocity vectors, this is to be expected as small changes in the bed topography can have a profound impact on the behaviour of the fluid. Further investigation will be done surrounding this by exploring more complex flow cases and using different test cases. Satellite imaging could be used to gain real time data which may be able to justify the differences in velocity and even aid in deciding whether the FBM generated bed is more true to life than the test case.

The method can be improved upon further by reducing the percentage difference between the FBM and Real beds in the beginning. The method can be optimised further by exploring more Hurst parameters (e.g. $H = 0.65$) as this may allow for faster runtimes and more accurate and closely related topographies.

APPENDIX

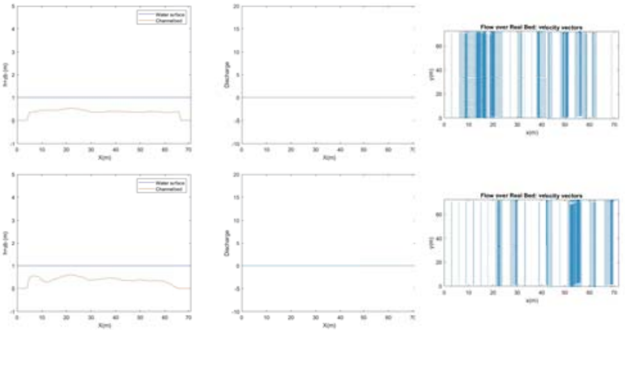


Fig. 8 FBM (a) Against Test Case (b) For a Stationary Flow Case after 1 Time Step

TABLE III
TIME TAKEN FOR PERCENTAGE DIFFERENCE CAP OF HURST PARAMETERS

H=0.6			H=0.9	
Cap	Difference	Time Taken	Difference	Time Taken
30%	26.356%	0.286747	20.558%	0.261346
30%	22.575%	0.146702	26.991%	0.118507
30%	29.425%	0.173842	28.135%	0.256904
30%	24.703%	0.172994	27.523%	0.270883
30%	26.907%	0.077948	22.014%	0.080453
30%	22.585%	0.093736	25.834%	0.901332
30%	21.664%	0.162416	25.357%	0.163796
30%	27.680%	0.217382	28.036%	0.765894
30%	23.464%	0.118588	28.796%	0.345014
30%	26.541%	0.201329	23.487%	0.183214
Average:	25.190%	0.1651684	25.673%	0.3347343
25%	22.104%	0.86736	24.044%	1.163692
25%	21.165%	0.161918	34.956%	0.272502
25%	23.686%	0.500552	20.632%	0.95775
25%	23.112%	0.181758	22.222%	0.445708
25%	20.997%	0.334041	21.703%	0.222179
25%	21.598%	0.170566	23.583%	1.763985
25%	22.692%	0.304982	24.526%	0.462329
25%	23.899%	0.173144	23.772%	0.970558
25%	23.283%	0.18609	23.036%	0.134203
25%	24.346%	0.244692	21.684%	0.265622
Average:	22.688%	0.3125103	24.016%	0.6658528
20%	18.913%	7.175192	17.984%	32.084597
20%	19.905%	0.269781	19.456%	2.584064
20%	19.562%	2.956529	19.709%	3.894293
20%	19.893%	0.77	16.506%	11.510269
20%	18.130%	3.966869	18.444%	7.236277
20%	19.560%	0.227337	19.341%	3.272515
20%	18.664%	10.901815	19.853%	9.160692
20%	19.502%	3.784419	18.417%	7.030988
20%	19.973%	11.12417	19.058%	4.760391
20%	19.140%	3.164217	19.488%	24.44816
Average:	19.324%	4.4340329	18.826%	10.5982246
17%	15.515%	217.53632	16.793%	1288.468401
17%	16.648%	82.281101	16.888%	367.974248
17%	16.924%	163.420519	15.296%	123.172032
17%	16.750%	1445.372839	16.091%	167.115428
17%	16.757%	810.586996	16.044%	214.569284
17%	16.068%	385.363203	16.761%	104.435453
17%	16.857%	106.741403	16.549%	538.433169
17%	16.447%	242.214792	16.836%	356.098545
17%	15.612%	537.96851	16.208%	654.015499
17%	16.958%	14.414493	16.529%	300.426337
Average:	16.454%	400.5900176	16.400%	411.4708396

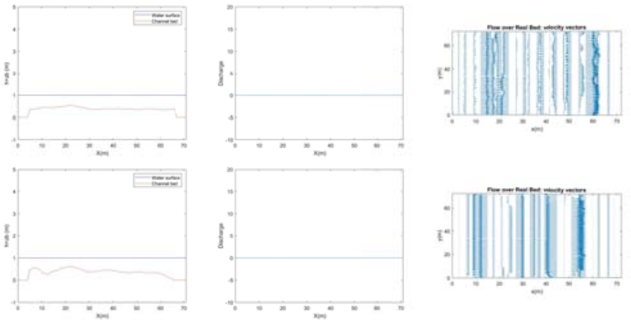


Fig. 9 FBM (a) Against Test Case (b) For a Stationary Flow Case after 500 Time Steps

REFERENCES

[1] Mandelbrot. B. B, Mandelbrot. B. B, The fractal geometry of nature (Vol. 1) *1982. New York: WH freeman.*

[2] A. M. Astakhov, V. A. Avetisov, S. K. Nechaev, and K. E. Polovnikov, *Fractal Dimension Meets Topology: Statistical and Topological Properties of Globular Macromolecules with Volume Interactions,* Macromolecules, vol. 54, no. 3, pp. 1281–1290, Feb. 2021, doi: 10.1021/acs.macromol.0c01717.

[3] A. N. Kolmogorov, *Wienersche spiralen und einige andere interessante Kurven in Hilbertscen Raum, C. R. (doklady),* 1940. Acad. Sci. URSS (N.S.) 26 115–118. MR0003441

[4] M. S. Taqqu, *Benoit Mandelbrot and Fractional Brownian Motion,* Statistical Science, vol. 28, no. 1, Feb. 2013, doi: 10.1214/12-sts389.

[5] B. B. Mandelbrot and J. w. van Ness. *Fractional Brownian motions, fractional noises and applications*, 1968, SIAM Rev. 10 422–437. MR0242239

[6] F. Family and T. Vicsek, *Dynamics of Fractal Surfaces*, 1991 ISBN 981-02-0720-4 page 45

[7] H. Zahouani, R. Vargiolu, and J.-L. Loubet, *Fractal models of surface topography and contact mechanics,* Mathematical and Computer Modelling, vol. 28, no. 4, pp. 517–534, 1998, doi: https://doi.org/10.1016/S0895-7177(98)00139-3.

[8] J. van Lawick van Pabst and H. Jense, *Dynamic Terrain Generation Based on Multifractal Techniques,* in High Performance Computing for Computer Graphics and Visualisation, 1996, pp. 186–203.

[9] A. Cristea and F. Liarokapis, *Fractal Nature - Generating Realistic Terrains for Games*. 1-8. 10.1109/VS-GAMES.2015.7295776. (2015).

[10] Indigo Renderer, 2004, Advanced Fractional Brownian Motion Noise, https://www.indigorenderer.com/indigo-technical-reference/indigo-shader-language-reference/built-functions-%E2%80%93-procedural-noise-fun-0, 16/03/2023

[11] B. T. Milne, *The utility of fractal geometry in landscape design,* Landscape and Urban Planning, vol. 21, no. 1, pp. 81–90, 1991, doi: https://doi.org/10.1016/0169-2046(91)90034-J.

[12] G. Franceschetti, D. Riccio, Editor(s): G. Franceschetti, D. Riccio, *Scattering, Natural Surfaces, and Fractals*, Academic Press, 2007, Pages 1-19, ISBN 9780122656552, https://doi.org/10.1016/B978-012265655-2/50001-5. (https://www.sciencedirect.com/science/article/pii/B9780122656552500015)

[13] D. Della-Bosca, D. Patterson, S. Costain, (2014) *Fractal Complexity in Built and Game Environments*. In: Pisan Y., Sgouros N.M., Marsh T. (eds) Entertainment Computing – ICEC 2014. ICEC 2014. Lecture Notes in Computer Science, vol 8770. Springer, Berlin, Heidelberg.

[14] N. Enriquez, *A simple construction of the fractional Brownian motion,* Stochastic Processes and their Applications, vol. 109, no. 2, pp. 203–223, 2004, doi: https://doi.org/10.1016/j.spa.2003.10.008.

[15] A. Patrice, and F. Sellan. *The Wavelet-Based Synthesis for Fractional Brownian Motion* Proposed by F. Sellan and Y. Meyer: Remarks and Fast Implementation. Applied and Computational Harmonic Analysis 3, no. 4 (October 1996): 377–83. https://doi.org/10.1006/acha.1996.0030.

[16] B. Jean-Marc, G. Lang, G. Oppenheim, A. Philippe, S. Stoev, and M. S. Taqqu. *Generators of Long-Range Dependent Processes: A Survey.* In Theory and Applications of Long-Range Dependence, edited by Paul Doukhan, Georges Oppenheim, and Murad S. Taqqu, 579–623. Boston: Birkhauser, 2003.

[17] A. Hayes, *Autoregressive Integrated Moving Average (ARIMA)*,2010, https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp

[18] MathWorks. 2022. *wfbm: Fractional Brownian Motion synthesis.* [online] $https://uk.mathworks.com/help/wavelet/ref/wfbm.html\#mw_3fcd5614-1ace-4b39-bcf0-a9b10550eae6$ [Accessed 1 September 2022].

[19] N. Clifford, N. Wright, GL. Harvey, A. Gurnell, O. Harmar, P. Soar. (2010). *Numerical Modeling of River Flow for Ecohydraulic Applications: Some Experiences with Velocity Characterization in Field and Simulated Data.* Journal of Hydraulic Engineering. 136. 1033-1041. 10.1061/(asce)hy.1943-7900.0000057.

[20] Zhou, J, Lattice Boltzmann Methods for Shallow Water Flows. *Springer-Verlag* 2004

[21] Zhou, J. G., An Elastic-Collision Scheme for Lattice Boltzmann Methods, *International Journal of Modern Physics C, vol. 12, no. 3, pp. 387–401, 2001. doi:10.1142/S0129183101001833.*