Geostatistics - Does the simulation method matter?

Julián M. Ortiz - https://julianmortiz.com/

March 13, 2025

Summary

No. All that matters is if it does what it is supposed to do.

1 Introduction

In this short note, I focus on the multiGaussian methods and the debate of whether to use turning bands or sequential Gaussian simulation (or any other multiGaussian method, for that matter). I review what are random functions (you can skip it if you just want the recipe), what simulation methods exist in the multiGaussian case for continuous variables (you can also skip this if you just want to know what these methods **should** do), and finally I address what to expect from a good algorithm implementation of any multiGaussian method.

The bottom line is that the simulation method used to create geostatistical realizations of a variable does not matter. You should stop asking what method is used or what methods does such or such software have implemented. **What you should be asking is whether the method works or not**. And this means, whether the methods follow what the theory says. Let me explain...

2 Random Function Models

Recall that in geostatistics, we need to make a leap of faith. We take the regionalized variable, which is the actual variable in the field we are interested in, and assume that at every spatial location $\boldsymbol{u}$, the value of the regionalized variable is a random variable $Z(\boldsymbol{u})$ (notice that we capitalize to emphasize the difference). This random variable is correlated with its neigbohrs in space, therefore, if we want to characterize these random variables, and then simulate them, we need to undertand the random function, that is, the collection of random variables in the domain of interest: $\{Z(\boldsymbol{u_i}), \boldsymbol{u_i} \in D\}$.

Geostatistical simulation methods are based on random function models. For example, there are many methods that are multiGaussian, that means they rely on a multiGaussian assumption. But what does this mean? In the case of a single variable Z, it means that there is an implicit assumption that if you take values of that variable and find pairs separated in space by a vector $\boldsymbol{h}$ (which means separated by a certain distance, the module of the vector, $|\boldsymbol{h}|$, and where the head $Z(\boldsymbol{u})$ and tail $Z(\boldsymbol{u}+\boldsymbol{h})$ of the vector point in a certain direction), the collection of all the pairs found $\{(Z(\boldsymbol{u_i}), Z(\boldsymbol{u_i} + \boldsymbol{h}), for\, i = 1, \ldots, N(\boldsymbol{h})\}$ will follow a bivariate Gaussian distribution (**Figure 1**). In other words, if you plot all these pairs, the scatter plot will show isodensity lines that follow an ellipsoidal shape. The correlation of this plot is related to the spatial continuity for that vector (it actually is the correlogram value for that vector, $\rho(\boldsymbol{h})$).

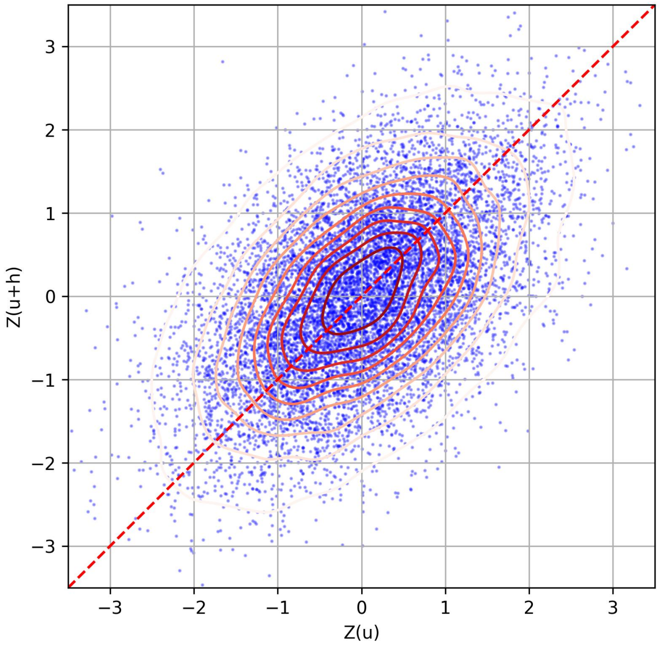

Figure 1: Example of a bivariate Gaussian plot with the relationship between points separated by a certain distance $\boldsymbol{h}$.

But the multiGaussian assumption means more that bivariate relationships. It also means that if you take triplets $(Z(\boldsymbol{u}), Z(\boldsymbol{u} + \boldsymbol{h_1}), Z(\boldsymbol{u} + \boldsymbol{h_2}))$, these will follow a trivariate

Gaussian distribution (the scatter plot will show ellipsoidal isodensity surfaces in three dimensions) (**Figure 2**). And this is also true for four, five and more points, but we cannot draw those cases.

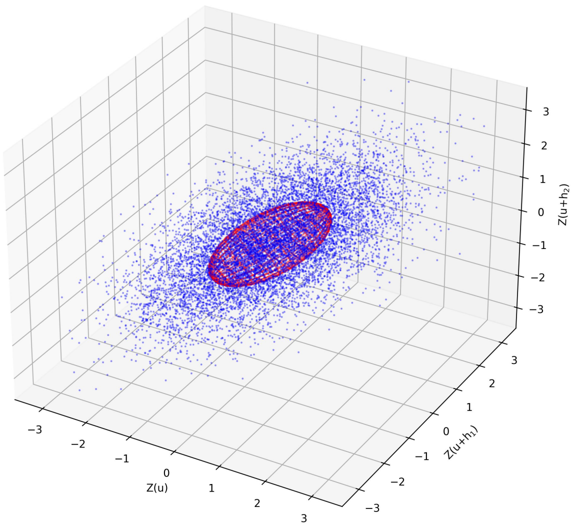

Figure 2: Example of a trivariate Gaussian plot with the relationship between points separated by distances $\boldsymbol{h_1}$ and $\boldsymbol{h_2}$.

3 Simulation Algorithms

There are many multiGaussian simulation algorithms. Some are exact and some are approximate. As everything in life, you cannot get everything: an exact method works in small cases and is limited by the number of nodes to simulate. An approximate method, can work on large number of nodes to simulate, but... it only gives an approximate result.

Among the multiGaussian methods, turning bands simulation and sequential Gaussian simulation are popular options that are implemented in commercial software. However, there are many more metods that rely on the multi-Gaussian assumption, as listed in Chiles and Delfiner book (see [1], Table 7.1), including:

- Sequential Gaussian: this method was originated from the need to simulate indicators, which was then expanded to the use of multiGaussian random functions. The original code was developed by Gomez-Hernandez and Srivastava [4]. A nice history of how this method came to life is provided by the same authors in a paper, part of a special issue in honor to Andre Journel [5].

- Matrix decomposition: this method is exact, but it is limited by the number of nodes and conditioning values considered, as it requires a matrix decomposition. It was originally introduced by Davis [2]. It is not used in practical applications in mining, although may be implemented as part of block simulation algorithms without you knowing.

- Turning bands: originally introduced by Matheron [8], and implemented in the book “Mining Geostatistics” by Journel and Huijbregts [7], this method relies on the generation of one-dimensional simulations over randomly oriented lines that are then projected and “averaged” in space over simulation node locations.

- Autoregressive: this model works only in 2D [1] and is not used in practical applications in mining.

- Moving average: this method works because averages tend to a normal distribution. The idea is to determine the appropriate weights assigned to nodes in a spatial configuration. Then, a convolution between points with white noise (i.e. pure random noise) and these weights is applied to obtain a “mocing average”, which becomes the simulated random values. These moving averages reproduce a certain covariance matrix, which is a result of the weights used in the first place. The fact that the method is a simple product of weights and simulated values at simulation nodes makes it very efficient [9].

- Continuous spectral: a stationary random function can be seen as a mixture of independent sinusoidal components at different frequencies [1]. Therefore, a random function can be simulated by considering an orthogonal random spectral measure, which requires simulating two real-valued variables that comply with certain conditions. This approach only works efficiently for some covariance functions, namely those parabolic near the origin, and it is not seen implemented in commercial software.

4 What to check?

As mentioned before, the methods described earlier have the same underlying assumption of multiGaussianity and should (in theory) achieve similar results. However, practicality usually mean that the implementation does not follow exactly what theory says.

For instance, in sequential simulation, one should use all the sample data and previously simulated values to ensure that the reproduction of the conditional probabilities from which we draw the subsequent simulated values is correct. In practice, we truncate to a maximum number of samples and/or previously simulated values. This has an impact on the histogram and variogram reproduction.

In turning bands, convergence is theoretically achieved with infinite randomly oriented lines. In practice, we just use a large number of approximately randomly-oriented simulated lines to project on the grid, so the result is approximate. Again this has impact on the histogram and variogram reproduction and on the ergodicity of the method (how each realization converges to the expected statistics of the random function model).

Furthermore, in practice, there may be trends in the data, thus the assumption (or decision) of stationarity may not work well. This means the modeller may decide limiting the neighborhood to avoid smearing the different high and low value zones.

So, what should you worry about? Simple. Simulations should reproduce the conditioning data at their locations, reproduce the global histogram and the variogram.

Two important aspects must be considered:

1. Gaussianity: in most cases, the original distribution is not Gaussian, so a normal score transformation is used. The resulting univariate distribution is guaranteed to be Gaussian, but the bivariate, trivariate and

multivariate (multi-point, really) distributions should be checked. Drawing h-scatter plots as depicted in **Figures 1** and **2** can help identify any significant departure from this assumption. There are other approaches to check for multiGaussianity, but in practice they are seldom used. One important note is that small departures from the precise expected histogram (a Gaussian distribution with mean 0 and vaiance 1), may generate biases in the back-transformation step. These departures may occur even when the average statistics match exactly the target values. Due to the fluctuations and the non-linear nature of the process of back-transformation (especially for distributions with very long tails, such as in the case of precious metals) the compounded effect of the realizations back-transformation may end up in a bias mean of the distribution in original units. Some post-processing may be needed in these cases.

2. Support: the information used for inferring the random function features is at sample support (point support). The histogram and variogram of normal scores reflect the statistical distribution and spatial correlation of point support samples. Simulation outcomes should therefore be checked at point support, before any change of support to block values has been done.

One would expect that trends are also followed in the resulting realizations (although simulations are not very robust to departures from stationarity, so detrending may be required as a pre-process). Swath plots should be checked and reflect the flustuations in the local mean, and also the expected uncertainty increase far from data and low uncertainty near conditioning locations.

In summary, a simulation is good if:

- **It reproduces the conditioning data** at their locations: all realizations should be “fixed” at conditioning locations and fluctuate away from conditioning points (see **Figure 3** where conditioning data reproduction is checked).

- **It reproduces the global histogram in Gaussian units at point support**. The point-support histogram of the simulated values prior to back-transformation (i.e. the histogram of the simulated normal score values) is standard Gaussian on average. Some realizations will have a mean that is not exactly zero, and a variance that is not exactly one, but the average mean of the realizations should be very close to zero and the average variance of the realizations should be very close to one.

- **It also reproduces the global histogram in original units at point support**. The back-transformed point-support histogram of the simulated values reproduces the original declustered histogram (see **Figure 4**).

- **It reproduces the normal scores point-support variogram model**. The variogram at point-support of the simulated values prior to back-transformation (i.e. the variogram of the simulated normal score values) should, on average, match the model inputted in the algorithm (**Figure 5**). It should be noted that the conditioning data will enforce the continuity, therefore if a poor model of the experimental variogram is used, the conditioning data will take control and the resulting variogram of simulated values will be closer to that of the conditioning data, particularly if these are abundant.

- **It reproduces the point-support variogram in original units**. The variogram at point-support of the simulated values in original units, after back-transformation, should match the data variogram. Notice that we can compare it with the experimental variogram of the original viarable or with the variogram model we may have for estimation. Small mismatches in histogram reproduction will be reflected as departures in the sill reproduction, since it is linked to the variance of the data.

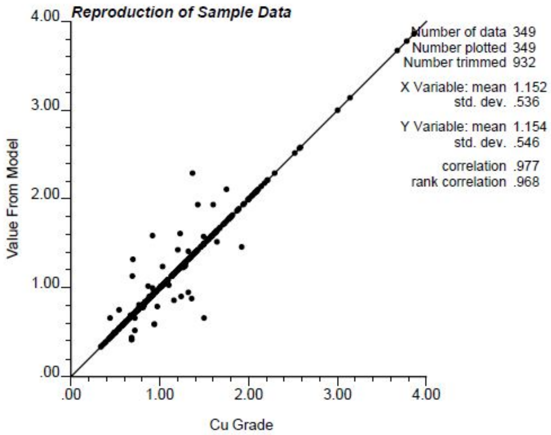

Figure 3: Check of reproduction of conditioning values at their locations. Departures are explained by the assignment of data to nodes.

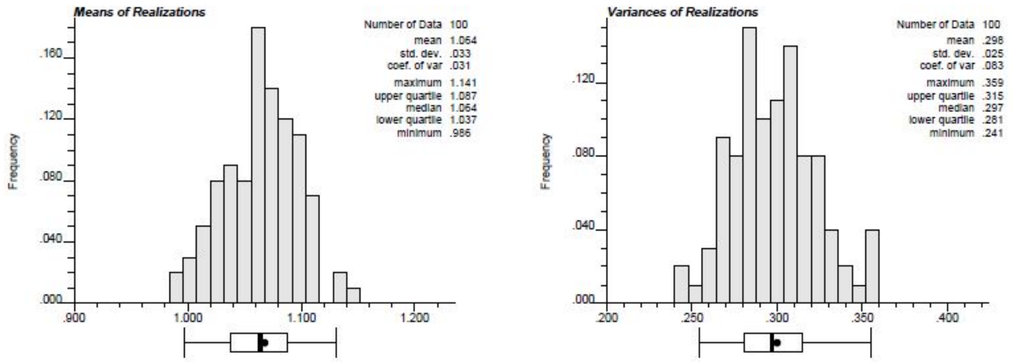

Figure 4: Check of histogram reproduction. Left: average of realization means; Right: average of realization variances.

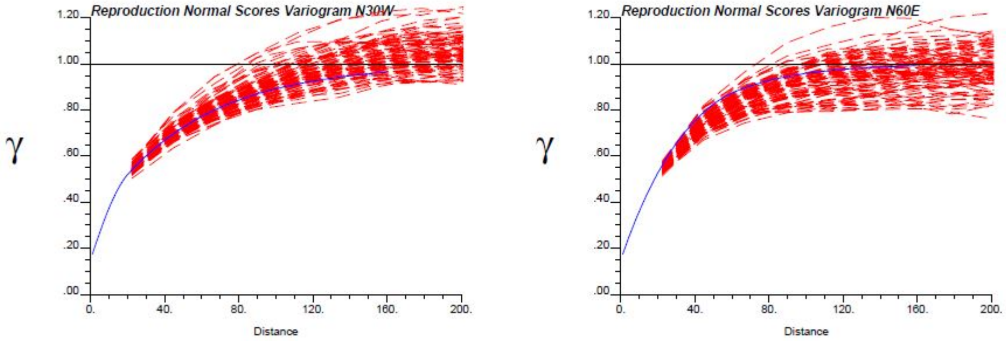

Figure 5: Check of variogram reproduction of normal scores in two directions.

If the software you use to simulate only provides block support back-transformed values, it will be hard to check if the quality of the simulation and the choice of parameters used is addequate. One approach to “validate” the approach is to feed the algorithm the data already normal score transformed and perform the simulation at point support (i.e. without discretization of the blocks). In this way, you can run a few realizations, check conditioning, histogram and variogram reproduction and, when you are satisfied with the results, you can re-run the realizations with the original data, so that the algorithm performs the transformation and the averaging. If your software does not allow for point support simulation, it is very hard to check that the realizations are converging to the expected statistics (histogram and variogram).

5 What makes an algorithm good?

An algorithm can be qualified as good if:

- It is easy to use.
- All relevant parameters can be changed by the user.
- Runs in reasonable time.
- Provides results with the expected statistics (histogram and variogram reproduction).

6 The small print

Obviously, making the point that the algorithm does not matter had the aim to provoke some discussion. In reality, algorithms have different pros and cons.

For instance, sequential Gaussian simulation is fairly easy to explain and teach and can be presented as a recursive application of Bayes' law to update conditional distributions from which simulation values are drawn (see [10]). The main drawback is that the conditioning data must be truncated to make it practical. Otherwise, the kriging system needed would increase with every new simulated node in the domain, becoming very slow and, at some point, impossible to solve. Despite this, it can be made very efficient by parallelization of the code [12, 13, 11].

On the other hand, turning bands is more difficult to understand and explain. Its derivation requires more advance mathematics and its convergence to multiGaussianity is based on the central limit theorem [6]. But, because the approach uses one dimensional lines to simulate a three dimensional field, it can be parallelized and made very efficient [3, 14].

In conclusion, only trust your algorithm if you can check the results.

References

[1] Chilès, J. P., and Delfiner, P. *Geostatistics Modeling Spatial Uncertainty*, second ed. John Wiley & Sons, New York, 2012.

[2] Davis, M. W. Production of conditional simulations via the LU decomposition of the covariance matrix. *Mathematical Geology 19*, 2 (1987), 91–98.

[3] Emery, X. A turning bands program for conditional cosimulation of cross-correlated gaussian random fields. *Computers & Geosciences 34* (2008), 1850–1862.

[4] Gómez-Hernández, J. J., and Srivastava, R. M. ISIM3D: An ANSI-C three dimensional multiple indicator conditional simulation program. *Computers & Geosciences 16*, 4 (1990), 395–410.

[5] Gómez-Hernández, J. J., and Srivastava, R. M. One step at a time: The origins of sequential gaussian simulation and beyond. *Mathematical Geosciences 53* (2021), 193–209.

[6] Journel, A. G. Geostatistics for conditional simulation of ore bodies. *Economic Geology 69* (1974), 673–687.

[7] Journel, A. G., and Huijbregts, C. J. *Mining Geostatistics*. Academic Press, New York, 1978.

[8] Matheron, G. The intrinsic random functions and their applications. *Advances in Applied Probability 5* (1973), 439–468.

[9] Oliver, D. S. Moving averages for gaussian simulation in two and three dimensions. *Mathematical Geology 27* (1995), 939–960.

[10] Ortiz, J. M. Introduction to sequential gaussian simulation. Tech. Rep. “Predictive Geometallurgy and geostatistics Lab, Annual Reort 2020, paper 2020-01”, Queen’s University, 2020.

[11] Peredo, O., and Herrero, J. R. Acceleration strategies for larg-scale sequential simulations using parallel neighbour search: Non-LVA and LVA scenarios. *Computers & Geosciences 160*, 105027 (2022).

[12] Peredo, O., Ortiz, J. M., and Herrero, J. R. Acceleration of the Geostatistical Software Library (GSLIB) by code optimization and hybrid parallel programming. *Computers & Geosciences 85A* (2015), 210–233.

[13] Peredo, O. F., Baeza, D., Ortiz, J. M., and Herrero, J. R. A path-level exact parallelization strategy for sequential simulation. *Computers & Geosciences 110* (2018), 10–22.

[14] Tolosana-Delgado, R., and Mueller, U. *Geostatistics for compositional data with R*. Springer, 2021.

Index