Computers and Geosciences 112 (2018) 121–132

Contents lists available at ScienceDirect

Computers and Geosciences

journal homepage: www.elsevier.com/locate/cageo

Research paper

Accelerating Sequential Gaussian Simulation with a constant path

Raphaël Nussbaumer [a,*], Grégoire Mariethoz [b], Mathieu Gravey [b], Erwan Gloaguen [c], Klaus Holliger [a]

[a] Institute of Earth Sciences, University of Lausanne, Lausanne, Switzerland
[b] Institute of Earth Surface Dynamics, University of Lausanne, Lausanne, Switzerland
[c] Centre Eau Terre Environnement, Institut National de la Recherche Scientifique, Québec, Canada

ARTICLE INFO

Keywords:
Simulation path
Sequential simulation
Sequential Gaussian Simulation
Constant path
Multi-grid approach
Parallelization

ABSTRACT

Sequential Gaussian Simulation (SGS) is a stochastic simulation technique commonly employed for generating realizations of Gaussian random fields. Arguably, the main limitation of this technique is the high computational cost associated with determining the kriging weights. This problem is compounded by the fact that often many realizations are required to allow for an adequate uncertainty assessment. A seemingly simple way to address this problem is to keep the same simulation path for all realizations. This results in identical neighbourhood configurations and hence the kriging weights only need to be determined once and can then be re-used in all subsequent realizations. This approach is generally not recommended because it is expected to result in correlation between the realizations. Here, we challenge this common preconception and make the case for the use of a constant path approach in SGS by systematically evaluating the associated benefits and limitations. We present a detailed implementation, particularly regarding parallelization and memory requirements. Extensive numerical tests demonstrate that using a constant path allows for substantial computational gains with very limited loss of simulation accuracy. This is especially the case for a constant multi-grid path. The computational savings can be used to increase the neighbourhood size, thus allowing for a better reproduction of the spatial statistics. The outcome of this study is a recommendation for an optimal implementation of SGS that maximizes accurate reproduction of the covariance structure as well as computational efficiency.

1. Introduction

Sequential Gaussian Simulation (SGS) is a popular method for generating stochastic values on a grid under the constraints of a statistical model and, possibly, some initially known values, herein referred to as hard data. SGS has been extensively used by practitioners because of its intuitive theoretical basis, its simple numerical implementation, and its high flexibility (e.g. Gómez-Hernández and Journel, 1993; Pebesma and Wesseling, 1998). Arguably, the major drawback of SGS is its computational cost. The exact estimation of kriging relies on taking into account all conditioning nodes, which results in large linear systems that need to be solved. For a square matrix of size n, common linear solvers have a computational complexity of $O(n^3)$ (Trefethen and Bau, 1997), which means that the computational effort is proportional to the cube of the matrix dimension. Therefore, the sequential simulation of a grid with N nodes represents an $O(N^4)$-type problem (Dimitrakopoulos and Luo, 2004).

Various attempts have been undertaken to reduce the associated computational cost. The most widespread approach is the so-called limited or moving neighbourhood, that is, the approximation of the kriging estimate by using only a limited number of conditioning points referred to as the neighbours (e.g. Isaaks and Srivastava, 1989; Deutsch and Journel, 1992; Goovaerts, 1997). This reduces the computational complexity of SGS to $O(k^3N)$, where k denotes the number of neighbours. This approach is rooted in the observation that neighbours which are located far away from the simulated point receive small or even vanishing weights. This effect originates from the rapid decrease of correlation with distance inherent to most covariance functions and is enhanced by the presence of intermediate neighbours screening the influence of those behind (e.g. Chilès and Delfiner, 1999). However, the omission of neighbours has shown to bias the simulation covariance matrix (Emery and Peláez, 2011; Nussbaumer et al., 2017), which in turn results in artifacts in the realizations (e.g. Meyer, 2004). Recent works on reducing such detrimental effects, while limiting the neighbourhood size and optimizing the computational efficiency, include those of Gribov and Krivoruchko (2004), Rivoirard and Romary (2011) and Dimitrakopoulos and Luo (2004).

An alternative to reducing the size of the kriging covariance matrix is

* Corresponding author.
E-mail address: raphael.nussbaumer@unil.ch (R. Nussbaumer).

https://doi.org/10.1016/j.cageo.2017.12.006
Received 2 May 2017; Received in revised form 5 September 2017; Accepted 14 December 2017
Available online 16 December 2017
0098-3004/

to approximate it. Barry and Kelley Pace (1997) formulate covariance-based kriging, which leads to the inversion of sparse symmetric matrices. Sparse matrix solvers considerably improve the computational performance, but this approach is limited to simulations based on covariance functions with a finite range. Furrer et al. (2006) and Memarsadeghi and Mount (2007) further increase the sparsity of the matrix by tapering the covariance for large lag-distances. Related approaches comprise the approximate iterative method (Billings et al., 2002), the low rank approximation (Kammann and Wand, 2003), the Sherman-Morrison-Woodbury formula (Sakata et al., 2004), and fast summation methods (Memarsadeghi et al., 2008; Srinivasan et al., 2008).

Another approach is to only consider simulations whose covariance function is from a limited set of easily solvable covariance models. Omre et al. (1993) propose the screening sequential simulation, which provides exact simulations for covariance models with the Markov property, such as, for example, the exponential model in 1D. Hartman and Hössjer (2008) approximate the simulated Gaussian field with a set of Gaussian Markov random fields (Rue and Tjelmeland, 2002), which can be simulated exactly and efficiently. Finally, Cressie and Johannesson (2008) consider covariance models composed of a fixed number of basic non-stationary functions. This technique is also referred to as fixed-rank kriging. A related approach is the predictive processes method (e.g. Banerjee et al., 2008).

A more general technique to cope with the high computation costs of SGS is parallelization, which reduces the computation time by splitting the work among several cores (Vargas et al., 2007; Mariethoz, 2010; Nunes and Almeida, 2010; Rasera et al., 2015). It is important to note that parallelization does not reduce the computational burden, but merely spreads it over several cores, and hence is just a useful complement to the other techniques.

The approach explored in this study aims at decreasing the overall computational cost by taking advantage of the large number of realizations typically needed in geostatistical applications. Indeed, an uncertainty assessment can only be performed with an ensemble of realizations spanning the variability of outcomes. When the simulation path, that is, the order in which the nodes are simulated, is kept identical among multiple realizations, the neighbourhood configurations of each simulated node are also identical throughout these realizations. Because the kriging weights are computed solely with the relative distances between nodes, a constant neighbourhood configuration produces the same kriging weights. Therefore, these weights only need to be computed once and then can be re-used for all realizations. This reduces the computational effort of each additional realization to simple matrix multiplications.

While some works outline the advantages of using a constant path (e.g. Verly, 1993), the overwhelming majority still discourages its use, because of the risk to draw correlated realizations, and rather advocate a randomized path to explore the solution space more homogenously (e.g. Deutsch and Journel, 1992; Goovaerts, 1997). Conversely, Cáceres et al. (2010); Boisvert and Deutsch (2011) reported that using a constant path in SGS does not result in a significant reduction of the space of uncertainty for neither first- nor second- order statistics, while allowing for compelling reductions in computational cost. However, both studies are based on empirical evidence and hence the generic validity of their findings remains to be verified.

In the present work, we seek to provide a thorough understanding of the implications of changing the simulation path in order to assess the constant path method. The paper is organized as follows. We begin by presenting a methodological description of randomized path simulations (section 2), followed by the implementation of a constant path method (section 3) and the quantification of the associated computational gains (section 4). Finally, we discuss some limitations of the covariance matrix evaluation (section 5).

2. Theory of randomized paths simulations

In order to understand the implications of generating stochastic realizations based on the same simulation path, the links between the random function (RF) Z, the realizations z, and the path p_i need to be explored in some detail.

2.1. Definition of a random function

In probability theory, a random variable (RV) denoted X is a deterministic function mapping the set of possible outcomes Ω of a random phenomenon to their values, usually a real number $\mathbb{R}$,

$$X : \Omega \rightarrow \mathbb{R} \tag{1}$$

$$\omega \mapsto x.$$

In the definition of a RV, Ω has to be a probability space, which implies that each possible outcome ω has a well-defined probability. Thus, the probability $P(X \leq x_T)$ is defined by the set of events $\{\omega \in \Omega : X(\omega) \leq x_T\}$.

For instance, a RV describing the sum of two rolled dice n_1 and n_2 is defined as the function mapping every possible outcome (n_1, n_2) to the measure $n_1 + n_2$

$$X(\{n_1, n_2\}) = n_1 + n_2. \tag{2}$$

With this formalism, the probability of the sum of two dice being 5 is defined as

$$\begin{aligned} P(X = 5) &:= P(\{n_1, n_2 \in \{1,2,3,4,5,6\} : n_1 + n_2 = 5\}) \\ &= P(\{1,4\},\{2,3\},\{3,2\},\{4,1\}) = 4/6^2 = 1/9. \end{aligned} \tag{3}$$

A realization $x^{(l)}$ is the value observed from a RV X given a specific outcome of the random phenomenon, also called random variate, ω_l

$$x^{(l)} = X(\omega_l). \tag{4}$$

2.2. Sequential Gaussian Simulation

GS is an algorithm whose purpose is to produce realizations $z^{(l)}(\boldsymbol{u})$ of a *regionalized multi-Gaussian random function (RF)* $Z(\boldsymbol{u})$.

1. A RF is a collection of indexed RV. If the indexation is multi-dimensional, the collection is usually referred to as random field instead.
2. A RF is called regionalized (Matheron, 1965) if it is distributed in a continuous space domain $D \subset \mathbb{R}^n$,

$$Z = \{Z(\boldsymbol{u}),\ \boldsymbol{u} \in D\}, \tag{5}$$

where $\boldsymbol{u}$ represents a space coordinate vector.

3. A RF is multi-Gaussian if any finite collection of its components has a multi-variate normal distribution. While this constraint is restrictive, it allows for the RF to be fully determined by its first- and second-order moments, that is, the mean $\boldsymbol{\mu}_Z$ and the covariance matrix $\boldsymbol{C}_Z$

$$Z \sim \mathcal{N}(\boldsymbol{\mu}_Z, \boldsymbol{C}_Z). \tag{6}$$

SGS takes advantage of this multi-Gaussian property to produce realizations of Z. It iteratively visits each node of the grid, computes the kriging estimate and variance error σ_E based on previously simulated nodes and samples a value from the corresponding conditional probability distribution. A newly simulated node thus becomes a conditioning node for the next one to be simulated. Mathematically, this can be summarized as

$$Z(\boldsymbol{u}_i) = \sum_{j=1}^{i-1} \lambda_j(\boldsymbol{u}_i) Z(\boldsymbol{u}_j) + \sigma_E(\boldsymbol{u}_i) U(\boldsymbol{u}_i), \quad \forall i = 1, \ldots, n, \tag{7}$$

where U is a standard Gaussian vector used for randomly sampling the conditional distribution and λ_j are the kriging weights which define the influence of the conditioning nodes.