Spring 1-1-2018

Simulation of Nonstationary Gaussian Process by Consecutive Conditioning

Kwan Ho Lee

University of Colorado at Boulder, jennysbk@gmail.com

Follow this and additional works at: https://scholar.colorado.edu/appm_gradetds

Part of the Applied Mathematics Commons, and the Physical Processes Commons

Recommended Citation
https://scholar.colorado.edu/appm_gradetds/136
Simulation of Nonstationary Gaussian Process by Consecutive Conditioning

by

Kwan Ho Lee

B.A., State University of New York at Geneseo, 2015

A thesis submitted to the Faculty of the Graduate School of the University of Colorado in partial fulfillment of the requirements for the degree of Master of Science Department of Applied Mathematics 2018
This thesis entitled:
Simulation of Nonstationary Gaussian Process by Consecutive Conditioning
written by Kwan Ho Lee
has been approved for the Department of Applied Mathematics

Prof. William Kleiber

Prof. Jem Corcoran

Prof. Brian Zaharatos

Date ______________________

The final copy of this thesis has been examined by the signatories, and we find that both
the content and the form meet acceptable presentation standards of scholarly work in the
above mentioned discipline.
This thesis aims to develop the method of consecutive conditioning, which is used to directly simulate a stochastic process given an arbitrary covariance function. As a method for simulating stochastic processes, consecutive conditioning is useful in at least in three respects. While most methods require modeling of the covariance function prior to simulation, consecutive conditioning can be used with any arbitrary covariance function, thus introducing less error into the simulation than other methods. Second, consecutive conditioning allows us to perform very fast computations during simulation and can be used even by people who are not experts in modeling, unlike other methods which require substantial statistical work prior to simulation. Finally, this method can be used to simulate both stationary and nonstationary processes, which is particularly useful since the majority of real-world physical processes are nonstationary.

With the Kullback-Leibler divergence in hand, we validate the consecutive conditioning method as follows. After executing our method on a simulated distribution, we compare the resulting distribution with the true distribution for calculating the KL values. Then, we demonstrate that the consecutive conditioning works well on different covariance functions by applying it to a different series of simulations. First, we use a consecutive conditioning with several different covariance functions to simulate two time points of a stochastic process, then compare the results to determine the best covariance function for our method. Finally, we use our method to generate five time points from a stochastic process in both uninitialized and initialized cases, then evaluate the results.
Dedication

To Mom and Dad
Acknowledgements

First of all I would like to thank my advisor, Will Kleiber for his patience throughout the writing process and for answering all of my questions, no matter how silly they were at the time. Thanks to my family for their love and support throughout my education. In particular, I would like to thank my dad and mother for their financial contributions to my graduate school career. Most importantly, I want to thank my mom for encouraging me to continue pursuing my masters degree. None of this would have happened without her belief in my ability.
Contents

Chapter

1 Introduction 1

1.1 Simulation 2

1.1.1 Simulation of Stationary Random Fields 2

1.1.2 Simulation of Nonstationary Random Fields 3

1.2 Literature Review 3

1.3 Challenges 4

2 Method: Consecutive Conditioning Simulation 6

2.1 Conditional Simulation and Kriging 6

2.1.1 Kriging 6

2.1.2 Conditional Simulation 8

2.2 Consecutive Conditioning 9

3 Validation of Method 14

3.1 Kullback Leibler divergence 14

3.2 Simulation at Two Time Points 15

3.2.1 Matern 16

3.2.2 Squared exponential 18

3.2.3 Cauchy 19

3.3 Uninitialized Simulation 21
Figures

3.1 Top: Showing how the Matern covariance function works in KL value if the range parameter is fixed and the smoothness parameter varies between 10 and 100. Bottom: Showing how the Matern covariance function works in KL value if the smoothness parameter is fixed and the range parameter varies between 10 and 100.

3.2 Showing how the SE covariance function works in KL value if the range parameter varies between $[1, 4]$.

3.3 Top Left: Showing how the Cauchy covariance function works in KL value if the LRD parameter is 1. Top Right: Showing how the Cauchy covariance function works in the KL value if the LRD parameter is 2. Bottom Left: Showing how the Cauchy covariance function works in the KL value if the LRD parameter is 3. Bottom Right: Showing how the Cauchy covariance function works in the KL value if the LRD parameter is 4.

3.4 $(v, a)$ implies the KL divergence value by the range and the smoothness parameter when $a$ varies between $[0, 10]$ and $v$ varies between $[0, 10]$.

3.5 Top: Showing how the SE covariance function works in KL value if the range parameter $a$ varies between $[0, 5]$. Bottom: Showing how the SE covariance function works in KL value if the range parameter $a$ varies between $[0, 100]$.
3.6 \((\alpha, \beta)\) implies the KL divergence value by the smoothness and the long-range dependence parameter when the range is fixed as 0.5.

3.7 \((v, a)\) implies the KL divergence value by the range and the smoothness parameter when \(a\) varies between \([0, 10]\) and \(v\) varies between \([0, 10]\).

3.8 Top: Showing how the SE covariance function works in the KL divergence value if the range parameter \(a\) varies between \([0, 5]\). Bottom: Showing how the SE covariance function works in the KL divergence value if the range parameter \(a\) is \([0, 100]\).

3.9 \((\alpha, \beta)\) implies the KL divergence value by the smoothness and the long-range dependence parameter when the range is fixed as 0.5.

3.10 \((v, a)\) implies percentage by the range and the smoothness parameter when \(a\) varies between \([0, 10]\) and \(v\) varies between \([0, 10]\).

3.11 Top: Showing percent improvement if the range parameter \(a\) varies between \([0, 5]\). Bottom: Showing percent improvement if the range parameter \(a\) varies between \([0, 100]\).

3.12 \((\alpha, \beta)\) implies percent improvement of the KL by the smoothness and the LRD parameter when the range is fixed as 0.5.
Chapter 1

Introduction

Society’s reliance on technology has grown exponentially over the past few decades, and as a result datasets today are much larger and more detailed than ever before. The large quantities of data we now have at our disposal are incredibly useful when modeling stochastic phenomena in the real world, such as weather forecasts and traffic management. However, in the real world, the physical processes are not stationary, hence the traditional stationary simulation algorithms are not directly applicable.

Stationary random fields play a fundamental role in spatial statistics, but unfortunately stationarity is often violated when working with real data. This issue is more pressing with the recent big data sets and high resolution sensing for which nonstationarity can be clearly visible. This raises a difficult problem for spatial statisticians when they try to estimate and model the dependency structure in random fields.

There are two different types of processes in spatial statistics: stationary and nonstationary. Simulation methods for stationary processes are well established. Among many different simulation of stationary methods, possibly the most popular is circulant embedding [1, 3, 20]. This method allows us to achieve fast high resolution simulations but it is restricted only to stationary process. While simulation with stationary processes works well with the circulant embedding, there are only few approaches to perform high resolution simulation with nonstationary processes.

The layout of this thesis is as follows. In chapter 1 we discuss the nonstationary
simulation in further detail, presenting methods employed in the literature and discussing specific challenges to this problem, such as issues introduced by the complexity of modeling or estimating. The main discussion of the Chapter 2 centers on our method of consecutive conditioning simulation. In the Chapter 3, we evaluate the performance of our method by validating it with a series of simulations.

1.1 Simulation

A simulation is a realization of a random process or a set of random variables. In other words, it is a draw from the multivariate distribution that defines the process. The act of simulating something requires that a model must be developed first. And this model represents the key characteristics and behaviors of the selected physical process. While the simulation represents the behavior of the system over time, the model represents the system itself.

A mathematical model is a representation of reality, and a simulation is an approximate solution to, or a draw from that model. By changing variables in the simulation, it is possible to estimate the future behavior of the system. There are two main reasons why we do statistical simulation. First, to understand how model definitions affect process behavior (i.e., to just look at the simulations to see if they look reasonable). Second, to estimate difficult probabilities (e.g., the probability of maximum over the whole domain is greater than 2, which is analytically hard to derive).

1.1.1 Simulation of Stationary Random Fields

Simulation of random fields is an important part in spatial analyses. Let \( \{Z(s) : s \in D \subset \mathbb{R}^n\} \) stand for a random function used as a model. If the stochastic process \( Z \) is said to be stationary, then it requires that \( E[Z(s)] \) is spatially constant and that the covariance function \( Cov(Z(s_1), Z(s_2)) = C(s_1, s_2) = C(s_1 - s_2) \) is a function of the lag vector separating
two spatial locations. Stationarity implies that the process is invariant under placement of the origin. Some of the most popular simulation methods for stationary processes include circulant embedding [4, 3, 20], turning bands [13], and random coins [17].

1.1.2 Simulation of Nonstationary Random Fields

A random field is classified as nonstationary for one of three reasons, either the mean is not constant, the covariance is not a function of the lag between spatial locations, or both. In this work, we focus only on covariance nonstationarity. There are two methods for performing simulation of nonstationary process: explicitly using the Cholesky factor and writing down a model for the nonstationary process (e.g., convolution-process by Higdon [6]) that allows an easy simulation.

1.2 Literature Review

Before we discuss the challenges presented in simulation of nonstationary process, we outline a few popular approaches to simulation of nonstationary process found in the literature. We begin with the approach suggested by Huang [7].

In [7], the proposed approach is Karhunen-Loeve expansions which is based on the eigen-decomposition of the covariance function. In the Karhunen-Loeve expansion method, the orthogonal deterministic basis functions and their coefficients magnitudes are the eigenfunctions and eigenvalues of the covariance function. The main idea of the Karhunen-Loeve expansion is therefore to obtain the eigenvalues and eigenfunctions of the covariance function by solving the homogeneous Fredholm integral equation of the following, second kind of equation:

\[ \int_D C(x_1, x_2)f_i(x_1)dx_1 = \lambda_i f_i(x_2) \]

In order to apply Karhunen-Loeve as simulation of nonstationary process tool, we need to know the practicality of solving the Fredholm integral equation numerically, and the accur-
racy of the resulting solution.

The authors in [15] introduce a method of simulation of nonstationary process. The basic idea behind their approach is to use the polynomial chaos expansions to yield an expression of the process as a multi-dimensional Hermite polynomial in a set of uncorrelated Gaussian variables. This approach consists of computing a nonlinear transformation of a suitable Gaussian process. The key method of this approach is the development of a single algorithm for simulating non-Gaussian, non-stationary, and multi-dimensional stochastic processes.

Thirdly, [6] introduces a method of nonstationary simulation. The basic idea behind his approach is to use the process-convolution to develop nonstationary space-time models. Next, in the approach suggested by [14], they use compactly supported basis functions and a Markov random field specification for the random coefficients. This allows one to use sparse matrix techniques for covariance matrix inverses. Finally, there is the approach suggested by [16] and [9]. Their analysis is based on generating a stationary random field followed by a nonlinear deformation to produce a nonstationary field.

1.3 Challenges

Our research goal is to produce an appropriate simulation method that allows high resolution simulation, regardless of whether that method is applied to stationary or non-stationary covariance function. However, only few methods have been suggested in the literature, and all of these extant methods seem to be ineffective in some respects. First of all, the approach suggested by [7] can be applied to very few limited cases. More specifically, the K-L expansion method is only available to use when the exact eigenvalues and eigenfunctions of the covariance function are found. In reality, most cases that we encounter do not have such closed-form eigenfunctions and eigenvalues. Moreover, since there are complicated solving processes in an integral equation from the K-L expansion, it causes huge limitation on
using the K-L expansion method. Therefore, in our research, we aim to find a method that is applicable even to cases that do not have the exact eigenvalues and eigenfunctions. Secondly, the approach suggested by [14] has the advantage of being applicable to large datasets. Their basis constructions methods leads to fast computations for large datasets. However, they have limitations in pre-processing for modeling such as defining parameters in equations before simulating large spatial datasets because their approach produces covariance functions that are sums over a large number of terms. Next, while the approach presented in [6] generated nonstationary covariance functions as integrals, this model requires that people have to numerically approximate the integrals for almost all reasonable situations, which can be computationally expensive and people also incur numerical errors. Finally, in [16] and [9], the authors claim that it is difficult to estimate the deformation functions, and there is no guarantee that the target nonstationary covariance will actually have an inverse deformation that makes the process stationary.

The objectives of this research are to: (1) create a method with respect to nonstationary process that is derived from conditional simulation and simple kriging; and (2) evaluate the performance of this approach by using a series of simulations.

In the next chapter we will suggest an alternative method that can be simulated well given an arbitrary covariance function. However, given some data of which covariance is not clear to determine whether it is stationary or nonstationary to begin with, the other methods might actually be better since they set up a framework for modeling. Therefore, if an arbitrary covariance is given, then our method is better than the other approaches because we do not need to do substantial statistical modeling and estimation prior before simulation.
Chapter 2

Method: Consecutive Conditioning Simulation

2.1 Conditional Simulation and Kriging

2.1.1 Kriging

A major goal of a spatial statistical analysis is smoothing or prediction in order to estimate the underlying process at locations where observations are not necessarily available. The geostatistical method used in this analysis is called kriging [10]. Kriging is an interpolating method developed by a French mathematician Georges Matheron in 1960 [12]. It is based on the 1951 Master’s thesis of Danie G.Krige, after whom the method is named. Kriging, also known as Gaussian process regression or Wiener-Kolmogorov prediction, yields an interpolation function based on a covariance or variogram model derived from the data, rather than a theoretical model of the interpolating function. Since kriging is based on prior covariances, it is able to yield the best linear unbiased prediction.

The basic idea of kriging is to predict the value of a random function at a given point by computing a weighted average of known values of a function in the neighborhood of the point. Kriging is similar to prediction in time series: given past values, typically measured at regular intervals, predicting a signal at some future time. To analyze the signal, first, the spectrum is modeled. And then a filter, or predictor, is constructed. The main difference between the kriging approach and the time series approach is that, unlike the time series approach which is based on temporal setting, kriging takes place in a spatial setting where there is no concept of past or future [18].
Before describing the process of kriging, we must first introduce notation. The kriging predictor of \(Z(s_0)\) will be denoted by \(\hat{Z}(s_0)\). To make our notation simpler, we will express values of functions at sample locations by the subscripts of the locations. For example, \(Z_\alpha = Z(s_\alpha), \ m_\alpha = m(s_\alpha)\) will represent the mean value of \(Z_\alpha\), and \(\Sigma_{\alpha\beta} = Cov(s_\alpha, s_\beta)\) will represent the covariance between \(Z_\alpha\) and \(Z_\beta\).

Depending on the stochastic properties of the random field, we can derive many different methods for calculating the weights, and the different methods would produce different kinds of kriging. The most popular methods are ordinary, universal, and simple kriging. It is convenient to introduce matrix notation for the kriging section. For instance, let \(s_0\) represent a spatial location where we estimate the value of the random function, \(Z(s_0)\). Define the vector \(Z = (Z(s_1), \ldots, Z(s_n))^T\) and the covariance vector \(\Sigma_0 = Cov(Z(s_0), Z) = (C(s_0, s_1), C(s_0, s_2), \ldots, C(s_0, s_n))\), that is, the row vector of covariances between \(Z\) at the new location \(s_0\) and \(Z\) all of the observation locations. Similarly, define

\[
\Sigma = Cov(Z, Z) = (C(s_i, s_j))_{i,j=1}^n
\]

as the \(n \times n\) matrix of covariances between \(Z\) at all pairwise data locations. Ordinary kriging assumes an unknown constant mean over the neighborhood of \(s_0\). Universal kriging assumes the mean that follows a general polynomial trend. The universal kriging predictor is just a slightly generalized form of the ordinary kriging predictor. Finally, simple kriging assumes a known mean and covariance. In this research, we focus on the method of simple kriging rather than the other two kriging methods [2].

For simple kriging, the predictor \(\hat{Z}\) is defined as below:

\[
\hat{Z}(s_0) = \Sigma_0 \Sigma^{-1}(Z - m_0) + m_0
\]

or with the mean \(m_0 = 0\),

\[
\hat{Z}(s_0) = \Sigma_0 \Sigma^{-1}Z
\]
Since we do not know where $\Sigma_0 \Sigma^{-1}$ comes from, we want to show how to derive $\Sigma_0 \Sigma^{-1}$ from simple kriging predictor.

Let $w = (w_1, \ldots, w_n)^T$ be the optimal kriging weights and consider any other set of weights $v = w + u$. Then we have

$$\text{Var}(v' \mathbf{Z} - Z(s_0)) = \text{Var}(u' \mathbf{Z} + (w' \mathbf{Z} - Z(s_0)))$$

$$= \text{Var}(u' \mathbf{Z}) + \text{Var}(w' \mathbf{Z} - Z(s_0)) + 2\text{Cov}(u' \mathbf{Z}, w' \mathbf{Z} - Z(s_0)) \quad (2.1)$$

$$\geq \text{Var}(w' \mathbf{Z} - Z(s_0)) + 2\text{Cov}(u' \mathbf{Z}, w' \mathbf{Z} - Z(s_0)),$$

Then the second term is

$$\text{Cov}(u' \mathbf{Z}, w' \mathbf{Z} - Z(s_0)) = \text{Cov}(u' \mathbf{Z}, w' \mathbf{Z}) - \text{Cov}(Z(s_0), u' \mathbf{Z})$$

$$= u' \Sigma w - \Sigma_0 u$$

$$= w' \Sigma u - \Sigma_0 u \quad (2.2)$$

which is zero for any choice of $u$ when $w' = \Sigma_0 \Sigma^{-1}$. We have shown that this kriging weights, proved above, minimizes the predictive variance.

### 2.1.2 Conditional Simulation

Conditional simulation is a methodology originally developed within geostatistics that allows one to see the variability of a spatial process at locations between the observations. The interesting aspect of the conditional simulation is that simulated values can be generated at very closely spaced geographical positions covering the whole deposit, not only at the sampled locations.

We assume that $Z(s)$ is a mean zero Gaussian process. Suppose $Z(s)$ has been observed at locations $s_1, \ldots, s_n$ and our goal is to generate an augumented vector $\mathbf{Z}_\alpha = (Z(s_1), Z(s_2), \ldots, Z(s_n), Z(s_0))^T$ that is an exact realization of $Z(\cdot)$. We simulate the Gaussian process(spatial uncertainty), and at the same time interpolate the data(regression) so
as to capture something both. If we use a unconditional simulation here, then we make an "error" at the location where we have a simple kriging. In kriging, we do not make that error at the location, thus the difference between the kriging map and the unconditional simulation can be used to "correct" the unconditional simulation into a conditionized one. A conditional simulation at new location \( s_0 \) takes the following steps:

1. Unconditionally simulate a vector
   \[
   (Z_u(s_1), Z_u(s_2), \ldots, Z_u(s_n), Z_u(s_0))
   \]
   with the same covariance function as \( Z \).
2. Krige \( Z_u = (Z_u(s_1), Z_u(s_2), \ldots, Z_u(s_n))^T \) to \( s_0 \), we call this \( \hat{Z}_u(s_0) = \Sigma_0 \Sigma^{-1} Z_u \).
3. Form the conditional simulation
   \[
   Z_{cs}(s_0) = \hat{Z}(s_0) + (Z_u(s_0) - \hat{Z}_u(s_0))
   \]
To show that \( Z_{cs}(s) \) and \( Z(s) \) have the right covariance for any \( s_0 \), we claim \((Z(s_1), Z(s_2), \ldots, Z(s_n), Z_{cs}(s_0)) = (Z^T(s), Z_{cs}(s_0)) \) is a realization with the correct covariance structure in that \( \text{Var}(Z_{cs}) = C(0) \) and \( \text{Cov}(Z_{cs}, Z) = (C(s_0 - s_1), \ldots, C(s_0 - s_n)) = \Sigma_0 \). Note that \( E[Z_{cs}(s)] = 0 \). We have
\[
\text{Var}(Z_{cs}(s_0)) = \text{Var}(\Sigma_0 \Sigma^{-1} Z + Z_u(s_0) - \Sigma_0 \Sigma^{-1} Z_u)
= \Sigma_0 \Sigma^{-1} \Sigma_0^T + C(0) - \Sigma_0 \Sigma^{-1} \Sigma_0^T
= C(0) \quad \tag{2.3}
\]
At data locations \( s_0 = s_i \), we have
\[
Z_{cs}(s_i) = \hat{Z}(s_i) + (Z_u(s_i) - \hat{Z}_u(s_i)) = Z(s_i) + Z_u(s_i) - Z_u(s_i) = Z(s_i),
\]
so that the conditional simulation passes through the observations.

### 2.2 Consecutive Conditioning

We begin by investigating our consecutive conditioning from one dimension for an easy first approach. We then will extend our method and apply to two or more dimensions later.
The basic idea of this method is to split the domain $D \in \mathbb{R}$ into disjoint regions $D_1, \ldots, D_N$ such that $\bigcup_{i=1}^{N} D_i = D$. Here, it should be noted that pairwise unions $D_i \cup D_{i+1}$ are topologically connected for $i = 1, \ldots, N - 1$. To clarify, for example, $D_i = (a_i, a_{i+1}]$ which refers to half-open and half-closed interval.

By this initial algorithm invented by Kleiber, the initial simulation $\mathbf{Z}_1$ in domain $D_1$ is generated by direct Cholesky manipulations. The initial simulation $\mathbf{Z}_1$ is a collection of $\mathbf{Z}(s_i)$ at any set of locations $\{s_i\} \subset D_1$. Suppose we have a simulation $\mathbf{Z} = (\mathbf{Z}_1^T, \ldots, \mathbf{Z}_k^T)^T$ and each component of the simulation $\mathbf{Z}_i$ corresponds to each domain $D_i$, $i = 1, \ldots, k$. To generate a $k+1$ step from given $k$ simulations, we need to follow the traditional conditional simulation step to simulate unconditional simulations over all observed subdomains that are $(\mathbf{Z}_{u,1}^T, \mathbf{Z}_{u,2}^T, \ldots, \mathbf{Z}_{u,k}^T, \mathbf{Z}_{u,k+1}^T) = (\mathbf{Z}_u^T, \mathbf{Z}_{u,k+1}^T)^T$, and use kriging predictors $\hat{\mathbf{Z}}_{k+1}$ and $\hat{\mathbf{Z}}_{u,k+1}$ based on data $\mathbf{Z}$ and $\mathbf{Z}_u$, respectively. However, there is restriction on computing all domains $D_1, \ldots, D_{k+1}$ for simulating dense grids because this could be a very high-dimensional random vector. For this reason, what we suggest is instead unconditionally simulating based on the only nearest-neighbor domain. More specifically, it is to unconditionally simulate $(\mathbf{Z}_{u,k}^T, \mathbf{Z}_{u,k+1}^T)^T$ on domain $D_k \cup D_{k+1}$, and then to form the consecutive conditioned simulation

$$\mathbf{Z}_{CC,k+1} = \hat{\mathbf{Z}}_{k+1} + (\mathbf{Z}_{u,k+1} - \hat{\mathbf{Z}}_{u,k+1})$$

where $\hat{\mathbf{Z}}_{k+1}$ and $\hat{\mathbf{Z}}_{u,k+1}$ are the simple kriging predictors based on $\mathbf{Z}_k$ and $\mathbf{Z}_{u,k}$, respectively.

Similar to the previous section on conditional simulation, we assume $\mathbf{Z}_{k+1} = \mathbf{Z}_{CC,k+1}$, which implies that the following vector $(\mathbf{Z}_1^T, \ldots, \mathbf{Z}_{k+1}^T)^T$ is an approximate realization of the process on the domain $\bigcup_{i=1}^{k+1} D_i$. While our consecutive conditioning is similar to conditional simulation, there are some differences between consecutive conditioning and conditional simulation. First, if we set $\mathbf{Z}_{k+1} = \mathbf{Z}_{CC,k+1}$, we have the approximate realization of the process, whereas conditional simulation has an exact realization. Second, there is significant improvement in the consecutive conditioning method especially with respect to calculation time. For
instance, if we use conditional simulation method over the whole domain $D$ and each subdomain $D_i$ has $n$ spatial locations, then the conditional simulation on $D_k$ would need kriging and simulation calculations based on $kn$ locations. However, consecutive conditioning, which is our method, only needs $2n$ spatial locations in one dimension because we focus on nearest-neighbor domain while conditional simulation has to consider the whole domain. In two or more dimensions, consecutive conditioning follows the same outline as in one dimension above.

We expect that consecutive conditioning works well when the process has an exponential covariance function. As expected, the simulated fields will often have bias in correlation between disconnected blocks, regardless of whether decaying covariances are slower or faster. It is beneficial to consider a one dimensional process $s \in \mathbb{R}$, and joint simulation at subsets $D_i = \{i\}$, $i = 1, \ldots, n$ using consecutive conditioning method. Suppose we define the first time point as given in a stochastic process, and then calculate next time points with the general consecutive conditioning method. As a result, we have $Z_1, Z_{CC,2}, Z_{CC,3}, \ldots, Z_{CC,n}$. Then it is easy to show that $\text{Cov}(Z_1, Z_{CC,n}) = C(1)^{n-1}$. Before proving the $\text{Cov}(Z_1, Z_{CC,n}) = C(1)^{n-1}$, we must first introduce notation. A covariance matrix, $\Sigma$, is a matrix whose element in the $i, j$ position is the covariance between the $i$th and $j$th elements of a random vector. Hence we define $\text{Cov}(Z_i, Z_j) = \Sigma_{ij}$.

Let $P(n)$ be the $n$th step in the induction proof, then we are trying to prove $P(n) = \text{Cov}(Z_1, Z_{CC,n}) = (\Sigma_{12} \Sigma_{22}^{-1})^{n-2} \Sigma_{n(n-1)n} = C(1)^{n-1}$ where we derive simple kriging predictor $\hat{Z}_{CC,n} = \Sigma_{n(n-1)} \Sigma_{(n-1)(n-1)}^{-1} Z_{CC,n-1}$ from the Section 2.1.1.

(1) Base case is when $n = 2$

$$P(2) = \text{Cov}(Z_1, Z_{CC,2}) = \Sigma_{12}$$

Since part $B = (Z_{u,2} - \hat{Z}_{u,2})$ does not affect to $Z_1$, we could remove $B$ from covariance
function.

\[
\text{Cov}(Z_1, Z_{CC,2}) = \text{Cov}(Z_1, \Sigma_{21} \Sigma_{11}^{-1} Z_1 + (Z_{u,2} - \hat{Z}_{u,2}))
\]

\[
= \text{Cov}(Z_1, \Sigma_{21} \Sigma_{11}^{-1} Z_1 + B)
\]

\[
= \text{Cov}(Z_1, Z_1)(\Sigma_{21} \Sigma_{11}^{-1})^T
\]

\[
= \Sigma_{11} \Sigma_{11}^{-1} \Sigma_{12}
\]

\[
= \Sigma_{12}
\]

Hence, \(P(2)\) is true.

(2) For any integer \(k \geq 2\)

\[
P(k) = \text{Cov}(Z_1, Z_{CC,k}) = (\Sigma_{12} \Sigma_{22}^{-1})^{(k-2)} \Sigma_{(k-1)k} = C(1)^{k-1}
\]

We want to prove \((k + 1)\) is also true.

\[
P(k + 1) = \text{Cov}(Z_1, Z_{CC,k+1})
\]

\[
= \text{Cov}(Z_1, \Sigma_{k+1} \Sigma_{kk}^{-1} Z_{CC,k} + (Z_{u,k+1} - \hat{Z}_{u,k+1}))
\]

\[
= \text{Cov}(Z_1, \Sigma_{k+1} \Sigma_{kk}^{-1} Z_{CC,k})
\]

\[
= \text{Cov}(Z_1, Z_{CC,k})(\Sigma_{(k+1)k} \Sigma_{kk}^{-1})^T
\]

\[
= \text{Cov}(Z_1, Z_{CC,k}) \Sigma_{kk}^{-1} \Sigma_{k(k+1)}
\]

\[
= P(k) \Sigma_{kk}^{-1} \Sigma_{k(k+1)}
\]

\[
= (\Sigma_{12} \Sigma_{22}^{-1})^{(k-2)} \Sigma_{k-1,k} \Sigma_{k(k+1)} \Sigma_{kk}^{-1}
\]

\[
= (\Sigma_{12} \Sigma_{22}^{-1})^{(k-1)} \Sigma_{k-1,k}
\]

\[
= (\Sigma_{12} \Sigma_{22}^{-1})^{(k-1)} \Sigma_{k,k+1}
\]

\[
= (\Sigma_{12} \Sigma_{22}^{-1}) \Sigma_{k,k+1}
\]

\[
= C(1) \Sigma_{k,k+1}
\]

Since we know the process is stationary, then we have \(\Sigma_{12} = \Sigma_{23} = \Sigma_{34} = \cdots = \Sigma_{(k-1)k} = \Sigma_{k(k+1)}\) and \(\Sigma_{11} = \Sigma_{22} = \Sigma_{33} = \cdots = \Sigma_{kk}\) which implies \(\Sigma_{11}^{-1} = \Sigma_{22}^{-1} = \cdots = \Sigma_{kk}^{-1} = \cdots = \Sigma_{12}^{-1}\).
\[ \Sigma_{33}^{-1} = \ldots = \Sigma_{kk}^{-1} \]

Such consecutive conditioning will have an exact prescription of the correlation between \( D_1 \) and \( D_n \).
In this chapter we introduce different simulations with uninitialized and initialized method, then apply our method of simulations to each different covariance model.

3.1 Kullback Leibler divergence

In mathematical statistics, the Kullback-Leibler(KL) divergence is a measure of how one probability distribution is different from another probability distribution \[|11\]. The KL divergence from \(Q\) to \(P\) is often denoted \(D_{KL}(P||Q)\).

\(D_{KL}(P||Q)\) is the amount of information lost when \(Q\) is used to approximate \(P\). In applications, \(P\) typically represents the "true" distribution of data, observations, or a precisely calculated theoretical distribution, while \(Q\) typically represents model, or approximation of \(P\) \[1\]. A KL divergence value 0 indicates that the two distributions are identical whereas 1 indicates that the two distributions diversify. The KL divergence helps us to measure just how much information we lose when we choose an approximation.

Although the KL divergence measures the "distance" between two distributions, it is not a distance measure. This is because the KL divergence is not a metric measure. Since it is not symmetric, the KL from \(P\) to \(Q\) is generally not the same as the KL from \(Q\) to \(P\). Moreover, it need not satisfy triangular inequality. Nevertheless, \(D_{KL}(P||Q)\) is a non-negative measure. \(D_{KL}(P||Q) \geq 0\) and \(D_{KL}(P||Q) = 0\) if and only if \(P = Q\).

Suppose, for example, we have two multivariate normal distributions, with means \(\mu_0, \mu_1\)
and with (non-singular) covariance matrices $\Sigma_0, \Sigma_1$. If the two distributions have the same dimension, $k$, then the KL divergence between the distributions is as follows:

$$D_{KL}(\mathcal{N}_0||\mathcal{N}_1) = \frac{1}{2} \left[ \text{tr}(\Sigma_1^{-1}\Sigma_0) + (\mu_1 - \mu_0)^T\Sigma_1^{-1}(\mu_1 - \mu_0) - k + \ln \left( \frac{\det \Sigma_1}{\det \Sigma_0} \right) \right]$$

The main purpose of using the KL divergence in our research is to validate our method, by ensuring that our simulated distribution is "close enough" to the true distribution.

### 3.2 Simulation at Two Time Points

As proposed in Chapter 2, the general consecutive conditioning method for any $N$ is $Z_{CC,N} = \hat{Z}_N + (Z_{u,N} - \hat{Z}_{u,N})$. In this simulation, we define the first time point as given in a stochastic process, and then calculate the next time point with the general consecutive conditioning method. As a result, we have $Z(1)$ and $Z_{CC}(N)$.

Since we have the stochastic process at 2 time points such as $Z(1)$ and $Z_{CC}(N)$, we could form these components into one implied bivariate normal distribution $(Z(1), Z_{CC}(N))^T \sim \mathcal{N}_1(\mu, \Sigma_1)$ where $(N \geq 2)$ and $N$ is an integer.

To investigate bias from a given implied normal distribution, we compare the implied normal distribution to the actual stochastic process with 2 time points for creating target bivariate normal distribution such as

$$(Z(1), Z(N))^T \sim \mathcal{N}_0(\mu, \Sigma_0) = \begin{pmatrix} 0 \\ \begin{pmatrix} 1 \\ C(N-1) \end{pmatrix} \end{pmatrix},$$

where $(N \geq 2)$ and $N$ is an integer.

Then we substitute these two bivariate normal distributions into the KL divergence formula for finding its values. And these values clearly indicate how the implied normal distribution is different from the actual normal distribution.
The formula for the KL divergence of two bivariate normal distributions is

\[
D_{KL}(N_0||N_1) = \frac{1}{2} \left[ \text{tr}(\Sigma_1^{-1}\Sigma_0) + (\mu_1 - \mu_0)^T \Sigma_1^{-1}(\mu_1 - \mu_0) - k + \ln \left( \frac{\det \Sigma_1}{\det \Sigma_0} \right) \right]
\]

(3.1)

\[
= \frac{1}{2} \left[ 2(1 - C(N - 1)C(1)^{N-1}) \right] - 2 + \ln \left( \frac{1 - C(1)^{2(N-1)}}{1 - C(N - 1)^2} \right)
\]

3.2.1 Matérn

In our research, we will focus on parametric spatial models. These spatial models are composed of mean and covariance functions that can be described with a few parameters. Matérn correlation function is defined by

\[
C(r) = \frac{2^{1-v}}{\Gamma(v)} \left( \frac{r}{a} \right)^v K_v \left( \frac{r}{a} \right), a > 0, v > 0
\]

where \( K_v \) is a modified Bessel function of the second kind of order \( v \).

In statistics, the Matérn covariance is a covariance function used in spatial statistics, geostatistics, machine learning, image analysis, and other applications of multivariate statistical analysis on metric spaces. It is commonly used to define the statistical covariance between measurements made at two points that are \( r \) units distant from each other. If the distance is Euclidean distance, the Matérn covariance is also isotropic.

We are focusing on two parameters in the Matérn covariance function to determine its covariance function’s value, \( a \) is the range parameter and \( v \) is the smoothness parameter. As we mentioned in the Section 3.2, these KL values are composed of two bivariate normal distributions. The covariance function is defined as Matérn based on which the implied covariance matrix with the simulation gets its values. And the same definition is made in the target covariance matrix to get its value. Since this Matérn covariance function’s value depends on its parameter smoothness and range, we fixed the range parameter as 1 and set the different smoothness parameter values for top figure. Additionally, we fixed the smoothness parameter as 1 and set the different range parameter values for bottom figure. Hence we have the different KL values that vary by the two parameters.

We can find at least two things in Figure 3.1. First, the KL divergence in the Matérn
Figure 3.1: Top: Showing how the Matérn covariance function works in KL value if the range parameter is fixed and the smoothness parameter varies between 10 and 100. Bottom: Showing how the Matérn covariance function works in KL value if the smoothness parameter is fixed and the range parameter varies between 10 and 100.

increases as the value of the smoothness parameter and that of the range parameter increase. Second, we noticed that if $N$ is large, that is to say, if the time point $N$ is large, then the CC is closer to true process since the KL divergence is close to 0. This clearly implies that our method tends to work better in cases of rough, short range processes and long time point $N$. 
3.2.2 Squared exponential

Squared exponential correlation function is defined by

\[ C(r) = \exp \left( - \left( \frac{r}{\alpha} \right)^2 \right), \alpha > 0 \]

The squared exponential(SE) covariance function is infinitely differentiable, which means that the Gaussian process with this covariance function has mean square derivatives of all orders, and is thus very smooth. Stein argues that such strong smoothness assumptions are unrealistic for modelling many physical processes, and recommends the Matèrn covariance\[19\]. However, the squared exponential is probably the most widely-used kernel within the kernel machines field.

There is only one parameter in the squared exponential covariance function to determine its covariance function's value, \( \alpha \) is the range parameter. As we mentioned in the Section 3.2, these KL values are composed of two bivariate normal distributions. The co-
variance function is defined as SE based on which the implied covariance matrix with the simulation gets its values. And the same definition is made in the target covariance matrix to get its value. Since the SE covariance function’s value depends on its parameter range, we set the $N$ value as x-axis and the KL divergence value as y-axis. Hence we have different KL values that vary by only one parameter.

We can find at least two things in Figure 3.2. First, the KL divergence in the SE increases as the value of the range parameter increases. Second, we noticed that if $N$ is large, i.e., if the time point $N$ is large, then the CC is closer to true process since the KL divergence is close to 0. This clearly implies that our method tends to work well in cases of short range processes and long $N$ time point such as greater than 20.

3.2.3 Cauchy

Cauchy correlation function is defined by

$$C(r) = \left(1 + \left(\frac{r}{a}\right)^{\alpha}\right)^{-\frac{\beta}{\alpha}}, \quad a > 0, \beta > 0, 0 < \alpha \leq 2$$

There are three parameters in the Cauchy covariance function to determine its covariance function’s value, $a$ is the range parameter, $\beta$ is the long-range dependence parameter and $\alpha$ is the smoothness parameter. In the Cauchy model, long-range dependence (LRD) is a phenomenon that may arise in the analysis of spatial or time series data. It relates to the rate of decay of statistical dependence of two points with increasing time interval or spatial distance between points. A phenomenon is usually considered to have LRD if the dependence decays more slowly than an exponential decay.

Since we want to investigate how the LRD parameter functioning with the KL divergence value, we could fix $a$ as some constant 1 because $a$ is just functioning as scaling its distance and $\alpha$ is restricted to range 0 to 2. Then we investigate $\beta$ range separately as 0 to 1 and 1 to 4.

As we mentioned in the Section 3.2, these KL values are composed of two bivariate
normal distributions. The covariance function is defined as Cauchy based on which the implied covariance matrix with the simulation gets its values. And the same definition is made in the target covariance matrix to get its value. Since this Cauchy covariance function’s value depends on its parameter smoothness and LRD, we fixed the smoothness parameter as 0.5 and set the different LRD parameter values for the above figure. Hence we have the different KL values that vary by the LRD parameter.

We focus more on the long-range dependence parameter than the other two parameters. In Figure 3.3, we find that the highest KL divergence values exist on a very low long-range dependence value such as 1, and that KL divergence rapidly decreases as the value of the $N$ increases. This implies that our method works better in cases of higher the long-range dependence processes and the larger $N$ point.
3.3 Uninitialized Simulation

As proposed in Chapter 2, the general consecutive conditioning method for any $N$ is $Z_{CC,N} = \hat{Z}_N + (Z_{u,N} - \hat{Z}_{u,N})$. In the uninitialized simulation, we define the first time point as given in a stochastic process, and then calculate next time points with the general consecutive conditioning method. As a result, we have $Z(1), Z_{CC}(2), Z_{CC}(3), \cdots, Z_{CC}(N)$. In our case, we explore the utility of the method on a small vector where we can explicitly calculate implied covariance relationships. This can become difficult when $N$ is very large as the resulting covariance matrix will have $N^2$ elements. This is why we generate 5 time points resulting in 25 elements in our covariance matrix.

In this section, we will show the difference between the uninitialized case and the initialized case of using the consecutive conditioning. Uninitialization is the process by which all unspecified solution field values are not assigned at the beginning of a simulation. In our research, the uninitialized simulation is composed of one real-valued stochastic process $Z(1)$ and its several consecutive conditioning stochastic process $Z_{CC}(2), Z_{CC}(3), Z_{CC}(4)$, and $Z_{CC}(5)$.

Since we have samples of the stochastic process at 5 time points such as $Z(1), Z_{CC}(2), Z_{CC}(3), Z_{CC}(4)$, and $Z_{CC}(5)$, we could form these components into one implied normal distribution

$$(Z(1), Z_{CC}(3), Z_{CC}(5), Z_{CC}(2), Z_{CC}(4))^T \sim \mathcal{N}_1(0, \Sigma_1)$$

We ordered this way because this order lets us compute the implied covariance matrix more easily when we do initialization in the next section. To investigate bias from a given implied normal distribution, we compare the implied normal distribution to the actual stochastic process with 5 time points for creating target normal distribution such as

$$(Z(1), Z(3), Z(5), Z(2), Z(4))^T \sim \mathcal{N}_0(0, \Sigma_0)$$

Then we substitute these two multivariate normal distributions into the KL divergence formula for finding its values. And these values clearly indicate how the implied normal
distribution is different than the actual normal distribution.

The formula for Kullback-Leibler divergence of two multivariate normal distributions is

\[
D_{KL}(\mathcal{N}_0||\mathcal{N}_1) = \frac{1}{2} \left[ \text{tr}(\Sigma_1^{-1}\Sigma_0) + (\mu_1 - \mu_0)^T \Sigma_1^{-1}(\mu_1 - \mu_0) - k + \ln\left(\frac{\det\Sigma_1}{\det\Sigma_0}\right) \right]
\]

\[
= \frac{1}{2} \left[ \text{tr}(\Sigma_1^{-1}\Sigma_0) - 5 + \ln\left(\frac{\det\Sigma_1}{\det\Sigma_0}\right) \right]
\]

where we assume \(\mu_0 = \mu_1 = 0\).

To calculate the above the formula, we generate the target covariance matrix and the implied covariance matrix using the CC to find the KL divergence values.

The target covariance matrix is:

\[
\begin{bmatrix}
Z(1) \\
Z(3) \\
Z(5) \\
Z(2) \\
Z(4)
\end{bmatrix}
= \begin{bmatrix}
1 & C(2) & C(4) & C(1) & C(3) \\
C(2) & 1 & C(2) & C(1) & C(1) \\
C(4) & C(2) & 1 & C(3) & C(1) \\
C(1) & C(1) & C(3) & 1 & C(2) \\
C(3) & C(1) & C(1) & C(2) & 1
\end{bmatrix}
= \Sigma_0
\]

The implied covariance matrix (Uninitialized) is:

\[
\begin{bmatrix}
Z(1) \\
Z_{CC}(3) \\
Z_{CC}(5) \\
Z_{CC}(2) \\
Z_{CC}(4)
\end{bmatrix}
= \begin{bmatrix}
1 & C(1)^2 & C(1)^4 & C(1) & C(1)^3 \\
C(1)^2 & 1 & C(1)^2 & C(1) & C(1) \\
C(1)^4 & C(1)^2 & 1 & C(1)^3 & C(1) \\
C(1) & C(1) & C(1)^3 & 1 & C(1)^2 \\
C(1)^3 & C(1) & C(1) & C(1)^2 & 1
\end{bmatrix}
= \Sigma_1
\]
Figure 3.4: \((v, a)\) implies the KL divergence value by the range and the smoothness parameter when \(a\) varies between \([0, 10]\) and \(v\) varies between \([0, 10]\).

3.3.1 Matèrn

The KL values in Figure 3.4 are expressed in terms of different colors. For instance, the dark blue area refers to 0 while the deep red area refers to 20, as is found in the vertical bar on the right side of the figure. As we mentioned in the Section 3.3, these KL values are composed of two multivariate normal distributions. Here, the covariance function is defined as Matèrn based on which the implied covariance matrix with the uninitialized simulation gets its values. And the same definition is made in the target covariance matrix to get its value. Since this Matèrn covariance function’s value depends on its parameter smoothness and range, we set the smoothness parameter as x-axis and the range parameter as y-axis. Hence we have the different KL values that vary by the two parameters.

We can find at least two things in Figure 3.4. First, the KL divergence in the Matèrn increases as the value of the smoothness parameter and that of the range parameter increase. Second, the KL divergence value is an inverse exponential type relationship between the
smoothness and the range parameters determined by its gradation form. This simulation, however, does not work well especially when the range value is higher than 10. This clearly implies that our method tends to work better in cases of rough and short range processes, i.e., cases where the values of the smoothness and the range parameter are lower than 1.

### 3.3.2 Squared exponential

Figure 3.5 shows the relationship between the KL values and the range parameter that comes from the SE covariance function. As we mentioned in the Section 3.3, these KL values
are composed of two multivariate normal distributions. The covariance function is defined as the SE based on which the implied covariance matrix with the uninitialized simulation gets its values. And the same definition applies to the target covariance matrix to get its value. Since the SE covariance function’s value depends on its parameter range, we set the range parameter as x-axis and the KL divergence value as y-axis. Hence we have the different KL values that vary by only one parameter.

Again, we can find two results from Figure 3.5. First, the KL divergence in SE increases as the value of the range parameter increases. For instance, the range value between 0 and 1 clearly shows that it is close to 2 the KL divergence value. By contrast, secondly, the range value above 1 contributes to increasing the KL divergence value: as the range value increases, the KL divergence value increases rapidly until 20, but the speed of increase becomes slower once the range value increases over 20. This implies that our method tends to work well in cases of short range process where the range parameter is between 0 and 1.

### 3.3.3 Cauchy

Since we want to investigate how the LRD parameter functioning with the KL divergence value, we could fix $a$ as some constant 0.5 because $a$ is just functioning as scaling its distance and $\alpha$ is restricted to range 0 to 2. The KL values in Figure 3.6 are expressed in terms of different colors. For instance, the dark blue area refers to 0 while the deep red area refers to 2.4, as is shown by the vertical bar on right side of the figure. As we mentioned in the Section 3.3, these KL values are composed of two multivariate normal distributions. The covariance function is defined as Cauchy based on which the implied covariance matrix with the uninitialized simulation get its values. And the same definition is made in the target covariance matrix to get its value. Since this Cauchy covariance function’s value depends on its parameter smoothness and LRD, we set the smoothness parameter as x-axis and the LRD parameter as y-axis. Hence we have the different KL values that vary by two parameters.

As we can see from Figure 3.6, the KL divergence values in Cauchy are mostly lower
than those in the two previous covariance models that we have discussed above. Since we simulate our method based on the Cauchy covariance model, we focus on the long-range dependence parameter than the other two parameters. In Figure 3.6, we find that the highest KL divergence values exist on a very low long-range dependence value such as near 0, and that KL divergence slowly decreases as the value of the smoothness parameter increases until its value reaches 1. This implies that our method works better in cases of higher the long-range dependence processes where the value of long-range dependence is higher than 1 in the Cauchy.

3.4 Initialized Simulation

As is proved in Chapter 2, the general consecutive conditioning method for any \( N \) is \( Z_{CC,N} = \hat{Z}_N + (Z_{u,N} - \tilde{Z}_{u,N}) \). In the initialized simulation, we define every other time point as given in a stochastic process, and then calculate empty or even time points with the general
consecutive conditioning method. As a result, we have $Z(1), Z_{CC}(2), Z(3), Z_{CC}(2) \cdots, Z(N)$ if $N$ is natural odd number. In our case, we explore the utility of the method on a small vector where we can explicitly calculate implied covariance relationships. This can become difficult when $N$ is very large as the resulting covariance matrix will have $N$ elements. This is why we generate 5 time points resulting in 25 elements in our covariance matrix.

To overcome bias from the uninitialized case from the previous section, we propose a simple approach such as initialization. Initialization is the process by which not all specified solution field values are assigned at the beginning of a simulation. We refer to these values as initial values. In our research, the initialized simulation is composed of three time points in a real-valued stochastic process, namely, $Z(1), Z(3), Z(5)$, and the two time points that are calculated by the consecutive conditioning. Thus, $Z(1), Z(3)$, and $Z(5)$ are initial values in this simulation approach. As we mentioned in Chapter 2, we generate $Z_{CC}(2)$ by the conditional simulation conditioned only on $Z(1)$ and $Z(3)$. Likewise we generate $Z_{CC}(4)$ by the conditional simulation conditioned only on $Z(3)$ and $Z(5)$. We could expect from these that there would be substantial reduction in bias by using the initialization.

Similarly, we have samples of the stochastic process at 5 time points such that $Z(1), Z_{CC}(2), Z(3), Z_{CC}(4)$, and $Z(5)$, then we form these components into one implied normal distribution

$$(Z(1), Z(3), Z(5), Z_{CC}(2), Z_{CC}(4))^T \sim \mathcal{N}_1(0, \Sigma_1)$$

To investigate bias from a given implied normal distribution, we generate actual stochastic process with 5 time points for creating target normal distribution such as

$$(Z(1), Z(3), Z(5), Z(2), Z(4))^T \sim \mathcal{N}_0(0, \Sigma_0)$$

Then we substitute these two multivariate normal distributions into the KL divergence formula for finding its values. And these values clearly indicate how the implied normal distribution is different than the actual normal distribution. We should generate target
covariance matrix and implied covariance matrix for finding the KL divergence values. The
target covariance matrix is defined in previous section.

The implied covariance matrix (Initialized) is:

\[
\begin{bmatrix}
Z(1) \\
Z(3) \\
Z(5) \\
Z_{cc}(2) \\
Z_{cc}(4)
\end{bmatrix} = \\
\begin{bmatrix}
1 & C(2) & C(4) & C(1) & \frac{C(1)}{1+C(2)}(C(2) + C(4)) \\
C(2) & 1 & C(2) & C(1) & C(1) \\
C(4) & C(2) & 1 & \frac{C(1)}{1+C(2)}(C(2) + C(4)) & C(1) \\
C(1) & C(1) & \frac{C(1)}{1+C(2)}(C(2) + C(4)) & 1 & (\frac{C(1)}{1+C(2)})^2(1 + 2C(2) + C(4)) \\
\frac{C(1)}{1+C(2)}(C(2) + C(4)) & C(1) & C(1) & (\frac{C(1)}{1+C(2)})^2(1 + 2C(2) + C(4)) & 1
\end{bmatrix}
\]

which we denote \( \Sigma_1 \).
3.4.1 Matérn

Figure 3.7: \((v, a)\) implies the KL divergence value by the range and the smoothness parameter when \(a\) varies between \([0, 10]\) and \(v\) varies between \([0, 10]\).

Similar to Figure 3.4, we can find at least two things in Figure 3.7. First, KL divergence in Matérn increases as the value of the smoothness parameter and that of the range parameter increase. Second, KL value is an inverse exponential type relationship between the smoothness and the range parameters determined by its gradation form. This simulation, however, does not work well especially when the range value is higher than 10. The difference between Figure 3.4 and Figure 3.7 at the same condition is that the overall KL value is smaller. This clearly implies that our method tends to work better in cases of rough and short range processes, i.e., cases where the values of the smoothness and the range
Figure 3.8: Top: Showing how the SE covariance function works in the KL divergence value if the range parameter $a$ varies between [0, 5]. Bottom: Showing how the SE covariance function works in the KL divergence value if the range parameter $a$ is [0, 100].

parameter are lower than 1.

### 3.4.2 Squared exponential

Similar to Figure 3.5, we can find two results from Figure 3.8. First, KL divergence in SE increases as the value of the range parameter increases. For instance, the range value between 0 and 1 clearly shows that it is close to 2 KL divergence value. By contrast, secondly, the range value above 1 contributes to increasing KL divergence value: as the range value increases, the KL divergence value increases rapidly until 20, but the speed of increase
Figure 3.9: \((\alpha, \beta)\) implies the KL divergence value by the smoothness and the long-range dependence parameter when the range is fixed as 0.5.

becomes slower once the range value increases over 20. The difference between Figure 3.5 and Figure 3.8 at the same condition is that the overall KL value is smaller. This implies that our method tends to work well in cases of short range process where the range parameter is between 0 and 1.

3.4.3 Cauchy

Similar to Figure 3.6, we can see KL divergence values in Cauchy are mostly lower than those in the two previous covariance models that we have discussed above. Since we simulate our method based on the Cauchy covariance model, we focus on long-range dependence parameter than the other two parameters. In Figure 3.9, we find that the highest KL divergence values exist on a very low long-range dependence value such as near 0, and that KL divergence slowly decreases as the value of the smoothness parameter increases until its value reaches 1. The difference between Figure 3.6 and Figure 3.9 at the same condition
is that the overall KL value is smaller. This implies that our method works better in cases of higher long-range dependence processes where the value of long-range dependence is higher than 1 in Cauchy.

3.5 Improvement

In this section, we compare the Figures from the Section 3.2 to the Section 3.3. We know that, other things being equal, one simulation is better than the other if the KL value that it produces is smaller than the other. Hence, initialized simulation is a better approach than uninitialized simulation.

We can easily recognize that the overall KL value decreases from the Section 3.2 figures to the Section 3.3 figures. However, it is difficult to evaluate how much it is improved from the Section 3.2 figures to the Section 3.3 figures because we do not know how bad KL value is if it is greater than 1. For we only know that KL value 1 indicates that the two distributions behave in such a different manner. For a more precise measurement of improvement, here, we will use the following equation:

$$1 - \frac{\text{KL divergence value(Initialized)}}{\text{KL divergence value(Uninitialized)}}$$

According to this formula, the more the result of calculation is close to 1 or 100 percent, the more there is huge improvement on initialized simulation compared to uninitialized simulation. By contrast, if the result of calculation is close to 0, it means that there is almost no improvement made by initialized simulation.

3.5.1 Matérn

We can find at least two things in Figure 3.10. First, percentage of improvement reach up to almost 90 between the smoothness value around 2 and the range value greater than 6. Second, percentage remains steady around 50 percent when the smoothness value is greater
Figure 3.10: \((v, a)\) implies percentage by the range and the smoothness parameter when \(a\) varies between \([0, 10]\) and \(v\) varies between \([0, 10]\).

than 4. This clearly implies that initialized simulation tends to work better than uninitialized simulation in cases of rough and long range processes.

3.5.2 Squared exponential

We can find two results from Figure 3.11. First, percentage of improvement decreases from 65 to 45 when the range value is between 0.4 and 3. Secondly, percentage slightly increases from 45 to 50 when the range value is greater than 3. This implies that initialized simulation tends to work better than uninitialized simulation in cases of short range processes where the range parameter is around 0.5.

3.5.3 Cauchy

In Figure 3.12, we see that the cauchy covariance has complicated improvement compared to other covariance functions. And we can find that percentage of improvement is high
Figure 3.11: Top: Showing percent improvement if the range parameter $a$ varies between $[0, 5]$. Bottom: Showing percent improvement if the range parameter $a$ varies between $[0, 100]$.

when LRD value is under 1. And percentage has decreased when smoothness value is between 0 and 1. This implies that initialized simulation tends to work better than uninitialized simulation when LRD process is restricted to interval $[0, 1]$. 
Figure 3.12: \((\alpha, \beta)\) implies percent improvement of the KL by the smoothness and the LRD parameter when the range is fixed as 0.5.
Chapter 4

Conclusion

As we have seen, simulation of Gaussian process is very important especially in expanding era of large spatial datasets. Since most physical processes are not stationary, many traditional methods of simulating stationary processes are not applicable. In this thesis, we proposed a new algorithm, which we call consecutive conditioning, that allows us to generate approximate realizations from a nonstationary process. The consecutive conditioning method heavily relies on conditional simulation and kriging. In Chapter 3 we apply our method in several different situations to determine where it works best. Using the KL divergence as our metric comparison, we applied the consecutive conditioning with several different covariance functions and analyzed the results. After simulating two time points from a stochastic process, we discovered that the consecutive conditioning works well with the SE covariance function and does not work well with the Matèrn covariance function. Next, we applied initialization to the previous analysis and compared the results from using different covariance functions. We discovered that the SE and Matèrn covariance cases improves drastically with initialization, while the Cauchy covariance only improves at certain interval of each parameter. In validation, we generated an implied normal distribution based on uninitialized simulation. We explored the utility of the method on a small vector where we could explicitly calculate implied covariance relationships. To investigate bias from a given implied normal distribution, we compared the implied normal distribution to the target normal distribution by using the KL divergence. To overcome bias from the previous
approach, we generated an implied normal distribution based on initialized simulation. In this case, not all specified solution field values were assigned at the beginning of a simulation. We referred to these values as initial values such as time point 1, 3 and 5. We performed the same analysis as before, using the KL divergence to measure the simulation outcome using three different covariance functions. The results showed that the Cauchy covariance function works better than the other covariance functions on both simulations approaches. Finally, we applied initialization to the previous method and evaluated the improvement in our simulation results by comparing the KL values from both simulations.

We suggest that future research should focus on modifying the previous initialized simulation for the purpose of minimizing the KL divergence value. For example, we could broaden the gap between time points given in a stochastic process e.g. time points 1, 6, 10, ···. The next step would be to form an implied multivariate normal distribution for comparison to the target normal distribution, which we suspect could improve the KL divergence values we saw earlier. Finally, we recognize that there is much research to be done on the consecutive conditioning in higher dimensions, and we leave this as a suggestion for future researchers.
Bibliography


