The Turning Bands Method

Péricles Lopes Machado

Universidade Federal do Rio Grande do Sul

eu@gogo40.com

March 16, 2016

Overview

Introduction

- Turning Bands is member of a family of algorithms to reduce the simulation of a Gaussian random function to the simulation of independent random functions called *Spectral Methods*.
- Turning Bands is an algorithm to transform the simulation of a Gaussian random function with covariance C in M simulations of independent stochastic processes with covariances C_θ (where θ is a direction and C_θ is called *Spectral representation of the covariance* C).
- To generate the simulations, M lines are created in different directions in a standard unit sphere in $\mathbb{R}^d$, 1-D simulations are computed in each directions and the results are linearly combined to generate the simulation.

Introduction

- Each covariance type is simulated using a specific algorithm (for example, spherical covariances are simulated using the dilution method). [Emery, 2006]
- To simulate covariances that are result of a linear combination of basic models (spherical, exponential, Gaussian, nugget, etc...), we simulate independently each model and the results are linearly combined.
- Because the 1-D simulation in each node is completely independent, we can easily parallelize the simulation in space without any synchronization technique.

Gaussian variable properties

Many spectral methods are based on these four results from probability [Lantuéjoul, 2002]:

Definition of a Gaussian random function

A random function is said to be Gaussian if any linear combination of its variables follows a Gaussian distribution.

Characterization of the spatial distribution of a Gaussian random function

The spatial distribution of a Gaussian random function is totally characterized by its mean value and its covariance function.

Gaussian variable properties

Independent Gaussian vectors

Two Gaussian vectors are independent if and only if they are uncorrelated.

Linear combination of I.I.D random variables

Let $Y_1, ..., Y_n, ...$ be a sequence of independent and identically distributed (I.I.D.) random variables. If their mean m is finite, then the strong law of large numbers says that the average $(Y_1 + ... + Y_n)/n$ converges almost surely towards m.

$$\lim_{n \to +\infty} \frac{Y_1 + ... + Y_n}{n} = m \tag{1}$$

Gaussian variable properties

Linear combination of I.I.D random variables

In the case when their variance σ^2 is also finite, the central limit theorem states that the distribution of $(Y_1 + \ldots + Y_n)/n$ tends to be Gaussian around its mean value.

$$\lim_{n \to +\infty} P \left\{ \frac{\frac{Y_1 + \ldots + Y_n}{n} - m}{\frac{\sigma}{\sqrt{n}}} < y \right\} = G(y) \tag{2}$$

where G denotes the standard Gaussian distribution function.

Essentially, these results tell us that the simulation of ANY Gaussian random function can be reduced to the simulation of a sequence of independent and identically distributed (I.I.D.) random variables.
These results are interesting, but they don't answer us some important questions:

- How to generate the sequence of i.i.d. random variables?
- How to simulate a random function with covariance function C_θ?
- How many simulations of stochastic processes are needed for the method to converge to the expected Gaussian distribution?

Some algorithms has been developed to answer the previous questions like Spectral method, dilution method, tesselation method and the turning bands method.

Remarks and Definitions

- The Turning Bands algorithm has been designed by Matheron [Matheron, 1973].
- How has been said previously, the turning bands is a technique to reduce the simulation of a Gaussian random function with covariance C to the simulations of independent stochastic processes with covariance C_θ, where θ is the direction parameter.
- The Turning Bands method can be seen as a generalization of the spectral method [Lantuéjoul, 2002]. The only difference between Turning Band and spectral method is that the spectral method uses exclusively cosine functions to build the spectral representation of the covariances.

Remarks and Definitions

- The relationship between the simulation of independent stochastic processes and the simulation of the original Gaussian random function is expressed by

$$Y^{(n)}(x) = \frac{1}{\sqrt{n}} \sum_{k=1}^{n} X_k(< x, \theta_k >), x \in \mathbb{R}^d \tag{3}$$

where $(\theta_n, n \in \mathbb{N})$ is a sequence of directions of a unit sphere S_d and $(X_n, n \in \mathbb{N})$ is a sequence of independent stochastic processes with covariance C_{θ_n}. $\rho_n =< x, \theta_n >$ is the projection of the vector x in direction θ_n and $X_n(\rho_n)$ is the spectral representation of the Gaussian random function $Y^{(n)}(x)$ in the direction θ_n.

Remarks and Definitions

- The covariance $C^{(n)}(h)$ of $Y^{(n)}(x)$ is described by

$$C^{(n)}(h) = \frac{1}{n}\sum_{k=1}^{n} C_{\theta_k}(< h, \theta_k >) \tag{4}$$

where $C_{\theta_n}(< h, \theta_n >)$ is the spectral representation of the covariance $C^{(n)}(h)$.

- If the covariance C is isotropic, i.e., can be written as $C(h) = C_d(|h|)$ for some scalar function C_d defined on $\mathbb{R}^+$, then the relationship between C_1 and C_d is given by

$$C_d(r) = 2\frac{(d-1)\omega_{d-1}}{d\omega_d}\int_0^1 (1-t^2)^{(\frac{d-3}{2})} C_1(tr)dt \tag{5}$$

where ω_d stands, as usual, the d-volume of the unit ball in $\mathbb{R}^d$. [Lantuéjoul, 2002]

Remarks and Definitions

- If $d = 3$, the equation (5) is reduced to

$$C_3(r) = \int_0^1 C_1(tr)dt \tag{6}$$

or equivalently

$$C_1(r) = \frac{d}{dr} rC_3(r) \tag{7}$$

Curiously, for $d = 2$ the relationship is more complicated

$$C_2(r) = \frac{1}{\pi} \int_0^\pi C_1(rsin(\theta))d\theta \tag{8}$$

and

$$C_1(r) = 1 + r \int_0^{\pi/2} \frac{d}{dr} C_2(rsin\theta)d\theta \tag{9}$$

The algorithm

Keeping in mind the previous results, the Turning Bands algorithm can be written as [Lantuéjoul, 2002]:

The Turning Bands Algorithm

1. Generate a set of directions $\theta_1, ..., \theta_n$ such that $\frac{1}{n}\sum_{k=1}^{n}\delta_{\theta_k} \approx \varpi$.
2. Generate independent standard stochastic processes $X_1, ..., X_n$ with covariances functions $C_{\theta_1}, ..., C_{\theta_n}$.
3. Compute $\frac{1}{\sqrt{n}}\sum_{k=1}^{n} X_k(< x, \theta_k >)$ for any $x \in D$.

In this algorithm, D is the simulation domain, δ_{θ_n} is a distribution where $\sum_{k=1}^{n}\delta_{\theta_k}$ converges weakly to ϖ, the uniform distribution over S_d (unit sphere in $\mathbb{R}^d$).

The algorithm

It's very important to notice that the Turning Bands Algorithm does not determine a method to generate the directions, or how to build the stochastic processes with the covariance C_θ! This algorithm is only a very high level description of the processing steps to generate a simulation. So, to implement this algorithm is needed to solve a set of problems related with each step of the algorithm. These problems are:

1. How to generate a set of directions satisfying $\frac{1}{n}\sum_{k=1}^{n}\delta_{\theta_k} \approx \varpi$?
2. How to simulate a stochastic process satisfying C_{θ_n}?
3. How to execute the data conditioning?

Generating the directions

There are three main ways to generate directions θ_n satisfying the TB condition:

- Arrange θ_n regularly on S_d. This approach is efficient in $d = 2$, but for dimensions greater than 2 this solution doesn't work well. In $d = 3$, for example, the maximum number of regular directions is equal to 15 [Lantuéjoul, 2002].
- Generate θ_n using a uniform distribution in S_d works. But, using this method, the convergence of $\frac{1}{n}\sum_{k=1}^{n}\delta_{\theta_k}$ is very slow, i.e., we need many lines to generate good results.

Generating the directions

- Use a sequence with weak discrepancy. For $d = 3$, [Freulon, 1994] propose the sequence:

$$u_n = \frac{a_0}{2} + \frac{a_1}{4} + ... + \frac{a_p}{2^{p+1}}$$
$$v_n = \frac{b_0}{3} + \frac{b_1}{9} + ... + \frac{b_p}{3^{q+1}} \qquad (10)$$
$$\theta_n = \left(\cos(2\pi u_n)\sqrt{1-v_n^2}, \sin(2\pi u_n)\sqrt{1-v_n^2}, v_n\right)$$

where $a_i = 0, 1$ and $b_j = 0, 1, 2$ and $n = a_p...a_2 a_1 a_0 = b_q...b_2 b_1 b_0 = a_0 + 2a_1 + ... + 2^p a_p = b_0 + 3b_1 + ...3^q b_q$. In other words, a_i and b_j are the digits of the binary and ternary representation of n, respectively.

Generating the directions

The Freulon's algorithm produces θ_n as far as possible from $\theta_1, ..., \theta_{n-1}$ and fills the sphere S_d as fast as possible. How we can see in the Figs. 1, 2, 3, using 1000 bands is possible to generate a dense set of points filling the sphere S_d.
In the practice, the Freulon's algorithm assure a fast convergence to the TB algorithm. Generally, to large datasets, using 1000 or 2000 directions it is possible to generate good results. To generate other configurations of bands, we can rotate the bands using a random rotation.

Generating the directions

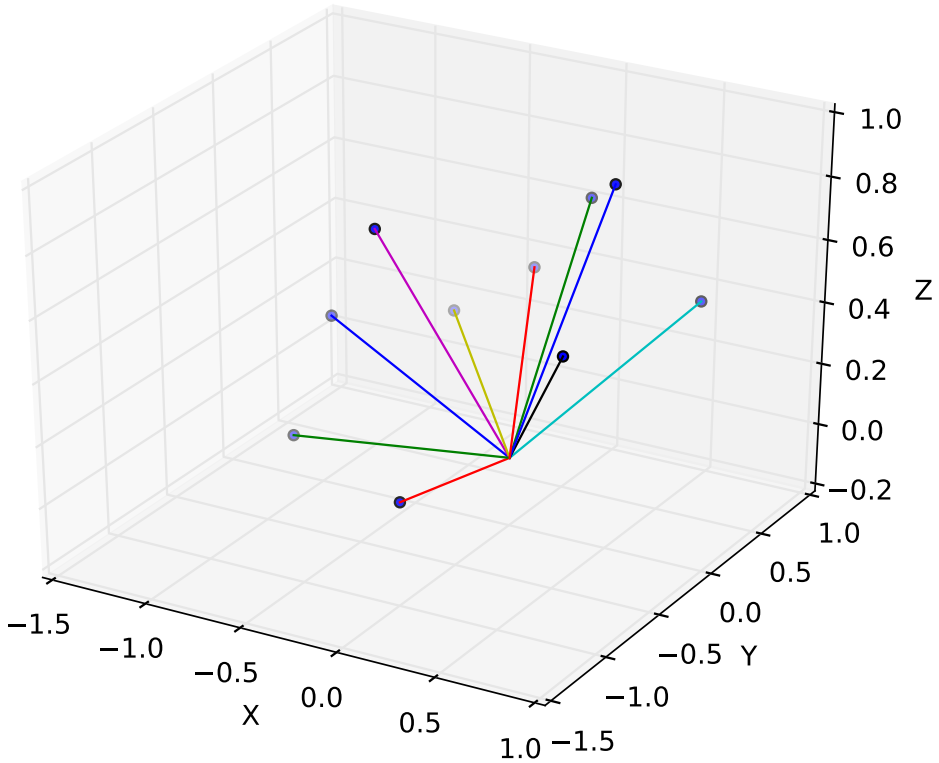

Figure: 10 bands generated using the Freulon's algorithm

Generating the directions

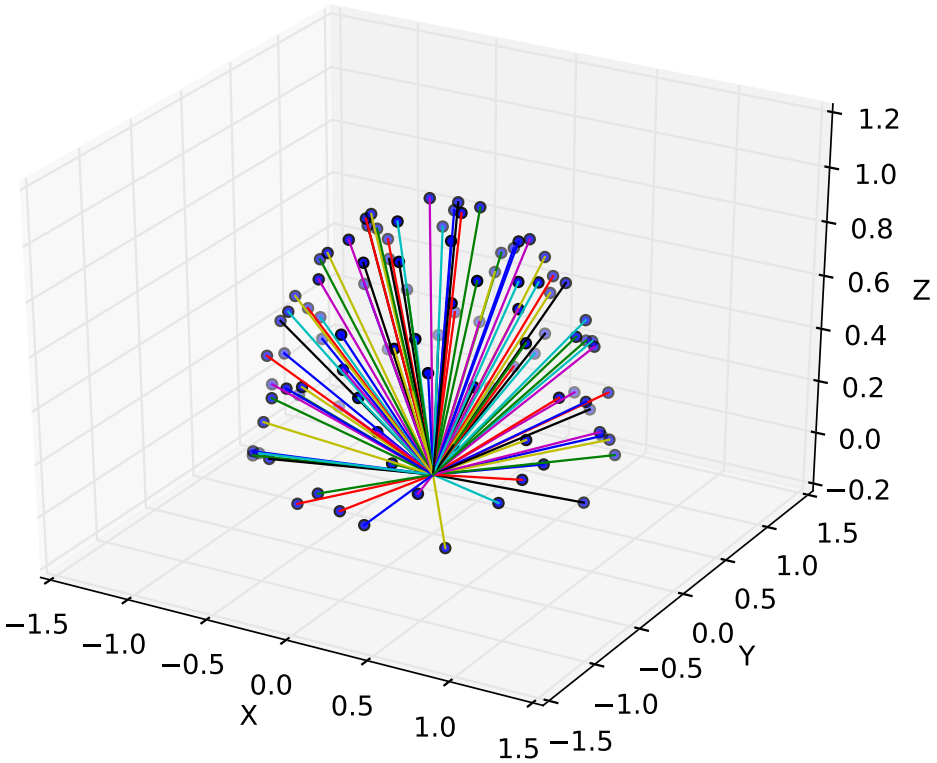

Figure: 100 bands generated using the Freulon's algorithm

Generating the directions

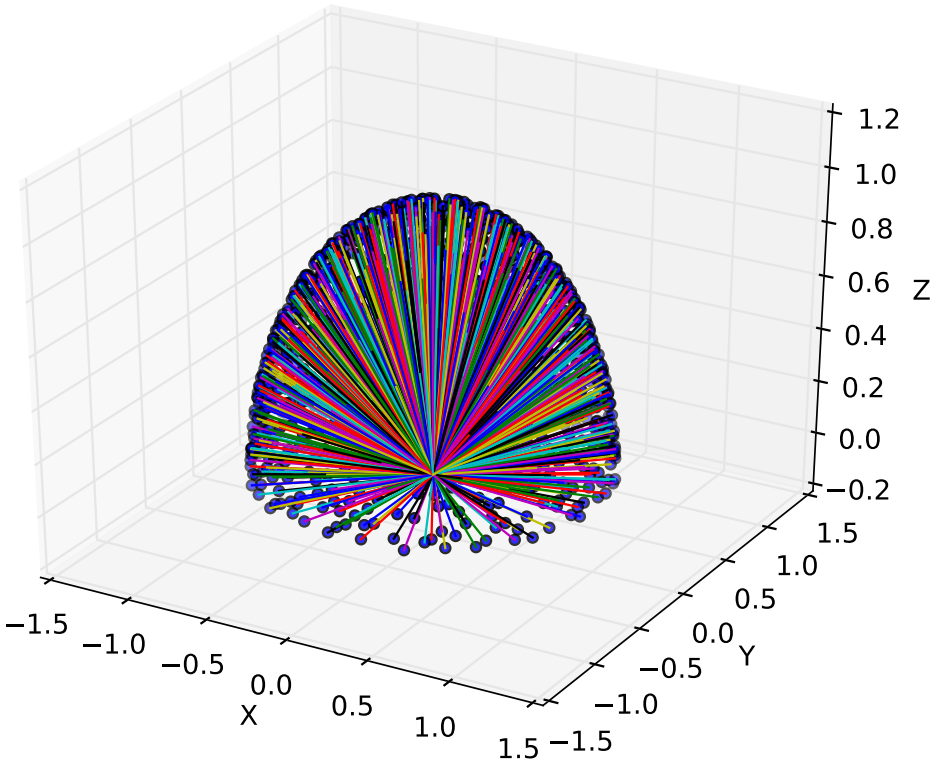

Figure: 1000 bands generated using the Freulon's algorithm

Generating the unidimensional simulations

Now, we have a solution to generate the bands to fill the unit sphere in $\mathbb{R}^3$. But, we have other problem: how to generate in each band a stochastic process with covariance function C_1?

Generating the unidimensional simulations

The Turning Bands algorithm doesn't determine an algorithm to generate the independent stochastic process in each band. Hence, in practice, for each model and band we generate independently the stochastic process using the appropriated algorithm to the specific model. After the execution of all independent simulations in a direction θ_n, the results are linearly combined to generate the final result in this direction. [Emery, 2006] presents 15 algorithms to generate 15 different types of Covariance models. In this presentation, we will present just the 3 models: nugget, spherical and the Gaussian.

Generating the nugget covariance

To generate a stochastic process with a covariance that is the spectral representation of the nugget covariance is enough to generate a random number with Gaussian distribution $N(0, C)$ with mean 0 and variance C.

Generating the spherical covariance

The algorithm to generate the stochastic process associated to the *Spherical covariance*

$$C_y(r) = C(1 - \frac{3r}{2a} + \frac{r^3}{2a^3})1_{0 \leq r \leq a} \tag{11}$$

with sill C and range (scale factor) a is

Algorithm to generate a stochastic process with spherical covariance

1. choose an offset on the real axis at random (uniformly in $[0, a[$) and divide this axis in intervals with length a.
2. within each interval, draw a linear function with an equal probability of being increasing or decreasing; the slope sign is independent from one interval to another one.
3. the slope value is $2\sqrt{\frac{3C}{n}}$, where n is the number of bands.

Generating the spherical covariance

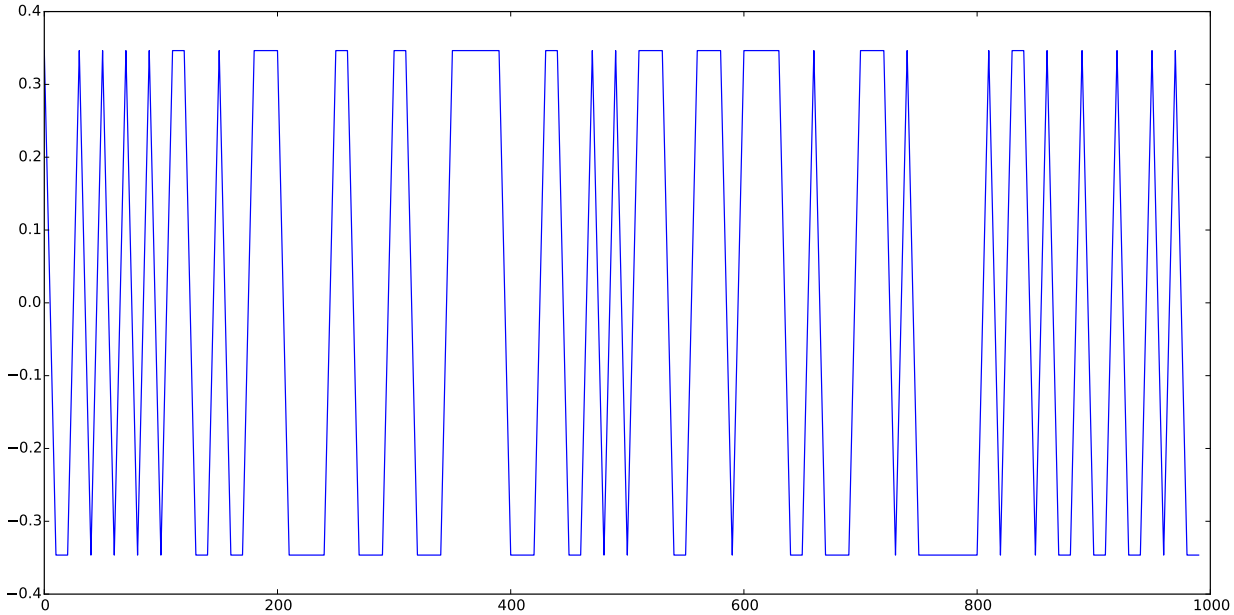

Figure: Simulation of the stochastic process associated to the *Spherical covariance* with $C = 10$, $a = 10$

Generating the spherical covariance

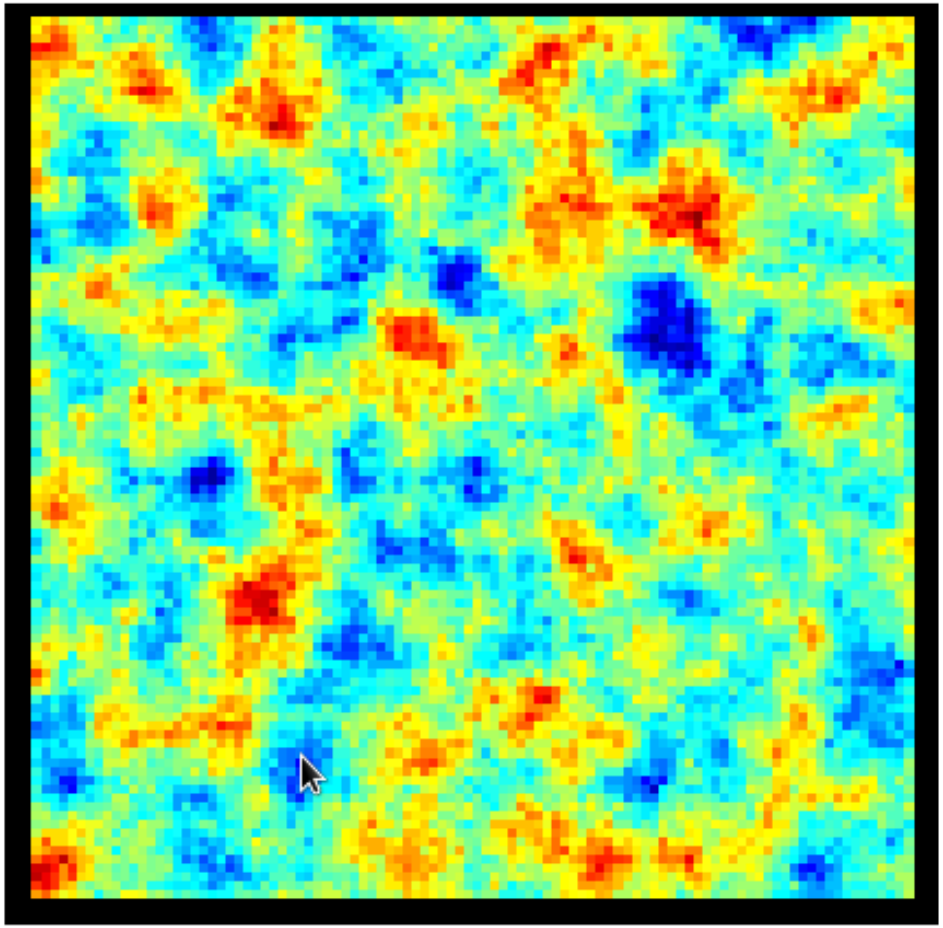

Figure: Unconditional simulation using a *Spherical covariance* with $C = 10$, $a = 10$ and 1000 lines.

Generating the Gaussian covariance

We can use the continuous spectral method [Emery, 2006] to generate the associated stochastic process to the Gaussian covariance

$$C_Y(r) = C \exp\left\{-\left(\frac{r}{a}\right)^2\right\} \tag{12}$$

with sill C and scale factor a (practical range $a\sqrt{3}$). The algorithm is the following:

1. Generate a Gaussian vector T and an uniform random phase ϕ defined in $[0, 2\pi[$. In $\mathbb{R}^3$, the Gaussian vector can be computed as $T = (t_1, t_2, t_3)$, where $t_i, i = 1, 2, 3$ is a Gaussian random number with distribution $N(0, 1)$.
2. The stochastic process is simulated using the equation $Y_\theta(X) = \sqrt{2C}\cos(2\pi(< x, T > +\phi))$, where $X \in \mathbb{R}^3$.

Generating the Gaussian covariance

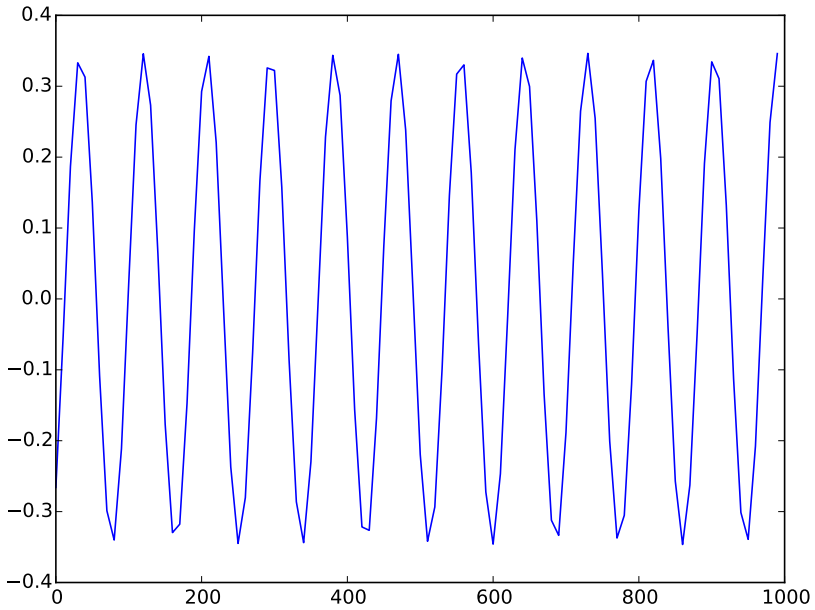

Figure: Simulation of the stochastic process associated to the *Gaussian covariance* with $C = 10$, $a = 10$

Generating the Gaussian covariance

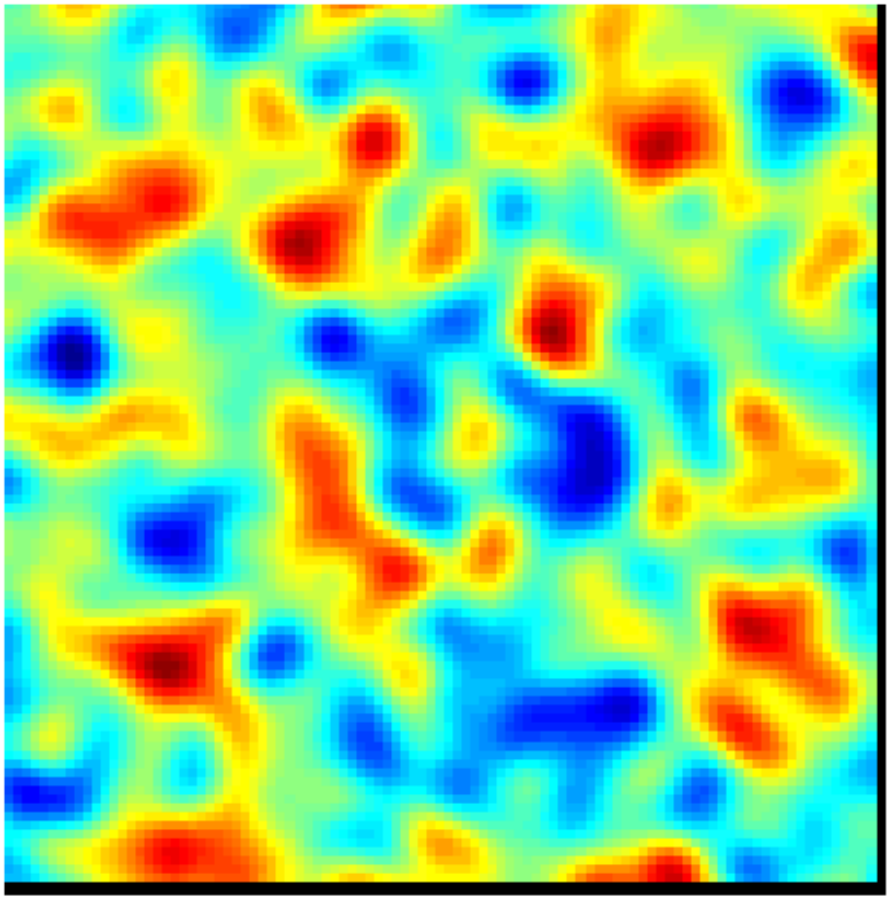

Figure: Unconditional simulation using a *Gaussian covariance* with $C = 10$, $a = 10$ and 1000 lines.

Conditioning the simulations

In the previous slides, we presented some techniques to simulated the stochastic process associated with some types of covariances. Now, we will present the algorithm to generate the conditional simulations using Turning Bands (or other type of algorithm) in the simulation domain D [Lantuéjoul, 2002]. In this algorithm, CD stands the conditioning data-set.

Conditional simulation of a Gaussian Random Function

1. Calculate the kriged estimates $y^*(x) = \sum_c \lambda_c(x) y(c)$ for each $x \in D$.
2. Simulate a gaussian random function (using Turning Bands or other type of simulation algorithm) with mean 0 and covariance C in the domain D and at the conditioning points. Let $(z(x), x \in D)$ and $(z(c), c \in CD)$ be the generated values.
3. Calculate the kriged estimates $z^*(x) = \sum_c \lambda_c(x) z(c)$ for each $x \in D$.
4. Return $(y^*(x) + z(x) - z^*(x), x \in D)$.

Final Remarks

- From the operational point of view, the Turning Bands is very similar to other Gaussian simulation algorithms like the *Sequential Gaussian Simulation*. The only difference is the parameter *number of lines (bands)*.
- The quality (convergence velocity) of the Turning Bands is directly related to the number of lines and to the algorithm used to generated the band directions.
- In the practice, using the Freulon's algorithm, a number of lines greater than 1000 is enough to generate good results [Emery, 2006].
- When the number of lines is not appropriated, artifacts are generated.
- The Turning Bands is a *share nothing* algorithm, so it's very easy to implement a highly efficient distributed version of this algorithm. The SGeMS has a very good implementation using openMP.

Final Remarks

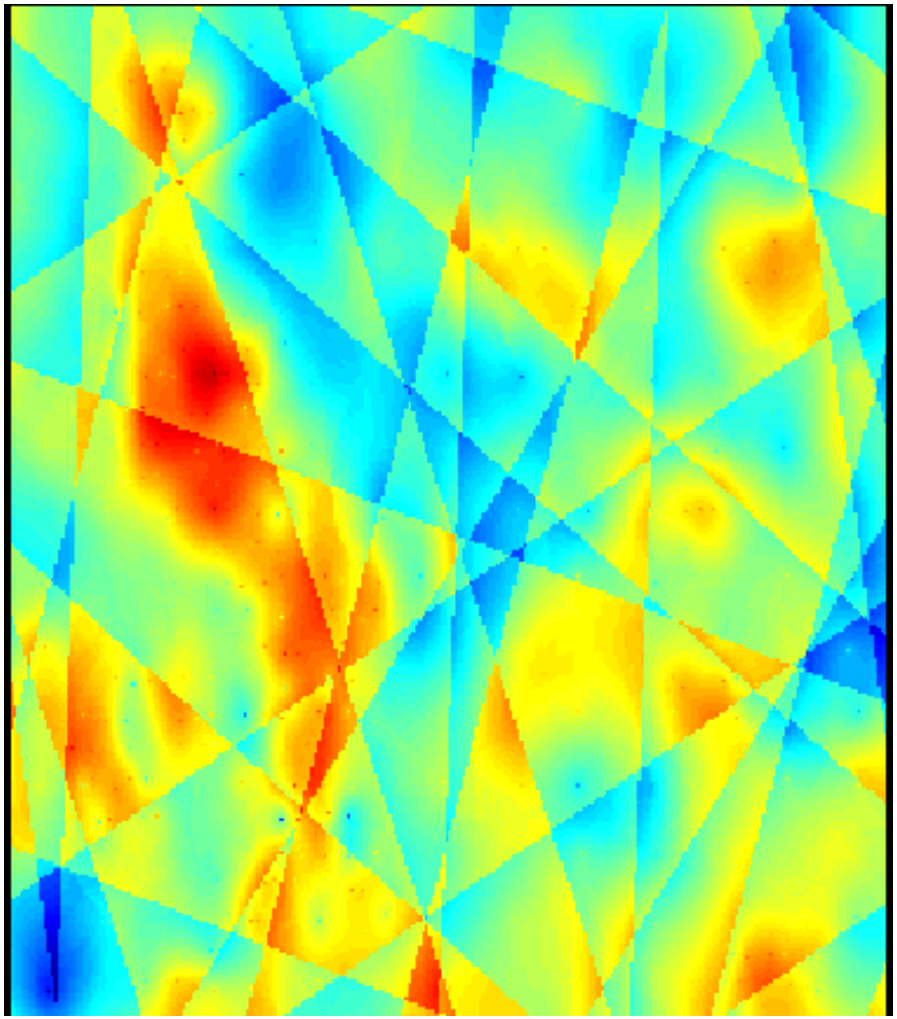

Figure: Conditional simulation of the Walker lake data-set with number of lines equals to 10.

Final Remarks

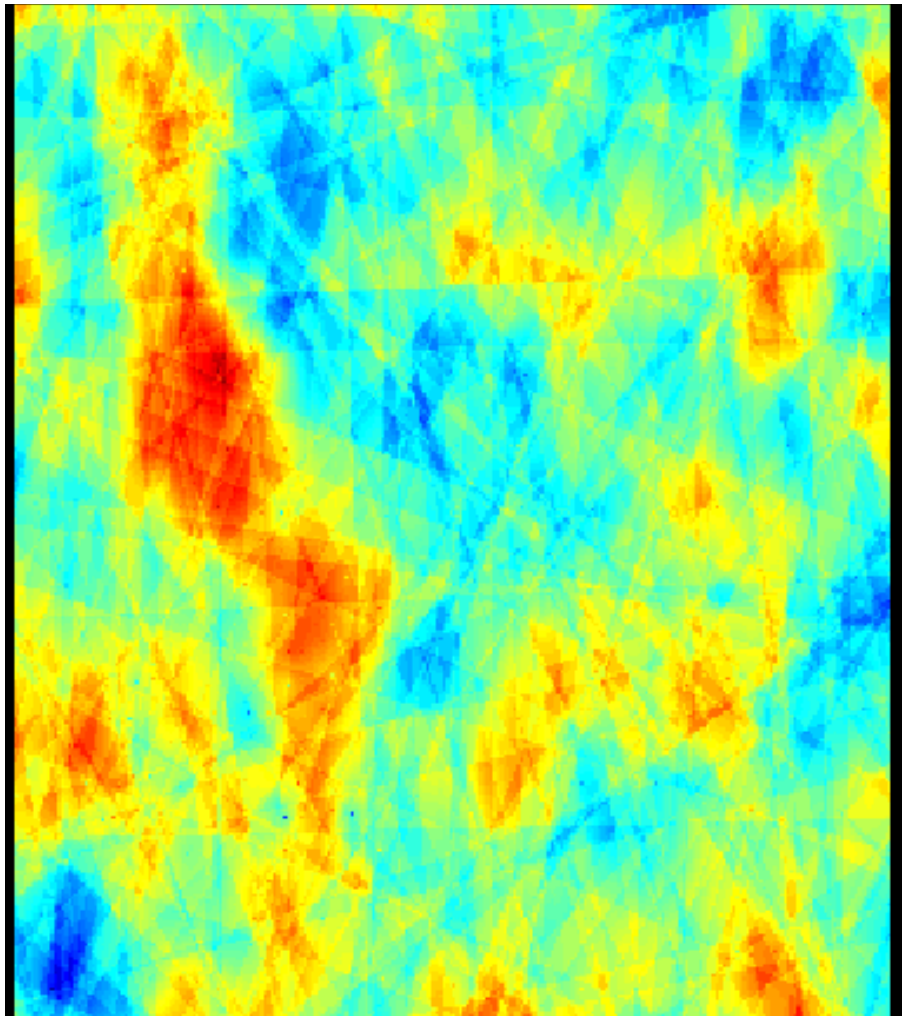

Figure: Conditional simulation of the Walker lake data-set with number of lines equals to 100.

Final Remarks

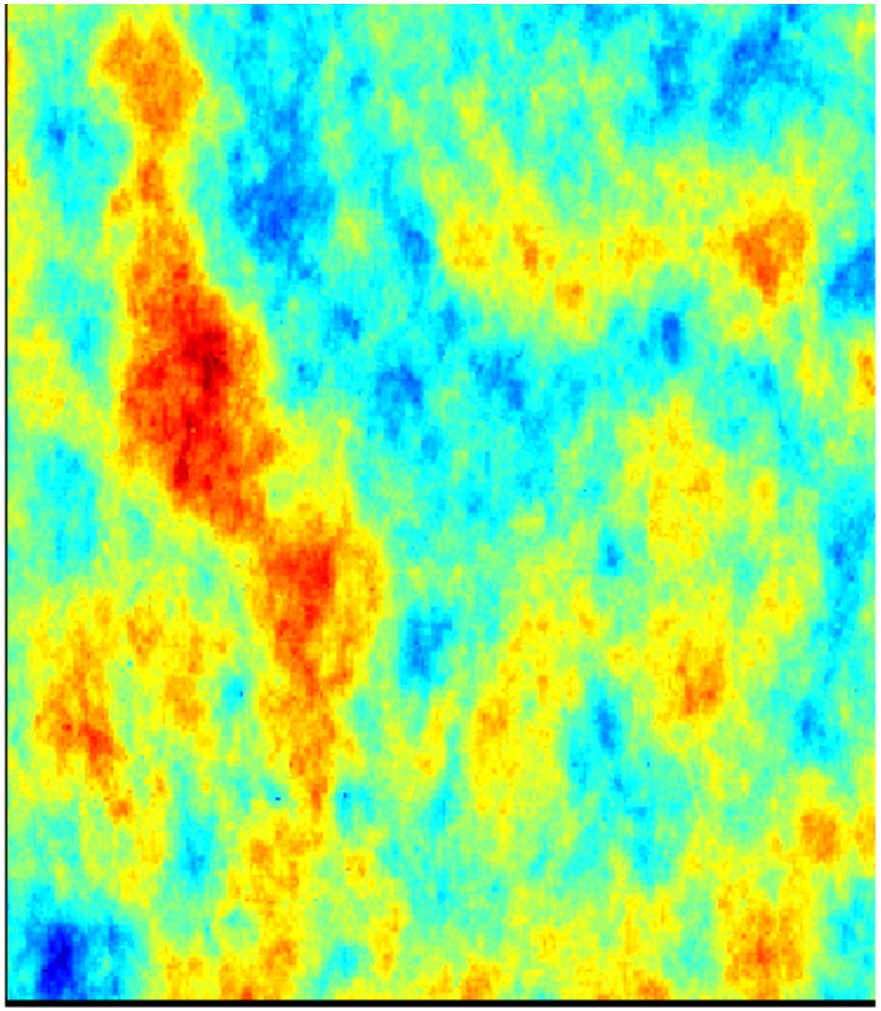

Figure: Conditional simulation of the Walker lake data-set with number of lines equals to 1000.

References

Christian Lantuéjoul (2002)

Geostatistical Simulation - Models and Algorithms

Springer, ISBN 3-540-42202-1

Xavier Emery, Christian Lantuéjoul (2006)

TBSIM: A computer program for conditional simulation of three-dimensional Gaussian random fields via the turning bands method

Computer and Geosciences 32, Elsevier, 2006, 1615-1628pp,

G. Matheron (1973)

The Intrinsic Random Functions and Their Applications

Advanced Applications probability

Xavier Freulon (1994)

Conditional simulation of a Gaussian random vector with non linear and/or noisy observations

Geostatistical Simulations, Kluwer, Dordrecht, 57-71.

The End