

Using Moving Average Methods for Simulating Large Fields

Vahid Dehdari

There are different Geostatistical methods for generating realizations. Each of these methods has different advantages and disadvantage. As an example Cholesky decomposition is based on decomposing covariance matrix for generating realizations. When the dimension of problem is large, decomposing covariance matrix would be very difficult. This paper shows procedure for generating realizations using moving average method which is similar to the Cholesky decomposition method. In this method, decomposing numerical covariance matrix can be avoided by finding square root of covariance matrix analytically. This can be done using Fourier transform properties. Using this method, generating realizations from large filed could be possible in a reasonable time. This paper shows the procedure by considering different examples.

Introduction

Generating a three-dimensional random field with covariance $C(r)$ could be done with the turning bands method by simulating along lines of the one dimensional random field with covariance $C^{(1)}(s)$.

Journel (1974) showed that the harmonic analysis theorem analysis of a second-order stationary process and Fourier analysis of the covariance function $C^{(1)}(s)$ permits calculation of a one dimensional realization $y(u)$ with $C^{(1)}(s)$ covariance matrix. This Fourier analysis can be difficult. He mentioned that the usual three-dimensional models $C(r)$, e.g. spherical or exponential are such that the corresponding one-dimensional covariance $C^{(1)}(s)$ can be expressed in the form of a convolution product of a function $f(u)$ and its transpose $f^T(u) = f(-u)$

$$C^{(1)}(s) = f * f^T = \int_{-\infty}^{+\infty} f(u).f(u + s)du$$

This means that one-dimensional simulation can be done using a moving average method with a weight function equal to $f(u)$. Oliver et.al (2008) showed that this method is closely related to the Cholesky method. In Cholesky decomposition covariance matrix should be decomposed for finding lower triangular matrix L . Consider L is cholesky decomposition of a 1D exponential covariance matrix. In this example practical range of the exponential covariance matrix is three grid cells.

$$L = \begin{bmatrix} \ddots & \ddots \\ b & c & d & e & f & 0 & 0 & 0 & 0 & 0 & 0 \\ a & b & c & d & e & f & 0 & 0 & 0 & 0 & 0 \\ 0 & a & b & c & d & e & f & 0 & 0 & 0 & 0 \\ 0 & 0 & a & b & c & d & e & f & 0 & 0 & 0 \\ 0 & 0 & 0 & a & b & c & d & e & f & 0 & 0 \\ 0 & 0 & 0 & 0 & a & b & c & d & e & f & 0 \\ 0 & 0 & 0 & 0 & 0 & a & b & c & d & e & f \\ \ddots & \ddots \end{bmatrix}$$

Except for the several first and last rows, other rows are similar to each other and just shifted one array to the right in next rows. It can be shown that each element of the covariance matrix can be found from the following formula:

$$c_{i,j} = \sum_k l_{i,k}l_{k,j}^T = \sum_k l_{i,k}l_{j,k} = \sum_k f(k - i)f(k - j)$$

By setting $m = k - i$ we can write:

$$c_{i,j} = \sum_m f(m)f(m - (j - i))$$

or

$$c(h) = \sum_m f(m)f(m-h)$$

This is the definition of convolution. Actually each element of the covariance matrix can be found by convolution of one row of L with a column of L^T . This means that simulation can be carried out as moving averages with the weighting function $f(m)$. Based on this formula we can write:

$$C = f * f^T$$

In above formula $*$ means convolution. f is a weight function. It is difficult to find this function for 2 and 3-D covariance matrices. In mathematics, the Fourier transform of the convolution of two matrices is equal to the multiplication of the Fourier transforms of them with each other:

$$F(f * f^T) = F(f)F(f^T)$$

To find the weight function f we need to find the Fourier transform of the analytical covariance matrix. The weight function is equal to the square root of the Fourier back transform of this function. After finding the weight function, unconditional realizations can be found using the following formula:

$$m_i = m_{prior} + f * z_i$$

This formula is similar to the formula for generating unconditional realizations using Cholesky decomposition. Again z_i is random vector from Gaussian distribution with zero mean and variance 1. The only difference between this method and Cholesky method is that convolution of weight matrix with z_i should be computed instead of computing multiplication of L with z_i . As before, conditional realizations can be found from the following equation:

$$z_s(u) = z_k^*(u) + [z_{uc}(u) - z_{uc}^*(u)].$$

Where $z_k^*(u)$ can be found by kriging using the data values and $z_{uc}^*(u)$ can be found by another kriging using unconditional simulated values at the same location as data. Also $z_{uc}(u)$ is an unconditional realization.

Fig.1 shows a simple example of convolution of two matrices f and g :

Comparing Cholesky and Moving Average Methods

The most important disadvantage of Cholesky method is related to decomposing the covariance matrix when the dimension of the field is large. In this case the covariance matrix would be large and computationally decomposing this large matrix would be impossible. Also when we have 2D or 3D field, finding the weight function is not easy. In moving average method, decomposing covariance matrix can be avoided by finding these weights. The next section shows a procedure for finding the required weights.

Applicable Method for Finding Weights in Moving Average Method

The weights could be found by using the following procedure:

1. Select an appropriate type of covariance matrix in two or three dimension
2. Find the Fourier transform of this covariance matrix analytically
3. Find the square root of result in step 2
4. Find the Fourier back transform of function in step 3 analytically. This function is our kernel function and weights can be found from this function easily
5. After finding the kernel matrix, its convolution with a random vector from Gaussian distribution using zero mean and unit variance should be found
6. Add the prior mean to the results. The result is an unconditional realization with specified prior mean and covariance

This method is an efficient method. The kernel function for common covariance functions in two or three dimension has been found.

Different authors find different kernel functions for different covariance functions. Journal (1978) found analytical weight function for 1D spherical covariance matrix. Formula for spherical covariance matrix is:

$$C(r) = \begin{cases} K[1 - \frac{3r}{2a} + \frac{r^3}{2a^3}], & \forall r \in [0, a] \\ 0, & \forall r \geq a \end{cases}$$

where $K = C(0)$. In one dimensional case, covariance matrix can be shown by the following formula:

$$C^1(s) = \begin{cases} K[1 - \frac{3s}{a} + \frac{2s^3}{a^3}], & \forall s \in [0, a] \\ 0, & \forall s \geq a \end{cases}$$

Journal proposed the following formula for weight function of 1D spherical covariance:

$$f(u) = \begin{cases} \sqrt{(12K/a^3)} \cdot u, & \forall u \in [-\frac{a}{2}, +\frac{a}{2}] \\ 0, & \text{if not} \end{cases}$$

Also for the exponential model, the covariance matrix can be shown with the following formula:

$$C(r) = Ke^{-\frac{r}{a}}$$

Again, the 1D exponential covariance matrix is equal to

$$C^1(s) = K(1 - \frac{s}{a})e^{-\frac{s}{a}}$$

Journal proposed the following formula for the weight function of 1D exponential covariance:

$$f(u) = \begin{cases} 2\sqrt{\frac{K}{a}} \cdot (1 - \frac{u}{a})e^{-\frac{u}{a}}, & \forall u \geq 0 \\ 0, & \forall u < 0 \end{cases}$$

Finally, the Gaussian covariance can be shown using the following formula:

$$C(r) = Ke^{-\frac{r^2}{a^2}}$$

The corresponding one dimensional covariance is

$$C^1(s) = K(1 - \frac{2s^2}{a^2})e^{-\frac{s^2}{a^2}}$$

Journal assigned the following weight function for this 1D covariance matrix:

$$f(u) = \left(\frac{16K}{a^3}\right)\sqrt{\pi}e^{-\frac{2u^2}{a^2}} \cdot u$$

Oliver (1995) considered different covariance function in 2D and 3D domain and proposed different kernel functions for each of them. He showed that the Hankel transform can be used for finding n -dimensional Fourier transform of a covariance matrix:

$$F(k) = (2\pi)^{-n/2}k^{1-n/2} \int_0^\infty r^{\frac{n}{2}}C(r)J_{\frac{n}{2}-1}(rk)dr$$

Where $J_p(x)$ is a Bessel function of first type, of order p . After that weight function can be found from inverse of square root of Fourier back transform:

$$f(r) = (2\pi)^{-n/2}r^{1-n/2} \int_0^\infty k^{\frac{n}{2}}\sqrt{F(k)}J_{\frac{n}{2}-1}(rk)dk$$

Using this procedure, Oliver found analytical kernel function for different covariance functions in 2D and 3D cases. Results for 2D and 3D cases are shown in Tables 1-2:

Table 1: Square root kernels for two-dimensional covariance functions

Covariance model	Kernel of 2D covariance $C(r)$	Kernel of square root of covariance $f(r)$
Exponential	$exp(-r/a)$	$K_1(\frac{r}{a})(2\pi^2a^3r)^{-1/4}\Gamma(\frac{3}{4})^{-1}$
Gaussian	$exp(-r^2/a^2)$	$(4/a^2\pi)^{1/2}exp(-2r^2/a^2)$

Whittle's	$(r/a)K_1(r/a)$	$(1/a)K_0(r/a)$
Circular	$1 - \frac{2r}{a\pi} \left(1 - \frac{r^2}{a^2}\right)^{\frac{1}{2}} - \frac{2}{\pi} \arcsin \frac{r}{a}$ <p style="text-align: center;"><i>for</i> $r \leq a$</p> <p style="text-align: center;"><i>0 for</i> $r > a$</p>	$2/(a\pi^{1/2}) \text{ for } r \leq a/2$ <p style="text-align: center;"><i>0 for</i> $r > a/2$</p>

Table 2: Square root kernels for three-dimensional covariance functions

Covariance model	Kernel of 3D covariance $C(r)$	Kernel of square root of covariance $f(r)$
Exponential	$exp(-r/a)$	$r^{-1}(2\pi a)^{-1/2} exp(-r/a)$
Gaussian	$exp(-r^2/a^2)$	$(4/a^2\pi)^{3/4} exp(-2r^2/a^2)$
Spherical	$1 - \frac{3r}{2a} + \frac{r^3}{2a^3} \text{ for } r \leq a$ <p style="text-align: center;"><i>0 for</i> $r > a$</p>	$\left(\frac{6}{a^3\pi}\right)^{1/2} \text{ for } r \leq \frac{a}{2}$ <p style="text-align: center;"><i>0 for</i> $r > a/2$</p>

Examples

A number of examples will now be considered to show the procedure to simulate realizations using this method. Let's start with a 2D porosity field on a uniform lattice of 100 × 100 gridblocks that is assumed to be multinormal with mean being 0.25 and variance 0.1. An isotropic correlation with range 15 will be assumed for this field. In this example, generating unconditional realization of porosity using moving average with different covariance matrices can be considered. In this example the dimension of each realization is 10⁴ × 1. Using these realizations, the covariance matrix can be found numerically. In this case the dimension of covariance matrix is 10⁴ × 10⁴. Cholesky decomposition of this covariance matrix is time consuming. For this reason moving average can be used.

Based on the analytical kernel of square root of covariance formula, $f(r)$, kernel matrices can be found. Fig.2 shows different kernel matrices (based on different covariance matrices) for this simple example. In this figure you can see different kernels and the difference between shape of them.

Oliver et.al (2006) showed that for finding dimensions of filtering array a cut-off criteria, s , can be assumed and r should be increased until $\frac{f(r=r_{max})}{f(0)} < s$, this r_{max} represent the distance within which $f(r)$ cannot be cut off. From the formula given below, the maximum values for x and y can be found:

$$r = \sqrt{x^2 + y^2}$$

If the domain is isotropic, $x = y = \sqrt{r^2/2}$. The dimension of the kernel matrix is approximately 2 times this value. The Gaussian filter acts similar to a blur filter in graphical software.

Unconditional realizations can be found based on finding the convolution of different filters with a random vector from Gaussian distribution with zero mean and unit variance. Also, the prior mean should be added to the generated realizations. For this example, in Fig.3 you can see a 100 × 100 vector of z generated from normal distribution with zero mean and unit variance.

By finding the convolution of filters in Fig.2 with Fig.3 and adding prior mean to the generated realization we can find final unconditional realizations. Fig.4 shows final unconditional realizations using different type of filters.

In the second example an anisotropic case will be considered. In this example, the porosity field on a uniform lattice of 150 * 150 gridblocks is assumed to be multinormal with mean 0.25 and variance 0.1. The

principle correlation is along - 60 direction and equivalent to the length of 40 gridblocks. The shortest correlation range in the perpendicular direction has the length of 20 gridblocks. In this case we want to find Gaussian covariance matrix using moving average method.

Oliver et.al (2006) showed that when anisotropy exists, the coordinate system can be rotated so that the longest correlation direction is along coordinate axis, then stretch an axis to make the spatial correlation isotropic in the new coordinate system. Transformation can be shown in the following matrix form:

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \alpha \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

where α is anisotropy ratio. In Fig.5 you can see Gaussian filter for this example. Also Fig.6 shows a simulated unconditional realization using the moving average method. These cases are for small models. Also in reality, the covariance matrix is combination of different types of covariance matrices and also the size of the domain is much larger than this one. Finding analytical Fourier transform and its square root is difficult if the field covariance is a combination of two covariance matrices. In this case the following formula for generating unconditional realizations can be used:

$$m = m_{prior} + f_1 * z_1 + f_2 * z_2.$$

Consider a more realistic example of size 1000×1000 . The mean is 0.25 and variance is 0.1. An isotropic correlation with range 200 will be assumed for this field. In this example, the covariance matrix is a combination of Gaussian and exponential covariance matrices. Fig.7 shows filters for different covariance matrices.

Each filter has different dimensions. For this reason, the size of the random Gaussian realizations with zero mean and unit variance would be different. As a result two random realizations for these two filters should be generated. After that using different seed numbers, different unconditional realizations can be generated. Fig. 8 shows one of these realizations for this example.

Conclusions

Moving average is an efficient method for generating 2D or 3D realizations. In this method there is no need for generating and decomposing covariance matrix numerically. Analytical covariance matrix and its analytical square root can be used for generating realizations. This paper shows formula for analytical square root of different covariance matrices in 2D and 3D dimensions. Also the normal approach of nested structures makes it easy to simulate different constituents of the linear model of regionalization, then add them to each other. Efficiency of this method has been considered by showing different examples.

References

Journel, A. and C. Huijbregts, Mining geostatistics, *New York*, 1978.
 Oliver, D., Moving averages for Gaussian simulation in two and three dimensions, *Mathematical geology*, **27**(8), 939-960, 1995.
 Oliver, D. S., A. C. Reynolds, and N. Liu, *Inverse Theory for Petroleum Reservoir Characterization and History Matching*, first edition, Cambridge University Press, Cambridge, 2008.

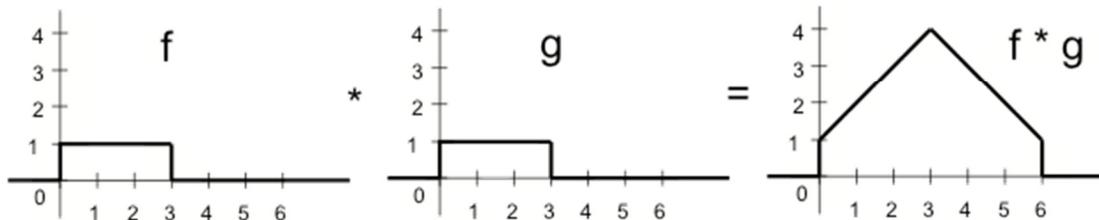


Figure 1: Convolution of matrix f with matrix (Ref: www.math.cornell.edu)

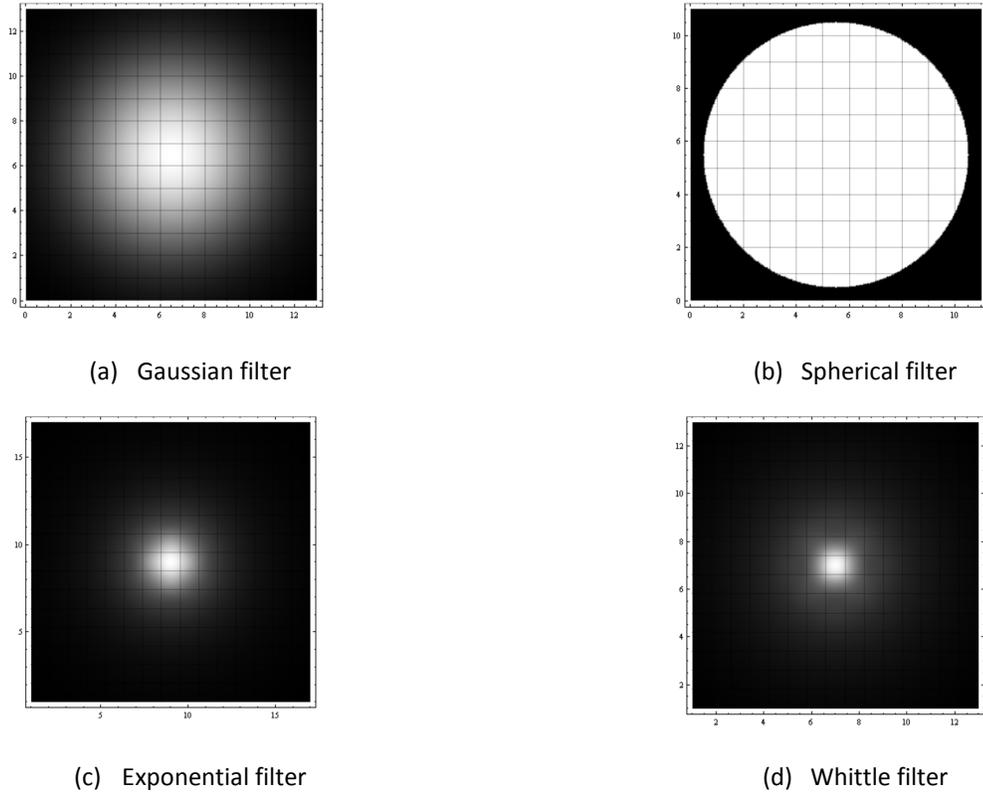


Figure 2: Filters of different covariance matrices

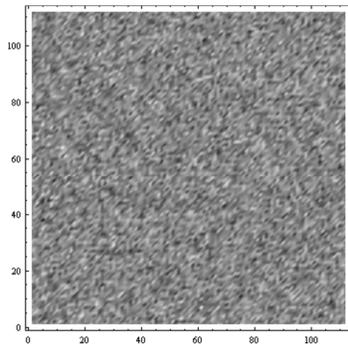
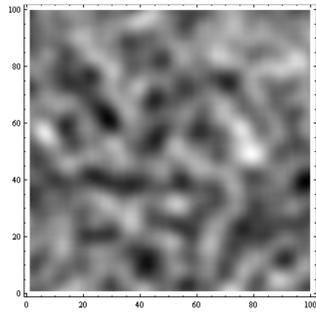
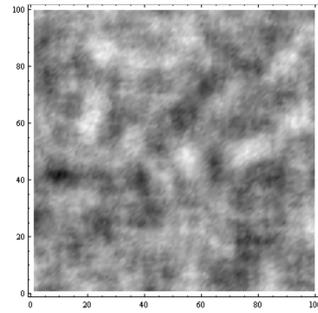


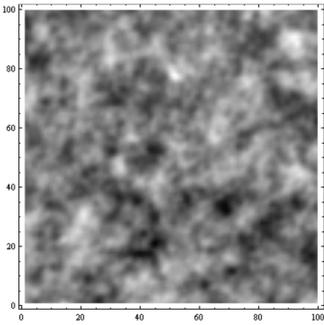
Figure 3: Realization from normal distribution with zero mean and unit variance



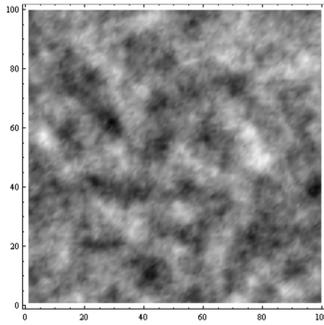
(a) Gaussian realization



(b) Spherical realization



(c) Exponential realization



(d) Whittle realization

Figure 4: unconditional realizations using moving average method

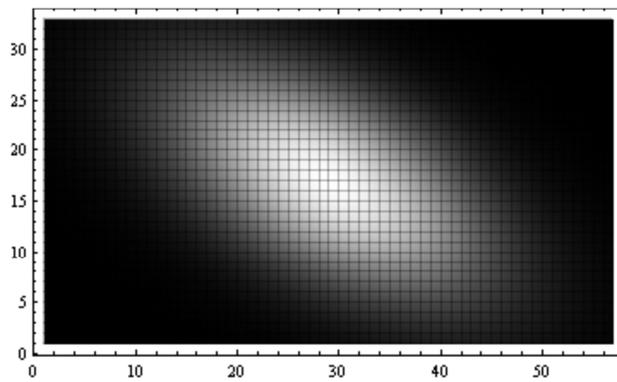


Figure 5: Gaussian filter with principal correlation along -60° direction

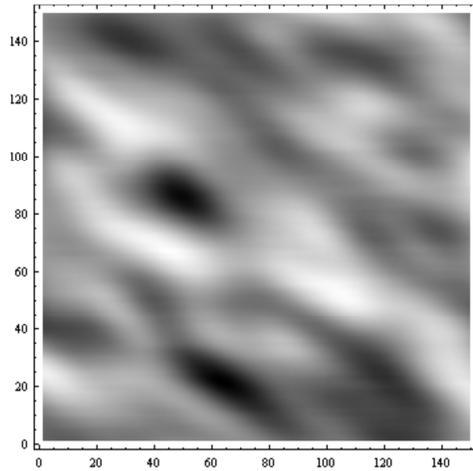
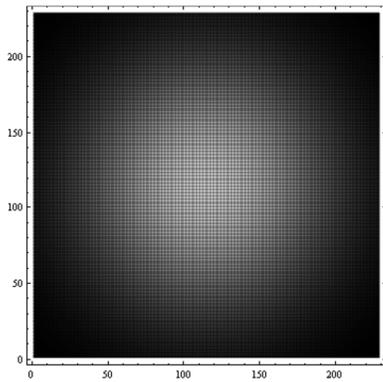
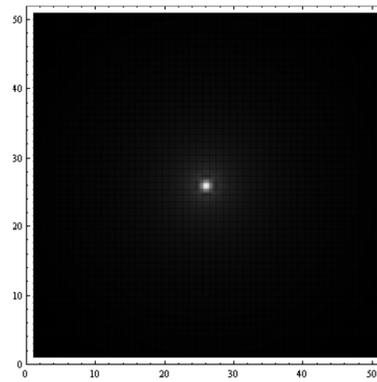


Figure 6: Unconditional realization with principal correlation along -60° direction



(a) Gaussian filter



(b) Exponential filter

Figure 7: Filters for example of mix covariances

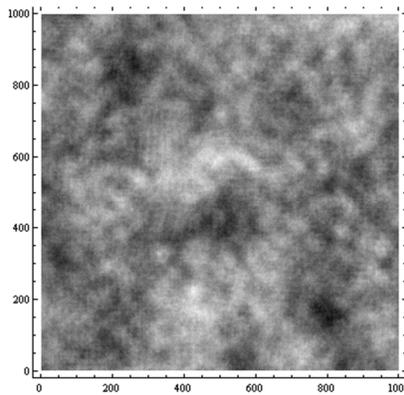


Figure 8: Unconditional realization for an example with mixed covariance