

Article

# The Sequential Generation of Gaussian Random Fields for Applications in the Geospatial Sciences <sup>†</sup>

John Dolloff and Peter Doucette \*

Sensor Geopositioning Center, National Geospatial-Intelligence Agency (contractors), 7500 GEOINT Dr, Springfield, VA 22150, USA; E-Mail: john.t.dolloff.ctr@nga.mil

<sup>†</sup> “Approval number” assigned by authors’ organization: PA case #14-350

\* Author to whom correspondence should be addressed; E-Mail: peter.j.doucette.ctr@nga.mil.

Received: 8 February 2014; in revised form: 21 May 2014 / Accepted: 26 May 2014 /

Published: 16 June 2014

---

**Abstract:** This paper presents practical methods for the sequential generation or simulation of a Gaussian two-dimensional random field. The specific realizations typically correspond to geospatial errors or perturbations over a horizontal plane or grid. The errors are either scalar, such as vertical errors, or multivariate, such as  $x$ ,  $y$ , and  $z$  errors. These realizations enable simulation-based performance assessment and tuning of various geospatial applications. Both homogeneous and non-homogeneous random fields are addressed. The sequential generation is very fast and compared to methods based on Cholesky decomposition of an *a priori* covariance matrix and Sequential Gaussian Simulation. The multi-grid point covariance matrix is also developed for all the above random fields, essential for the optimal performance of many geospatial applications ingesting data with these types of errors.

**Keywords:** geospatial; random field; errors; sequential; simulation; covariance matrix; strictly positive definite correlation function

---

## 1. Introduction and Motivation

This paper identifies a specific and practical subclass of homogeneous Gaussian two-dimensional (2D) random fields and presents a simple, fast, sequential method to generate discrete realizations over a  $pxq$  (horizontal) grid for the purpose of Monte Carlo simulation-based analyses. Let us term this

method Fast Sequential Simulation (FSS) for brevity of further description. FSS can be considered an extension of the sequential generation of a first order Gauss-Markov process from a 1D function of time to a 2D function of horizontal space. Although FSS was derived independently, it is also demonstrated a special case of Sequential Gaussian Simulation which is commonly used in the Geostatistics community. In particular, FSS is an unconditional simulation with simplicity and speed due to both exponential correlation in the spatial directions and an ordered generation over an evenly spaced grid of horizontal locations. Although other applications of Sequential Gaussian Simulation are more general (conditional or unconditional, irregularly spaced points, random generation paths, arbitrary valid correlation functions, *etc.*), many applications do not require these generalities but do require speed, preferably with a simple and direct implementation.

The paper first addresses scalar random fields, *i.e.*,  $z(k, l)$  where  $z$  typically represents a scalar error or perturbation at grid location  $(k, l)$ . The desired variance and spatial correlations for the  $z(k, l)$  are specifiable with FSS, and the multi-grid point covariance matrix derived. The paper then generalizes FSS to the generation of multivariate Gaussian two-dimensional random fields, *i.e.*,  $X(k, l)$ , where  $X$  is a vector of arbitrary dimension  $n$ . Finally, the paper generalizes FSS results even further to non-homogeneous Gaussian two-dimensional random fields, where the variance and spatial correlations are a function of grid location  $(k, l)$ . Some of the practical techniques presented for the sequential generation of both multivariate and non-homogeneous random fields are believed to be new and somewhat innovative.

Example realizations of scalar, multivariate, homogenous, and non-homogenous random fields are presented throughout the paper, as well as various theoretical properties, insights, and proofs, the latter contained in appendices. FSS is also compared to equivalent methods based on (1) Cholesky decomposition of a pre-computed *a priori* covariance matrix; and (2) Sequential Gaussian Simulation as implemented in various statistical packages. FSS is demonstrated to be many orders of magnitude faster than all of these other generation methods, as well as being a simpler implementation.

The ability to simulate errors across a horizontal grid with specifiable expected magnitudes (variance) and interrelationships (correlations) is an important capability in support of the Geospatial Sciences and supported by the FSS method presented in this paper. For example, the errors can represent elevation errors across a Digital Elevation database, horizontal errors in the location of vertices across a GIS database, horizontal and vertical errors in the locations of control points across a control point database, *etc.* All of these errors are essentially a function of horizontal location, *i.e.*, representable as a two-dimensional random field.

These simulated errors can be used to modify corresponding “truth” data in a simulation environment. Subsequent performance of various down-stream applications can then be meaningfully assessed, including modification (tuning) of their algorithms for optimal and reliable performance. Alternatively, in an operational environment, the applications themselves can have an embedded simulation capability in order to represent the effects of errors in their input data of known (specifiable) *a priori* characteristics. The effect is relative to the application’s output product and usually represented graphically. The simulation of tens of millions of errors within a few seconds and hundreds of millions within 30 s on a laptop computer is desired.

Previously, relevant errors have sometimes been simulated as homogeneous errors solely as an assumption for reduced complexity and/or increased speed. However, many realistic applications

correspond to data with non-homogeneous error characteristics; for example, data sets previously fused from other data sets with differing error (accuracy) characteristics. This paper addresses both types of errors. The non-homogeneous techniques presented in this paper essentially preserve the speed associated with the technique presented for the homogeneous case, typically reducing the speed by only a factor of two or three. The corresponding non-homogeneous characteristics are not totally general, but still adequate for many applications.

Finally, a common theme throughout this paper is the practical computation and need for a multi-grid point covariance matrix corresponding to  $z(k, l)$  or  $X(k, l)$  at multiple grid point locations. It is used by various applications to predict the accuracy of their input data and properly weight it within their various algorithms.

The authors of [1–5] discuss random fields in general, including their generation or simulation. Generation techniques include those based on Cholesky decomposition and Sequential Gaussian Simulation. In addition, these references discuss interpolation of a random field's realization based on Kriging. These references are relatively standard in the geostatistics community. They, along with other references from this community, are referenced per specific topic throughout the remainder of this paper, including appendices.

As detailed later, FSS was derived independently of Gaussian Sequential Simulation but is equivalent in specified circumstances. FSS is also directly related to both generalized multi-grid point covariance matrices [6] and strictly positive definite correlation functions [7] that have applications in the Geospatial community. Recent applications of FSS include evaluating conflation methods [8] and various geospatial algorithms [9].

### *Roadmap*

Sections 1–4 of this paper define the scalar homogeneous Gaussian 2D random field, the fast sequential generation algorithm FSS, and related practical aspects. Section 5 compares FSS to more typical generation methods such as those based on Cholesky decomposition or Sequential Gaussian Simulation, particularly with respect to timing or throughput. Section 6 then extends the FSS technique to a multivariate homogeneous Gaussian 2D random field, and Section 7 to a non-homogeneous Gaussian 2D random field.

## **2. A Scalar Gaussian 2D Random Field and Its Sequential Generation**

In this section of the paper, we define a scalar, homogeneous, Gaussian two-dimensional (2D) random field, typically corresponding to perturbations or errors. We also present FSS, the algorithm for the fast and sequential generation of a discrete and specific realization of this random field that can be used for various Monte Carlo simulation-based analyses.

We assume that the random field corresponds to  $z$ -error for specificity. Indices  $k$  and  $l$  correspond to a two-dimensional grid in the  $y$ - $x$  horizontal plane:  $k$  is in the “ $y$ ” direction and is the first grid index,  $l$  is in the “ $x$ ” direction and is the second grid index. Specifically,  $z(k, l)$  corresponds to  $z$ -error in meters at grid point  $(k, l)$ .

## 2.1. Statistical Characteristics

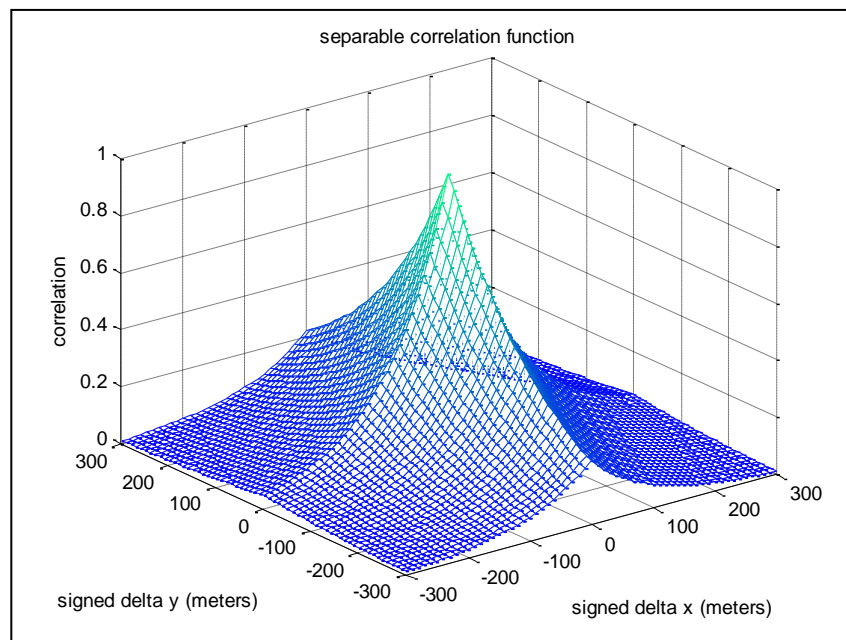
$z(k, l)$  is normally (Gaussian) distributed, and has a mean value of zero and a specifiable one-sigma  $\sigma_z$ , *i.e.*, is normally distributed  $N(0, \sigma_z)$  for all grid locations  $(k, l)$ . Its spatial correlation across the grid is separable, *i.e.*, has the (normalized) correlation function  $\rho(\Delta k, \Delta l)$ , where  $\Delta k$  and  $\Delta l$  are the absolute values of the component-wise differences in the  $(k, l)$  location of two arbitrary grid points. This function is represented as:

$$\rho(\Delta k, \Delta l) = \rho(\Delta y, \Delta x) = e^{-\Delta k \delta_y / T_y} e^{-\Delta l \delta_x / T_x} \quad (1)$$

where  $T_y$  and  $T_x$  are specifiable spatial correlation distance constants (meters) and  $\delta_y$  and  $\delta_x$  specifiable grid spacing (meters) in the  $y$  and the  $x$  directions in the horizontal plane, respectively. Note that  $\Delta y = \Delta k \delta_y$  and  $\Delta x = \Delta l \delta_x$ .

Figure 1 presents an example of  $\rho(\Delta y, \Delta x)$ , with  $T_y = 200$  m,  $T_x = 100$  m, and  $\delta_y = \delta_x = 10$  m. The spatial correlation  $\rho(\Delta y, \Delta x)$  is applicable to any pair of grid points within the entire grid that are separated by  $\Delta y$  meters in the  $y$ -direction and  $\Delta x$  meters in the  $x$ -direction. The use of two different spatial correlation distance constants allows for specification of different correlation characteristics in each of the horizontal directions.

**Figure 1.** An example of the separable spatial correlation function; in the plot of  $\rho(\Delta y, \Delta x)$ ,  $\Delta y$  and  $\Delta x$  have signed values.



Regarding the *a priori* statistics of  $z(k, l)$  in a more formal manner:

$$E\{z(k, l)\} = 0 \text{ (meters)}$$

$$E\{z(k, l)^2\} = \sigma_z^2 \text{ (meters-squared)} \quad (2)$$

$$E\{z(k, l)z(k + \text{or} - \Delta k, l + \text{or} - \Delta l)\} = \rho(\Delta k, \Delta l)\sigma_z^2 \text{ (meters-squared)}$$

at an arbitrary location  $(k, l)$  within the grid.

Note that  $E\{\}$  is the expectation operator, and  $\Delta k \geq 0$ ,  $\Delta l \geq 0$ ,  $T_y > 0$ ,  $T_x > 0$ ,  $\delta_y > 0$ ,  $\delta_x > 0$ ,  $|\rho(\Delta k, \Delta l)| < 1$  if  $\Delta k \neq 0$  or  $\Delta l \neq 0$ , and  $\rho(\Delta k, \Delta l) = 1$  when  $\Delta k = \Delta l = 0$ . Section 3.0 of this paper also presents the covariance matrix associated with two or more  $z(k, l)$ , each associated with a different grid point  $(k, l)$ .

## 2.2. Core Grid-Generation Equation

Equation (3) is the core grid-generation equation for FSS:

$$z(k+1, l+1) = r \cdot z(k+1, l) + s \cdot z(k, l+1) - r \cdot s \cdot z(k, l) + u(k+1, l+1) \quad (3)$$

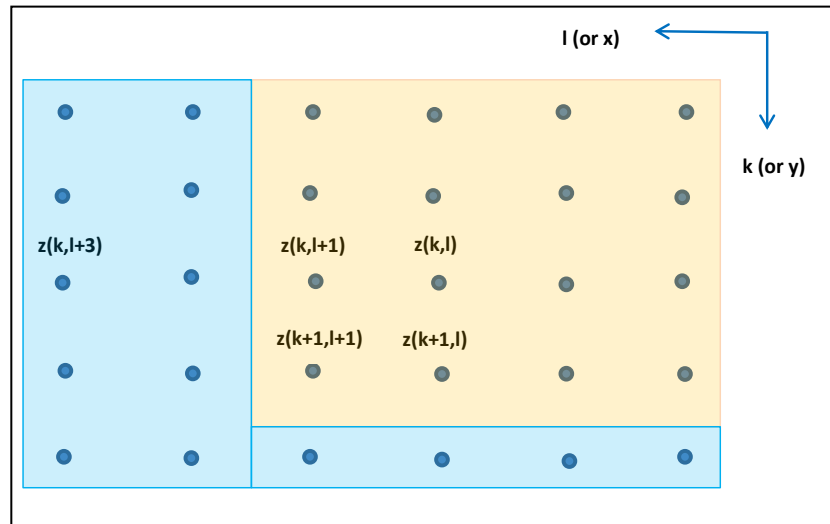
The integers  $k$  and  $l$  correspond to points in the grid,  $s = e^{-\delta_y/T_y}$ , and  $r = e^{-\delta_x/T_x}$ .  $u(k, l)$  is a random sample of Gaussian white noise, and is normally distributed  $N(0, \sigma_u)$ , where  $\sigma_u^2 = (1 - s^2)(1 - r^2)\sigma_z^2$ . That is, given a desired  $s$ ,  $r$ , and  $\sigma_z$ , a corresponding value of  $\sigma_u$  is computed per the above.

$s$  is the spatial correlation of the scalar error  $z$  between adjacent grid points in the  $k$  (or  $y$ ) direction ( $0 \leq s < 1$ , unit-less), and  $r$  is the spatial correlation of the scalar error  $z$  between adjacent grid points in the  $l$  (or  $x$ ) direction ( $0 \leq r < 1$ , unit-less). Therefore, we can also write the spatial correlation function  $\rho(\Delta k, \Delta l)$  as a function of grid units only:

$$\rho(\Delta k, \Delta l) = s^{\Delta k} r^{\Delta l} \quad (4)$$

Figure 2 illustrates the grid of errors  $z(k, l)$  generated based on Equation (3) and corresponding to a specific realization of the scalar, homogeneous, two-dimensional random field.

**Figure 2.** Horizontal grid of  $z$  errors.



All errors in the light orange area affect the error  $z(k+1, l+1)$  and those in the light blue area do not. Also, based on the specific sequential grid generation algorithm presented in Section 2.3, an error in the light blue area (e.g.,  $z(k, l+3)$ ) may be generated prior to  $z(k+1, l+1)$  even though it does not affect it.

### 2.3. Sequential Generation Algorithm for Realization over a $pxq$ Grid

The following presents a specific FSS algorithm for a discrete realization of  $z(k, l)$  over a  $pxq$  grid based on Equation (3):

$$z(1,1) = \text{random\_}N(0, \sigma_z);$$

$$z(2,1) = sz(1,1) + \text{random\_}N(0, \sigma_u);$$

$$z(1,2) = rz(1,1) + \text{random\_}N(0, \sigma_u);$$

$$z(2,2) = rz(2,1) + sz(1,2) - rsz(1,1) + \text{random\_}N(0, \sigma_u).$$

$$z(1,q) = rz(1, q-1) + \text{random\_}N(0, \sigma_u);$$

$$z(2,q) = rz(2, q-1) + sz(1,q) - rsz(1, q-1) + \text{random\_}N(0, \sigma_u).$$

The above completes rows 1 and 2. Generate row 3 as follows:

$$z(3,1) = sz(2,1) + \text{random\_}N(0, \sigma_u);$$

$$z(3,2) = rz(3,1) + sz(2,2) - rsz(2,1) + \text{random\_}N(0, \sigma_u).$$

$$z(3,q) = rz(3, q-1) + sz(2,q) - rsz(2, q-1) + \text{random\_}N(0, \sigma_u).$$

Repeat row 3-type processing for rows 4 through  $p$ .

Note that, in general, “ $\text{random\_}N(0, \sigma_a)$ ” corresponds to a random number (realization) from a  $N(0, \sigma_a)$  probability distribution; for example, in matlab this is implemented as “ $\text{sigma\_}a * \text{randn}(1,1)$ ”.

Appendix A of this paper presents a direct proof that the above FSS algorithm generates a realization of a two-dimensional random field with the specifiable statistical properties presented in Section 2.1. Appendix B also demonstrates its mathematical equivalence to a corresponding Sequential Gaussian Simulation approach for completeness. The latter must specify separable exponential correlation in the spatial directions and a fixed grid with a specific ordered path across it for generation of the realization. Also, depending on how it is implemented, it may or may not take advantage of the need to use the realization at only three previously generated grid points for generation at the current grid point, as does FSS. If it did in an efficient manner using pre-computed Kriging weights and little overhead due to flexibility and complexity, its speed could approach that of FSS.

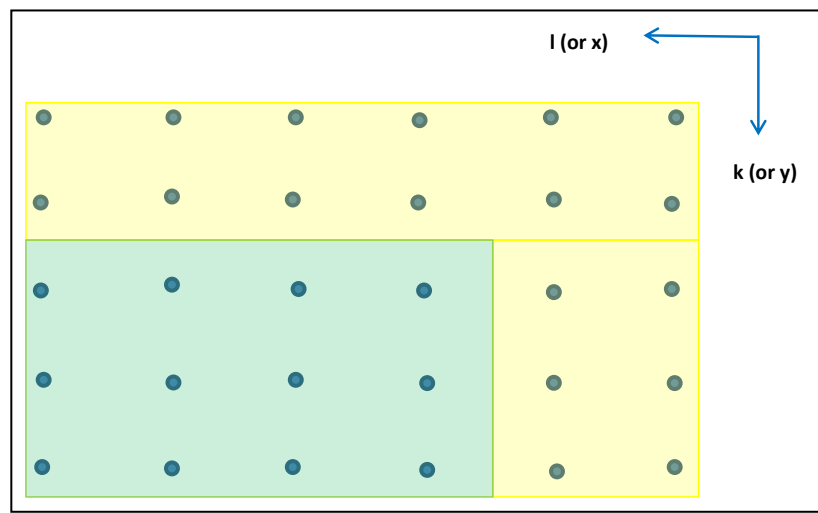
#### 2.3.1. Grid Spacing

Equation (3) and the above algorithm should typically incorporate grid spacing  $\delta_y$  and  $\delta_x$  equivalent to approximately one-ninth or less their respective spatial correlation distance constant, insuring at least 0.9 correlation with an adjacent grid point, i.e.,  $s = e^{-\delta_y/T_y} \geq e^{-1/9} \cong 0.9$  and  $r = e^{-\delta_x/T_x} \geq e^{-1/9} \cong 0.9$ , or equivalently,  $\delta_y \leq T_y/9$  m and  $\delta_x \leq T_x/9$  m. Of course, this “rule-of-thumb” is application dependent. For example, if very high spatial correlation between adjacent grid points is of interest, spacing should be closer.

### 2.3.2. Grid Buffer

As shown in Appendix A of this paper, the statistical properties of the  $z(k, l)$  are based on steady-state properties over an assumed infinite horizontal grid. Thus, for an actual application (realization) necessarily using a finite grid, the “final” grid should have a “buffer” on two edges of the computed grid to ensure that steady-state has essentially been reached. This is illustrated in Figure 3, where the buffer is yellow and the final grid is green.

**Figure 3.** Grid buffer (yellow), computed grid (yellow + green), final grid (green).



Placement of the buffer corresponds to the specific sequential grid generation algorithm presented earlier that starts at the top of the grid and always proceeds from right to left. The width of the top buffer should correspond to the equivalent of approximately two times the spatial correlation distance constant in the  $y$ -direction, and the width of the side buffer should correspond to the equivalent of approximately two times the spatial correlation distance constant in the  $x$ -direction.

More specifically,  $\text{width of top buffer (\# grid rows)} = \frac{2T_y}{\delta_y}$  and  $\text{width of side buffer (\# grid columns)} = 2T_x/\delta_x$ . Or equivalently, if  $s$  and  $r$  equal the value 0.9, 19 grid rows and 19 grid columns. This will ensure generation of errors throughout the final grid with the desired statistical properties.

### 2.4. Example Realizations: Surface Plots

This section presents surface plots of the  $z$ -error over a subset of a 2D final grid generated using the sequential algorithm of Section 2.3. Example 1 corresponds to specified  $\sigma_z = 10$  m, and specified  $s = r = 0.95$  (thus  $\sigma_u = 0.975$  m). Assuming a grid spacing in both the  $y$  and  $x$  directions of 1 m ( $\delta_y = \delta_x = 1$ ), this corresponds to spatial distance constants equal to  $T_y = T_x = 19.5$  m.

In this particular case, the spatial distance constants  $T_y$  and  $T_x$  were derived from the specified  $s$  and  $r$ , given assumed grid spacing  $\delta_y$  and  $\delta_x$ , not *vice versa*. The spatial distance constants were computed for information only. That is, there are two basic but equivalent approaches for the specification, application, and interpretation of spatial correlation, the particular approach selected based on convenience:

Approach 1—specify spatial correlation by the values of  $s$  and  $r$  (unitless) directly, implement Equation (3), and then interpret location-dependent results in terms of grid units. Given assumed grid spacing  $\delta_y$  and  $\delta_x$  (meters), the spatial distance constants  $T_y$  and  $T_x$  (meters) can be derived for information purposes only.

Approach 2—specify spatial correlation by the values of  $T_y$  and  $T_x$  (meters) and grid spacing  $\delta_y$  and  $\delta_x$  (meters), compute  $s$  and  $r$ , implement Equation (3), and then interpret location-dependent results in terms of y-x horizontal space in meters. The approach works well when the generated random field is to correspond to the *a priori* statistics and spatial resolution of a specific application of interest in the Geospatial Sciences.

Figure 4 presents the realization results of Example 1 based on Approach 1. Note that the remaining realization examples in this paper use Approach 1 as well, as it is most convenient.

**Figure 4.** Example 1—Realization of z-error with high spatial correlation between adjacent grid points.

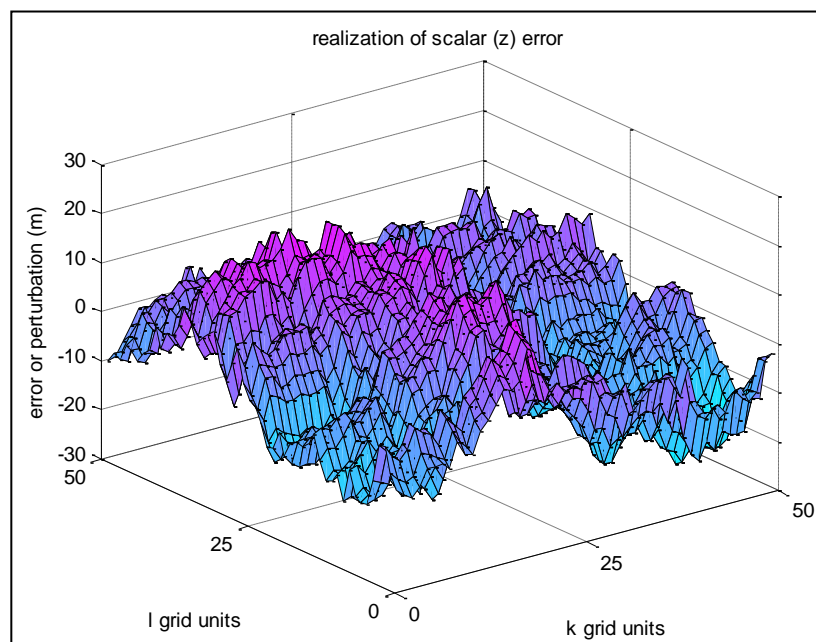


Figure 5 corresponds to Example 2, a new realization with the same  $\sigma_z = 10$  m, but with  $s = r = 0.1$  (thus  $\sigma_u = 9.9$  m).

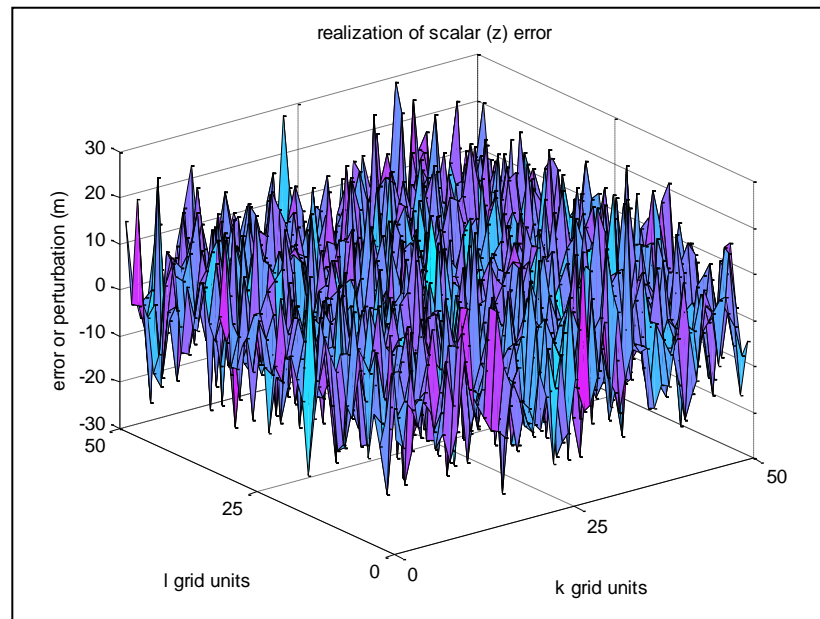
Figure 6 corresponds to Example 3, a new realization with the same  $\sigma_z = 10$  m, but with  $s = r = 0.999$  (thus  $\sigma_u = 0.02$  m).

As expected, the above realizations over portions of the grid do not have a mean z-error of zero nor a standard deviation about the mean of 10 m. However, when sample statistics are computed over numerous realizations, the corresponding mean and standard deviation approach 0 m and 10 m, respectively, matching the common *a priori* statistics used to generate the realizations.

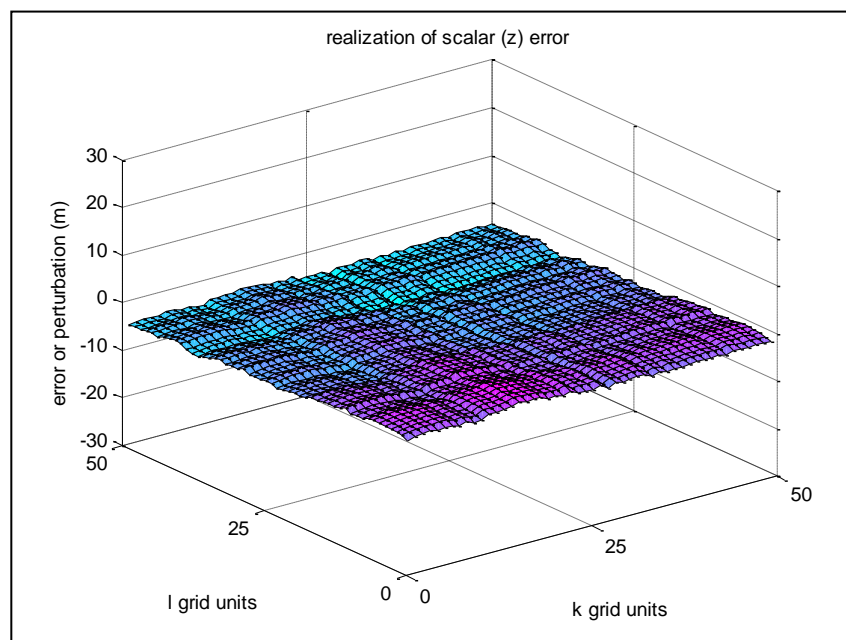
Finally, Figure 7 below presents Example 4, a new realization with the same  $\sigma_z = 10$  m, but with  $s = 0.1$  and  $r = 0.95$  (thus,  $\sigma_u = 3.107$ ), *i.e.*, different spatial correlations in the two directions.



**Figure 5.** Example 2—Realization of z-error with low spatial correlation between adjacent grid points.



**Figure 6.** Example 3—Realization of z-error with very high spatial correlation between adjacent grid points.

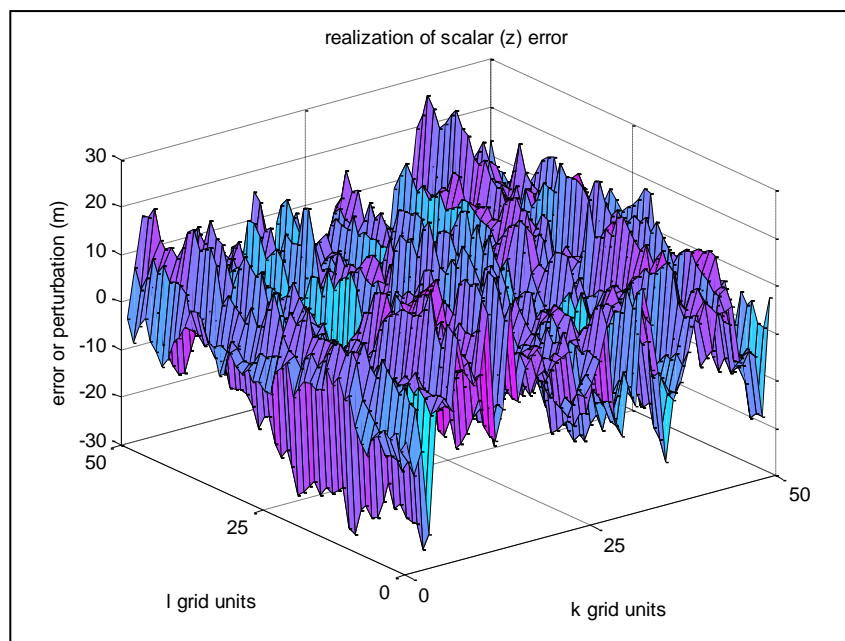


## 2.5. Example Realizations: Sample Statistics

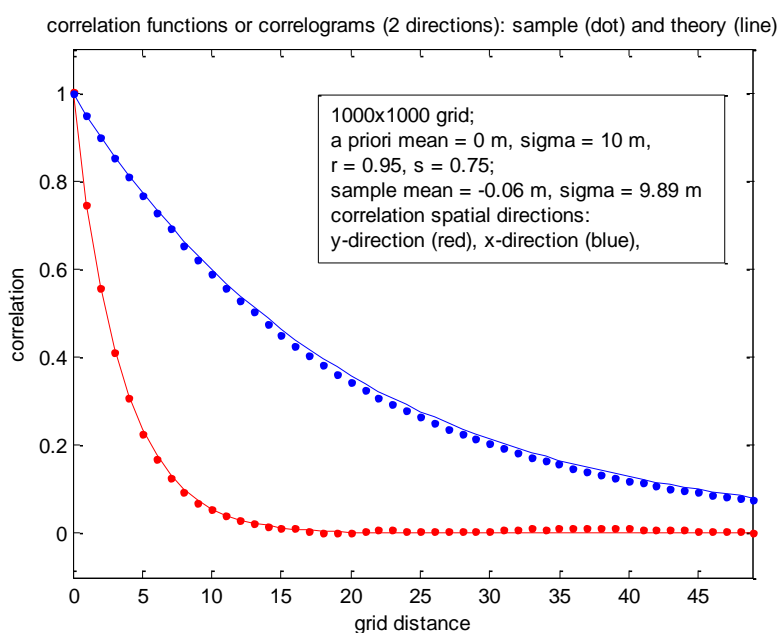
Sample correlation functions or correlograms were computed for two independent realizations across a final z-grid  $1000 \times 1000$  in size. *A priori* statistics were specified with a fixed mean value of 0 and a standard deviation about the mean of  $\sigma_z = 10$  m. The first realization corresponded to *a priori* correlations represented by  $r = 0.95$  and  $s = 0.75$ , and is presented in Figure 8. Correlation functions

were plotted as a function of horizontal distance in the x-direction and horizontal distance in the y-direction, and are different as expected per the values of  $r$  and  $s$ .

**Figure 7.** Example 4—Realization of  $z$ -error with different spatial correlations.



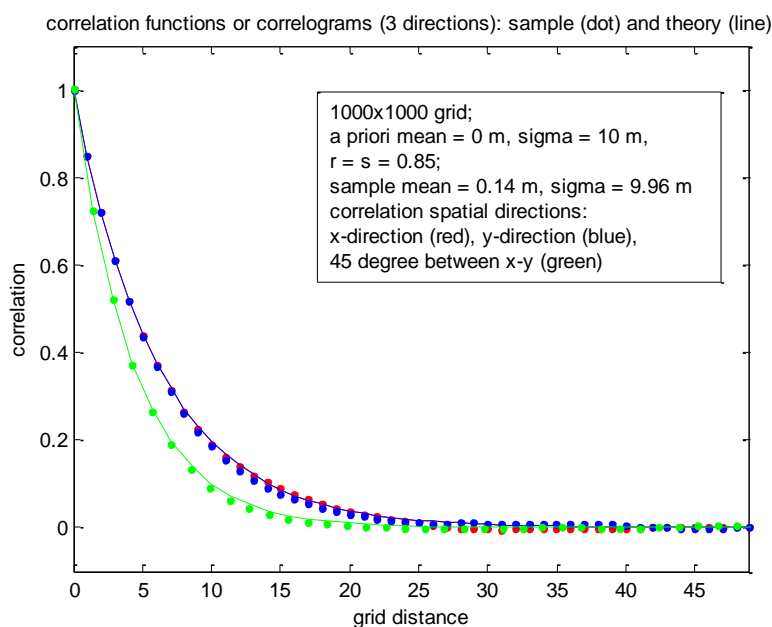
**Figure 8.** Sample statistics corresponding to  $1000 \times 1000$  grid with different  $a$  priori correlations.



The second realization corresponded to  $r = s = 0.85$ , and is presented in Figure 9. Three different horizontal directions were evaluated:  $x$ ,  $y$ , and  $45$  degrees between. Note that results for the latter are different even though  $r = s$  because the FSS correlation model is not isotropic. (Note that  $45$  degrees yields the direction with maximum difference.) In general, both plots demonstrate nearly identical results between the true and sample correlation functions—not unexpected because FSS has virtually

no approximations and because the random field is ergodic and the number of samples within a given realization large.

**Figure 9.** Sample statistics corresponding to  $1000 \times 1000$  grid with the same *a priori* correlations for the x and y directions, evaluated across three different directions.



**Figure 10.** Semivariograms corresponding to  $200 \times 200$  grid with the same *a priori* correlations for the x and y directions, evaluated across the y-direction for five different realizations.

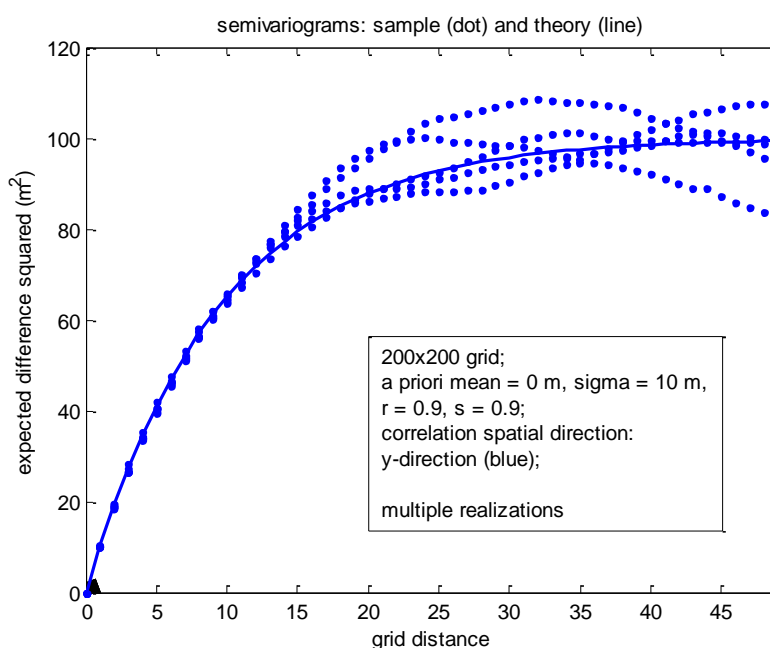


Figure 10 corresponds to a set of five independent realizations, this time over a much smaller  $200 \times 200$  grid. *A priori* statistics were identical to those corresponding to Figure 9 except that  $r = s = 0.9$ . Also, the sample and theoretical statistics computed correspond to the semivariogram,

typically of interest in the geostatistics community, computed in the y-direction across the horizontal grid. (See [5] for example, definitions of the correlogram and semivariogram.) Note the reasonable variability of the sample semivariograms corresponding to each of the five realizations about the common theoretical semivariogram.

### 3. Multi-Grid Point Covariance Matrix

If there are  $m$  specific scalar errors  $z(k, l)$  of interest associated with  $m$  arbitrary and different grid locations  $(k, l)$ , their corresponding  $m \times m$  (joint) covariance matrix is symmetric and positive definite (valid) since all the grid point errors have the same variance  $\sigma_z^2$  and have inter-grid point correlation between pairs corresponding to a normalized strictly positive definite correlation function (spdcf)  $\rho(\Delta l, \Delta k) = \rho(\Delta y, \Delta x) = e^{-\Delta l \delta_y / T_y} e^{-\Delta k \delta_x / T_x}$ , i.e., the multi-grid point covariance matrix equals:

$$P = E\{ZZ^T\} = \sigma_z^2 \begin{pmatrix} 1 & \rho(\Delta y_{12}, \Delta x_{12}) & \dots & \rho(\Delta y_{1m}, \Delta x_{1m}) \\ \rho(\Delta y_{21}, \Delta x_{21}) & 1 & \dots & \rho(\Delta y_{2m}, \Delta x_{2m}) \\ \dots & \dots & \dots & \dots \\ \rho(\Delta y_{m1}, \Delta x_{m1}) & \rho(\Delta y_{m2}, \Delta x_{m2}) & \dots & 1 \end{pmatrix} \quad (5)$$

where  $Z$  is an  $m \times 1$  vector such that  $Z^T = [z_1, \dots, z_m]$  and the  $z_i = z(k_i, l_i)$ ,  $i = 1, \dots, m$ , correspond to an ordered list of the  $m$  grid point locations. Also,  $\Delta y_{ij} = \Delta y_{ji}$  and  $\Delta x_{ij} = \Delta x_{ji}$  are the y and x distances in meters in the horizontal plane between the ordered points  $i, j \in \{1, \dots, m\}$ ;  $\sigma_z^2$  directly multiplies each element of the  $m \times m$  matrix. (Alternatively, the spatial correlation function and distances could have been written based on grid units.)

Note that the above is an *a priori* covariance matrix corresponding to the various  $z(k, l)$  considered as random variables, not specific realizations. See Reference [7] regarding the properties of a spdcf such that the above  $m \times m$   $P$  is guaranteed a valid (symmetric and positive definite) covariance matrix regardless the size of  $m$ . In general, just because the absolute value of correlation between two arbitrary grid point locations is less than 1.0, this by itself does not ensure a valid multi-grid point covariance matrix for  $m > 2$ .

The FSS generation of a realization of  $z(k, l)$  over a 2D grid as presented in Section 2.3 did not require use of the explicit multi-grid point covariance matrix in the generation process. So, why is the calculation of this covariance matrix of interest in terms of specific applications of generated errors or perturbations? A major reason is as follows: An “analysis module” may generate the simulated grid of errors and apply a subset to “truth data” and then pass the composite data to a “down-stream” application such that its performance can be assessed in the presence of errors. Many such applications also require knowledge of the multi-grid point covariance matrix corresponding to the composite data for purposes of proper weighting of the composite data in various estimation procedures (Kalman Filter and Weighted Least Squares estimators) for the parameters (state vectors) of interest to the application [6]. Of course, these applications can simply be passed, along with the composite data, the corresponding  $\sigma_z^2$  and the parameters that define  $\rho(\Delta k, \Delta l) = \rho(\Delta y, \Delta x) = e^{-\Delta k \delta_y / T_y} e^{-\Delta l \delta_x / T_x}$ , such that the applications can then build the appropriate multi-grid point covariance matrix themselves.

### Homogeneity and Gaussian Joint Probability Density

The scalar errors  $z(k, l)$  generated using the FSS sequential generation technique are Gaussian distributed as they are a linear combination of the  $u(k, l)$ , which are Gaussian distributed by definition. (The linear combination is demonstrated explicitly in Appendix A.) Also, the corresponding grid of  $z$ -errors corresponds to a wide-sense homogenous random field since the variance and correlation of errors are invariant of specific absolute grid location(s). In addition, since the errors are Gaussian distributed, a wide-sense homogeneous random field is equivalent to a homogeneous random field [2].

Any finite collection of  $z(k, l)$  at different grid locations  $(k, l)$  contained in the  $m \times 1$  vector  $Z$  has a corresponding Gaussian joint probability density function defined as follows:

$$f(Z) = (2\pi)^{-m/2} (\det(P))^{-1/2} e^{-(Z^T P^{-1} Z)} \quad (6)$$

where  $P$  is the multi-grid point covariance matrix, and  $\det$  is the matrix determinant. Thus, probabilities can be assigned in a straightforward manner to any absolute or relative confidence interval of interest.

Finally, it is worth noting that all of the multi-grid point covariance matrices computed per this paper are valid, regardless the specific underlying probability distribution of errors. This is true for both the scalar error of Sections 1–5, multivariate errors of Section 6, and non-homogenous errors of Section 7. This is discussed in References [6,7], which also allow for the use of any valid family of spdcf. In addition, the authors of [6] discuss the importance of such covariance matrices, other generation methods for such covariance matrices, and how to generate corresponding probability error ellipsoids. Note that in Reference [6], these covariance matrices are more generally termed “multi-state vector error covariance matrices”.

## 4. Interpolation into the Grid

The FSS technique as described in Section 1 and 2 of this paper generates a realization of a random field at grid point locations only. This is perfectly adequate for many applications since the grid can be very large and dense. However, if scalar errors are desired between grid point locations, interpolation of the  $z(k, l)$  at the four enclosing grid point locations may be performed. Also, the multi-grid point covariance (Equation (5)) can be easily modified for a corresponding set of interpolated points. Simply modify the distances between grid points to corresponding distance between the locations of the interpolated points. These distances may be represented in either non-integer units of grid spacing or corresponding  $y$  and  $x$  distances in meters in the horizontal plane.

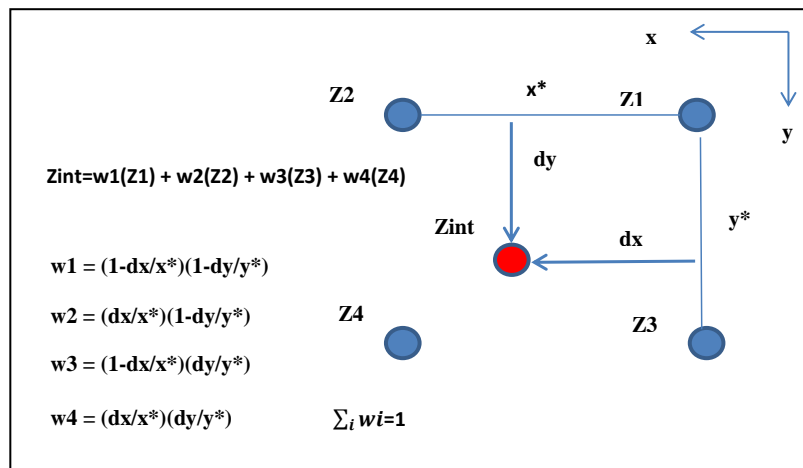
### Related Effects

Dependent on the interpolation method, the actual *a priori* statistics (one-sigma value, spatial correlations) for the interpolated  $z$  may be somewhat different than the assumed values as represented by the modified multi-grid point error covariance discussed above. The latter is consistent with the *a priori* statistics of the random field assuming no interpolation. (For *a priori* statistics, the interpolated value of  $z$  is treated as a random variable, not an estimate of a realization.)

For nearest neighbor interpolation, there are no explicit differences between the actual and assumed statistics as the location of the point is assumed a (nearest) grid point location. Of course, there can be implicit differences: for example, if two points for interpolation are close but not identical in location, they can be assigned to the same grid point with corresponding 100% spatial correlation between their errors whereas the actual spatial correlation is less.

For bi-linear interpolation, there are differences due to the “averaging” of uncorrelated components of error in the surrounding  $z(k, l)$  used during bi-linear interpolation. The higher the correlation between adjacent grid points (the larger  $s$  and  $r$ ), the less the effect (differences). If the recommended *a priori* correlation between adjacent grid points (0.9 or greater per Section 2.3.1) is used during grid generation, the effect is minimal. For example, if  $r = s = 0.9$ , the actual *a priori* one-sigma value for the interpolated points is 0%–5% less than the *a priori* one-sigma value for the grid points, and the actual *a priori* correlation between interpolated points 0%–3% less than for grid points at corresponding distances. The actual value for the percentage difference is dependent on how close the interpolated point(s) is to a grid point. Figure 11 illustrates bilinear interpolation.

**Figure 11.** Bilinear interpolation of four surrounding grid points.



## 5. Comparison to Alternate Generation Methods

This paper presents FSS, a fast and efficient sequential method for the generation of a 2D grid of errors or perturbations. There is also an implied, associated multi-grid point covariance matrix (e.g., Equation (5)), but this covariance matrix is not needed in the generation process. On the other hand, the spatial correlation of errors with this generation method is limited to a specific spdcf family of spatial correlation functions, *i.e.*,  $\rho(\Delta k, \Delta l) = \rho(\Delta y, \Delta x) = e^{-\Delta k \delta_y / T_y} e^{-\Delta l \delta_x / T_x}$ . (Albeit, reasonably general in that the distance constants  $T_y$  and  $T_x$  are specifiable).

There are two other general approaches to the generation (simulation) over a 2D grid: (1) matrix square roots; and (2) Sequential Gaussian Simulation. The latter is sequential and, as mentioned previously, more general than FSS. The former is also more general, but not sequential. They are described in more detail in the next subsection.

### 5.1. Timing Comparisons among Simulation Techniques

Figure 12 shows CPU time comparisons among five different techniques for simulating perturbations for a square grid by varying the number of points  $n$ ; where  $\sqrt{n}$  is the number of points along one side of the grid. Computation times were measured with a PC laptop with Intel i5 dual core 2.3 GHz CPUs and 8 GB of memory.

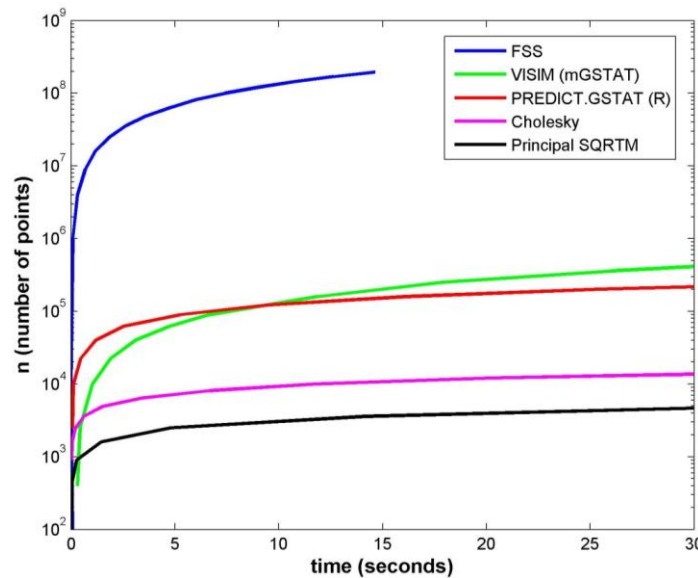
The objective was to measure the CPU performance of the main computation (sans overhead setup) for each method. Efforts were made to match the modeling parameters among all methods as closely as possible. Testing assumed unconditional, homogeneous, and isotropic models only (actually, for FSS the model was approximately isotropic, as  $T_x = T_y$ ). The following describes what main computations were timed in each of the five methods, and are listed in ascending order of computational speed gain according to Figure 12.

(1) Principal matrix square root (using Matlab function SQRTM)

$$\epsilon_z = \sqrt{\Sigma_z} \mathbf{r} \quad (7)$$

where  $\sqrt{\Sigma_z}$  is the  $n \times n$  principal matrix square root of  $\Sigma_z$ ;  $\mathbf{r}$  is a  $n \times 1$  vector realization of  $n$  independent  $N(0,1)$  distributed random variables, and  $\epsilon_z$  is the  $n \times 1$  vector of perturbations corresponding to the random variable  $z$  or  $z(k, l)$  over the grid.  $\Sigma_z$  was assumed to be a full, and positive definite matrix, i.e., the *a priori* covariance matrix corresponding to the random field at the  $n$  different points in the grid. Matlab uses the Schur decomposition technique to compute SQRTM for a general square matrix, which can be further sped up for symmetric and real matrices.

**Figure 12.** Time comparison among methods for unconditional simulation of a scalar random field  $z(k, l)$  over a 2D square grid (Note that FSS is cutoff at ~15 s due to reaching the system memory limit).



(2) Cholesky decomposition (using Matlab function CHOL)

$$\epsilon_z = L \mathbf{r} \quad (8)$$

where  $L$  is the lower triangular  $n \times n$  matrix from the Cholesky decomposition,  $\Sigma_z = LL^*$ , where  $L^*$  is the conjugate transpose of  $L$ ;  $\mathbf{r}$  is a  $n \times 1$  vector realization of  $n$  independent  $N(0,1)$  distributed random variables, and  $\epsilon_z$  is the  $n \times 1$  vector of perturbations corresponding to the random variable  $z$  or  $z(k, l)$  over the grid.  $\Sigma_z$  was assumed to be a full and positive definite matrix.

(3) PREDICT.GSTAT (version 1.0, 19 April 2014) [10] is the algorithm based upon Pebesma [11] as implemented and tested in the “R” (version 3.0.2, 64 bit) statistical package. The following R script is an example of parameters used to time unconditional simulation on a  $100 \times 100$  grid. Note that only the execution of the PREDICT function was timed.

```
#unconditional simulations on a 100 x 100 grid using gstat
library(gstat)
library(sp)
# create structure
xy <- expand.grid(1:100,1:100)
names(xy) <- c('x','y')
# define the gstat object (spatial model)
g.dummy <- gstat(formula=z~1, locations=~x+y, dummy=T, beta=0,
model=vgm(psill=1,model='Exp',range=10,nugget=0), nmax=2)
# time one simulations based on the gstat object
ptm <- proc.time()
yy <- predict(g.dummy, newdata=xy, nsim=1)
proc.time() - ptm
```

(4) VISIM in mGstat [12] is a sequential simulation code based on GSLIB (Geostatistical Software LIBrary, Stanford Center for Reservoir Forecasting, Stanford University) [13] for sequential Gaussian and direct sequential simulation. mGstat is a geostatistical Matlab toolbox available as open source that allows access to VISIM (among other algorithms) via a Matlab interface. The parameter file used to measure VISIM performance can be found in Appendix C.

(5) Fast Sequential Simulation (FSS) is the technique described in this paper, and was coded and timed as a Matlab function. The main required parameters used were grid spacing ( $\delta = 1$ ), standard deviation for the random variable ( $\sigma_z = 1$ ), and the spatial distance correlation constants ( $T_x = T_y = 10$ ), as all described in Sections 2.3–2.4 of this paper.

## 5.2. Discussion of Timing Results

The principal matrix square root (SQRTM) and Cholesky decomposition (CHOL) methods were provided to serve as a starting benchmark. While they are the least practical for large  $n$ , their main benefit is providing an exact solution for any spatial distribution of points and any *a priori* spatial statistics (valid covariance matrix). Figure 12 shows Cholesky providing roughly half an order of magnitude speed gain over SQRTM.

PREDICT.GSTAT and VISIM provide implementations of standard geostatistical techniques for Sequential Gaussian Simulation. Figure 12 shows that they provide comparable speed performance, and are 1–2 orders of magnitude faster than SQRTM and Cholesky. Their main advantage is providing broad flexibility for general purpose modeling among conditional and unconditional simulations. Moreover, additional speed efficiencies can be achieved when simulating multiple realizations with a fixed parameter set, which is not captured in Figure 12. E.g., following a single random path through the locations, PREDICT.GSTAT reuses results for each of the subsequent simulations [10].



FSS is the technique proposed in this paper. The two main advantages of FSS are (1) speed gain, e.g., three orders of magnitude faster than the next fastest technique as shown in Figure 12; and (2) simplicity of operation, e.g., requiring only three main parameters. Note that the FSS curve in Figure 12 is cutoff at ~15 s, which was due to reaching the memory limit for the grid size ( $n > 2 \times 10^8$  points). However, this constraint can be easily overcome by performing the computation with a local moving window *versus* storing the entire grid into system memory. The speed gain of FSS makes simulation of considerably denser grids more practical compared to the other methods. With this capability, our conjecture is that for those applications requiring interpolation, less expensive bilinear or nearest neighbor interpolation could be adequate for very dense grids *versus* more expensive Kriging in coarser grids. The variogram (correlation) model is constrained to an exponential function with FSS, which makes it less flexible than the GSTAT (Sequential Gaussian Simulation) methods. However, the tradeoff in speed gain and simplicity of implementation offers practical and useful advantages to motivate a potentially broader community of users.

## 6. Extension of FSS to a Multivariate Gaussian 2D Random Field

The FSS core grid generation equation, Equation (3), can be extended from a scalar error  $z$  to a (multivariate)  $n \times 1$  error vector  $X$  over a 2D grid in a straightforward manner. The more general case is presented directly below, with special but practical subcases presented in following subsections that include simpler notation.

$$X(k+1, l+1) = RX(k+1, l) + SX(k, l+1) - RSX(k, l) + U(k+1, l+1) \quad (9)$$

$$\text{where diagonal } R = \begin{bmatrix} r_1 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & r_n \end{bmatrix}, 0 < r_i < 1, i = 1, \dots, n; \text{ diagonal } S = \begin{bmatrix} s_1 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & s_n \end{bmatrix}, 0 < s_i < 1,$$

$$i = 1, \dots, n;$$

$$E\{X(k, l)X(k, l)^T\} = P_X, \text{ the } n \times n \text{ covariance matrix};$$

$$E\{X(k, l)X(k + \Delta k, l + \Delta l)^T\} = P_X S^{\Delta k} R^{\Delta l}, \{X(k, l)X(k - \Delta k, l + \Delta l)^T\} = S^{\Delta k} P_X R^{\Delta l},$$

$$E\{X(k, l)X(k + \Delta k, l - \Delta l)^T\} = R^{\Delta l} P_X S^{\Delta k}, E\{X(k, l)X(k - \Delta k, l - \Delta l)^T\} = S^{\Delta k} R^{\Delta l} P_X, \text{ for } \Delta k \geq 0 \text{ and } \Delta l \geq 0;$$

$$R^{\Delta l} = \begin{bmatrix} r_1^{\Delta l} & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & r_n^{\Delta l} \end{bmatrix} = \begin{bmatrix} e^{-\Delta l \delta_x / T_{x1}} & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & e^{-\Delta l \delta_x / T_{xn}} \end{bmatrix},$$

$$S^{\Delta k} = \begin{bmatrix} s_1^{\Delta k} & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & s_p^{\Delta k} \end{bmatrix} = \begin{bmatrix} e^{-\Delta k \delta_y / T_{y1}} & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & e^{-\Delta k \delta_y / T_{yn}} \end{bmatrix};$$

and  $P_U = E\{U(k, l)U(k, l)^T\}$  must be a valid (symmetric and positive definite) covariance matrix which satisfies the following:

$$P_U = H * P_X, \text{ the Hadamard product (term by term product) of two } n \times n \text{ matrices, where}$$

$$H = \begin{bmatrix} \dots & \dots & \dots \\ \dots & (1 - s_i s_j)(1 - r_i r_j) & \dots \\ \dots & \dots & \dots \end{bmatrix} \text{ and } i, j \text{ correspond to matrix row } i, \text{ column } j.$$

Note that the above constraint that  $P_U$  is a positive definite matrix is not satisfied for all possible combinations of  $s_i$ ,  $r_i$ , and desired (valid) steady state error covariance  $P_X$ , in which case Equation (9) and its statistics are no longer valid.

The corresponding derivation of the *a priori* statistics for the above multivariate homogeneous Gaussian 2D random field, including the constraint for  $P_U$ , is somewhat complicated and presented in Appendix D.

The actual grid generation algorithm associated with Equation (9) is as described previously in Section 2.3 of this paper associated with Equation (3), except that  $random\_N(0, \sigma_z)$  is replaced by  $P_X^{1/2} random\_v$ , and  $random\_N(0, \sigma_u)$  is replaced by  $P_U^{1/2} random\_v$ , where the superscript  $1/2$  corresponds to principal matrix square root and  $random\_v$  is the realization of an independent  $n \times 1$  random vector with each component an independent realization of a scalar random variable that is distributed  $N(0, 1)$ . Of course,  $S$  replaces  $s$ ,  $R$  replaces  $r$ , and  $X$  replaces  $z$ , as well.

Finally, for reasons similar to those presented in Section 3.1 for the scalar random field, the above errors  $X(k, l)$  are multivariate Gaussian distributed and correspond to a homogeneous random field. Again, see Reference [1].

### 6.1. Common Spatial Correlation Subcase

The following is a special, but practical, subcase of Equation (9) where the constraint on  $P_U$  is always satisfied:

$$X(k+1, l+1) = RX(k+1, l) + SX(k, l+1) - RSX(k, l) + U(k+1, l+1) \quad (10)$$

where  $R = rI_{n \times n}$ ,  $S = sI_{n \times n}$ ,  $P_U = (1 - s^2)(1 - r^2)P_X$ .

This, in turn, leads to a simple form for the cross covariance and corresponding spatial spdcf:  $E\{X(k, l)X(k+or-\Delta k, l+or-\Delta l)^T\} = \rho(\Delta k, \Delta l)P_X$ ; that is, all  $n$  components of  $X(k, l)$  have common inter-grid (spatial) correlation via a common (scalar) spdcf  $\rho(\Delta k, \Delta l) = s^{\Delta k} r^{\Delta l} = e^{-\Delta k \delta_y / T_y} e^{-\Delta l \delta_x / T_x}$ .

Note that Equation (10) allows for a full  $n \times n$  covariance matrix  $P_X$ , i.e., there can be non-zero intra-component correlations among the  $n$  components of  $X(k, l)$ . For example, assume that  $n = 3$  and  $X^T = [x \ y \ z]$  consists of error components  $x$ ,  $y$ , and  $z$ . Furthermore, at an arbitrary grid point location  $(k, l)$  the  $z$ -component of error is correlated  $+0.10$  with the  $y$ -component of error and the same  $x$ -component of error is correlated  $-0.60$  with the  $z$ -component of error. Of course, all  $n$ -choose-2 (a value of 3 for the case  $n = 3$ ) combinations of correlation among error components must correspond to a symmetric and positive definite  $P_X$ .

### Multi-Grid Point Covariance Matrix

For the special case of common spatial correlation, the corresponding  $mn \times mn$  multi-grid point covariance matrix for a collection of  $X(k, l)$  at  $m$  arbitrary grid points  $(k, l)$  has a convenient and valid representation:

$$P = E\{\Lambda \Lambda^T\} = \begin{bmatrix} P_X & \rho(\Delta y_{12}, \Delta x_{12})P_X & \dots & \rho(\Delta y_{1m}, \Delta x_{1m})P_X \\ \rho(\Delta y_{21}, \Delta x_{21})P_X & P_X & \dots & \rho(\Delta y_{2m}, \Delta x_{2m})P_X \\ \dots & \dots & \dots & \dots \\ \rho(\Delta y_{m1}, \Delta x_{m1})P_X & \rho(\Delta y_{m2}, \Delta x_{m2})P_X & \dots & P_X \end{bmatrix} \quad (11)$$

where  $\Lambda$  is an  $mn \times 1$  vector such that  $\Lambda^T = [X_1^T \ \dots \ X_m^T]$  and the  $X_i = X(k_i, l_i)$ ,  $i = 1, \dots, m$ , correspond to an ordered list of the  $m$  grid point locations. The  $n \times n$  cross-covariance terms

$\rho(\Delta y_{ij}, \Delta x_{ij})P_X$  consist of each element of  $P_X$  multiplied by the scalar value  $\rho(\Delta y_{ij}, \Delta x_{ij})$ ,  $i, j = 1, \dots, m$ .

## 6.2. Diagonal Covariance Subcase

Another special, but practical, subcase of Equation (9) is when the specified  $P_X$  is a diagonal matrix. This allows for any values of  $0 < s_i < 1$  and  $0 < r_i < 1$ , *i.e.*, different specifiable spatial correlations for each of the two directions for each of the  $n$  error components. Additionally, of course, this allows for different variances specified along the diagonal elements of  $P_X$ ; also, the constraint on  $P_U$  is always satisfied. Note that this special case is simply equivalent to the scalar case for each of the  $n$  components applied independently.

The resultant system of equations are identical to Equation (9) except that we have the following diagonal matrices:

$$P_X = \begin{bmatrix} \sigma x_1^2 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & \sigma x_n^2 \end{bmatrix}$$

$$E\{X(k, l)X(k + or - \Delta k, l + or - \Delta l)^T\} = S^{\Delta k} R^{\Delta l} P_X = \begin{bmatrix} s_1^{\Delta k} r_1^{\Delta l} \sigma x_1^2 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & s_n^{\Delta k} r_n^{\Delta l} \sigma x_n^2 \end{bmatrix} \quad (12)$$

where  $s_v^{\Delta k} r_v^{\Delta l} = \rho_v(\Delta k, \Delta l) = \rho_v(\Delta y, \Delta x) = e^{-\Delta k \delta_y / T_{yv}} e^{-\Delta l \delta_x / T_{xv}}$ ,  $v = 1, \dots, n$ .

Note that each component of  $X(k, l)$  has its own spatial correlation function with specifiable distance constants.

### 6.2.1. Example Realizations

This section presents quiver plots of multivariate error over a subset of a 2D final grid generated using the sequential algorithm discussed in Section 6. The multivariate error corresponds to a two-dimensional vector ( $n = 2$ ). The corresponding covariance matrix is diagonal with a common variance for the two components of error for convenience, *i.e.*,  $P_X = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_1^2 \end{bmatrix}$ . Similarly, spatial correlations corresponding to a specific spatial direction are common for convenience, *i.e.*,  $s_1 = s_2$  and  $r_1 = r_2$ . Figure 13 (automatically scaled) corresponds to  $\sigma_1 = \sigma_2 = 10$  m, and  $s_1 = s_2 = 0.95$  and  $r_1 = r_2 = 0.95$ . Figure 14 (automatically scaled) corresponds to  $\sigma_1 = \sigma_2 = 10$  m, and  $s_1 = s_2 = 0.95$  and  $r_1 = r_2 = 0.5$ .

### 6.2.2. Multi-Grid Point Covariance Matrix

For the special case of a diagonal covariance matrix  $P_X$ , the corresponding  $mn \times mn$  covariance matrix for a collection of  $X(k, l)$  at  $m$  arbitrary grid points  $(k, l)$  has a convenient and valid representation:

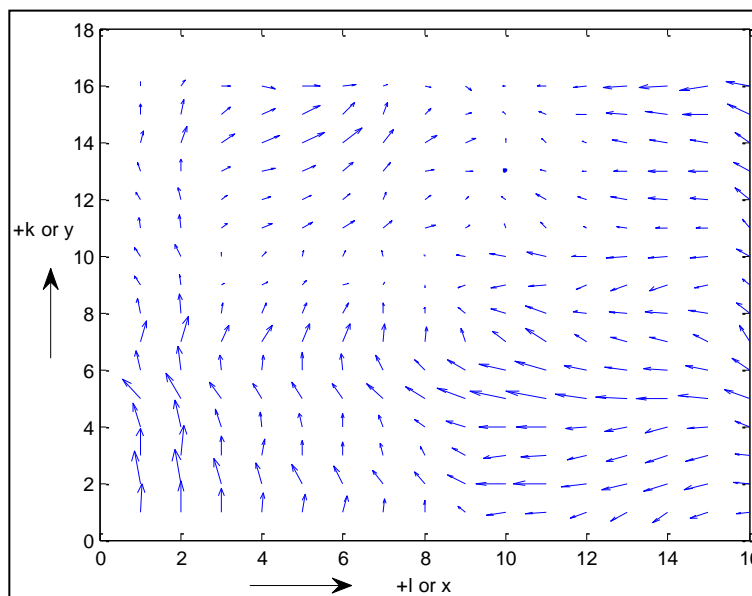
$$P = E\{\Lambda \Lambda^T\} = \begin{bmatrix} P_X & H_{12} * P_X & \dots & H_{1m} * P_X \\ H_{21} * P_X & P_X & \dots & H_{2m} * P_X \\ \dots & \dots & \dots & \dots \\ H_{m1} * P_X & H_{m2} * P_X & \dots & P_X \end{bmatrix} \quad (13)$$

where  $\Lambda$  is an  $m \times n$  vector such that  $\Lambda^T = [X_1^T \dots X_m^T]$  and the  $X_i = X(k_i, l_i)$ ,  $i = 1, \dots, m$ , correspond to an ordered list of the  $m$  grid point locations. Also,  $*$  corresponds to the matrix Hadamard (element by element) product, the  $n \times n$  diagonal matrix  $H_{ij} = \begin{bmatrix} \rho_1(\Delta y_{ij}, \Delta x_{ij}) & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & \rho_n(\Delta y_{ij}, \Delta x_{ij}) \end{bmatrix}$ , and  $\rho_v(\Delta y_{ij}, \Delta x_{ij})$  corresponds to the spatial correlation function associated with component  $v = 1, \dots, n$  of  $X(k, l)$ .

### 6.3. General Case with Constraint Enforced

Referring back to the general case of Section 6, the following presents two examples for  $n = 2$ . Assume that the two components of error correspond to  $x$ -error and  $y$ -error for specificity. From Equation (9),  $s_1, s_2, r_1, r_2$  can have any combination of values such that each is within the positive interval  $(0,1)$  and  $P_U$ , a function of the desired  $P_X$  and the  $s_1, s_2, r_1, r_2$ , is a symmetric and positive definite matrix.

**Figure 13.** Realization of two-dimensional multivariate errors over a 2D grid: high spatial correlation in the grid's  $k$  or  $y$ -direction and high spatial correlation in the grid's  $l$  or  $x$ -direction.



Subcase 1: Assume that  $s_1 = s_2 = s$  and  $r_1 = r_2 = r$ , to narrow down the possible combinations; therefore,

$$P_U = \begin{bmatrix} (1-s^2)(1-r^2) & (1-s^2)(1-r^2) \\ (1-s^2)(1-r^2) & (1-s^2)(1-r^2) \end{bmatrix} * P_X = (1-s^2)(1-r^2)P_X \quad (14)$$

which is always positive definite for any  $s$  and  $r$ .

Subcase 2: Assume that  $s_1 = r_1$  and  $s_2 = r_2$ , therefore,

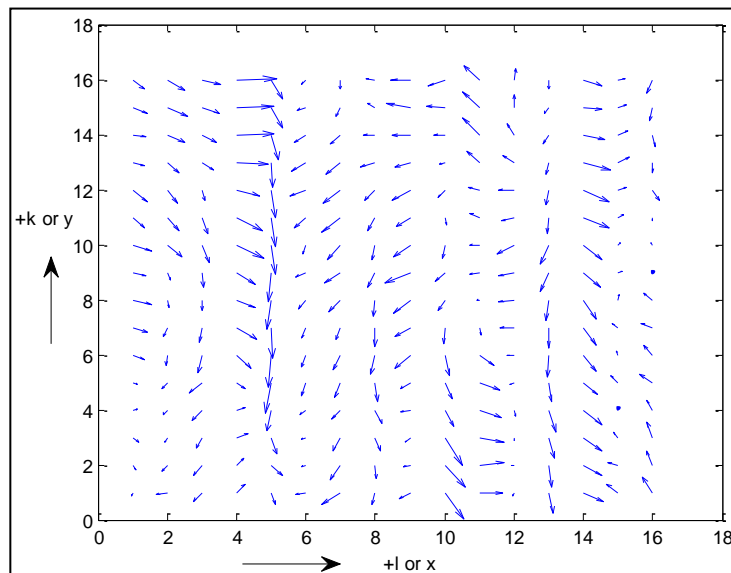
$$P_U = \begin{bmatrix} (1-s_1^2)^2 & (1-s_1s_2)^2 \\ (1-s_1s_2)^2 & (1-s_2^2)^2 \end{bmatrix} * P_X \quad (15)$$

which is positive definite if

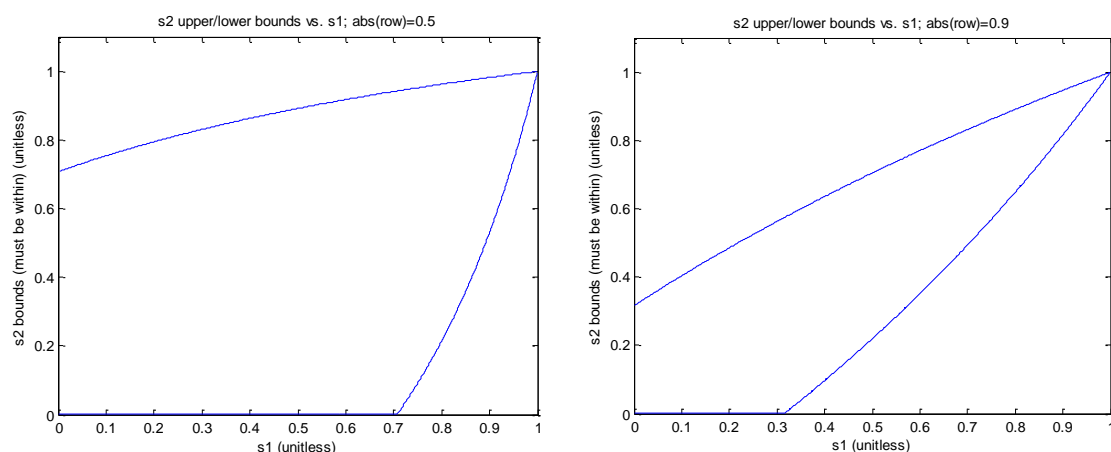
$$\left| \frac{P_U(1,2)}{(P_U(1,1)P_U(2,2))^{1/2}} \right| < 1, \text{ or } \left| \frac{\rho(1-s_1s_2)^2}{(1-s_1^2)(1-s_2^2)} \right| < 1, \text{ where } P_X = \begin{bmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix} \quad (16)$$

The left portion of Figure 15 plots the upper and lower bounds for  $s_2$  given the desired value of  $s_1$  and assuming that  $|\rho| = 0.5$ ; the right side assuming that  $|\rho| = 0.9$ .

**Figure 14.** Realization of two-dimensional multivariate errors over a 2D grid: high spatial correlation in the grid's  $k$  or  $y$ -direction and lower spatial correlation in the grid's  $l$  or  $x$ -direction.



**Figure 15.** Flow-down of constraints to spatial correlation bounds.



As can be seen from the above, the larger the absolute value of the correlation  $\rho$  between the error components  $x$  and  $y$ , the closer  $s_2 = r_2$  must be to  $s_1 = r_1$ .

Note that any multi-grid point covariance matrix for this general case must be assembled “term-by-term” using the *a priori* statistics presented in Equation (9), *i.e.*, there is no convenient functional form for the cross-covariances in the multi-grid point covariance matrix similar to those presented for the special case of common spatial correlation among the components of  $X(k, l)$  and the special case of a diagonal covariance matrix  $P_X$  presented earlier.

## 7. Extension of FSS to a Non-Homogeneous 2D Random Field

This section of the paper extends the FSS of a scalar homogeneous Gaussian 2D random field to a scalar non-homogeneous Gaussian 2D random field. In particular, the specified values for  $\sigma_z$ ,  $s$ , and  $r$  (variance and spatial correlation parameters) corresponding to  $z(k, l)$  are either explicitly or implicitly a function of grid location  $(k, l)$ . There is no one “right way” to do the extension.

Two general methods are presented below, each practical but with different characteristics regarding the form of non-homogeneity represented. Each method can also compute a corresponding multi-grid point covariance matrix, necessary for many applications as discussed earlier in Section 3. For one method, this covariance matrix is exact, for the other, an approximation. The best technique, when both non-homogeneity characteristics and possible multi-grid point covariance matrix approximations are taken into account, is application dependent. (The methods presented below can also be extended in a straightforward manner to multivariate non-homogeneous Gaussian 2D random fields.)

### 7.1. Method 1: Convex Combination

The core grid generation equation and corresponding sequential algorithm for scalar errors (Sections 2.2 and 2.3) is simply exercised  $n$  different times, either sequentially with the results saved temporally, or in parallel in order to save storage. (Of course, the grid size and spacing remains constant each time.) The number of times is typically two, *i.e.*,  $n = 2$ . Each uses a different set of specified  $\sigma_z$ ,  $s$ , and  $r$ . Thus, after the above is performed there are  $n$  grids, each homogenous and in accordance with the  $\sigma_z$ ,  $s$ , and  $r$  specified for use with that particular grid. Each grid is uncorrelated with the others.

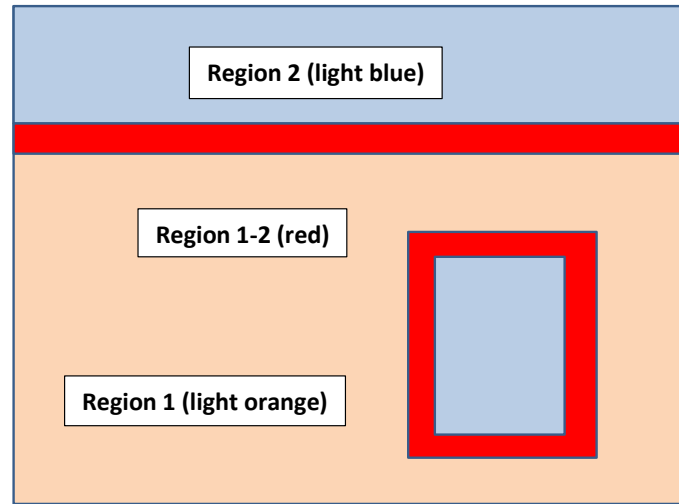
The  $n$  grids of  $z(k, l)$ , designated  $z_i(k, l)$  for  $i = 1, \dots, n$ , are then combined based on a convex combination into a final grid of  $z(k, l)$ . That is, at each  $(k, l)$  location in the  $p \times q$  grid:

$$z(k, l) = \sum_{i=1}^n \sqrt{w_i(k, l)} z_i(k, l), \text{ where } 0 \leq w_i(k, l) \leq 1 \text{ and } \sum_{i=1}^n w_i(k, l) = 1 \quad (17)$$

The specification of the  $w_i(k, l)$  values, also symbolized as  $w_{ikl}$  for convenience, can be as simple or as complicated as appropriate over the locations across the  $p \times q$  grid. However, their recommended values are in accordance with the following:

Let us assume that  $z(k, l)$  is to be exclusively the value  $z_i(k, l)$  across the various  $(k, l)$  in *Region i* of the  $pxq$  grid; hence, in this region, all  $w_{ikl} = 1$ . In addition, let us define *Region i-j* as a “buffer region” from *Region i* into *Region j*. In this buffer region,  $w_{kl}$  varies linearly from 1–0 corresponding to the  $(k, l)$  at the start to the end of the buffer region, respectively. Furthermore, of course,  $w_{jkl} = 1 - w_{ikl}$  throughout *Region i-j*. Finally,  $w_{ikl} = 0$  for all locations  $(k, l)$  in *Region j*. See Figure 16 as an example for  $n = 2$ .

Note that the width of the buffer region *Region i-j* should be at least twice the maximum of the corresponding spatial distance constants associated with  $z_i(k, l)$  and  $z_j(k, l)$ , expressed in grid unit. This ensures reasonable spatial correlation across the buffer region. If there were no buffer region, the spatial correlation between two points, one anywhere in *Region i* and the other anywhere in *Region j*, would be 0, *i.e.*, there would be an unwanted abrupt change across the boundary of the two regions.

**Figure 16.** Example of region layout over a  $p \times q$  grid.

### 7.1.1. Multi-Grid Point Covariance Matrix

Assume that  $m$  different scalar  $z(k, l)$  in the (final) grid are of interest regarding a corresponding multi-grid point covariance matrix. Each of these  $z(k, l)$  corresponds to their own unique  $(k, l)$  location in the grid, and are ordered in a known fashion sequentially for  $j = 1, \dots, m$ , and placed into an  $m \times 1$  vector  $Z$ , where  $Z^T = [z_1 \dots z_m]$ . Such a vector can also be defined for the same ordered locations for each of the  $m$  different realizations as  $Z_i, i = 1, \dots, n$ . Therefore, based on Equation (17) we have:

$$Z = \sum_{i=1}^n W_i Z_i, \quad (18)$$

where  $W_i$  is an  $m \times m$  diagonal matrix for  $i = 1, \dots, n$ , with the appropriate values of  $\sqrt{w_i(k, l)}$  down its diagonal.

For example, if the first component of  $Z$  corresponds to grid location  $(k, l) = (10, 20)$ , the first diagonal component of  $W_i$  equals  $\sqrt{w_i(10, 20)}$ .

Let us represent the corresponding  $m \times m$  multi grid point covariance matrix for the  $Z_i$  as  $P_i$ . (See Section 3 for how this matrix is computed given the corresponding  $\sigma_z, s$ , and  $r$ .) The  $m \times m$  multi grid point covariance matrix for  $Z$  is computed as follows:

$$P = E\{ZZ^T\} = \sum_{i=1}^n W_i (E\{Z_i Z_i^T\}) W_i = \sum_{i=1}^n W_i (P_i) W_i \quad (19)$$

A nice feature of Method 1 is that the above representation for the multi-grid point covariance matrix  $P$  is exact.  $P$  also corresponds to the following:

$$P = E\{ZZ^T\} = \begin{bmatrix} \sigma_{Z_1}^2 & \rho_{12} \sigma_{Z_1} \sigma_{Z_2} & \dots & \rho_{1m} \sigma_{Z_1} \sigma_{Z_m} \\ - & \sigma_{Z_2}^2 & \dots & \rho_{2m} \sigma_{Z_2} \sigma_{Z_m} \\ & - & \dots & \dots \\ & & - & \sigma_{Z_m}^2 \end{bmatrix} \quad (20)$$

where  $Z$  is an  $m \times 1$  vector such that  $Z^T = [z_1 \dots z_m]$  and the  $z_i = z(k_i, l_i)$ ,  $i = 1, \dots, m$ , correspond to the ordered list of the  $m$  grid point locations;  $\rho_{j_1 j_2}$  corresponds to the explicit correlation between two such points; the matrix entries “-” indicate symmetry.

### 7.1.2. Typical Statistics

The *a priori* statistics for a point and a point pair in the final  $z(k, l)$  grid are readily determined by the appropriate entries of a multi-grid point covariance matrix in which the two points are referenced. For convenience, results are summarized for a typical case as follows:

Assume a total of two regions and the typical values for  $w1_{kl}$  in *Region 1* ( $w1_{kl} = 1$ ),  $w1_{kl}$  in *Region 2* ( $w1_{kl} = 0$ ), and  $w1_{kl}$  in *Region 1–2* ( $w1_{kl} = 1 \rightarrow 0$ ). Let us designate the *a priori* one-sigma and correlation functions for the homogeneous  $z_1(k, l)$  and  $z_2(k, l)$  across the grid as  $\sigma_{z1}$  and  $\rho_1(\Delta k, \Delta l)$ , and  $\sigma_{z2}$  and  $\rho_2(\Delta k, \Delta l)$ , respectively, for convenience. We have the following location-dependent statistics for the final combined  $z(k, l)$ :

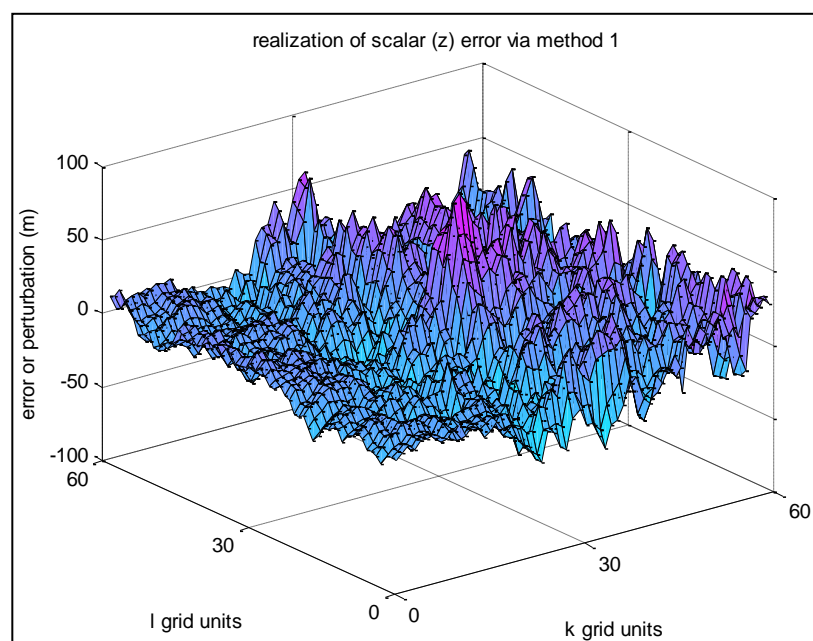
One-sigma value  $\sigma_z$ : a point in *Region 1*,  $\sigma_{z1}$ ; a point in *Region 2*,  $\sigma_{z2}$ ; a point in *Region 1–2*,  $(w1_{kl}\sigma_{z1}^2 + (1 - w1_{kl})\sigma_{z2}^2)^{1/2}$ .

Spatial correlation function  $\rho(\Delta k, \Delta l)$  value for a pair of points: both in *Region 1*,  $\rho_1(\Delta k, \Delta l)$ ; both in *Region 2*,  $\rho_2(\Delta k, \Delta l)$ , one in *Region 1* and one in *Region 2*, 0; one in *Region 1* and one in *Region 1–2*,  $(\frac{\sigma_{x1}\sqrt{w1_{kl}}}{\sigma_x})\rho_1(\Delta k, \Delta l)$ , etc.

### 7.1.3. Example Realizations

The following non-homogeneous realization combines two homogeneous realizations with  $\{r1 = s1 = 0.9, \sigma_{z1} = 10 \text{ m}\}$  and  $\{r2 = s2 = 0.9, \sigma_{z2} = 30 \text{ m}\}$ , respectively. *Region 1* of the displayed portion of the final grid consists of  $k=1-10$ , *Region 1–2*  $k = 11-30$ , and *Region 2*  $k = 31-60$ . (For each region, the corresponding  $l = 1-60$ .) Use of the typical assignment scheme for the values of  $w1_{kl}$  (and  $w2_{kl} = 1 - w1_{kl}$ ) was employed. Figure 17 below presents the results.

**Figure 17.** A smooth transition from *Region 1* to *Region 2*, each with their own specified *a priori* statistics.





The same experiment was performed (but different realization) except that there was no *Region 1–2*, i.e., a non-typical scheme. Figure 18 below presents the results.

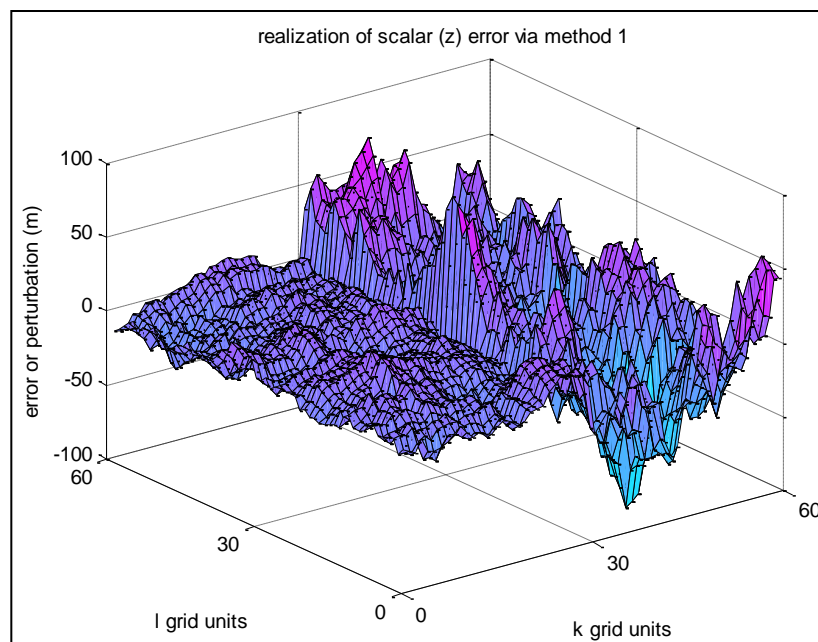
## 7.2. Method 2: Functional Variation of a Priori Statistics

With the second method, the core grid generation equation and corresponding sequential algorithm for scalar errors (Sections 2.2 and 2.3) is implemented only once, but modified as follows:

$$z(k+1, l+1) = r(k+1, l+1) \cdot z(k+1, l) + s(k+1, l+1) \cdot z(k, l+1) - r(k+1, l+1) \cdot s(k+1, l+1) \cdot z(k, l) + u(k+1, l+1) \quad (21)$$

where  $u(k, l)$  is a random sample of Gaussian white noise distributed  $N(0, \sigma_u(k, l))$ , and where  $\sigma_u^2(k, l) = (1 - s(k, l)^2)(1 - r(k, l)^2)\sigma_z^2(k, l)$ . Also,  $s(k, l) = e^{-\delta_y/T_y(k, l)}$ , and  $r(k, l) = e^{-\delta_x/T_x(k, l)}$ , that is, the spatial distance constants can be considered a function of  $(k, l)$  as well.

**Figure 18.** An abrupt transition between *Region 1* and *Region 2*, each with their own specified *a priori* statistics.

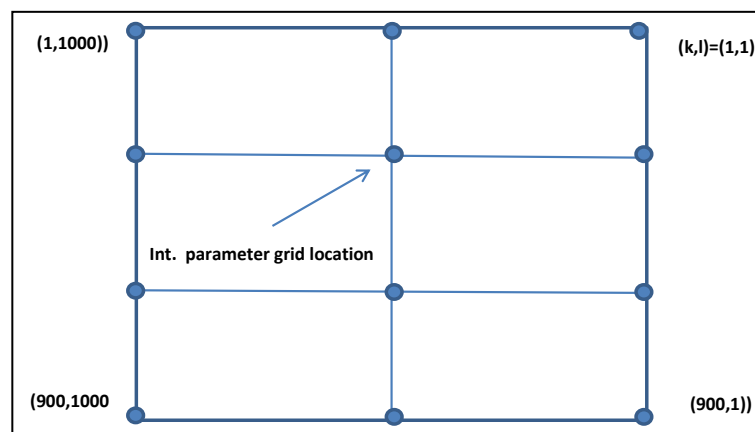


As indicated above, the values for  $\sigma_z$ ,  $s$ , and  $r$ , and hence  $\sigma_u$ , are a function of the grid location  $(k, l)$ . In addition, for Method 2, they are determined by the bilinear interpolation of such specified values over a less-dense grid overlaying the grid of errors to be generated. For example, if the 2D  $p \times q$  grid of errors to be generated is  $900 \times 1000$ , the grid for interpolation might be an evenly spaced  $4 \times 3$  parameter grid overlying the denser grid. Each of the corresponding 12 parameter grid points contains the specified values for  $\sigma_z$ ,  $s$ , and  $r$  for the corresponding local region around the parameter grid point. Note that  $\sigma_u$  is a function of the interpolated values of  $\sigma_z$ ,  $s$ , and  $r$ ; hence, is also recalculated in Equation (21) for every grid location  $(k, l)$ . See Figure 19 for a graphical representation of the interpolation parameter grid. Each interpolation parameter grid location contains a unique set of values for  $\sigma_z$ ,  $s$ , and  $r$ .

Also, the spacing between interpolation parameter grid points should be at least twice the maximum of the corresponding spatial distance constants associated with that grid point and the other interpolation parameter grid points immediately surrounding it, expressed in grid unit. This ensures that both the desired and the computed approximation of the *a priori* statistics corresponding to the  $z(k, l)$  across the dense grid are approximately met and reasonably reliable (see Section 7.2.2), respectively. (This also assumes that the appropriate buffer relative to the “final” grid is included as well—see Section 2.3.2)

Once defined appropriately, Equation (21) is then implemented via a direct counterpart to the algorithm described in Section 2.3. The latter simply utilizes the appropriate  $\sigma_z$ ,  $s$ , and  $r$  values that vary with  $(k, l)$  location.

**Figure 19.** An example of an Interpolation Parameter Grid.



### 7.2.1. Example Realizations

The following examples correspond to a  $7 \times 7$  parameter interpolation grid overlaying a  $90 \times 90$  2D grid. The results corresponding to a  $60 \times 60$  displayed portion of the final grid are presented.

All 49 sets of  $\{s, r, \sigma_z\}$  parameters were identical except for four sets corresponding to an interior rectangle near the center of the final grid. Let us term the 45 common sets as Group 1 and the other four common sets as Group 2. In Figure 20 below, the Group 1 set contain values  $\sigma_z = 10$  m,  $s = r = 0.9$ . Group 2 sets contain values  $\sigma_z = 50$  m,  $s = r = 0.9$ .

In Figure 21 below, the Group 1 sets contain  $\sigma_z = 10$  m,  $s = r = 0.1$ . Group 2 sets contain  $\sigma_z = 50$  m,  $s = r = 0.95$ .

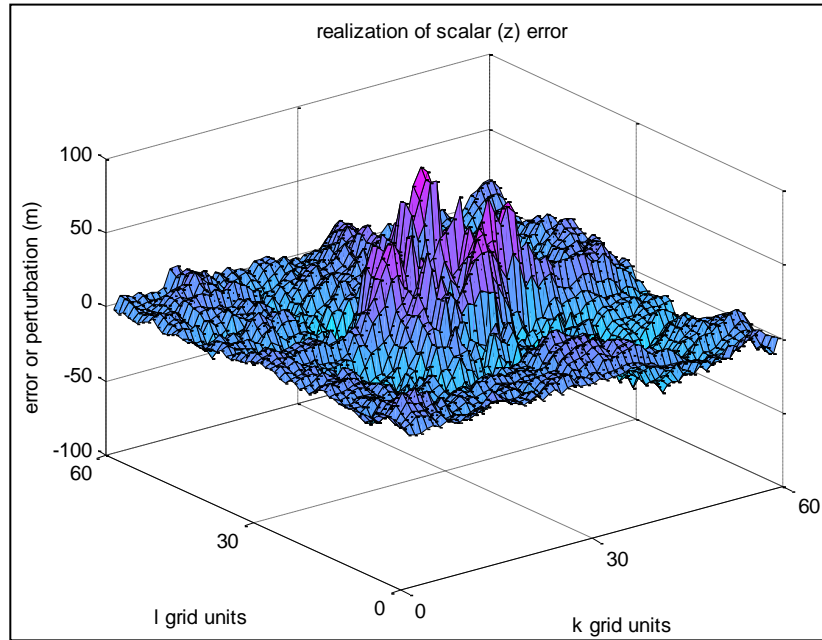
### 7.2.2. Statistics and Multi-Grid Point Covariance Matrix

Corresponding *a priori* statistics are no longer straightforward for this method, but can be approximated. In particular,  $\sigma_z$  corresponding to a specific location  $(k, l)$  is the corresponding bilinear interpolated value. The spatial correlation function corresponding to  $m$  different locations  $(k, l)$  is the average of  $m$  spatial correlation functions, each corresponding to the bilinear interpolated values for  $s$  and  $r$  for that location. Of course, these statistics reflect non-homogeneity, *i.e.*, are a function of the specific  $(k, l)$  locations of interest. The corresponding  $m \times m$  approximation for the multi-grid point covariance matrix corresponding to scalar errors at  $m$  different grid locations is represented as follows:

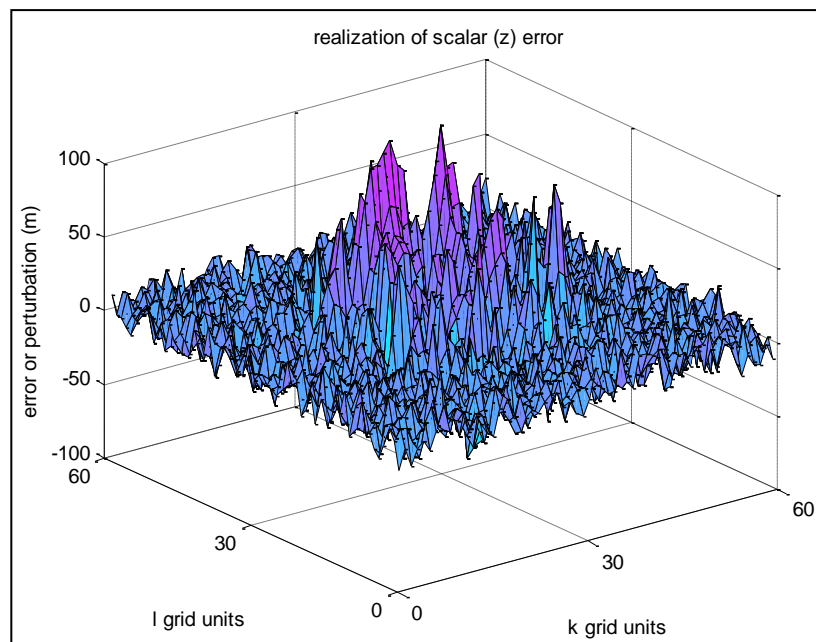
$$P = \{ZZ^T\} = \begin{bmatrix} \sigma z_1^2 & \overline{\rho(\Delta y_{12}, \Delta x_{12})} \sigma z_1 \sigma z_2 & \dots & \overline{\rho(\Delta y_{1m}, \Delta x_{1m})} \sigma z_1 \sigma z_m \\ - & \sigma z_2^2 & \dots & \overline{\rho(\Delta y_{2m}, \Delta x_{2m})} \sigma z_2 \sigma z_m \\ & - & \ddots & - \\ & - & - & \sigma z_m^2 \end{bmatrix} \quad (22)$$

where  $Z$  is an  $m \times 1$  vector such that  $Z^T = [z_1 \dots z_m]$  and the  $z_i = z(k_i, l_i)$ ,  $i = 1, \dots, m$ , correspond to an ordered list of the  $m$  grid point locations.

**Figure 20.** Non-homogeneous scalar realization—different variances.



**Figure 21.** Non-homogeneous scalar realization—different variances and spatial correlations.



$\overline{\rho(\Delta y_{ij}, \Delta x_{ij})} = \left(\frac{1}{m}\right) \sum_{v=1}^m \rho_v(\Delta y_{ij}, \Delta x_{ij})$ , and the individual spatial correlation functions are defined by their corresponding interpolated values for  $s$  and  $r$ .

Because the average of a collection of strictly positive definite correlation functions (spdcfs) is an spdcf itself, the above is guaranteed a valid covariance matrix regardless of the fact that the various  $\sigma z_i$  can vary in value (Reference [7]).

Note that if the  $m$  grid points consist of widely spaced subgroups of points such that the scalar error at any grid point in one subgroup has (approximated) low correlation (e.g., less than 0.1) with the scalar error at any grid point in any other subgroup, a higher fidelity representation for the multi-grid point covariance matrix can be achieved as follows: Use the representation in Equation (22) to compute a “sub-multi-grid point covariance matrix” for each subgroup of grid points, and then combine them into the (final) multi-grid point covariance matrix by placing (in order) each sub-multi-grid point covariance matrix down the block diagonals with values of zero for all off-diagonal (cross-covariance) blocks.

Finally, to generate a multi-grid point covariance matrix for a non-homogeneous multivariate  $X(k, l)$  instead of that for a non-homogeneous scalar  $z(k, l)$ , the same general procedure presented in this section can be extended in a straightforward manner using methods described in References [6,7].

## 8. Summary and Conclusions

Practical methods for the sequential generation of two-dimensional random fields were presented, and their corresponding multivariate covariance matrices derived. The corresponding methods were based on FSS, which was also compared to Sequential Gaussian Simulation and other approaches. Although less general, FSS was shown to be clearly superior in terms of speed and simplicity, primarily due to assumed separable exponential spatial correlation and simple ordered generation over an evenly spaced grid. FSS methods presented in the paper are applicable to performance assessment and tuning of geospatial applications in a simulation environment, as well as near-real time display of the effect of errors on applications by the applications themselves.

## Acknowledgments

The authors would like to thank Michael Lenihan and Christopher O'Neill of the Sensor Geopositioning Center (SGC) of the National Geospatial-Intelligence Agency (NGA) for their programmatic support.

## Author Contributions

Both authors made significant contributions throughout the manuscript.

## Conflicts of Interest

The authors declare no conflict of interest.

## References

1. Chiles, J.; Delfiner, P. *Geostatistics: Modeling Spatial Uncertainty*; Wiley: New York, USA, 1999.
2. Christakos, G. Simulation of Natural Processes. In *Random Field Models in Earth Sciences*; Dover: New York, NY, USA, 2005; pp. 295–336.
3. Cressie, N.; Wikle, C.K. *Statistics for Spatio-Temporal Data*; Wiley: Hoboken, NJ, USA, 2011.

4. Goovaerts, P. *Geostatistics for Natural Resources Evaluation*; Oxford University Press: Oxford, UK, 1997.
5. Schabenberger, O.; Gotway, C.A. Simulation of Random Fields. In *Statistical Methods for Spatial Data Analysis*, 1st ed.; Chapman & Hall/CRC: Boca Raton, FL, USA, 2005; pp. 405–420.
6. Dolloff, J. The full multi-state vector error covariance matrix: Why needed and its practical representation. *Proc. SPIE* **2013**, doi:10.1117/12.2015121.
7. Dolloff, J.; Lofy, B.; Sussman, A.; Taylor, C. Strictly positive definite correlation functions. *Proc. SPIE* **2006**, 6235, doi:10.1117/12.663967.
8. Doucette, P.; Dolloff, J.; Zuzelski, R.; Lenihan, M.; Mosko, D. Evaluating conflation methods using uncertainty modeling. *Proc. SPIE* **2013**, doi:10.1117/12.2015321.
9. Doucette, P.; Dolloff, J.; Spizler, A. Experiments with Fast Sequential Simulation for Assessment of Geospatial Algorithms. In Proceedings of 11th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences, East Lansing, MI, USA, 8–11 July 2014, in press.
10. Pebesma, E.J. Package “Gstat” (Version 1.0-19); 2014. Available online: <http://cran.r-project.org/web/packages/gstat/gstat.pdf> (accessed on 30 May 2014).
11. Pebesma, E.J. Multivariable geostatistics in S: The gstat package. *Comput. Geosci.* **2004**, *30*, 683–691.
12. Hansen, T.M. MGstat (Version 0.99); 2011. Available online: <http://mgstat.sourceforge.net/mGstat.pdf> (accessed on 30 May 2014).
13. Deutsch, C.; Journel, A. *GSLIB: Geostatistical Software Library and User's Guide*; Oxford University Press: New York, NY, USA, 1992; p. 340.
14. Tran, T. Improving variogram reproduction on dense simulation grids. *Comput. Geosci.* **1994**, *20*, 1161–1168.
15. Bar-Shalom, Y.; Fortmann, T. *Tracking and Data Association*; Academic Press: San Diego, CA, USA, 1988.

## Appendix A: Derivation of Statistics for the FSS Scalar Gaussian 2D Random Field

The following derives the steady-state variance ( $\sigma_z^2$ ) and spatial correlation ( $\rho(\Delta k, \Delta l)$ ) associated with  $z(k, l)$ , assumed generated using Equation (3) from the main body of this paper. These *a priori* statistics were also summarized earlier in Equation (2).

First we show the relationship of  $z(k, l)$  with the various white noise (uncorrelated between grid points) samples  $u(k, l)$  across the grid. After that, we calculate the corresponding statistical properties of  $z(k, l)$  based on those for  $u(k, l)$ .

### A.1. Relationship of Core Grid-Generation Equation with Underlying Random Samples

The following derives the relationship of Equation (3), the core-grid generation equation for  $z(k, l)$ , with the underlying random samples  $u(k, l)$  across the (infinite) grid.

Define:

$$z(k, l) = \sum_{i \leq k} \sum_{j \leq l} s^{k-i} r^{l-j} u(i, j) \quad (\text{A1})$$

a convolution over a grid of uncorrelated random numbers, with the point  $z(k, l)$  in the lower left corner of the kernel (see the light orange area in Figure 2).

It follows that:

$$z(k+1, l+1) = rz(k+1, l) + sz(k, l+1) - rsz(k, l) + u(k+1, l+1) \quad (\text{A2})$$

(Note that Equation (A2) is a repeat of Equation (3) from the main body of this paper for convenience.)

#### A.1.1. Proof of Relationship

The following presents a proof of Equation (A-2) by direct expansion using Equation (A1):

Left side of Equation (A2),

$$\begin{aligned} z(k+1, l+1) &= \sum_{i \leq k+1} \sum_{j \leq l+1} s^{k+1-i} r^{l+1-j} u(i, j) = \\ &u(k+1, j+1) + ru(k+1, l) + s(u(k, l+1) + sr \sum_{i \leq k} \sum_{j \leq l} s^{k-i} r^{l-j} u(i, j)) \end{aligned}$$

Right side of Equation (A2),

$$\begin{aligned} rz(k+1, l) + sz(k, l+1) - rsz(k, l) + u(k+1, l+1) &= \\ r \sum_{i \leq k+1} \sum_{j \leq l} s^{k+1-i} r^{l-j} u(i, j) + s \sum_{i \leq k} \sum_{j \leq l+1} s^{k-i} r^{l+1-j} u(i, j) - rs \sum_{i \leq k} \sum_{j \leq l} s^{k-i} r^{l-j} u(i, j) + \\ u(k+1, l+1) &= \\ r \left( u(k+1, l) + s \sum_{i \leq k} \sum_{j \leq l} s^{k-i} r^{l-j} u(i, j) \right) + s \left( u(k, l+1) + r \sum_{i \leq k} \sum_{j \leq l} s^{k-i} r^{l-j} u(i, j) \right) - \\ rs \left( \sum_{i \leq k} \sum_{j \leq l} s^{k-i} r^{l-j} u(i, j) \right) + u(k+1, l+1) &= ru(k+1, l) + su(k, l+1) + u(k+1, l+1) + \\ (rs + sr - rs) \sum_{i \leq k} \sum_{j \leq l} s^{k-i} r^{l-j} u(i, j) &= u(k+1, l+1) + ru(k+1, l) + su(k, l+1) + \\ sr \sum_{i \leq k} \sum_{j \leq l} s^{k-i} r^{l-j} u(i, j) \end{aligned}$$

The left and right sides of Equation (A2) are equal, hence Equation (A2) is correct. This proved that Equation (A1)  $\Rightarrow$  Equation (A2); see Section A.3 for an informal proof that Equation (A2)  $\Rightarrow$  Equation (A1) for completeness.

#### A.2. Derivation of Statistics

In addition, the following *a priori* statistics apply:

$$E\{z(k, l)\} = 0, \text{ i.e., a mean value of zero; } \quad (\text{A3})$$

$$E\{z(k, l)^2\} \equiv \sigma_z^2 = \frac{\sigma_u^2}{(1-s^2)(1-r^2)}, \text{ or more generally, } E\{z(k, l)z(k+or-\Delta k, l+or-\Delta l)\} = s^{\Delta k} r^{\Delta l} \sigma_z^2.$$

Note that this set of equations implies that:

$$\text{For a given } \sigma_z, \sigma_u = (1-s^2)^{1/2} (1-r^2)^{1/2} \sigma_z; \quad \rho(\Delta k, \Delta l) = s^{\Delta k} r^{\Delta l}, \text{ or equivalently, } \rho(\Delta k, \Delta l) = \rho(\Delta y, \Delta x) = e^{-\Delta k \delta_y / T_y} e^{-\Delta l \delta_x / T_x}.$$

## A.2.1. Detailed Derivations

The following derives Equation (A3) by the statistical properties of Equation (A1):

$$\begin{aligned} E\{z(k, l)\} &= E\left\{\sum_{i \leq k} \sum_{j \leq l} s^{k-i} r^{l-j} u(i, j)\right\} = \sum_{i \leq k} \sum_{j \leq l} s^{k-i} r^{l-j} E\{u(i, j)\} = 0; \\ E\{z(k, l)^2\} &= E\left\{\left(\sum_{i \leq k} \sum_{j \leq l} s^{k-i} r^{l-j} u(i, j)\right)\left(\sum_{i \leq k} \sum_{j \leq l} s^{k-i} r^{l-j} u(i, j)\right)\right\} = \\ &= \sum_{i \leq k} \sum_{j \leq l} s^{2(k-i)} r^{2(l-j)} E\{u(i, j)^2\} = \sigma_u^2 \sum_{i \leq k} \sum_{j \leq l} s^{2(k-i)} r^{2(l-j)} = \\ &= \sigma_u^2 \left(\sum_{i \leq k} s^{2(k-i)}\right) \left(\sum_{j \leq l} r^{2(l-j)}\right) = \\ &= \sigma_u^2 (1 + s^2 + s^4 + \dots)(1 + r^2 + r^4 + \dots) = \frac{\sigma_u^2}{(1-s^2)(1-r^2)}; \end{aligned}$$

The above utilized the following: by definition,  $E\{u(k, l)\} = 0$ ,  $E\{u(k, l)^2\} = \sigma_u^2$ , and  $E\{u(k, l)u(p, q)\} = 0$  for  $p \neq k$  or  $q \neq l$ ; by the properties of a geometric series,  $\sum_{k=0}^{\infty} a^k = \frac{1}{1-a}$ , where  $0 < a < 1$ , which is applicable since  $0 < a = s^2 = e^{-2\delta_y/T_y} < 1$  and  $0 < a = r^2 = e^{-2\delta_x/T_x} < 1$  in the above.

Define  $m = \Delta k > 0$  and  $n = \Delta l > 0$  for convenience.

$$\begin{aligned} E\{z(k+m, l+n)z(k, l)\} &= E\{z(k, l)z(k+m, l+n)\} = \\ &= E\left\{\left(\sum_{i \leq k} \sum_{j \leq l} s^{k-i} r^{l-j} u(i, j)\right)\left(\sum_{i \leq k+m} \sum_{j \leq l+n} s^{k+m-i} r^{l+n-j} u(i, j)\right)\right\} = \\ &= E\left\{\left(\sum_{i \leq k} \sum_{j \leq l} s^{k-i} r^{l-j} u(i, j)\right)\left(s^m r^n \sum_{i \leq k} \sum_{j \leq l} s^{k-i} r^{l-j} u(i, j)\right)\right\} \end{aligned}$$

The latter representation is due to the fact that  $E\{u(k, l)u(p, q)\} = 0$  for  $p \neq k$  or  $q \neq l$ .

Thus,

$$E\{z(k, l)z(k+m, l+n)\} = s^m r^n E\left\{\left(\sum_{i \leq k} \sum_{j \leq l} s^{k-i} r^{l-j} u(i, j)\right)\left(\sum_{i \leq k} \sum_{j \leq l} s^{k-i} r^{l-j} u(i, j)\right)\right\} = s^m r^n \sigma_z^2.$$

Similarly,

$$\begin{aligned} E\{z(k-m, l+n)z(k, l)\} &= E\{z(k, l)z(k-m, l+n)\} = \\ &= E\left\{\left(\sum_{i \leq k} \sum_{j \leq l} s^{k-i} r^{l-j} u(i, j)\right)\left(\sum_{i \leq k-m} \sum_{j \leq l+n} s^{k-m-i} r^{l+n-j} u(i, j)\right)\right\} = \\ &= E\left\{\left(\sum_{i \leq k-m} \sum_{j \leq l} s^{k-i} r^{l-j} u(i, j)\right)\left(\sum_{i \leq k-m} \sum_{j \leq l} s^{k-m-i} r^{l+n-j} u(i, j)\right)\right\} = \\ &= E\left\{s^m \left(\sum_{i \leq k-m} \sum_{j \leq l} s^{k-m-i} r^{l-j} u(i, j)\right) r^n \left(\sum_{i \leq k-m} \sum_{j \leq l} s^{k-m-i} r^{l-j} u(i, j)\right)\right\} = \\ &= E\left\{s^m r^n \left(\sum_{i \leq k-m} \sum_{j \leq l} s^{k-m-i} r^{l-j} u(i, j)\right)\left(\sum_{i \leq k-m} \sum_{j \leq l} s^{k-m-i} r^{l-j} u(i, j)\right)\right\} = \\ &= s^m r^n E\left\{\left(\sum_{p \leq k} \sum_{j \leq l} s^{k-p} r^{l-j} u(p-m, j)\right)\left(\sum_{p \leq k} \sum_{j \leq l} s^{k-p} r^{l-j} u(p-m, j)\right)\right\} = \\ &= s^m r^n E\left\{\sum_{p \leq k} \sum_{j \leq l} s^{2(k-p)} r^{2(l-j)} E\{u(p-m, j)^2\}\right\} = \\ &= s^m r^n \sigma_u^2 E\left\{\sum_{p \leq k} \sum_{j \leq l} s^{2(k-p)} r^{2(l-j)}\right\} = s^m r^n \sigma_u^2 \left(\sum_{i \leq k} s^{2(k-i)}\right) \left(\sum_{j \leq l} r^{2(l-j)}\right) = \\ &= s^m r^n \sigma_z^2. \end{aligned}$$

By the same procedure, it is also follows that

$$E\{z(k+m, l-n)z(k, l)\} = E\{z(k, l)z(k+m, l-n)\} = s^m r^n \sigma_z^2.$$

Thus,

$$E\{z(k, l)z(k+or-\Delta k, l+or-\Delta l)\} = s^m r^n \sigma_z^2 = s^{\Delta k} r^{\Delta l} \sigma_z^2,$$

or alternatively,

$$E\{z(k, l)z(k+or-\Delta k, l+or-\Delta l)\} = e^{-\Delta k \delta_y / T_y} e^{-\Delta l \delta_x / T_x} \sigma_z^2.$$

### A.3. Further Relationship of the Core Grid-Generation Equation with Underlying Random Samples

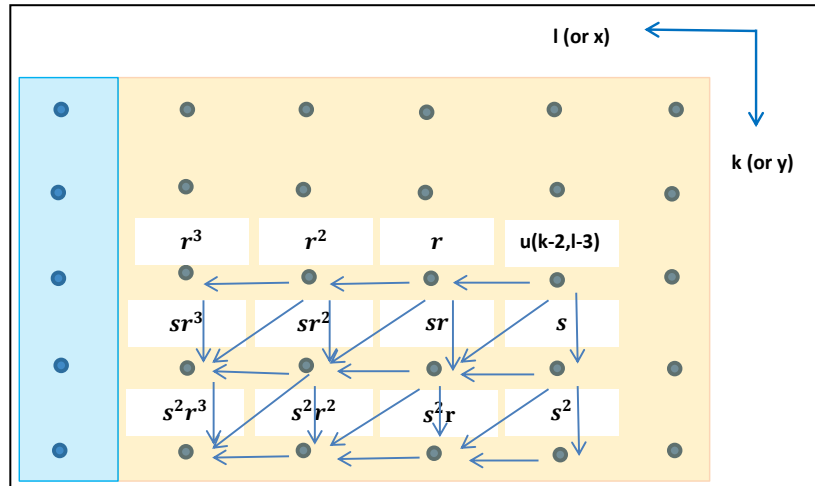
The following presents an informal proof that Equation (A2)  $\Rightarrow$  Equation (A1). That is, Equation (A2) for the sequential generation of the scalar random field  $z(k, l)$  implies the representation of  $z(k, l)$  as a weighted sum (kernel) of an infinite grid of  $u(k, l)$  as represented by Equation (A1).

Figure A1 demonstrates that each random sample  $u(i, j)$  in the light orange area contributes  $s^{k-i} r^{l-j} u(i, j)$  to  $z(k, l)$  via Equation (A2). In the following figure,  $i = k - 2$  and  $j = l - 3$  for specificity.

Note that, for example, in the above figure the  $u(k - 2, l - 3)$  multiplier corresponding to grid location  $(k - 1, l - 2)$  is equal to:  $sr^2 = r(sr) + s(r^2) - rs(r)$ , via applications of Equation (A2). This same three-term paradigm is also applicable to all other grid locations with an incoming “diagonal arrow”.

Finally, the multiplier  $s^2 r^3$  corresponds to the contribution of  $u(k - 2, l - 3)$  to  $z(k, l)$ , i.e.,  $z(k, l) = s^2 r^3 u(k - 2, l - 3) + \text{other } u \text{ terms}$ , or more generally,  $z(k, l) = \sum_{i \leq k} \sum_{j \leq l} s^{k-i} r^{l-j} u(i, j)$ , i.e., Equation (A1).

**Figure A1.** The “route” of  $u(k - 2, l - 3)$  to  $z(k, l)$ , the lower left grid location in the orange area.



## Appendix B: Mathematical Comparison of FSS to Sequential Gaussian Simulation

Sequential Gaussian Simulation is based on drawing random numbers from the appropriate (conditional) probability distributions, and since distributions are Gaussian, equivalent to the use of the appropriate conditional mean and variance. These, in turn, correspond to the minimum mean square estimate, or simple Kriging interpolation, detailed as follows. Let us assume that a horizontal grid of realizations as per the pattern specified in Section 2.2 has already been generated and that the next point in the ordered grid is to be generated based on these values. Let us specify  $X_1$  as the  $n \times 1$  vector of generated values and  $X_2$  as the scalar value to be generated:

$$\begin{aligned} X_2 &= P_{21} P_{11}^{-1} X_1 \\ \sigma_{X_2}^2 &= \sigma_z^2 - P_{21} P_{11}^{-1} P_{12} \end{aligned} \quad (B1)$$

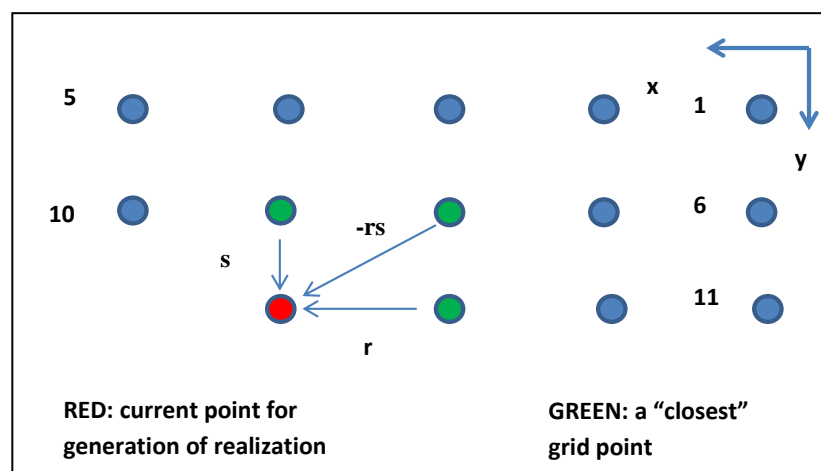


$X_2$  is the best estimate of the realization at the appropriate horizontal location.  $X_2 + \text{random\_}N(0, \sigma_{X_2})$  is the corresponding simulated value, where  $\text{random\_}N(0, \sigma_{X_2})$  corresponds to a mean-zero Gaussian random number with variance  $\sigma_{X_2}^2$ , i.e., the variance of the Kriging solution relative to the value of the true realization.

(More details can be found in the literature for simple Kriging [1,3], conditional and unconditional simulation [1], sequential simulation [14], Sequential Gaussian Simulation [4], and Gaussian probability distributions, the conditional mean, and the conditional variance [15].)

Let us now assume specific grid point locations of interest and a Gaussian two-dimensional random field with *a priori* standard deviation  $\sigma_z = 10\text{ m}$  and separable exponential spatial correlation with correlation between adjacent grid point locations  $r = e^{-1/10}$  and  $s = e^{-1/20}$ . A representative set of 14 grid locations ( $n = 13$ ) are represented graphically as follows:

**Figure B1.** The example grid.



We first generate the realizations for  $X_1$  and then  $X_2$  based on the FSS sequential method presented in this paper (Section 2), and obtain:  $X_1^T = [7.15\ 6.20\ 5.45\ 6.88\ 8.08\ 8.66\ 8.46\ 5.92\ 8.16\ 11.28\ 8.88\ 9.99\ 8.34]$  and  $X_2 = [9.81]$ . (Components 1–13 of  $X_1$  correspond to grid points #1–13, and  $X_2$  corresponds to the red point in Figure B1.) During generation of  $X_2$ , the additive Gaussian random number  $u$  per Equation (3) was equal to  $-0.39$  corresponding to the standard deviation  $\sigma_u = \sigma_z \sqrt{1 - r^2} \sqrt{1 - s^2} = 1.31$ . (Also, using the symbology of Equation (3),  $X_2$  corresponds to  $z(k + 1, l + 1)$  and  $u$  to  $u(k + 1, l + 1)$ .)

We now implement the Kriging equations (B1) for Sequential Gaussian Simulation using the value of  $X_1$  generated by FSS and detailed in the previous paragraph. In addition, the same additive Gaussian random number  $u$  will be added to the Kriging solution for  $X_2$  per the Sequential Gaussian Simulation procedure since, as will be demonstrated below,  $\sigma_{X_2} = \sigma_u$ . In support of the Kriging solution, the *a priori* cross-covariance matrix between  $X_2$  and  $X_1$  and the *a priori* covariance matrix for  $X_1$  are computed in accordance with the assumed statistics ( $\sigma_z, r, s$ ) of the random field presented previously. Correspondingly, both  $P_{21}$  and  $P_{11}$  are full (no zero elements), but the product  $P_{21}P_{11}^{-1}$  is only non-zero for the three elements of  $X_1$  which correspond to the nearest three grid locations 8, 9, and 13 to the point to be simulated. (It also follows that a pre-computed “compressed”  $1 \times 3$  version of  $P_{21}P_{11}^{-1}$  can actually be used as common Kriging weights for the realizations at the three nearest grid points.)

In particular, and based on the point locations laid out in Figure B1:

$$P_{21} = \sigma_z^2 [r^3 s^2 \quad r^2 s^2 \quad rs^2 \quad \dots \quad r^2 \quad r] \quad (1 \times 13), \text{ and } P_{11} = \sigma_z^2 \begin{bmatrix} 1 & r & \dots & r^2 s^2 \\ \dots & 1 & \dots & rs^2 \\ \dots & \dots & \dots & 1 \end{bmatrix} \quad (13 \times 13).$$

Therefore, given the values of  $r$  and  $s$  specified earlier,  $P_{21}P_{11}^{-1} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ -rs \ s \ 0 \ 0 \ 0 \ r] = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ -86 \ 0.95 \ 0 \ 0 \ 0 \ 0.90]$ , and the solution (with additive random number) is  $P_{21}P_{11}^{-1}X_1 - 0.39 = 9.81$ , identical to that generated using the FSS method of this paper. In addition,  $\sigma_{X_2} = 1.31$ , which is equal to  $\sigma_u$ . These equivalences and the use of only the nearest three grid points were enabled due to both the separable exponential spatial correlation and the regular grid of realizations generated in a simple, ordered fashion. (Thus, for example, if grid point #8 were moved +0.5 grid units in the y-direction, there would be 7 instead of 3 non-zero scalar weights.)

The above was an arbitrary, but specific, example. A formal analytic proof that the Kriging weight row vector  $P_{21}P_{11}^{-1}$  has only non-zero weights  $-rs$ ,  $s$ , and  $r$  corresponding to the three nearest grid points is relatively easy and done by direct verification that  $P_{21} = WP_{11}$ , where  $W$  is a  $1 \times n$  row vector consisting of all zeroes except for the non-zero scalar weights  $-rs$ ,  $s$ , and  $r$  at the appropriate locations corresponding to the three nearest grid points (see Figure B1). Similarly, in order that  $\sigma_z = \sigma_u$ ,  $P_{21}P_{11}^{-1}P_{12}$  must equal  $\sigma_z^2(r^2 + s^2 - r^2s^2)$ , which is easily verified by direct evaluation of  $P_{21}P_{11}^{-1}P_{12} = WP_{12}$ .

Thus, this appendix has both demonstrated and proven that FSS is equivalent to Gaussian Sequential Simulation under appropriate circumstances.

## Appendix C: VISIM Parameter File

```
Parameter file for VISIM
*****
0 # - conditional simulation (3=Vol,2=point,1=Vol+Point, 0=Uncon)
nodata # - file with conditioning data
1 2 3 4 # - columns1 for X,Y,Z,vr,wt,sec.var
dummy # - Geometry of volume/ray
dummy # - Summary of volgeom.eas
-1 1e+21 # - trimming limits for conditioning data
0 -1 -1 -1 -1 0 0 # - debugging level: -1,0,1,2,3,
read_covtable,read_lambda,read_volnh,read_randpath,do_cholesky,do_error_sim
visim.out # - file for output
1 # - number of realizations to generate
0 # - ccdf. type: 0-normal, 1-target
reference.eas # - target histogram file
1 0 # - columns for variable and weights
-3.50 3.50 100 # - min_Gmean, max_Gmean, n_Gmean
0.00 2.00 100 # - min_Gvar, max_Gvar, n_Gvar
170 0 # - nq, do_discrete
500 1.0000 1.0000 # - nx,xmn,xsiz
500 1.0000 1.0000 # - ny,ymn,ysiz
```

```

1 1.0000 1.0000 # - nz,zmn,zsiz
69069 # - random number seed
0 8 # - min and max original data for sim
28 # - number of simulated nodes to use
3 32 0.001 # - volNH method(0,1,2) nusevols, covlevel
3 # - Random path
1 # - assign data to nodes (0=no, 1=yes)
0 # - maximum data per octant (0=not used)
30.0000 30.0000 0.0000 # - radius for search ellipsoid
0.0000 0.0000 0.0000 # - angles for search ellipsoid
10.00000000000 1.00000000000 # - global mean and variance
1 0.001000000 # - nst, nugget effect
2 0.9990000000 45.000000 0.0000 0.0000 # - it,cc,ang1,ang2,ang3
249.5000 249.5000 0.0000 # - a_hmax, a_hmin, a_vert
0.000 100.000 # - zmin,zmax (tail extrapolation for target histogram)
1 0.000 # - lower tail option, parameter
1 3.000 # - upper tail option, parameter

```

## Appendix D: Derivation of Statistics for the Multivariate Gaussian 2D Random Field

Equation (9) of Section 6 in the main body of this paper is a straightforward multivariate extension of the FSS scalar random field Equation (3) of Section 1.2. In this appendix we derive the corresponding statistics associated with Equation (9).

Equation (D1) below is a straightforward multivariate extension of the Equation (A1) of Appendix A:

$$X(k, l) = \sum_{i \leq k} \sum_{j \leq l} S^{k-i} R^{l-j} U(i, j), \text{ a vector of assumed dimension } v \times 1 \quad (\text{D1})$$

And based on Equation (D1), we derive the corresponding *a priori* multivariate statistics as follows:

$$\begin{aligned}
& E\{X(k, l)X(k + m, l + n)^T\} = \\
& E\left\{\left(\sum_{i \leq k} \sum_{j \leq l} S^{k-i} R^{l-j} U(i, j)\right) \left(\sum_{i \leq k+m} \sum_{j \leq l+n} S^{k+m-i} R^{l+n-j} U(i, j)\right)^T\right\} = \\
& E\left\{\left(\sum_{i \leq k} \sum_{j \leq l} S^{k-i} R^{l-j} U(i, j)\right) \left(\sum_{i \leq k} \sum_{j \leq l} S^{k+m-i} R^{l+n-j} U(i, j)\right)^T R^n S^m\right\} = \\
& \left(\sum_{i \leq k} \sum_{j \leq l} (S^{k-i} R^{l-j}) P_U (S^{k-i} R^{l-j})^T\right) R^n S^m = \\
& P_X S^m R^n, \text{ dimension } vxv, \text{ where}
\end{aligned}$$

$$P_X = A * P_U, A = \begin{bmatrix} \cdots & \cdots & \cdots \\ \cdots & (1 - s_p s_q)^{-1} (1 - r_p r_q)^{-1} & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix}, p, q \in \{1, \dots, v\}.$$

The  $p, q$  term of the above Hadamaker product ( $A * P_U$ ) corresponds to

$$\begin{aligned}
& \sum_{i \leq k} \sum_{j \leq l} s_p^{k-i} r_p^{l-j} P_U(p, q) r_q^{l-j} s_q^{k-i} = \\
& P_U(p, q) \sum_{i \leq k} s_p^{k-i} s_q^{k-i} \sum_{j \leq l} r_p^{l-j} r_q^{l-j} = \\
& P_U(p, q) \sum_{i \leq k} (s_p s_q)^{k-i} \sum_{j \leq l} (r_p r_q)^{l-j} = \\
& (1 - s_p s_q)^{-1} (1 - r_p r_q)^{-1} P_U(p, q).
\end{aligned}$$

Equivalently,  $P_U = H * P_X$ , where  $H = \begin{bmatrix} \cdots & \cdots & \cdots \\ \cdots & (1 - s_p s_q)(1 - r_p r_q) & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix}$ . Also,  $P_U$  must be positive definite in order that  $P_X$  is positive definite via the earlier summation. (Note that, in the above,  $S^m R^n = R^n S^m$ , i.e., they commute since they are diagonal matrices.) Also,  $E\{X(k, l)X(k - m, l + n)^T\} =$

$$E\left\{\left(\sum_{i \leq k} \sum_{j \leq l} S^{k-i} R^{l-j} U(i, j)\right) \left(\sum_{i \leq k-m} \sum_{j \leq l+n} S^{k-m-i} R^{l+n-j} U(i, j)\right)^T\right\} =$$

$$E\left\{\left(\sum_{i \leq k-m} \sum_{j \leq l} S^{k-m-i} R^{l-j} U(i, j)\right) \left(\sum_{i \leq k-m} \sum_{j \leq l} S^{k-m-i} R^{l-j} U(i, j)\right)^T R^n\right\} =$$

$$(S^m \sum_{i \leq k-m} \sum_{j \leq l} (S^{k-m-i} R^{l-j}) P_U (S^{k-m-i} R^{l-j})^T) R^n = S^m P_X R^n.$$

Similarly,

$$E\{X(k, l)X(k + m, l - n)^T\} = R^n P_X S^m, \text{ and}$$

$$E\{X(k, l)X(k - m, l - n)^T\} = S^m R^n P_X.$$

Further note that  $E\{X(k, l)X(k + m, l + n)^T\} = E\{(X(k + m, l + n)(X(k, l)^T)^T\}$ , etc.

In addition,  $E\{X(k, l)X(k + m, l + n)^T\} = E\{X(k - m, l - m)X(k, l)^T\}$ , since  $P_X S^m R^n = (S^m R^n P_X)^T$ , as required for wide-sense homogeneity (see Reference [7]).

© 2014 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).